BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER SYSTEMS
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

HIGH PERFORMANCE COMPUTING IN ULTRASOUND CANCER TREATMENT
VYUŽITÍ SUPERPOČÍTAČŮ PŘI ULTRAZVUKOVÉ LÉČBĚ RAKOVINY

HABILITATION THESIS
HABILITAČNÍ PRÁCE

AUTHOR **JIRI JAROS, PhD**
AUTOR PRÁCE

BRNO 2017

Preface

According to the Czech Society of Oncology, more than 73,000 tumour diseases are newly diagnosed in the Czech Republic every year and this number is continuing to grow. A very promising alternative to the standard treatment procedures is a non-invasive high intensity focused ultrasound (HIFU), also known as focused ultrasound surgery. The critical component of the effective HIFU treatment is the preoperative treatment planning to precisely place the focus at a desired position and determine an appropriate dosage. However, modelling the ultrasound propagation in human body is both physically complex and computationally challenging.

This thesis summarises original research in the area of large-scale acoustic model development, I have been involved in since 2011. During this period, my colleagues and I have published a large number of scientific papers, fourteen of which are presented in this thesis.

First, the acoustic model accounting for combined effect of heterogeneity, non-linearity and absorption is discussed. Next, the numerical solution using a corrected k-space pseudospectral method is introduced, and its main benefits and drawbacks are discussed. Consequently, the implementation, validation and performance of large-scale distributed simulation codes are outlined. The focus is put on two approaches of the simulation domain decomposition, namely the global domain decomposition with a superior accuracy but inherited communication bottleneck, and the novel local Fourier basis decomposition making a compromise between the numerical accuracy and the communication overhead. The performance, scaling and simulation cost of the developed simulation codes are evaluated on best supercomputers with thousands of processor cores and hundreds of graphics processing units. Finally, several clinical applications in the prostate, kidney and brain are presented and their impact discussed.

The accurate acoustic model and distributed simulation codes presented in this work have opened the door for disruptive science by allowing to simulate HIFU in domains comprising more than 8,000 cm^3 in a clinically meaningful time. This constitutes more than 250× bigger simulation domain than ones commonly used at the time I joined this project.

Brno, March 1st, 2017 Jiri Jaros

Acknowledgements

First of all, I would like to thank my colleague and a very good friend Bradley E. Treeby who initiated me into the mysteries of ultrasound modelling. Since the first day we met in Canberra, he has been a permanent source of stimulation and motivation for my work. I would also like to thank Ben T. Cox for a lot of useful discussion about acoustics and numerical methods. My great thanks must go to Alistair P. Rendell who provided me with a great deal of advice and constructive criticism which significantly influenced my research and academic career.

I am also grateful to Vaclav Dvorak who has been my mentor for more than a decade. Special thanks must go to Lukas Sekanina for supporting my research and helping me cut loose and start a new research group. I would also like to thank my colleagues Michal Bidlo, Richard Ruzicka and Vaclav Simek for many discussions and ideas we shared. I would like to appreciate all my PhD, graduate and undergraduate students who have put their shoulders to the wheel. Finally, I would like to thank all the other members of the Department of Computer System, Faculty of Information Technology, Brno University of Technology which I have had the pleasure to work with.

My heartfelt appreciation must go to my partner Marta Cudova for her love, support, motivation, and inspiration. I would also like to thank my dearest parents Jirina Jarosova and Bohuslav Jaros for their endless encouragement and steadfast support in all my endeavours and for believing in me. Last but not least, I would like to thank all friends from music bands, I have had the pleasure to play with, for keeping me sane and providing welcome distractions.

Contents

Chapter 1

Introduction

According to the Czech Society of Oncology[1], more than 73,000 tumours diseases are newly diagnosed in the Czech Republic every year and this number is continuing to grow. Sadly, almost 27,000 patients succumb to the disease every year. Unfortunately, current cancer treatment procedures including external beam radiation therapy (EBRT), chemotherapy and surgical interventions have severe limitations and side effects (radiation and drug dosage limits, operability, repeatability, long-lasting consequences) that reduce the chances of successful cure [26].

A very promising alternative to the standard treatment procedures is a non-invasive high intensity focused ultrasound (HIFU), also known as focused ultrasound surgery [1, 19, 44]. The technique works by sending a focused beam of ultrasound into the tissue, typically using a large transducer. At the focus, the acoustic energy is sufficient to cause cell death in a localised region while the surrounding tissue is left unharmed [36]. Simply said, HIFU "cooks" or ablates the target tissue at the focus of the ultrasound beam by thermal and cavitation effects. In recent years, HIFU has been used in clinical trials for the treatment of tumours in many organs, including the prostate, pancreas, uterus, kidney, liver, bone (periosteum), breast, and brain [18, 101].

To be effective, the HIFU application for tissue ablation requires tools for dosimetry therapy planning, and real-time feedback of the intended and actual target tissues. Typically, the therapy planning approach involves the use of pretreatment imaging data, defining the target and surrounding tissues by manual or semiautomatic segmentation, development of a 3D anatomy model of the region of interest from segmentation or registration with a reference dataset, simulation of the HIFU beam and thermal dosimetry around the target tissue, display and 3D visualization of imaging and simulation data, and review of the treatment plan options [2].

In principle, the simulation of the HIFU beam and the thermal dosimetry can be calculated using appropriate acoustic and thermal models [64, 88]. The most general approach for ultrasound simulation is to directly solve the equations of continuum mechanics while the thermal models can be solved using Penne's bioheat equation and appropriate tissue models [12, 66]. However, the modelling problem is both physically complex and computationally challenging. For example, the heterogeneous material properties of human tissue can cause the ultrasound beam to become strongly distorted [21, 78], the exact values for the material properties and their temperature dependence are normally unknown [12], and the rate and mechanism for tissue damage are both temperature and cell specific [3, 48].

[1] http://www.linkos.cz/en/czech-society-for-oncology/

Moreover, the tissue realistic acoustic models are extremely computationally demanding problem due to the large size of the region of interest in relation to the size of the acoustic wavelength. Furthermore, the generated acoustic pressures are of sufficient magnitude that the wave propagation is nonlinear, and the tissue absorption and related heating is strongly frequency dependant [32, 89]. For example, a typical MR-Guided HIFU operation may cover a domain of the size of 25 × 25 × 25 cm encompassing the HIFU transducer and the treatment area. At low focal intensities, nonlinear effects cause energy to be generated up to at least the 10th harmonic [98]. At very high focal intensities where strongly shocked waves are produced, as many as 600 harmonics might be required to model the focal heating accurately [46]. Thus, the frequency content of the propagating ultrasound waves can be very broadband.

If the governing equations describing the HIFU field are solved on a uniform Cartesian grid where the grid spacing is defined to meet the Nyquist limit of two points per minimum wavelength, the resulting grid sizes can exceed 10^{12} grid points even for a low focal intensity HIFU applications [32]. One 3D matrix of this size in single precession consumes almost 4 TB of computer memory, making many simulations intractable. Even the world No. 1 supercomputer Sunway TaihuLight featuring 1.3 PB of memory[2] might not be fast enough to deliver the treatment plans in clinically useful time-limits [62]. Therefore, new approaches are needed to allow accurate large-scale ultrasound simulations using more economical computational resources.

This habilitation thesis summarises my research into large-scale acoustic models and the development work on the acoustic k-Wave toolbox[3]. First, the acoustic model accounting for combined effect of heterogeneity, nonlinearity and absorption is discussed. Next, the numerical solution using a corrected k-space pseudospectral method is introduced, and its main benefits and drawbacks are discussed. Consequently, the implementation, validation and performance of large-scale distributed simulation codes are outlined. The focus is laid on two approaches of the simulation domain decomposition, namely the global domain decomposition with a superior accuracy but inherited communication bottleneck, and the novel local Fourier basis decomposition making a compromise between the numerical accuracy and the communication overhead. The performance, scaling and simulation cost of the developed simulation codes are evaluated on best supercomputers with thousands of CPU cores and hundreds of GPUs. The impact of the developed acoustic toolbox is demonstrated on several clinical applications including patient selection for the prostate salvage HIFU treatment, accurate focus placement in the kidney, transcranial ultrasonic neurostimulation, or emerging photoacoustic imaging.

[2] National Supercomputing Center in Wuxi, China, November 2016

[3] http://www.k-wave.org

Chapter 2

Tissue Realistic Modelling of Ultrasound Propagation

The tissue realistic models of ultrasound wave propagation in the human body have to take into account many specific aspects. The human body is composed of many different kinds of tissue which can be categorized either as soft or hard. Soft tissue includes tendons, ligaments, fascia, skin, fibrous tissues, fat, muscles, nerves and blood vessels. Hard tissues of humans are bones, skull, tooth enamel, dentin, and cementum.

Soft tissue is normally very hydrated because of the ground substance and has properties very similar to water (e.g., density and sound speed). Soft tissues is a compressible medium supporting longitudinal wave propagation. Hard tissue (also termed calcified tissue) is tissue which is mineralized and has a firm intercellular matrix. This tissue supports both longitudinal and transversal waves.

The HIFU treatment procedures on many organs usually involve both kinds of tissue. For example, intracranial operations and deep brain stimulation have to cope with strong beam aberration and reflection caused by the skull. Similarly, treatment of organs inside or in a close proximity of the rib cage has to prevent prefocal heating due to the strong scattering by the ribs. The clinically relevant acoustic model thus have to incorporate both the soft and hard tissue models, often referred to as fluid and elastic models, respectively.

2.1 Ultrasound Propagation in Soft Tissue

The development of accurate models for ultrasound propagation in soft tissue requires the consideration of three important factors. (1) In most cases the amplitude of the acoustic waves is sufficiently large that the wave propagation is nonlinear. (2) The material properties of biological tissue (e.g., the sound speed, density, acoustic absorption and nonlinearity parameters) are weakly heterogeneous, with variations between the different soft tissue types and water on the order of 5% [99]. (3) The tissue is absorbing at ultrasonic frequencies with the absorption following a frequency power law. In the context of nonlinear wave propagation, an accurate model of acoustic absorption is of particular importance as the generation of higher frequency harmonics via nonlinearity is delicately balanced with their absorption. Only by capturing accurate absorption, the energy deposition and thermal dose threshold in cumulative equivalent minutes leading to tissue devitalisation can be determined.

2.1.1 Governing equations

Our research team developed and released a full-wave nonlinear ultrasound model based on the k-space pseudospectral method as part of the open-source k-Wave Acoustics Toolbox [32, 83, 89]. Following [89], the required governing equations can be written as three-coupled first-order partial differential equations derived from the conservation laws and a Taylor series expansion for the pressure about the density and entropy

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} &= -\frac{1}{\rho_0}\nabla p + \mathbf{F} \ , && \text{(momentum conservation)} \\ \frac{\partial \rho}{\partial t} &= -\rho_0 \nabla \cdot \mathbf{u} - \mathbf{u}\cdot\nabla\rho_0 - 2\rho\nabla\cdot\mathbf{u} + \mathrm{M} \ , && \text{(mass conservation)} \\ p &= c_0^2 \left(\rho + \mathbf{d}\cdot\nabla\rho_0 + \frac{B}{2A}\frac{\rho^2}{\rho_0} - \mathrm{L}\rho\right) \ . && \text{(pressure-density relation)} \end{aligned} \tag{2.1}$$

Here $\mathbf{u}$ is the acoustic particle velocity, $\mathbf{d}$ is the acoustic particle displacement, p is the acoustic pressure, ρ is the acoustic density, ρ_0 is the ambient (or equilibrium) density, c_0 is the isentropic sound speed, and B/A is the nonlinearity parameter which characterises the relative contribution of finite-amplitude effects to the sound speed. These equations account for cumulative nonlinear effects (nonlinear effects that build up over space and time) up to second order in the acoustic variables, equivalent to the Westervelt equation [25, 96]. All the material parameters are allowed to be heterogeneous. Two linear source terms are also included, where $\mathbf{F}$ is a force source term which represents the input of body forces per unit mass in units of $\mathrm{N\,kg^{-1}}$, and M is a mass source term which represents the time rate of the input of mass per unit volume in units of $\mathrm{kg\,m^{-3}\,s^{-1}}$.

The nonlinear term in the mass conservation equation accounts for a convective nonlinearity in which the particle velocity affects the wave velocity can be written as

$$-2\rho\nabla\cdot\mathbf{u} \approx \frac{2}{\rho_0}\rho\frac{\partial\rho}{\partial t} = \frac{1}{\rho_0}\frac{\partial\rho^2}{\partial t} \approx \frac{1}{\rho_0 c_0^4}\frac{\partial p^2}{\partial t} \ . \tag{2.2}$$

The nonlinear term is written as a spatial gradient of the particle velocity. This is significant because spatial gradients can be computed accurately using spectral methods, and do not require any additional storage [89].

The operator L in the pressure-density relation in Eq. (2.1) is an integro-differential operator that accounts for acoustic absorption that follows a frequency power law of the form $\alpha = \alpha_0\omega^y$. This type of absorption has been experimentally observed in human soft tissues, where y is typically between 1 and 2 [8]. The operator has two terms both dependent on the fractional Laplacian and is given by [10, 84]

$$\mathrm{L} = \tau\frac{\partial}{\partial t}\left(-\nabla^2\right)^{\frac{y}{2}-1} + \eta\left(-\nabla^2\right)^{\frac{y+1}{2}-1} \ . \tag{2.3}$$

Here τ and η are absorption and dispersion proportionality coefficients given by $\tau = -2\alpha_0 c_0^{y-1}$ and $\eta = 2\alpha_0 c_0^{y}\tan(\pi y/2)$, where α_0 is the power law prefactor in $\mathrm{Np\ (rad/s)^{-y}\,m^{-1}}$, and y is the power law exponent. The two terms in L separately account for power law absorption and dispersion for $0 < y < 3$ and $y \neq 1$ [84, 85].

2.1.2 Numerical solution

Closely connected with the development of accurate governing equations for describing ultrasound propagation in tissue is the issue of their efficient solution. In a standard finite difference method, spatial gradients are computed locally based on the function values at neighbouring grid points. As an alternative, it is also possible to calculate spatial gradients globally using the function values across the whole domain via spectral methods. This increases the accuracy of the gradient calculation and thus reduces the number of grid points required per wavelength for a given level of accuracy. For smoothly varying fields, spatial gradients can be calculated with spectral accuracy up to the Nyquist limit (two grid points per wavelength). Often the spectral calculation of spatial gradients is combined with a leapfrog finite-difference scheme to integrate forwards in time (calculate temporal gradients). However, the finite difference approximation introduces unwanted numerical dispersion into the solution that can only be controlled by reducing the size of the time step or increasing the order of the approximation [20, 50, 51].

Fortunately, for the standard linear wave equation valid for homogeneous and lossless media, an exact finite difference scheme for the temporal derivative exists. This can be used to derive an exact pseudospectral discretization of both the second-order wave equation [91] and the corresponding coupled first-order conservation equations and pressure-density relation [80]. This approach is known as the k-space pseudospectral method (or simply the k-space method), because the difference between the exact and standard finite difference approximations reduces to an operator in the spatial frequency domain (referred to herein as the k-space operator). In the case of heterogeneous and absorbing media, the temporal discretization is no longer exact. However, if these perturbations are small, the k-space operator still reduces the unwanted numerical dispersion associated with the finite difference approximation of the time derivative (see [89]).

The discrete governing equations can be found in [32]. For regularly spaced Cartesian grids, the spatial gradients can be computed efficiently using the fast Fourier transform (FFT). To prevent waves from wrapping around the domain a split-field anisotropic Perfectly Matched Layer (PML) can be applied [80]. To further improve numerical accuracy, the grids, acoustic pressure and velocity are defined on, were also spatially and temporally staggered [80]. Further details can be found in Appendix A.1.

2.1.3 Model validation

The developed fluid model was validated by means of numerical accuracy analysis and experimental evaluation. In the limit of linear wave propagation in a lossless and homogeneous medium, the k-space pseudospectral discretization of the governing equations, Eq. (2.1), is exact and the algorithm is unconditionally stable. However, in the case of heterogeneous media the phase correction provided by k-space operator will no longer be exact, and unwanted numerical dispersion will be introduced into the solution. For soft biological tissue where the medium parameters are only weakly heterogeneous, the k-space method remains an apposite numerical technique with error below 1% for 3 grid points per wavelength and the level of parameter heterogeneity of 10%. Another source of numerical error is the PML layer with a typical attenuation of the signal at the domain edges of 100 dB [80].

Fig. 2.1 provides an insight of the numerical accuracy for classic finite difference time domain, pseudospectral and k-space corrected pseudospectral methods. Overall, for three-dimensional simulations, using a fourth-order accurate finite difference scheme requires

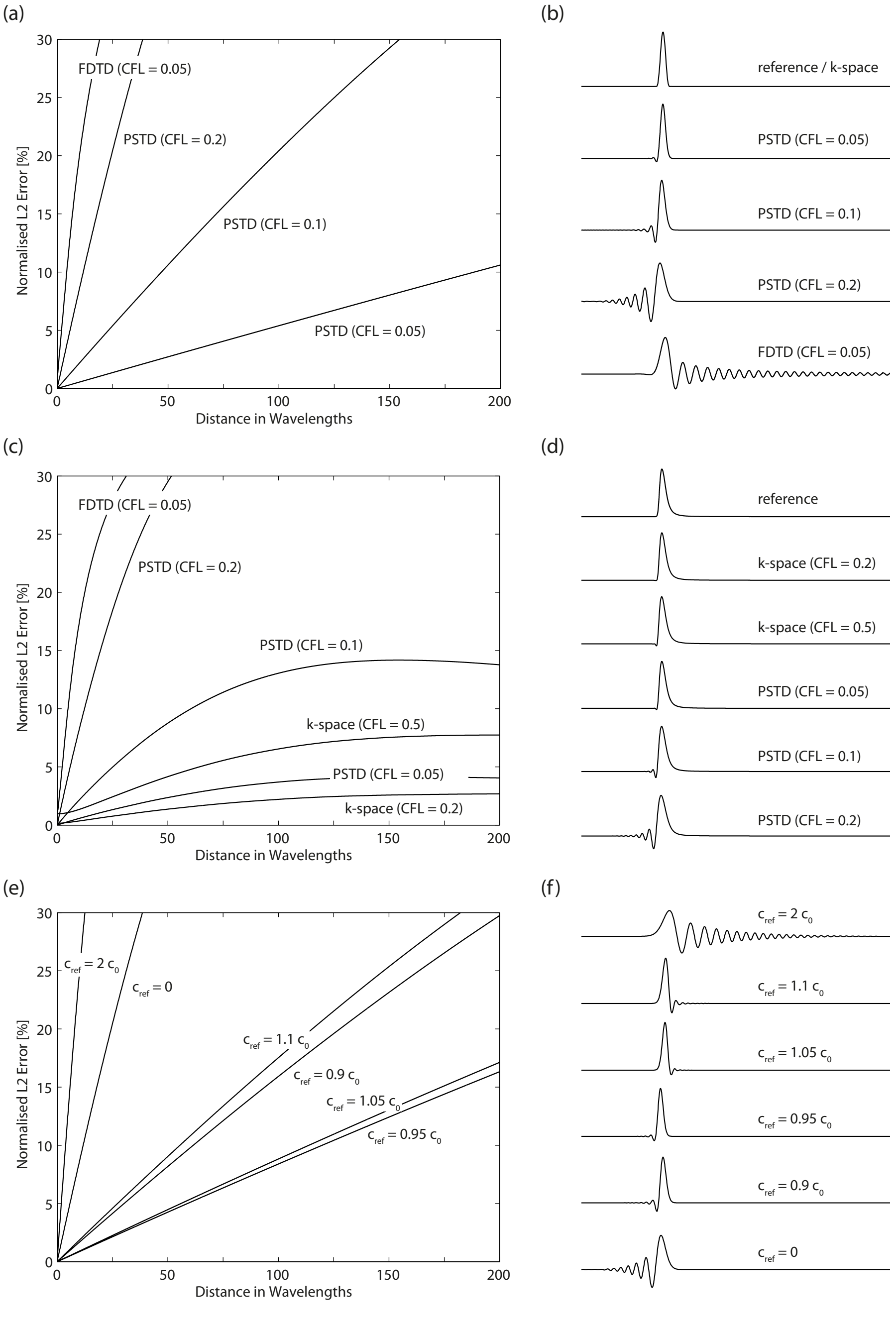


Figure 2.1: (a) Comparison of the accumulation of numerical error with distance for a linear simulation in a homogeneous and lossless medium using pseudospectral time domain (PSTD) and finite different time domain (FDTD) schemes for different Courant-Fredrichs-Lewy (CFL) numbers. In this case the k-space pseudospectral scheme is exact. (b) Corresponding time domain signals after propagating the input pulse a distance of 200 wavelengths. (c) Accumulation of numerical error for a nonlinear simulation in a homogeneous absorbing medium. In this case, the k-space pseudospectral scheme is no longer exact, however, the numerical error is significantly reduced compared to the PSTD scheme. (d) Corresponding time domain signals. (e) Accumulation of numerical error for a linear simulation in a homogeneous and lossless medium with variations in the reference sound speed. (f) Corresponding time domain signals.

around a 10-fold increase in the total number of grid points to achieve the same level of accuracy, while using a first-order scheme requires a 100-fold increase.

In conclusion, the accuracy of the k-space model is dependent on several parameters which can be tuned. First, the number of grid points used per wavelength will control whether the computational grid can support the propagation of the generated harmonics. In turn, the rate at which these harmonics are produced will depend on the shock parameter (for example, the source strength and the coefficient of nonlinearity), while the rate at which they are absorbed will depend on the power law absorption parameters. Finally, the Courant-Fredrichs-Lewy (CFL) number will control the amount of unwanted numerical dispersion introduced by the finite difference time step, as well as the accuracy with which the nonlinearity and absorption terms in the pressure-density relation are computed [53].

The developed k-space model implemented in the k-Wave toolbox was also cross-validated against the Iterative Contrast Source method (INCS) [15, 28] on two test configurations. In both configurations, a square piston excited a three-cycle Gaussian-modulated tone burst with a center frequency of 1 MHz and a source pressure of 750 kPa. The medium in the first configuration was homogeneous (water with power law attenuation with an exponent of 1.5 and a magnitude of 0.75 dB/cm at 1 MHz). In the second configuration, the medium was made inhomogeneous by putting a hollow cylinder (speed of sound equal to 1540 m/s) and a solid sphere (parameter of nonlinearity equal to 1) in the course of the radiated beam. In both cases, the results obtained with INCS and k-Wave were in excellent agreement, with maximum local differences in the order of 0.5-0.6 dB in the significant parts of the field. Because both methods are computationally quite different, it is improbable that these both suffer from the same systematic error. Hence it is established that both methods are correct and highly accurate, and are suitable tools for performing precise simulations and generating accuracy benchmarks. More details can be found in [16].

The k-Wave toolbox was also validated using experimental measurements of the ultrasound fields produced by a diagnostic ultrasound transducer and recorded by a membrane hydrophone. Measurements were performed in both deionised water and olive oil, the latter exhibiting power law absorption characteristics similar to human tissue. Steering angles of 0° and 20° were also tested, with propagation distances on the order of hundreds of acoustic wavelengths. The simulated and experimental results showed a close agreement in both the time and frequency domains, see [94] and Appendix A.2.

2.2 Ultrasound Propagation in Hard Tissue

Modelling the ultrasound propagation in hard tissue is in many aspects similar to the that in soft tissue, yet several additional aspects have to be considered to make the model clinically relevant. (1) Bones have a high acoustic absorption coefficient and a compressional sound speed over twice as high as soft tissue. This leads to the attenuation, reflection and aberration of propagating wavefronts and a reduction in focusing quality [41]. (2) In an elastic medium, the propagation of both compressional (longitudinal) and shear (transversal) waves has to be taken into account to precisely calculate the absorption and related heat deposition.

2.2.1 Governing equations

The elastic model developed and implemented in the k-Wave toolbox uses the Kelvin-Voigt model of viscoelasticity and the pseudospectral time domain solution [90].

The governing equations are based on two coupled first-order equations describing the stress and particle velocity within an isotropic medium. The equations only capture linear wave propagation (the nonlinear wave propagation is still under development). For absorbing media, the Kelvin-Voigt model of viscoelasticity is used. Followed [90] and Appendix A.3, the governing equations can be written using Einstein summation notation as

$$\frac{\partial \sigma_{ij}}{\partial t} = \lambda \delta_{ij} \frac{\partial v_k}{\partial x_k} + \mu(\frac{\partial v_i}{\partial x_i} + \frac{\partial v_j}{\partial x_i}) + \chi \delta_{ij} \frac{\partial^2 v_k}{\partial x_k \partial t} + \eta(\frac{\partial^2 v_i}{\partial x_j \partial t} + \frac{\partial^2 v_j}{\partial x_i \partial t}). \tag{2.4}$$

Here σ is the stress tensor, λ and μ are the Lamè parameters where μ is the shear modulus, and χ and η are the compressional and shear viscosity coefficients, and v_i is the particle velocity. To model the propagation of elastic waves, this is combined with an equation expressing the conservation of momentum. Written as a function of stress and particle velocity, this is given by

$$\frac{\partial v_i}{\partial t} = \frac{1}{\rho_0} \frac{\partial \sigma_{ij}}{\partial x_j}. \tag{2.5}$$

2.2.2 Numerical solution

A computationally efficient model for elastic wave propagation in absorbing media can be constructed based on the explicit solution of the coupled equations given in Eqs. (2.4)-(2.5) using the Fourier pseudospectral method [9, 52], similarly to the fluid model solution. This uses the Fourier collocation spectral method to compute spatial derivatives, and a leapfrog finite-difference scheme to integrate forwards in time. The accuracy is again improved by using temporally and spatially staggered grids. The k-space correction term can again be used to correct the phase error, however, it is very memory and computationally demanding in this case. A multi-axial perfectly matched layer (M-PML) is implemented to allow free-field simulations using a finite-sized computational grid [55]. Further details can be found in [90] or Appendix A.3.

2.2.3 Model validation

The work on the elastic model validation is still in progress. So far, we have attempted to quantify the impact of the numerical dispersion, the effectiveness of the PML, and the accuracy of the representation of step-changes in medium properties on the numerical accuracy of compressional waves propagation. We also compared the finite difference time domain scheme with the pseudospectral (PSTD) and k-space pseudospectral methods [74, 75]. The test case was inspired by transcranial transmission of ultrasound using the time-reversal focusing with medium including several types of tissue (water, skull, fat and brain tissue). The medium was not weakly heterogeneous any more and contained large step-changes in medium properties (e.g., sound speed in fat and bone is 1430 ms^{-1} and 3200 ms^{-1}, respectively [8]).

The experimental work has shown that even in a strongly heterogeneous medium, the k-space method can significantly reduce memory requirements by only needing 4 point per wavelength, compared to 11.2 and 17.9 for FDTD and PSTD methods, respectively. When taking into account multiple reverberations, this grows up to 5.4, 28.6 and 25.1 points per wavelength for k-space, PSTD and FDTD methods, respectively.

2.3 Related Work

Since clinically relevant ultrasound simulations are prohibitively complex from both the physical and computational point of view, simplifying assumptions are frequently made. For modelling the beam patterns from ultrasound transducers, a common approach is to only consider one-way (or forward) wave propagation (see [28] for a recent review). If the problem is axisymmetric, the governing equations can also be solved in 2D to dramatically reduce computer resources [77]. However, these approaches are unable to account for all aspects of nonlinear wave propagation in heterogeneous media. For the simulation of diagnostic ultrasound images, a Green's function method is also often used [37]. In this case, the scattering medium is modelled as series of point sources in a homogeneous background. However, this does not account for more complex acoustic phenomena, for example, multiple scattering or nonlinearity.

Over the last half a century, a large number of researchers have contributed to an extensive body of knowledge on the nonlinear propagation of acoustic waves [17]. However, despite the long history of nonlinear acoustics, most rigorous derivations are based on the assumption of a homogeneous medium with thermoviscous absorption. In particular, there have been very few attempts to consider acoustic heterogeneities at the level of the governing equations such as the one presented in [13].

The most common approach to modelling heterogeneous medium parameters is to assume that the effects of nonlinearity and heterogeneity are sufficiently small that their interactions can be neglected. For example, Hallaj et al. [24] and Pinton et al., [68] both utilized a Westervelt equation augmented with the heterogeneous density term from the linear wave equation. Jing and Cleveland [40] presented a similar wave equation including local nonlinear effects. This was then reduced to a Khokhlov–Zabolotskaya–Kuznetsov (KZK) equation suitable for heterogeneous media. An analogous approach was taken by Verweij and Huijssen [92] and Jing and Clement [42] where both the nonlinearity and heterogeneity terms were introduced as contrast source terms. Similarly, Averyanov et al., supplemented a linear parabolic wave equation for heterogeneous media with additional terms describing the effects of nonlinearity and absorption [4]. While the accuracy of these wave equations for modelling nonlinear wave propagation in weakly heterogeneous media is well established, they do not provide heterogeneous forms of the conservation and pressure-density equations which can be solved as a set of coupled first-order equations.

Classical lossy wave equations based on the inclusion of viscosity and thermal conduction into the governing equations yield an acoustic absorption term that is proportional to frequency squared. However, the absorption mechanisms in soft biological tissue are significantly more complex (including vibrational, structural, and chemical relaxations) which leads to an experimentally observed attenuation of the form

$$\alpha = \alpha_0 \omega^y, \tag{2.6}$$

where the power law exponent y is typically in the range 1-1.5 [81]. To account for this difference, the thermoviscous absorption term can be replaced with an alternate loss term. There have been several attempts by Blackstock [5], Szabo [79] or Nachmann [57] but with frequency range limitations, extreme memory requirements given by time domain convolution, or requires unknown relaxation times for biological materials.

2.3.1 Contributions and future work

The contribution of our work to the field on nonlinear acoustics in human body can be summarised as follows:

1. Derivation of a full-wave acoustic model for soft tissues accounting for combined effect of nonlinearity, heterogeneity and power law absorption.
2. Cross-validation and experimental evaluation of the fluid model.
3. Efficient numerical solution of the fluid model using k-space corrected pseudospectral method.
4. Derivation of a full-wave acoustic model for hard tissues accounting for combined effect of heterogeneity and absorption.
5. Efficient numerical solution of the elastic model using pseudospectral method.

There are a lot of challenges our research is facing to:

1. Derivation of the nonlinear elastic model.
2. Reduction of the computational complexity of the k-space method for the elastic model.
3. Derivation of the coupled fluid-elastic model minimising the computational complexity and allowing for more precise wave propagation in large domains.

Chapter 3

Development of Large-scale Ultrasound Simulation Codes

The computational complexity of realistic ultrasound simulations primarily lies in the scale of the problem. As spatial and temporal resolutions used in typical HIFU applications is on the order of 0.1 mm and 10 ns, respectively, the ultrasound simulation code has to operate on billions of grid points over hundreds of thousands of time steps [32]. Applying even a single operation per grid point per time step leads to the order of Peta (10^{15}) operations required to complete the simulation. However, there are thousands of operations necessary to be applied on a single grid point. This takes us to the order of Exa (10^{18}) operation required. Moreover, terabytes of data are to be accommodated in memory and processed every single time step. Clearly, there is absolutely no way how to satisfy such requirements by common workstations or servers while still meeting the clinically meaningful time span.

Fortunately, the performance of supercomputers, steadily grows. While the TOP #1 supercomputer of 1996 ASCI Red[1] opened the Tera-scale era by calculating over 10^{12} FLOPS (floating point operations per second), the Nvidia Tesla K20X[2] offered the same performance in a single graphics card in 2012. Currently, the best supercomputer is the Chinese Sunway TaihuLight[3] with the theoretical performance of 125 PetaFLOPS and over 1.3 PB of main memory. It is impressive to see the theoretical performance has grown more than 100,000 times over last two decades and we are rapidly approaching the exascale era expected in 2020 [73].

The current trend in supercomputing is towards fat and hybrid nodes interconnected with a fast low latency network, called computer clusters. Fat nodes are based on the Non-Uniform Memory Access (NUMA) architecture with usually 2-4 sockets each of which attached with 8-16 CPU cores and 64-128 GB of memory. Hybrid nodes contain one or more accelerators, often Nvidia GPUs [97] or Intel Xeon Phi accelerators [35]. The interconnection networks are based on either proprietary (CRAY) or commodity (Infiniband, Ethernet) technologies with bandwidth up to 100 Gb/s [45, 72]. This design has many benefits such as high integration, energy efficiency and good scaling. However, the developers are exposed to quite a complicated system with a deep memory hierarchy, many levels of parallelism, and an increasing gap between the compute performance and memory, peripheral and interconnection bandwidth [60].

[1] https://www.top500.org/featured/systems/asci-red-sandia-national-laboratory/

[2] https://www.hpcwire.com/2012/11/12/nvidia_unveils_1-3_teraflop_gpu_for_supercomputing/

[3] https://www.top500.org/system/178764

3.1 Simulation Framework Overview

Over last couple of years, we have been developing large scale implementations of the acoustic models described in Secs 2.1 and 2.2. The goal has always been the same: **"Make the simulation code as fast as possible while keeping the accuracy and simulation cost at an acceptable level"**. This has led to the development of several variants of the simulation code targeted on different domain sizes and computing platforms.

For domain sizes up to 512^3 grid points, the best platform is currently the Nvidia Pascal architecture which provides up to 24 GB of memory and compute performance over 12 TeraFLOPS[4]. For common GPU cards, the maximum domain size that fits into the on-board memory stays between 256^3 and 384^3 grid points. If the accelerator is not available, a single cluster node with 16 cores and 128 GB of RAM can process a domain of 1024^3 grid points, however in terms of days. More realistic domain sizes for a single node stay below 512^3 grid points. For larger simulations, a distributed code joining many nodes has to be used. However, here the inter-node communication comes into play ultimately limiting the code scaling. To deal with the communication complexity, we have investigated traditional global domain decompositions allowing to divide the work over up to 8192 cores and preserve the relative numeric error around 10^{-5}. Alternatively, we have been working on novel local domain decomposition techniques that decrease the communication overhead from the quadratic to linear complexity, but for the cost of reduced accuracy with relative errors around 10^{-3}.

3.1.1 Simulation work flow

Although the simulation codes are architecture specific, they share a common work flow. During the pre-processing phase, the input data for the simulation is generated. This involves defining the domain discretisation, defining the spatially varying material properties (e.g., using a CT scan of the patient [76]), defining the properties of the ultrasound transducer and drive signal and defining the desired output data (e.g., the peak positive pressure or time-averaged acoustic intensity in the region of the HIFU target [82]). The simulation phase involves reading the input data, running the actual simulation following the acoustic model used, and storing the output data. The post-processing phase involves analysing the (potentially large) output files and presenting this data in a human-readable form.

3.1.2 Simulation code description

The simulation code consists of several modules. The open-source HDF5 library was chosen to manipulate the input and output files because of its ability to organise complex datasets in heterogeneous computing environments [47]. The library provides both serial and parallel interfaces where the parallel one is able to reach I/O bandwidth over 25 GB/s on parallel lustre file systems [27].

The key part of the simulation codes is the gradient calculation. While finite difference methods only apply basic arithmetic operation on a few grid points in a close proximity, pseudospectral and k-space methods require the Fourier transform to be taken over the whole domain. In case of pseudospectral methods, the Fourier transforms are only calculated over one dimension (may be along x, y or z dimension in the 3D space). However, the k-space correction term requires all Fourier transforms to be three dimensional [83]. This

[4] http://www.nvidia.com/object/tesla-supercomputing-solutions.html

naturally leads to an increased computational complexity of the gradient calculation for pseudospectral and k-space methods to the order of $O(N^3 logN)$, see [14] for more details. Fortunately, considerable effort has already been devoted to the development of efficient Fast Fourier Transform (FFT) libraries, namely MIT FFTW [49], Nvidia cuFFT [43] and Intel MKL library [29].

In total, the simulation codes consist of 14 FFTs per time step for absorbing medium while only 10 FFTs for lossless medium. Since all quantities in the spatial space are defined on the real domain, the time and space complexity of the FFT calculation can be reduced by replacing full complex-to-complex FFTs by real-to-complex or complex-to-real transform.

Apart from the FFTs, only simple element-wise matrix operations spreading over approximately 100 compute kernels are executed. These operation can be effectively vectorised using SIMD instruction extensions such as SSE or AVX, however, these kernels are mostly memory bound.

3.1.3 Estimation of memory complexity

The approximate memory usage for a particular grid size Nx × Ny × Nz and floating point values stored in single precision can be estimated using the formula

$$\text{memory usage [GB]} \approx \frac{(13+A)NxNyNz + (7+B)\frac{Nx}{2}NyNz}{1024^3/4} + \text{input} + \text{output}. \quad (3.1)$$

Here the constants $A = [0, 9]$ and $B = [0, 2]$ depend on which material properties are heterogeneous. The parameter “input” is the size of the user defined input data (transducer shape definition and driving signals). Similarly, “output” is the shape of the sensor (the area of interest) and the quantity recorded per time step (e.g., pressure, velocity or aggregated values such as maximums, root mean square, etc.).

The number 13 in the first term accounts for storing spatial quantities such as pressure, velocity, gradient of velocity, plus some scratch place while the number 7 in the second term accounts for storing quantities in the Fourier space, such as the k-space operator, the matrix of wave numbers and temporary scratch place. As the Fourier space matrices are symmetrical, only the first half is stored. More details can be found in [86].

The real memory consumption measured on a shared memory system, see Fig. 3.1(c), is in a very good agreement with this simple model. In the case of distributed simulation codes, the memory consumption may be influenced by the presence of MPI communication buffers. These buffers may reach a substantial size when the code is being executed on a large number of processor cores (thousands and more). If necessary, their size can be limited by the MPI runtime, however, with possible impact on the performance [100].

3.2 Shared Memory Simulation Codes

There have been developed three versions of the simulation code intended for systems with shared memory possibly extended by an accelerator. These simulation codes are meant for ordinary users without extensive computational resources. Thus, both Windows and Linux based operation systems are supported. All codes share the common framework described in Sec 3.1.

3.2.1 Simulation code for NUMA systems

The implementation for shared memory systems composed of multicore processors with either UMA or NUMA memory organisation was first released in 2012 as a part of the k-Wave toolbox to accelerate the reference MATLAB implementation [89]. The simulation code was implemented in the C++ language complying the 2011 standard and parallelised and vectorised using the OpenMP 4.0 library. The fast Fourier transforms are calculated either by FFTW or Intel MKL where the MKL is usually faster for processors with AVX2 instruction sets such as Intel Haswell and Skylake. For NUMA architectures, the NUMA first touch strategy was implemented to preserve memory locality [6].

Depending on the exact properties of the computer used and the simulation domain sizes chosen, the compiled OpenMP code typically outperforms the reference MATLAB solution on the order of 7 to 12 times, see Fig 3.1(a). The maximum relative numeric error keeps bellow 10^{-5} which is very close to the resolution of single precision floating point values. The code is able to operate on moderate domain sizes of up to 1024^3 and while executed on large NUMA systems such as SGI UV 2000 with hundreds of computer cores, the simulation time can be reduced to a few hours.

3.2.2 Simulation code for GPU systems

If the simulation domain is small enough to fit into the GPU onboard memory, the calculation can be accelerated by a GPU. The GPU simulation code, released in 2016 [86] is written in Nvidia CUDA 7.5 with the FFT calculated by the cuFFT library [43]. All the computational work is done by the GPU. The processor only servers to load input data and progressively stream the sampled output data on disk in a non-blocking manner.

The performance of the GPU code is usually 3 to 4 times higher than the reference MATALB version accelerated by the Parallel Computing Toolbox, see Fig. 3.1(b). The numeric error is comparable to the processor code.

3.2.3 Simulation code for Intel Xeon Phi accelerators

The OpenMP code was also ported on the Intel Xeon Phi coprocessor Knights Corner and executed in the native mode. Unfortunately, the performance of the code was very poor. After a deep investigation, the Intel's FFT implementation was found to be very inefficient for the domain sizes of interest [95]. Although the Xeon Phi theoretically offers almost twice as high compute performance as a 12 core Haswel CPU, the reality is much worse (nearly 3 times slower). Another issue comprises the I/O operation. Since the HDF5 library is single-threaded, only one core can stream data to the disk. Considering the very low performance of Xeon Phi's cores and the Amdahl's law, the overall performance is almost 5 times lower than a single Haswell CPU. This practically eliminates the Xeon Phi from realistic applications.

Fortunately, the new architecture of Xeon Phi called Knights Landing eliminates the FFT issue and offers comparable performance to a dual-socket node. At the time of being, there are only pre-production versions of these chips, therefore the real measurements will be conducted at the end of 2017.

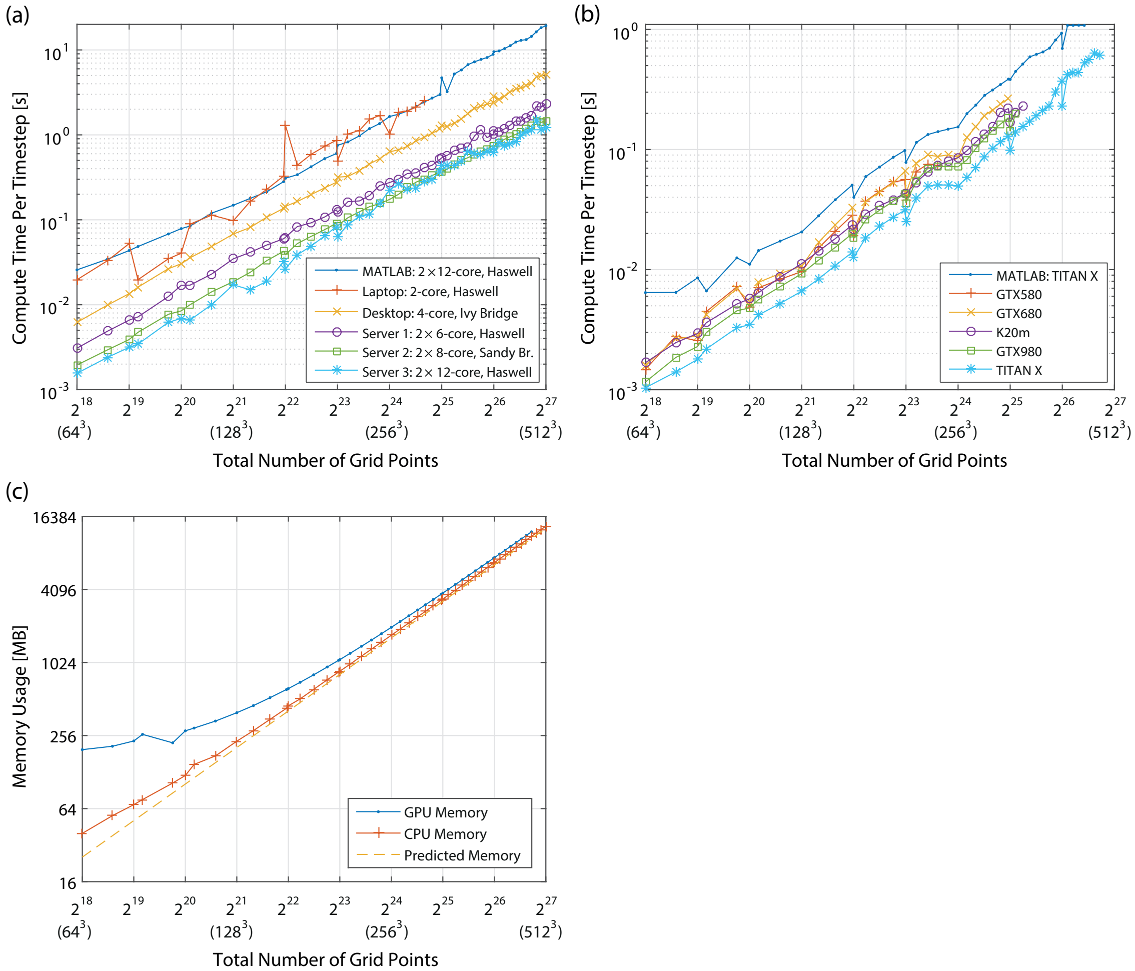


Figure 3.1: Compute times per time step for the nonlinear, heterogeneous absorbing simulations for different 3D grid sizes. (a) The comparison of the CPU code and the MATLAB version on several computers including a laptop, desktop and three servers. (b) The comparison of the GPU code and the GPU-accelerated MATLAB code on a set of high-end Graphics cards. (c) Memory consumption of the CPU and GPU code for the nonlinear, heterogeneous absorbing simulations for different 3D grid sizes.

3.2.4 Performance evaluation

The performance of the shared memory simulation codes was investigated using several systems covering a typical dual-core ultrabook, a quad-core desktop and three NUMA node configurations with 2×6, 2×8 and 2×12 cores of the Anselm and Salomon supercomputers[5]. The performance of the GPU code was investigated on a set of GPUs covering Nvidia's Fermi, Kepler and Maxwell architectures. Thorough details on the hardware, compilation process and the software stack can be found in the reference manual, see [86].

The execution times per time step for domain sizes growing from 64^3 to 512^3 grid points are shown in Fig. 3.1. The benchmarks simulate the propagation of an ultrasound wave in fully heterogeneous and absorbing medium accounting for nonlinear propagation, with the time varying acoustic pressure recorded over a single xy 2D plane. The domain sizes grow

[5] Supercomputer operated by the IT4Innovations National Supercomputing Center, www.it4i.cz

with multiples of 32 in order to preserve good memory alignment and keep the domain sizes favourable for the FFT libraries (FFT performance is very sensitive to the domain size). This is noticeable in Fig. 3.1, where several drops in execution time can be seen (execution is faster than expected) for domain sizes that are powers of two or with very low prime factors (2, 3, 5, 7). The difference between tested systems is clearly visible and can be used for execution time prediction on similar systems.

A practical lesson learnt from the performance comparison is that the optimised CPU and GPU codes can provide a significant speed-up. For example, even an ultrabook running a tuned code can be even faster than a supercomputer node running the reference Matlab version. Alternatively, a high-end Nivida GTX 980 card can provide almost twice as fast execution than a Salomon node with 24 Haswell cores while running a simulation on 320^3 grid points. This domain size can be sufficiently large for photoacoustic imaging applications or fast approximation of the focus position in the HIFU treatment (before submitting the large simulation on the cluster).

Let us note that the plots only show the execution time per a single time step. However, the number of time steps to be executed grows with the simulation domain size and the nonlinear effects according to the CFL number. For example, for diagnostic ultrasound simulations where a single reflection (echo) is only considered, the simulation has to capture the ultrasound propagation along the body diagonal and back. For example, two nonlinear simulations over domains of 128^3 and 512^3 with a CFL of 0.2 require at least 2217 and 8868 time steps, respectively.

Another important observation is that the simulation code is strongly memory bound. Therefore, good performance is subject to high memory bandwidth, large cache memories and good spatial and temporal data locality of the developed C++ and CUDA codes. For example, comparing GTX 580 and Maxwell TITAN X, there is an increase in the theoretical memory bandwidth of about 1.7× (192 vs 336 GB/s), while the raw compute power increases by a factor of 7 (1.5 TFLOPS vs 11 TFLOPS). In comparison, the experimental measurements reveal that the TITAN X is usually 1.4 - 2.1 times faster than the GTX 580, which is on the order of the increase in memory bandwidth.

Finally, the memory consumption predicted by the simple model in Eq. (3.1) is in a very close agreement with the experimentally measured values. The discrepancy for the small domain sizes is caused by the size of the compiled code (on the order of 100 MB).

3.3 Traditional Distributed Memory Simulation Codes

Once the simulation domain cannot fit into the memory of a single node or the execution time becomes prohibitive, the only solution to handle larger domains is to employ distributed clusters, i.e., partition the domain over multiple computing nodes connected together via a fast interconnection network.

A natural way to partition the simulation domain is to use a geometric decomposition along one or multiple dimensions [70]. The subdomains are then assigned to particular compute nodes and executed independently. However, to propagate the ultrasound wave through multiple domains, the compute nodes have to periodically exchange some information. Here is the place where pseudospectral methods get into troubles. The global nature of the gradient calculation, in this case using the 3D fast Fourier transform (FFT), introduces additional challenges for the development of an efficient parallel code. Specifically, while the FDTD method only requires small quantities of data to be exchanged between the processes responsible for adjacent portions of the domain, performing an FFT requires a globally synchronising all-to-all data exchange with a quadratic complexity [49, 69, 23]. This communication can become a significant bottleneck in the execution of spectral models.

Following sections present two versions of the distributed simulation code based on the multi-dimensional domain decomposition and our attempt to deploy the code on CPU and GPU based clusters. We will call this domain decomposition as global, since the gradients are calculated over the whole domain, not only within subdomains.

3.3.1 One-dimensional domain decomposition for CPU clusters

The most straight forward domain decomposition for pseudospectral methods is the 1D slab decomposition directly supported by many FFT libraries such as FFTW [49] or Intel MKL [29]. The 1D decomposition partitions the 3D simulation domain along the z-dimension into 2D slabs. The slabs are then distributed over P MPI processes, where each MPI process corresponds to one physical CPU core. The total number of processes is constrained by $P \leq Nz$, where Nz is the number of grid points in the z-dimension (and thus the number of slabs).

The gradient calculation consists of taking the forward 3D FFTs, multiplying the spectrum with the matrix of wave numbers and k-space correction term, and taking the inverse FFTs. Since all the matrix element-wise operations are local, the only communication is done as part of the FFT calculation. Each 3D FFT is executed by first performing a series of 2D FFTs in the xy-plane (i.e., on each slab) using local data. This is then followed by an all-to-all global communication to perform a $z \leftrightarrow y$ transposition. This step is necessary as the FFT can only be performed on local data (it cannot stride across data belonging to multiple processes). The global transposition is consequently followed by a series of 1D FFTs performed in the transposed z-dimension, followed by another global transposition from $y \leftrightarrow z$ to return the data to its original layout. This chain of operations is illustrated in Figure 3.2.

Examining Figure 3.2, it is apparent that the last global $y \leftrightarrow z$ transposition of the forward FFT is paired with an identical but reverse transposition immediately after the element-wise operations. As the intervening operations are independent of the order of the individual elements, it is possible to eliminate these two transpositions such that operations in the spatial frequency domain are performed in transformed space [49]. This has a significant effect on performance, with compute times reduced by 35-40% depending on the number of processes used. Note, in this case, variables defined in the spatial frequency domain must instead be partitioned in transformed space along the y-dimension, with the total number of MPI processes constrained by $P \leq min(Ny, Nz)$. To avoid having idle processes during calculations involving either regular or transposed data, the number of processes P should ideally be chosen to be a divisor of both Ny and Nz.

Since, the distributed code is expected to generate hundreds of GB of output data, the care was taken to carefully optimise the I/O subsystem. The parallel interface of the HDF5 based on the top of MPI-I/O was used as a workhorse. The settings for I/O subsystem is tuned on-the-fly to fit best the simulation being executed and the cluster the code is being run on. This is achieved by data redistribution and staging before writing to disk. The sustainable bandwidth can reach over 10 GB/s [93]. More details can be found in [32], Appendix B.1.

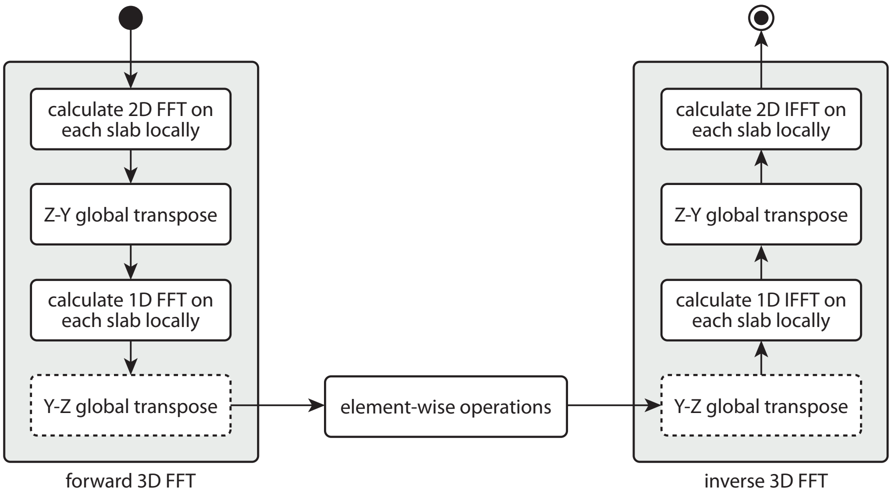

calculate 2D FFT on each slab locally
Z-Y global transpose
calculate 1D FFT on each slab locally
Y-Z global transpose
forward 3D FFT
element-wise operations
calculate 2D IFFT on each slab locally
Z-Y global transpose
calculate 1D IFFT on each slab locally
Y-Z global transpose
inverse 3D FFT

Figure 3.2: Chain of operations needed to calculate spatial gradients using the Fourier pseudospectral method. First, the 3D forward FFT is calculated, then element-wise operations within the spatial frequency domain are applied. Finally, the result is transformed back to the spatial domain using an inverse 3D FFT. The transposes depicted in the dashed boxes can be omitted if the element-wise matrix operations are performed in the transposed domain.

3.3.2 Two-dimensional hybrid domain decomposition for CPU clusters

Although the 1D decomposition allowed to run large simulations with a reasonable efficiency, it also imposed a restriction on the maximum number of processor cores that can be used $P \leq min(Ny, Nz)$. Therefore, even massive domains of 2048^3 cannot be calculated by more than 2048 computer cores. Moreover, as the communication complexity grows with a square of the number of processor cores, the efficiency for high core counts drops below acceptable values.

The 2D domain decomposition is supposed to alleviate both limitations. This approach has recently been implemented in two novel FFT libraries, PFFT [69] and P3DFFT [65]. Although this approach increases the maximum number of processor cores from N to N^2, it also requires another global communication. Nevertheless, these global transposition steps require communication only among subgroups of all compute cores. However, according to Pekurovsky [65], attention must be paid to the pencil placement over the computing cores to keep good locality and efficacy.

Unfortunately, both of these libraries adopt the pure MPI approaches which cannot exploit the shared memory within the cluster node. We therefore implemented a custom FFT wrapper on the top of the FFTW, see [59] or Appendix B.2. This wrapper first divides the domain along the first axis and distributes the slabs over MPI processes (usually one per socket or node). The slabs are further divided into pencils distributed among OpenMP threads. This approach allows to increases the number of cooperating cores by an order of magnitude while keeping the same communication complexity as the 1D decomposition (the second transposition is done in shared memory). The performance was evaluated on up to 2048 cores on Intel-based systems and 16,384 cores on a BlueGene system with a superior performance to the state-of-the-art FFT libraries, see Appendix B.2.

The whole simulation code was then adapted to support hybrid OpenMP/MPI parallelism by merging the OpenMP code with the 1D MPI code. The tests on SuperMUC with

8192 cores showed almost 4 fold decrease in the simulation time compared to 1024 cores. The efficiency is thus about 50%, which is an expected scaling of spectral methods [14]. More details can be found in [31], Appendix B.3.

3.3.3 Domain decomposition for multi-GPU systems

Motivated by the performance of the single GPU simulation code, we focused on the development of a multi-GPU implementation based on a traditional global domain decomposition. It was clear from the very beginning that the communication was going to become a serious issue. By integrating a GPU into a compute node, the compute power may increase by an order of magnitude and in the case of multi-GPU nodes even more. However, the node's interconnection bandwidth does not usually change significantly. Moreover, the communication among GPUs can cause further delays as multiple GPUs share PCI-Express links.

The first work towards the multi-GPU code was thus the survey of current multi-GPU systems, e.g., 7-GPU Tyan servers or 8-GPU nodes of the UK Emerald system[6]. The architecture evaluation published in [33], see Appendix B.4, revealed significant NUMA effects in accessing GPUs connected to different CPU socket and the congestion while using PCI-Express hubs with bandwidth reduction up to 40%.

Before implementing the whole simulation framework, we decided to develop a distributed multi-GPU 3D FFT library. Inspired by the work of Czechowski [14], we extended the Nvidia CUDA library by a layer allowing direct peer-to-peer transfers between GPUs and further accelerated the matrix transposition. However, despite all the effort and even reducing the communication going via node's main memory by almost 49%, the performance of 4 collaborating GPUs was only 12% higher than the performance of the CPU side, see [58]. In recent years, many other researchers have attempted to implement a multi-GPU FFT library, best of which is probably AccFFT [23], however, even in this case, the communication part takes almost 99% of the overall execution time when running on 128 GPUs distributed over multiple nodes.

The last hope for this branch of k-Wave is the Nvidia NVLink[7] interconnection very recently introduced in the high-end NVIDIA DGX-1 system. The NVlink interconnection makes a massive breakthrough in the multi-GPU systems by increasing the inter-GPU bandwidth from 12 GB/s up to 80 GB/s [61]. Nvidia claims that this interconnection is supposed to improve the performance of 3D FFT by a factor of 2.25. This could finally open the door for the multi-GPU implementation of the k-Wave toolbox based on the global domain decomposition.

3.3.4 Performance evaluation

The performance evaluation of 1D and 2D global domain decompositions was done on a set of realistic HIFU simulations executed on several supercomputer systems including VAYU (National Computational Infrastructure in Australia), Anselm and Salomon (IT4Innovations, Czech Republic) and SuperMUC (Leibniz Supercomputing Centre, Germany) with core counts ranging from 16 up to 8192. The benchmark set was designed to cover a wide range of domain sizes, from small simulations (256 × 256 ×256 grid points) that can be run on desktop systems, up to large-scale simulations (4096 × 2048 × 2048 grid

[6] 100 TFLOP multi-GPU system operated by Science & Engineering South, UK

[7] http://www.nvidia.com/object/nvlink.html

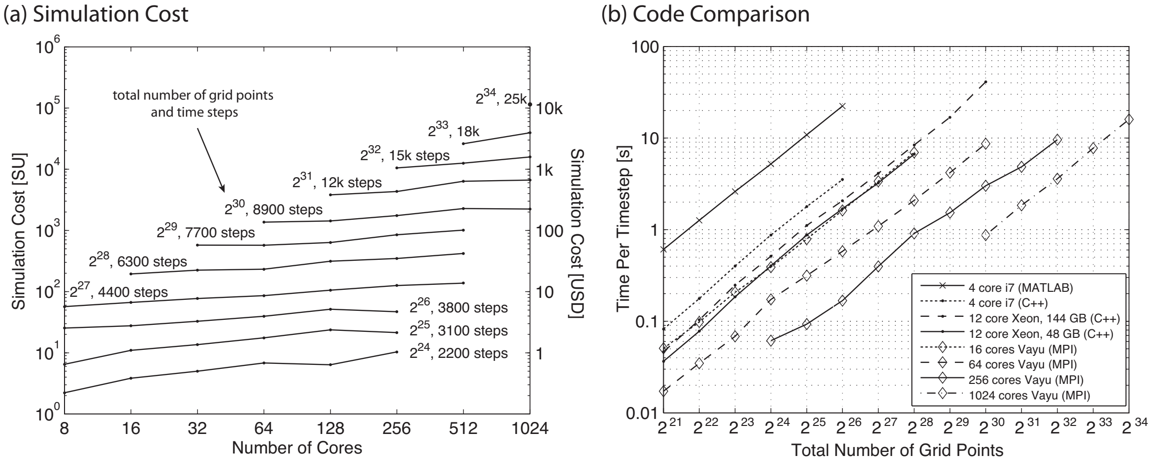


Figure 3.3: (a) The simulation cost in terms of service units (core-hours) displayed on the left y-axis and USD on the right y-axis. The number of time steps is derived from the time the wave needs to propagate along the diagonal length of the domain. (b) Execution time per time-step running different implementations of the k-space pseudospectral model on different machines.

points) that approach the limits of what is currently feasible using a supercomputer. We always investigated three main metrics: strong scaling, weak scaling and simulation cost.

The 1D domain decomposition shows good strong scaling behaviour with a speed-up of 1.7× whenever the number of cores is doubled. This means large-scale problems can be spread over increasing numbers of compute cores with a reasonable computational overhead. The overall efficiency of the parallel implementation is on the order of 60%. This is consistent with profiling data, which revealed that 30-50% of the execution time was associated with FFT communication, another 30-40% due to FFT operations and the remaining 15-20% due to element-wise matrix operations. Despite the fact that the cost of the underlying all-to-all communication step grows with the square of the number of processes, the experimental results showed relatively good weak scaling performance. The trends suggest that simulations with even larger grid sizes could be solved with reasonable efficiency using higher number of processor cores.

Extremely important for practical applications is the simulation cost. From Fig. 3.3(a), we can see that for a given grid size, the simulation cost remains fairly constant as a function of core count, with the ratio between the highest and lowest costs always less than 2. This allows to meet sharp deadlines of practical HIFU simulations with an acceptable price (here below 10,000 USD). For more details on the performance of 1D domain decomposition, see Appendix B.1.

Figure 3.3(b) shows the scale of the simulation domains that can be handled by different simulation codes. The distributed simulation code enabled to process two orders of magnitude larger domains in the same time than the shared memory code. This opens the door for new discoveries in many fields since the domain is not limited to a small part of the human body but may now contain, e.g., a whole human head. Alternatively, more precise simulation with a broad content of higher harmonics can be executed.

The behaviour of the hybrid OpenMP/MPI 2D domain decomposition was investigated in two configurations: one MPI process per node and 1 MPI process per socket. The performance measurements revealed that one process per socket is the best configuration allowing to increase the number of compute cores by 8-12. This configuration solves problems with data locality on NUMA architectures, since the OpenMP threads spawned by a process are always local to a single socket (NUMA region). The 2D domain decomposition proved to be very efficient when running on large numbers of compute cores. For example, simulation of 1024^3 grid points domain was accelerated by a factor of 4 when using 8192 cores and 2D decomposition compared to 1024 cores and 1D decomposition. On the other hand, we found out that whenever it is possible to use 1D decomposition, we can get better simulation times. This is related to the reduced network bandwidth since two MPI processes are not able to saturate the network. The 2D decomposition thus pays off when requesting more processor cores than the 1D decomposition supports, e.g., to meet time constraints. More details can be found in Appendices B.2 and B.3.

Finally, the performance of the multi-GPU simulation code was investigated. Unfortunately, although the calculation was accelerated by a factor of 2-3, the communication overhead prohibited to get a sensible performance benefit. Consequently, the multi-GPU code was not significantly faster than the CPU one, see Appendix B.5.

3.4 Modern Distributed Memory Simulation Codes

The global domain decompositions used in the k-Wave toolbox have significantly extended the range of tractable simulations and allowed to employ thousands of computer cores. However, for typical ultrasound simulations with grid sizes ranging from 512^3 to 2048^3 grid points, when distributed over more than 512 processor cores, over 50% of the execution time may be wasted waiting for data exchanges [32]. Adding further computer resources, e.g., more processor cores or GPU accelerators does not improve the performance significantly.

3.4.1 Multi-dimensional local Fourier basis domain decomposition

One way to overcome the global communication imposed by the Fourier spectral method is to use local Fourier bases as proposed by [30]. This allows the evaluation of derivatives to be split into multiple coupled subdomains, where the Fourier transforms for each subdomain are computed independently, followed by the exchange of data in an overlap or halo region. The spectral accuracy is maintained by forcing the local domains to be periodic through multiplication of the local data by a bell function. The bell function is equal to one within the physical domain, and tapers to zero within the overlap region [7].

The main benefit of the so called local domain decomposition is the reduction of the communication overhead from a quadratic to linear complexity. Instead of exchanging P^2 messages among P MPI processes, we only need to exchange $P \times 2$, 8 or 27 messages amongst direct neighbours in 1D, 2D or 3D dimensional decomposition, respectively. However, the reduced communication complexity is traded off for increased computational complexity. The local subdomain has to be wrapped up by an overlap (halo) region of such a size that meets the accuracy requirements, typically 8-16 grid points. Considering a local subdomain of 256^3 grid points extended by an overlap of 16 grid points in all three dimensions, the resulting subdomain size of 288^3 grid points comprises 1.42× more grid points. This introduces a non-negligible computational overhead. Making the local subdomain bigger would of course decrease the relative overhead, however, even 288^3 requires almost 3 GB of RAM which may be prohibitive for older GPUs and may limit the scaling. Moreover, the local domain decompositions suffer from a reduction in the numeric accuracy caused by calculating the derivatives only over the local subdomains.

3.4.2 Accuracy evaluation

The loss in numerical accuracy was thoroughly investigated using $L2$ and L_∞ error compared to a traditional domain decomposition. The tests consisted of propagating a broad band plane wave along the grid axis with the global domain divided into a given number of subdomains with a specified overlap size. The first observation was that even a overlap size of 32 is not sufficient to reach the machine precision. However, the equivalent accuracy of the PML is only on the order of 10^{-3} to 10^{-4}, even with optimized parameters [74]. Therefore, it is sufficient to maintain a similar level of accuracy for the domain decomposition. Thus an overlap of 16 grid points was chosen, which gives an error less than 10^{-4} when using two subdomains. The second observation is that the numerical error is invariant to the local domain size, thus if necessary, as small subdomains as 32^3 can be used for extreme scaling. Finally, the error increases linearly with the number of domain cuts the wave traverses, with a slope of ≈ 0.5 was examined. Thus, for typical sized problems (on the order of 2048 grid points in each dimension), up to 31 domain cuts (i.e., 32 subdomains if using 1D decomposition) can be used in each dimension with an overlap size of 16 grid points, and the error is still on the order of 10^{-3}. For 3D decomposition, this corresponds to 32,768 total subdomains (e.g., GPUs, cluster nodes/sockets). This means in practice, the level of achievable parallelism is not limited by the reduction in accuracy due to the use of local Fourier bases.

The numerical accuracy can further be tuned to minimise the necessary size of the overlap region. We are currently working on the evolutionary optimisation of the shape of the bell function. The first results have shown that we might be able to decrease the error introduced by a single domain interface by an order of magnitude. This is believed to enable us to reduce the overlap size down to 6-8 grid points even when using thousands of local subdomains. Secondly, knowing the orientation of the ultrasound transducer, we can decompose the domain primarily in the directions where there is minimal energy exchange (i.e., parallel with the direction along the wave traverses). Consequently, the wave will have to cross fewer interfaces while the domain can be still partitioned into a large number of subdomains. More details can be found in [34], see Appendix B.6.

3.4.3 Performance evaluation

The simulation code based on the local domain decomposition was developed in both CPU and GPU versions, evaluated on a standard set of benchmark simulations and compared with the traditional global domain decomposition. The CPU version was investigated on Salomon up to 6144 processor cores, while the GPU version was tested on Anselm with 16 Kepler GPUs, Emerald up to 128 Fermi GPUs, and very recently on the best European system Piz Daint up to 256 Pascal GPUs. Apart from strong scaling, weak scaling and related simulation cost, we also tested the influence of the number of dimensions the simulation domain is partitioned along and the impact of the overlap size on the simulation speed. Details can be found in [34], Appendix B.6.

Figure 3.4(b) presents the strong scaling of the multi-dimensional local domain decomposition on up to 6144 processor cores and compares it with the global domain decomposition with up to 1024 cores, see Fig. 3.4(c). From the first glance, it is evident the superiority of the local domain decomposition. The communication overhead was reduced by a great deal leading to a higher slope of the curves and better efficiency. For example, when using 128 sockets and a domain of 1024^3 grid points, the communication comprises less than 5% to the simulation time. This is a great progress compared to more than 60% for the global

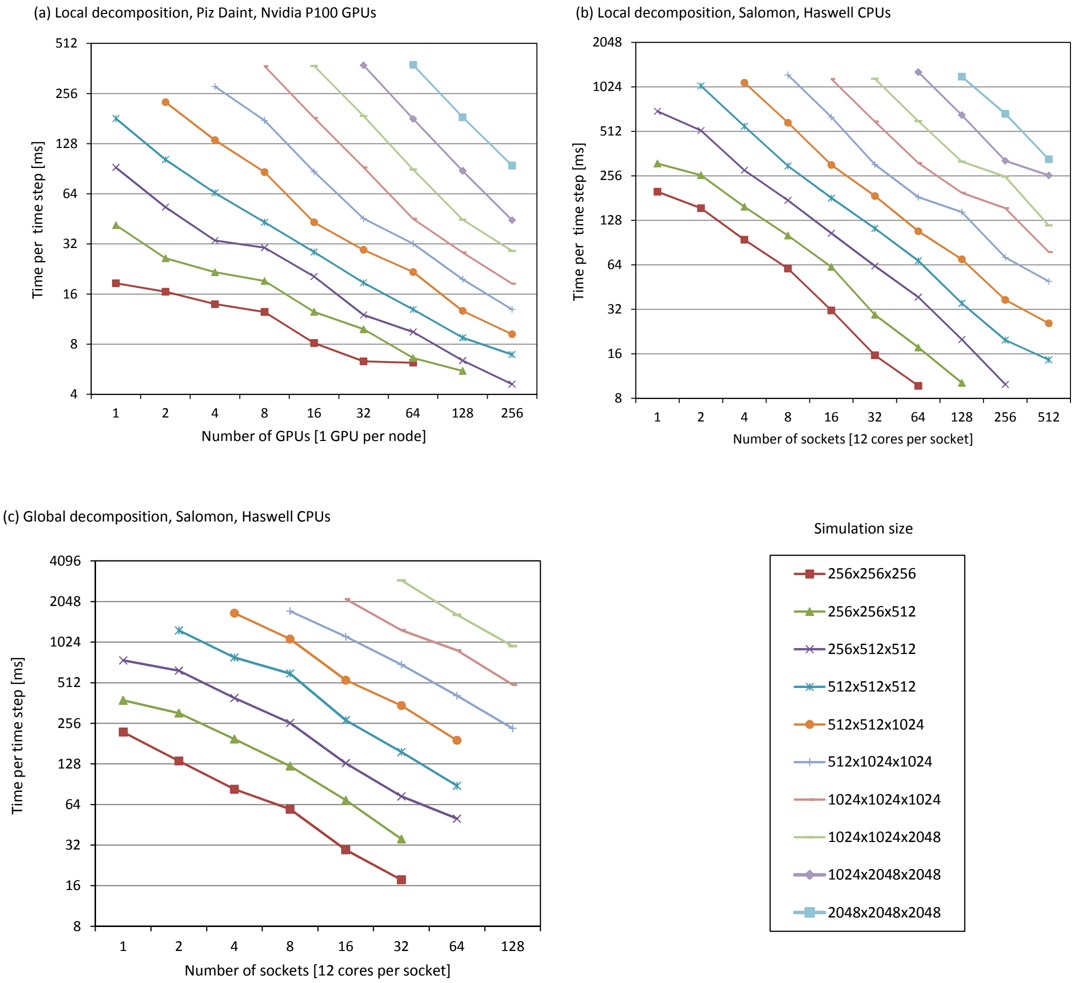


Figure 3.4: Strong scaling on typical simulation domain sizes for (a) multi-dimensional local domain decomposition using up to 256 Nvidia GPUs, (b) multi-dimensional local domain decomposition using up to 512 sockets, and (c) 1D global domain decomposition using up to 128 processor sockets on Salomon. The scaling of the local domain decomposition is limited by a minimal size of the subdomain (32^3). The global domain decomposition is limited by the size of the z dimension.

domain decomposition and the same simulation size. The execution time was consequently reduced from 0.5 s per time step to 0.2 s. Moreover, the local domain decomposition has no problems with scaling to higher numbers of cores. For example, for the same domain size and 512 sockets (6144 cores), the execution time can be reduced to 0.08 s and there is no evidence the code should not scale well even for higher number of processor cores.

The GPU results are stunning, see Fig 3.4(a). Here, the additional work hidden in calculating the overlaps virtually disappears thanks to much higher computational performance. The parallel efficiency for bigger domain is almost 100%. Looking at the domain size of 1024^3, only 8 graphics cards can beat 1024 processor cores running the global domain decomposition, which is incredible. However, what about taking 256 GPUs? We can then simulate a single time step of such a domain in 0.018 s, almost five times faster than using 512 sockets (256 nodes of Salomon). However, the true performance reveals when running large simulation domains. What used to be intractable with the global domain decomposition becomes a standard with the GPU simulation code. Employing thousands of GPUs, we will be very soon able to solve domains nobody would even dream of.

3.5 Related Work

The k-Wave toolbox now incorporates several versions of the optimised CPU and GPU codes allowing to routinely run nonlinear, absorbing simulations in heterogeneous media on domains of 2048^3. Although, the simulation code is not limited by any particular size, the simulation cost can very quickly reach 10,000 USD, such as simulations presented in Chapter 4.

Apart from k-Wave, which is a well established toolbox with more than 8,000 registered users, there are several other research teams working in this area. For example, Pinton et al. presented a solution to the Westervelt equation using a second-order in time, fourth-order in space finite difference scheme [68]. This was used to investigate the effects of non-linear wave propagation on HIFU therapy in the brain [67]. Simulations were run on 112 cores of a distributed cluster with grid sizes and run times on the order of 800^3 and 32 hours. Okita et al. used a similar model to study HIFU focusing through the skull, using 128 cores of a distributed cluster with grid sizes and run times on the order of $800 \times 600 \times 600$ and 2 hours [63]. They also demonstrated excellent weak-scaling results for a benchmark using up to 3.45×10^{11} grid points distributed across 98,304 cores on the K computer[8][62]. In another study, Pulkkinen et al. used a hybrid FDTD model to study transcranial focused ultrasound using grid sizes up to $1338 \times 1363 \times 1120$ running on 96 cores of a distributed cluster with compute times on the order of 40 hours [71].

SimSonic is a software suite for the simulation of ultrasound propagation based on FDTD computations of the elastodynamic equations [54]. One of its application was the study of the propagation phenomena of ultrasonic waves in prototypes cylindrically shaped implants and to investigate the sensitivity of their ultrasonic response to the surrounding bone biomechanical properties. Unfortunately, the toolbox only supports shared memory systems and is thus unable to process comparable simulation domains as k-Wave.

Finally, the KZKTexas is a widely used time-domain computer code developed at the University of Texas at Austin to model axisymmetric sound beams in fluids. Since being axisymmetric, the code is not able to process fully heterogeneous 3D domains [11].

All the mentioned codes are based on finite difference methods that require much denser grid size. Consequently the physical volume is much smaller compared to k-Wave. Let us Compare the biggest reported domain size of $(1338 \times 1363 \times 1120)$ presented in [71] with our production run of $1536 \times 1024 \times 2048$, which is more than 1.57 times bigger. While Pulkkinen needed almost 40 hours, we can calculate the output in 53 hours using 512 processor cores on Salomon with global domain decomposition, 5.6 hours using 6144 processor cores on Salomon, 9.5 hours using 128 Fermi GPUs on Emerald, and in less than 1 hour using 128 Pascal GPUs. In order to run a simulation similar to that one executed on the K Computer, we would need at least 37 TB of memory to store the simulation data. This would require a Top 10 class supercomputer such as Titan and at least 4096 GPUs. However, the simulation price would significantly exceed 100,000 USD.

[8] supercomputer installed at the RIKEN Advanced Institute for Computational Science in Japan

Apart from FDTD methods, Jensen at al. has been developing the Field II toolbox [38] based on the concept of spatial impulse responses. Field II is a MATLAB toolbox which can be used for linear ultrasound simulations only. The toolbox also includes an optimised shared memory simulation code [39]. However, the performance of the proposed method is far behind k-Wave. Field's execution times for even 2D domains of 20 × 50 mm are on the order of minutes.

Morales at al. released the MPARD, a high-frequency wave-based acoustic solver for very large compute clusters [56]. This toolbox is based on a novel scalable method for dividing acoustic field computations specifically for large-scale distributed memory clusters using parallel Adaptive Rectangular Decomposition (ARD). The ARD method requires a very low number of grid points per wave length, typically around 3. This parallel algorithm makes it possible to compute the sound pressure field for high frequencies in large environments that are thousands of cubic meters in volume. The performance of this system on large clusters with 16,000 cores on homogeneous indoor and outdoor allowed to run benchmarks up to 10 kHz with more than 4 billions of grid points. However, this toolbox does not account for absorption and thus cannot be used in HIFU simulations.

3.5.1 Contributions and future work

The efficient parallel implementations of the k-Wave toolbox have contributed to both the field of realistic ultrasound simulation and the field of high performance computing. Let me mention a few examples:

1. The large scale simulation codes have opened the door to disruptive science by making extremely large simulations viable with the help of current supercomputers. During the last decade we have been able to extend the simulation domain size by a factor of 250 while keeping the simulation time on the order of tens of hours.

2. The implementation of the local Fourier basis decomposition along with the performance and accuracy evaluation has allowed to efficiently use GPUs in pseduspectral methods for the first time. The reached parallel efficiency is almost perfect reducing the simulation time by a factor of 5 compared to a corresponding number processors nodes. Moreover, it allows to employ significantly more processor or GPUs nodes than traditional domain decompositions and decrease the simulation time by at least an order of magnitude. This work is expected to become a widely adopted approach in pseudospectral methods.

3. The simulation codes for desktop systems and mainstream GPUs have supported a lot of common users including researchers, academics and students without the possibility to use large supercomputing facilities.

4. The novel hybrid OpenMP/MPI library for the 3D fast Fourier transform improved the state of the art of 3D FFT calculation and can be adopted by other developers from various fields.

5. The development of the custom bell functions reduced the necessary overlap size by more than 50%. This decreased the size of the overlap region to reach an appropriate numerical accuracy. As a consequence, the communication and computation overhead could be significantly reduced.

The development of large ultrasound simulation codes has also risen a lot of questions to be addressed in the future:

1. The size of output data becomes extreme, reaching TBs. Therefore, novel on-the-fly compression mechanism are being looked for to reduce the output files and decrease the time spent in I/O.

2. At the time being, the simulation code only uses either CPUs or GPUs. However, this implies some resources are wasted. The future work is towards full employment of heterogeneous architectures.

3. The load balancing on heterogeneous architectures becomes a rapidly increasing issue when talking about heterogeneous architectures with dynamic set-up (clock frequency, shared inter-node and intra-node network, etc.).

4. When the domain contains both fluid and elastic tissue, it would be beneficial to decompose the domain according to the medium type and run the best optimised models for each subdomain. This leads to a complex model coupling problem.

5. The clinical applications of the model based treatment planning requires a complex tool chain composed of the user interface to set-up the simulation, send it off for processing on the cluster, collect the outputs, display them back to clinicians, and upload the treatment plan into a HIFU machine after approval. Although the work on this tool chain has begun, a few years of development is still ahead us.

Chapter 4

Applications of Ultrasound Modelling in Cancer Treatment

This chapter presents a small subset of practical applications, the k-Wave toolbox has been used for and our team has participate in. The efficient implementation, especially the large scale distributed simulation codes significantly expanded the range of applications. The scientific outcomes will be confronted with the HPC aspects, primarily the computation resources used and the price of the simulation in terms of core-hours or USD.

4.1 Focused Ultrasound Waves in Heterogeneous Tissue

When focused ultrasound waves propagate through the human body, the combined effects of nonlinearity, absorption, refraction, and scattering can significantly alter the position and shape of the focal volume compared to the response in water. This can affect the desired clinical outcome, and lead to adverse events such as near-field heating and skin burns. In this study, these wave phenomena are discussed and their respective importance is demonstrated through numerical simulations in the k-Wave toolbox. Both homogeneous and heterogeneous media are considered, including sonications of the kidney and liver through the muscles surrounding the abdominal cavity and ribs cage.

First, this study revealed the increase in the volume rate of heat deposition due to nonlinear effects is always less than 10%. Second, the optimal driving frequency for abdominal targets is in the range from 500 kHz to 1.3 MHz, depending on the depth of the target and the output parameter that is maximized. Third, the reduction in focal intensity due to absorption and refraction are of the same order, and together can reduce the focal intensity by a factor of ten. In particular, the skin and muscle layers can cause significant aberrations to the ultrasound beam compared to water. Finally, the effects of reflection due to soft-tissue interfaces are negligible. However, the inclusion of bone within the beam path can reduce the focal intensity by more than 70%.

Simulations were executed using the distributed version of k-Wave running on the Salomon supercomputer. For all simulations, the domain size (which determines the region of interest) was set to $165 \times 165 \times 220$ mm. Three grid resolutions were used, depending on the required maximum frequency supported by the spatial grid. While the smallest simulation domain comprised of $1152 \times 1152 \times 1536$ grid points and 10,137 time steps, the largest one spread over $3072 \times 3072 \times 4096$ grid points being simulated over 27,139 time steps. This is by far the largest ultrasound simulation executed to date. Using 1024 computer cores and 3.1 TB of memory, the code run for 8 days and 21 hours consuming over 218 thousand core-hours (approx 20,000 USD). More details can be found in [88], see Appendix C.1.

4.2 Salvage HIFU Treatment in the Prostate

High intensity focused ultrasound (HIFU) provides a non-invasive salvage treatment option for patients with recurrence after external beam radiation therapy (EBRT). As part of EBRT the prostate is frequently implanted with permanent fiducial markers. To date, the impact of these markers on subsequent HIFU treatment is unknown. The objective of this work was to systematically investigate, using computational simulations, how these fiducial markers affect the delivery of HIFU treatment.

We conducted a series of simulations modelling the propagation of ultrasound pressure waves in the prostate with a single spherical or cylindrical gold marker at different positions and orientations. For each marker configuration, a set of metrics (spatial-peak temporal-average intensity, focus shift, focal volume) was evaluated to quantify the distortion introduced at the focus. An analytical model was also developed describing the marker effect on the intensity at the focus. The model was used to examine the marker's impact in a clinical setting through case studies.

The simulations show that the presence of the marker in the pre-focal region causes reflections which induce a decrease in the focal intensity and focal volume, and a shift of the maximum pressure point away from the transducer's focus. These effects depend on the shape and orientation of the marker and become more pronounced as its distance from the transducer's focus decreases, with the distortion introduced by the marker greatly increasing when placed within 5 mm of the focus. The analytical model approximates the marker's effect and can be used as an alternative method to the computationally intensive and time consuming simulations for quickly estimating the intensity at the focus. A retrospective review of a small patient cohort selected for focal HIFU after failed EBRT indicates that the presence of the marker may affect HIFU treatment delivery.

The simulations were performed using the distributed version of k-Wave on Salomon. The actual hardware utilized for each simulation comprised either of 6 or 9 nodes[1]. Since 143 different marker positions had to be examined, we used low numbers of nodes to improve overall throughput and decrease the simulation cost as much as possible. Even though, the compute allocation necessary for this experimental study reached 5 million core-hours (almost 500,000 USD) with about 63 TB of generated data. More details can be found in [22], Appendix C.2.

4.3 Ultrasound Focus Placement in the Kidney

Kidney cancer is a severe disease which can be treated non-invasively using high-intensity focused ultrasound (HIFU) therapy. However, tissue in front of the transducer and the deep location of kidney can cause significant losses to the efficiency of the treatment. The effect of attenuation, refraction and reflection due to different tissue types on HIFU therapy of the kidney was studied using a nonlinear ultrasound simulation model. The geometry of the tissue was derived from a computed tomography (CT) dataset of a patient which had been

[1] Every node with two Intel Xeon E5-2680v3 processors, each equipped with 24 cores and 128 GB RAM, interconnected by a 7D Enhanced hypercube Infiniband network.

segmented for water, bone, soft tissue, fat and kidney. The combined effect of inhomogeneous attenuation and sound speed was found to result in an 11.0 dB drop in spatial peak temporal average (SPTA) intensity in the kidney compared to pure water. The simulation without refraction effects showed a 6.3 dB decrease indicating that both attenuation and refraction contribute to the loss in focal intensity. The losses due to reflections at soft tissue interfaces were less than 0.1 dB. Focal point shifting due to refraction effects resulted in -1.3, 2.6 and 1.3 mm displacements in x-, y- and z-directions respectively. Furthermore, focal point splitting into several smaller subvolumes was observed. The total volume of the secondary focal points was approximately 46% of the largest primary focal point. This could potentially lead to undesired heating outside the target location and longer therapy times.

Before performing the actual simulations, several convergence simulations were conducted in order to find the optimal grid size and temporal resolution. The computational grid consisted of $1200 \times 1200 \times 1200$ grid points (i.e., 22.2 cm × 22.2 cm × 22.2 cm) giving a spatial resolution of 185 μm which supported nonlinear harmonic frequencies up to 4 MHz. Perfectly matched layers (PML) were used on the edges of the grid. The simulation time duration was set to 260 μs with a temporal resolution of 8.15 ns giving a total of 31,876 time steps per simulation. The simulations were run using 400 computing cores for approximately 50 hours per simulation and requiring 200 GB of memory. One such a simulation used 20,000 core hours, which constitutes a price of 2,000 USD. The simulations were conducted using the computing facilities provided by advanced research computing (ARC) at the University of Oxford.

More information can be found in [78], see Appendix C.3. An extended study including also the effect of thermal heating has been submitted into IEEE Transactions on Biomedical Engineering and will be accepted after minor revision.

4.4 Transcranial Ultrasonic Neurostimulation

Non-invasive, focal neurostimulation with ultrasound is a potentially powerful neuroscientific tool that requires effective transcranial focusing of ultrasound to develop. Time-reversal (TR) focusing using numerical simulations of transcranial ultrasound propagation can correct for the effect of the skull, but relies on accurate simulations. Here, focusing requirements for ultrasonic neurostimulation are established through a review of previously employed ultrasonic parameters, and consideration of deep brain targets. The specific limitations of finite-difference time domain (FDTD) and k-space corrected pseudospectral time domain (PSTD) schemes are tested numerically to establish the spatial points per wavelength and temporal points per period needed to achieve the desired accuracy while minimizing the computational burden.

The results presented in this study are primarily relevant to the simulation of transcranial ultrasound propagation for TR targeting of deep brain structures with finely controlled ultra-sound for the purposes of neurostimulation. However, the criteria and simulations presented are also relevant to alternative low-intensity, transcranial ultrasonic therapies such as opening the blood-brain barrier with ultrasound, as well as existing transcranial HIFU ablation therapies. Use of appropriately discretized simulations will ensure accurate targeting and effective therapy as the field of ultrasonic neurostimulation develops.

The 2D simulations were carried on a single Nvidia Titan X GPU with domain size of 3780^2 over 258,462 time steps with a total simulation time of 10.6 hours. 3D simulations were executed on a large NUMA server SGI UV1, which is a part of on the Salomon supercomputer. The 3D domains comprised of 1024^3 grid points over 22,718 time steps, however massive input pressure sources was used for forward simulations (on the order of 100 GB). When executed by a single node of the SGI UV1, the simulation took 112.3 hours. When the whole machine with 112 cores was employed, the execution time dropped down to 9.7 hours. More details can be found in [75], see Appendix C.4.

4.5 Photoacoustic Imaging

Reconstructing images from measured time domain signals is an essential step in tomography-mode photoacoustic imaging. However, in practice, there are many complicating factors that make it difficult to obtain high-resolution images. These include incomplete or under-sampled data, filtering effects, acoustic and optical attenuation, and uncertainties in the material parameters. Here, the processing and image reconstruction steps implemented in the k-Wave toolbox are discussed. These include correction for acoustic and optical attenuation, spatial resampling, material parameter selection, image reconstruction, and log compression. The effect of each of these steps is demonstrated using a representative *in vivo* dataset.

Techniques discussed in this study such as automatic sound speed selection and acoustic attenuation compensation are fast and easy to apply, and noticeably improve the reconstructed images. Using the latest hardware and software advances, three-dimensional time reversal image reconstruction can also be performed on relatively large datasets in under 30 seconds using a single GPU. More details can be found in [87], see Appendix C.5.

Chapter 5

Conclusions

This thesis has provided an overview of the research activities I have been involved in for about 6 years. During this period, I have been developing the k-Wave acoustic toolbox[1]. Since the first beta release in 2010, k-Wave has rapidly become the de facto standard software in the field, with almost 8000 registered users in 60 countries (from both academia and industry). The toolbox now underpins a wide range of international research in ultrasound and photoacoustics, ranging from the reconstruction of clinical photoacoustic images to fundamental studies into the design of ultrasound transducers. The ultimate goal of my research is to create a robust toolbox that could be used by clinicians for HIFU treatment planning and photoacoustic imaging on everyday basis. I feel we are on the right way.

My contribution to the field of the ultrasound modelling in large tissue realistic domains can be summarised as follows: (1) The development of the acoustic model accounting for combined effect of nonlinearity, heterogeneity and abstraction. (2) The design, development and deployment of various simulation codes targeting different platforms ranging for laptops up to top supercomputers. (3) Novel decomposition techniques for spectral methods which improves the performance by a negligible reduction of the accuracy. (4) The support of several research teams in their clinical studies on the prostate, kidney, liver, or in the brain.

There are many open research questions for the future work. In the very near future, we are about to run the first production simulations on the best Europe supercomputer Piz Daint[2] and later on the US Titan[3]. The compute allocations we strive for are in millions of node hours and will support many application studies.

For the short term, we have a couple of open projects such as dynamic load balancing on heterogeneous architecture that is supposed to distribute workload over both CPUs and GPUs. We are also working on a model coupling framework that would allow us to divide the domain into fluid and elastic parts and use a specific model whenever possible to make the simulation faster and cheaper. This framework will also be used to tightly couple the ultrasound model with the thermal and tissue ones. By doing so, we could immediately project the changes in the tissue structure due to localised heating to the ultrasound wave propagation.

For a longer term, we would like investigate ParaReal techniques for the decomposition in time which has the potential to distribute the work over millions of compute cores.

[1] http://www.k-wave.org

[2] Swiss National Supercomputing Centre

[3] Oak Ridge National Laboratory

Bibliography

[1] Al-Bataineh, O.; Jenne, J.; Huber, P.: Clinical and future applications of high intensity focused ultrasound in cancer. *Cancer Treatment Reviews*. vol. 38, no. 5. 2012: pp. 346–353. ISSN 0305-7372. doi:10.1016/j.ctrv.2011.08.004.

[2] Amin, V.; Wu, L.; Long, T.; et al.: Therapy planning and monitoring of tissue ablation by high intensity focused ultrasound (HIFU) using imaging and simulation. In *Conference Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2008. ISBN 1557-170X. ISSN 1557-170X. page 4471. doi:10.1109/IEMBS.2008.4650204.

[3] Arora, D.; Skliar, M.; Roemer, R.: Minimum-Time Thermal Dose Control of Thermal Therapies. *IEEE Transactions on Biomedical Engineering*. vol. 52, no. 2. feb 2005: pp. 191–200. ISSN 0018-9294. doi:10.1109/TBME.2004.840471.

[4] Aver'yanov, M. V.; Khokhlova, V. A.; Sapozhnikov, O. A.; et al.: Parabolic equation for nonlinear acoustic wave propagation in inhomogeneous moving media. *Acoustical Physics*. vol. 52, no. 6. dec 2006: pp. 623–632. ISSN 1063-7710. doi:10.1134/S1063771006060017.

[5] Blackstock, D. T.: Generalized Burgers equation for plane waves. *The Journal of the Acoustical Society of America*. vol. 77, no. 6. jun 1985: pp. 2050–2053. ISSN 0001-4966. doi:10.1121/1.391778.

[6] Bolosky, W.; Fitzgerald, R.; Scott, M.: Simple but effective techniques for NUMA memory management. *ACM SIGOPS Operating Systems Review*. vol. 23, no. 5. nov 1989: pp. 19–31. ISSN 0163-5980. doi:10.1145/74851.74854.

[7] Boyd, J. P.: Asymptotic Fourier Coefficients for a C Bell (Smoothed-"Top-Hat") & the Fourier Extension Problem. *Journal of Scientific Computing*. vol. 29, no. 1. oct 2006: pp. 1–24. ISSN 0885-7474. doi:10.1007/s10915-005-9010-7.

[8] Cameron, J.: Physical Properties of Tissue. A Comprehensive Reference Book , edited by Francis A. Duck. *Medical Physics*. vol. 18, no. 4. jul 1991: pp. 834–834. ISSN 0094-2405. doi:10.1118/1.596734.

[9] Caputo, M.; Carcione, J. M.; Cavallini, F.: Wave Simulation in Biologic Media Based on the Kelvin-Voigt Fractional-Derivative Stress-Strain Relation. *Ultrasound in Medicine & Biology*. vol. 37, no. 6. jun 2011: pp. 996–1004. ISSN 0301-5629. doi:10.1016/j.ultrasmedbio.2011.03.009.

[10] Chen, W.; Holm, S.: Fractional Laplacian time-space models for linear and nonlinear lossy media exhibiting arbitrary frequency power-law dependency. *The Journal of*

the Acoustical Society of America. vol. 115, no. 4. apr 2004: pp. 1424–1430. ISSN 0001-4966. doi:10.1121/1.1646399.

[11] Cleveland, R.: Time-domain modeling of finite-amplitude sound in relaxing fluids. *The Journal of the Acoustical Society of America*. vol. 99, no. 6. 1996: page 3312. ISSN 0001-4966. doi:10.1121/1.414983.

[12] Connor, C. W.; Hynynen, K.: Bio-acoustic thermal lensing and nonlinear propagation in focused ultrasound surgery using large focal spots: a parametric study. *Physics in Medicine and Biology*. vol. 47, no. 11. 2002: pp. 1911–1928. ISSN 0031-9155. doi:10.1088/0031-9155/47/11/306.

[13] Coulouvrat, F.: New equations for nonlinear acoustics in a low Mach number and weakly heterogeneous atmosphere. *Wave Motion*. vol. 49, no. 1. 2012: pp. 50–63. ISSN 0165-2125. doi:10.1016/j.wavemoti.2011.07.002.

[14] Czechowski, K.; Battaglino, C.; McClanahan, C.; et al.: On the communication complexity of 3D FFTs and its implications for Exascale. In *ICS &apos;12: Proceedings of the 26th ACM international conference on Supercomputing*. New York, New York, USA: ACM Press. 2012. ISBN 9781450313162. page 205. doi:10.1145/2304576.2304604.

[15] Demi, L.; van Dongen, K. W. a.; Verweij, M. D.: A contrast source method for nonlinear acoustic wave fields in media with spatially inhomogeneous attenuation. *The Journal of the Acoustical Society of America*. vol. 129, no. 3. 2011: pp. 1221–1230. ISSN 0001-4966. doi:10.1121/1.3543986.

[16] Demi, L.; Treeby, B. E.; Verweij, M. D.: Comparison between two different full-wave methods for the computation of nonlinear ultrasound fields in inhomogeneous and attenuating tissue. In *IEEE International Ultrasonics Symposium, IUS*. IEEE. sep 2014. ISBN 9781479970490. ISSN 1948-5727. pp. 1464–1467. doi:10.1109/ULTSYM.2014.0362.

[17] Demi, L.; Verweij, M.: *Nonlinear Acoustics*. Acoustical Society of America. 2014. ISBN 9780444536327. 387–399 pp.. doi:10.1016/B978-0-444-53632-7.00218-5.

[18] Dogra, V. S.; Zhang, M.; Bhatt, S.: High-Intensity Focused Ultrasound (HIFU) Therapy Applications. 2009. doi:10.1016/j.cult.2009.10.005.

[19] Dubinsky, T. J.; Cuevas, C.; Dighe, M. K.; et al.: High-intensity focused ultrasound: Current potential and oncologic applications. *American Journal of Roentgenology*. vol. 190, no. 1. jan 2008: pp. 191–199. ISSN 0361-803X. doi:10.2214/AJR.07.2671.

[20] Filoux, E.; Levassort, F.; Calle, S.; et al.: Combined pseudospectral and finite-difference time-domain methods for ultrasonic transducers modeling. In *ICU Proceedings*. Vienna University of Technology. 2007. pp. 1–4. doi:10.3728/ICUltrasonics.2007.Vienna.1227_filoux.

[21] Gao, J.; Cochran, S.; Huang, Z.: Ultrasound beam distortion and pressure reduction in transcostal focused ultrasound surgery. *Applied Acoustics*. vol. 76. 2014: pp. 337–345. ISSN 0003-682X. doi:10.1016/j.apacoust.2013.06.003.

[22] Georgiou, P. S.; Jaros, J.; Payne, H.; et al.: Beam distortion due to gold fiducial markers during salvage high-intensity focused ultrasound in the prostate. *Medical Physics*. vol. 44, no. 2. feb 2017: pp. 679–693. ISSN 0094-2405. doi:10.1002/mp.12044.

[23] Gholami, A.; Hill, J.; Malhotra, D.; et al.: AccFFT: A library for distributed-memory FFT on CPU and GPU architectures. *CoCR*. vol. abs/1506.0. jun 2015. 1506.07933.

[24] Hallaj, I. M.; Cleveland, R. O.; Hynynen, K.: Simulations of the thermo-acoustic lens effect during focused ultrasound surgery. *The Journal of the Acoustical Society of America*. vol. 109, no. 5 Pt 1. 2001: pp. 2245–2253. ISSN 0001-4966. doi:10.1121/1.1360239.

[25] Hamilton, M.; Morfey, C.: Model Equations. In *Nonlinear Acoustics*, vol. 2, edited by M. F. Hamilton; D. T. Blackstock. Acoustical Society of America. 1998. ISBN 0-97440-6759. pp. 41–63.

[26] Harstell, W. F.; Scott, C. B.; Bruner, D. W.; et al.: Randomized trial of short-versus long-course radiotherapy for palliation of painful bone metastases. *Journal of the National Cancer Institute*. vol. 97, no. 11. 2005: pp. 798–804. ISSN 1460-2105. doi:10.1093/jnci/dji139.

[27] Howison, M.; Koziol, Q.; Knaak, D.; et al.: Tuning HDF5 for Lustre file systems. *IASDS '10 Proceedings of the Workshop on Interfaces and Abstractions for Scientific Data Storage*. vol. 5. 2012.

[28] Huijssen, J.; Verweij, M. D.: An iterative method for the computation of nonlinear, wide-angle, pulsed acoustic fields of medical diagnostic transducers. *The Journal of the Acoustical Society of America*. vol. 127, no. 1. jan 2010: pp. 33–44. ISSN 0001-4966. doi:10.1121/1.3268599.

[29] Intel Corporation: *Math Kernel Library 11.3 Developer Reference*. Intel Corporation. 2015. 2496 pp.

[30] Israeli, M.; Vozovoi, L.; Averbuch, A.: Spectral multidomain technique with Local Fourier Basis. *Journal of Scientific Computing*. vol. 8, no. 2. jun 1993: pp. 135–149. ISSN 0885-7474. doi:10.1007/BF01060869.

[31] Jaros, J.; Nikl, V.; Treeby, B. E.: Large-scale Ultrasound Simulations Using the Hybrid OpenMP / MPI Decomposition. In *Exascale Applications and Software Conference*. Edinburgh: Association for Computing Machinery. 2015. ISBN 978-3-319-14895-3. pp. 115–119.

[32] Jaros, J.; Rendell, A. P.; Treeby, B. E.: Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound. *International Journal of High Performance Computing Applications*. vol. 30, no. 2. may 2015: pp. 1094342015581024–. ISSN 1094-3420. doi:10.1177/1094342015581024. arXiv:1408.4675v1.

[33] Jaros, J.; Treeby, B. E.; Rendell, A. P.: Use of multiple GPUs on shared memory multiprocessors for ultrasound propagation simulations. In *Conferences in Research and Practice in Information Technology Series*, vol. 127, edited by J. Chen; R. Ranjan. Melbourne, Australia: ACS. 2012. ISBN 9781921770081. ISSN 1445-1336. pp. 43–52.

[34] Jaros, J.; Vaverka, F.; Treeby, B. E.: Spectral Domain Decomposition Using Local Fourier Basis: Application to Ultrasound Simulation on a Cluster of GPUs. *Supercomputing Frontiers and Innovations*. vol. 3, no. 3. nov 2016: pp. 39–54. ISSN 2313-8734. doi:10.14529/jsfi160305.

[35] Jeffers, J.; Reinders, J.: *Intel Xeon Phi Coprocessor High Performance Programming*. 1. Elsevier, Inc.. 2013. 409 pp.

[36] Jenne, J. W.; Preusser, T.; Günther, M.: High-intensity focused ultrasound: Principles, therapy guidance, simulations and applications. *Zeitschrift fur Medizinische Physik*. vol. 22, no. 4. dec 2012: pp. 311–322. ISSN 0939-3889. doi:10.1016/j.zemedi.2012.07.001.

[37] Jensen, J. A.: A model for the propagation and scattering of ultrasound in tissue. *The Journal of the Acoustical Society of America*. vol. 89, no. 1. jan 1991: pp. 182–190. ISSN 0001-4966. doi:10.1121/1.400497.

[38] Jensen, J. A.: FIELD: A Program for Simulating Ultrasound Systems. *Medical and Biological Engineering and Computing*. vol. 34, no. SUPPL. 1. 1996: pp. 351–352. ISSN 0140-0118.

[39] Jensen, J. A.: A multi-threaded version of Field II. In *IEEE International Ultrasonics Symposium, IUS*. IEEE. sep 2014. ISBN 9781479970490. ISSN 1948-5727. pp. 2229–2232. doi:10.1109/ULTSYM.2014.0555.

[40] Jing, Y.; Cleveland, R. O.: Modeling the propagation of nonlinear three-dimensional acoustic beams in inhomogeneous media. *The Journal of the Acoustical Society of America*. vol. 122, no. 3. 2007: page 1352. ISSN 1520-8524. doi:10.1121/1.2767420.

[41] Jing, Y.; Meral, C.; Clement, G.: Time-reversal transcranial ultrasound beam focusing using a k-space method. *Physics in medicine and biology*. vol. 57. 2012: pp. 901–917. ISSN 1361-6560. doi:10.1088/0031-9155/57/4/901. NIHMS150003.

[42] Jing, Y.; Wang, T.; Clement, G. T.: A k-space method for moderately nonlinear wave propagation. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*. vol. 59, no. 8. 2012: pp. 1664–1673. ISSN 0885-3010. doi:10.1109/TUFFC.2012.2372. 1105.2210.

[43] Jodra, J. L.; Gurrutxaga, I.; Muguerza, J.: A Study of Memory Consumption and Execution Performance of the cuFFT Library. *Proceedings - 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2015*. 2016: pp. 323–327. doi:10.1109/3PGCIC.2015.66.

[44] Kennedy, J. E.; Ter Haar, G. R.; Cranston, D.: High intensity focused ultrasound: surgery of the future? *The British journal of radiology*. vol. 76, no. 909. sep 2003: pp. 590–599. ISSN 0007-1285. doi:10.1259/bjr/17150274.

[45] Kerbyson, D. J.; Barker, K. J.; Vishnu, A.; et al.: A performance comparison of current HPC systems: Blue Gene/Q, Cray XE6 and InfiniBand systems. *Future Generation Computer Systems*. vol. 30, no. 1. 2014: pp. 291–304. ISSN 0167-739X. doi:10.1016/j.future.2013.06.019.

[46] Khokhlova, V. A.; Bessonova, O. V.; Soneson, J. E.; et al.: Bandwidth limitations in characterization of high intensity focused ultrasound fields in the presence of shocks. In *AIP Conference Proceedings*, vol. 1215. 2010. ISBN 9780735407589. ISSN 0094-243X. pp. 363–366. doi:10.1063/1.3367181.

[47] Koranne, S.: *Handbook of Open Source Tools*. Boston, MA: Springer US. 2010. ISBN 9781441977182. 484 pp.. doi:10.1007/978-1-4419-7719-9.

[48] Lepock, J. R.: Cellular effects of hyperthermia: relevance to the minimum dose for thermal damage. *Int J Hyperthermia*. vol. 19, no. 3. 2003: pp. 252–266. ISSN 0265-6736. doi:10.1080/0265673031000065042.

[49] Li, Q.; Jin, T.; Fu, Y.; et al.: The design and implementation of a topic-driven crawler. *Proceedings - Workshop on Intelligent Information Technology Application, IITA 2007*. vol. 93, no. 2. 2007: pp. 153–156. ISSN 1937-4771. doi:10.1109/IITA.2007.87. 1509.07821.

[50] Liu, Q. H.: The PSTD algorithm: A time-domain method requiring only two cells per wavelength. *Microwave and Optical Technology Letters*. vol. 15, no. 3. jun 1997: pp. 158–165. ISSN 0895-2477. doi:10.1002/(SICI)1098-2760(19970620)15:3<158::AID-MOP11>3.0.CO;2-3.

[51] Liu, Q. H.: The pseudospectral time-domain (PSTD) algorithm for acoustic waves in absorptive media. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*. vol. 45, no. 4. jul 1998: pp. 1044–1055. ISSN 0885-3010. doi:10.1109/58.710587.

[52] Liu, Q. H.: Large-scale simulations of electromagnetic and acoustic measurements using the pseudospectral time-domain (PSTD) algorithm. *IEEE Transactions on Geoscience and Remote Sensing*. vol. 37, no. 2 II. mar 1999: pp. 917–926. ISSN 0196-2892. doi:10.1109/36.752210.

[53] Mahgerefteh, H.; Rykov, Y.; Denton, G.: Courant, Friedrichs and Lewy (CFL) impact on numerical convergence of highly transient flows. *Chemical Engineering Science*. vol. 64, no. 23. 2009: pp. 4969–4975. ISSN 00092509. doi:10.1016/j.ces.2009.08.002.

[54] Mathieu, V.; Anagnostou, F.; Soffer, E.; et al.: Numerical simulation of ultrasonic wave propagation for the evaluation of dental implant biomechanical stability. *The Journal of the Acoustical Society of America*. vol. 129, no. 6. 2011: pp. 4062–4072. ISSN 0001-4966. doi:10.1121/1.3586788.

[55] Meza-Fajardo, K. C.; Papageorgiou, A. S.: On the stability of a non-convolutional perfectly matched layer for isotropic elastic media. *Soil Dynamics and Earthquake Engineering*. vol. 30, no. 3. mar 2010: pp. 68–81. ISSN 0267-7261. doi:10.1016/j.soildyn.2009.09.002.

[56] Morales, N.; Chavda, V.; Mehra, R.; et al.: MPARD: A high-frequency wave-based acoustic solver for very large compute clusters. *Applied Acoustics*. vol. 121. jun 2017: pp. 82–94. ISSN 0003-682X. doi:10.1016/j.apacoust.2017.01.009.

[57] Nachman, A. I. A.; Waag, R. R. C.; Smith, J. F. I.; et al.: An equation for acoustic propagation in inhomogeneous media with relaxation losses. *The Journal of the*

Acoustical Society of America. vol. 88, no. 3. 1990: pp. 1584–1595. ISSN 0001-4966. doi:10.1121/1.400317.

[58] Nandapalan, N.; Jaros, J.; Rendell, A. P.; et al.: Implementation of 3D FFTs across multiple GPUs in shared memory environments. In *Parallel and Distributed Computing, Applications and Technologies, PDCAT Proceedings*. Beijing: IEEE. dec 2012. ISBN 9780769548791. pp. 167–172. doi:10.1109/PDCAT.2012.79.

[59] Nikl, V.; Jaros, J.: Parallelisation of the 3D fast Fourier transform using the hybrid OpenMP/MPI decomposition. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8934. Springer International Publishing Switzerland 2014. 2014. ISBN 9783319148953. ISSN 1611-3349. pp. 100–112. doi:10.1007/978-3-319-14896-0_9.

[60] Novakovic, S.; Daglis, A.; Bugnion, E.; et al.: Scale-out NUMA. *Proceedings of the 19th international conference on Architectural support for programming languages and operating systems - ASPLOS '14*. 2014: pp. 3–18. ISSN 0362-1340. doi:10.1145/2541940.2541965.

[61] NVIDIA: NVIDIA NVLink High-Speed Interconnect: Application Performance. Technical Report November. NVIDIA Corporation. 2014.

[62] Okita, K.; Narumi, R.; Azuma, T.; et al.: The role of numerical simulation for the development of an advanced HIFU system. *Computational Mechanics*. vol. 54, no. 4. oct 2014: pp. 1023–1033. ISSN 0178-7675. doi:10.1007/s00466-014-1036-y.

[63] Okita, K.; Ono, K.; Takagi, S.; et al.: Development of high intensity focused ultrasound simulator for large-scale computing. *International Journal for Numerical Methods in Fluids*. vol. 65, no. 1-3. 2011: pp. 43–66. ISSN 0271-2091. doi:10.1002/fld.2470. fld.1.

[64] Paulides, M. M.; Stauffer, P. R.; Neufeld, E.; et al.: Simulation techniques in hyperthermia treatment planning. *International journal of hyperthermia : the official journal of European Society for Hyperthermic Oncology, North American Hyperthermia Group*. vol. 29, no. 4. 2013: pp. 346–357. ISSN 1464-5157. doi:10.3109/02656736.2013.790092. NIHMS150003.

[65] Pekurovsky, D.: P3DFFT: A Framework for Parallel Computations of Fourier Transforms in Three Dimensions. 2012. doi:10.1137/11082748X.

[66] Pennes, H.: Analysis of tissue and arterial blood temperatures in the resting human forearm. *Journal of applied physiology*. vol. 1, no. 2. 1948: pp. 93–122. ISSN 0021-8987. doi:9714612. arXiv:1011.1669v3.

[67] Pinton, G.; Aubry, J.-F.; Fink, M.; et al.: Effects of nonlinear ultrasound propagation on high intensity brain therapy. *Medical Physics*. vol. 38, no. 3. 2011: page 1207. ISSN 0094-2405. doi:10.1118/1.3531553.

[68] Pinton, G. F.; Dahl, J.; Rosenzweig, S.; et al.: A heterogeneous nonlinear attenuating full- wave model of ultrasound. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*. vol. 56, no. 3. 2009: pp. 474–488. ISSN 0885-3010. doi:10.1109/TUFFC.2009.1066.

[69] Pippig, M.: PFFT-An extension of FFTW to massively parallel architectures. *SIAM Journal on Scientific Computing*. vol. 35, no. 3. 2013: pp. 213–236. ISSN 1064-8275. doi:http://dx.doi.org/10.1137/120885887.

[70] Pitt-Francis, J.; Whiteley, J.: *Guide to Scientific Computing in C++ (Undergraduate Topics in Computer Science)*. Undergraduate Topics in Computer Science. London: Springer London. 2012. ISBN 1447127358. 262 pp.. doi:10.1007/978-1-4471-2736-9. arXiv:1011.1669v3.

[71] Pulkkinen, A.; Werner, B.; Martin, E.; et al.: Numerical simulations of clinical focused ultrasound functional neurosurgery. *Physics in medicine and biology*. vol. 59, no. 7. apr 2014: pp. 1679–700. ISSN 1361-6560. doi:10.1088/0031-9155/59/7/1679. NIHMS150003.

[72] Release, F.: Infiniband Architecture Specification Volume 1. *Architecture*. vol. 1, no. November. 2015.

[73] Requirements, A.; Computing, O. L.: APPROACHING EXASCALE : Application Requirements for OLCF Leadership Computing. Technical Report July. OAK RIDGE NATIONAL LABORATORY. Oak Ridge. 2013.

[74] Robertson, J. L.; Cox, B. T.; Treeby, B. E.: Quantifying numerical errors in the simulation of transcranial ultrasound using pseudospectral methods. In *IEEE International Ultrasonics Symposium, IUS*. IEEE. sep 2014. ISBN 9781479970490. ISSN 1948-5727. pp. 2000–2003. doi:10.1109/ULTSYM.2014.0498.

[75] Robertson, J. L. B.; Cox, B. T.; Jaros, J.; et al.: Accurate simulation of transcranial ultrasound propagation for ultrasonic neuromodulation and stimulation. *The Journal of the Acoustical Society of America*. vol. 141, no. 3. mar 2017: pp. 1726–1738. ISSN 0001-4966. doi:10.1121/1.4976339.

[76] Schneider, U.; Pedroni, E.; Lomax, A.: The calibration of CT Hounsfield units for radiotherapy treatment planning. *Physics in medicine and biology*. vol. 41, no. 1. 1996: pp. 111–24. ISSN 0031-9155. doi:10.1088/0031-9155/41/1/009. arXiv:1011.1669v3.

[77] Sparrow, V. W.; Raspet, R.: A numerical method for general finite amplitude wave propagation in two dimensions and its application to spark pulses. *J. Acoust. Soc. Am.*. vol. 90, no. 5. 1991: pp. 2683–2691. doi:10.1121/1.401863.

[78] Suomi, V.; Jaros, J.; Treeby, B. E.; et al.: Nonlinear 3-D simulation of high-intensity focused ultrasound therapy in the Kidney. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. aug 2016. ISSN 1557-170X. pp. 5648–5651. doi:10.1109/EMBC.2016.7592008.

[79] Szabo, T. L.: Time-Domain Wave-Equations for Lossy Media Obeying a Frequency Power-Law. *Journal of the Acoustical Society of America*. vol. 96, no. 1. 1994: pp. 491–500.

[80] Tabei, M.; Mast, T. D.; Waag, R. C.: A k-space method for coupled first-order acoustic propagation equations. *The Journal of the Acoustical Society of America*. vol. 111, no. 1 Pt 1. jan 2002: pp. 53–63. ISSN 00014966. doi:10.1121/1.1421344.

[81] Ter Haar, G.: Therapeutic ultrasound. 1999. doi:10.1016/S0929-8266(99)00013-0.

[82] Ter Haar, G.; Shaw, A.; Pye, S.; et al.: Guidance on Reporting Ultrasound Exposure Conditions for Bio-Effects Studies. 2011. doi:10.1016/j.ultrasmedbio.2010.10.021.

[83] Treeby, B. E.; Cox, B. T.: k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *Journal of Biomedical Optics*. vol. 15, no. 2. 2010: pp. 021–314. ISSN 1560-2281. doi:10.1117/1.3360308.

[84] Treeby, B. E.; Cox, B. T.: Modeling power law absorption and dispersion for acoustic propagation using the fractional Laplacian. *The Journal of the Acoustical Society of America*. vol. 127, no. 5. 2010: pp. 2741–48. ISSN 1520-8524. doi:10.1121/1.3377056.

[85] Treeby, B. E.; Cox, B. T.: A k-space Green's function solution for acoustic initial value problems in homogeneous media with power law absorption. *J Acoust Soc Am*. vol. 129, no. 6. 2011: pp. 3652–3660. ISSN 0001-4966. doi:10.1121/1.3583537.

[86] Treeby, B. E.; Cox, B. T.; Jaros, J.: k-Wave: A MATLAB toolbox for the time domain simulation of acoustic wave fields. Technical report. University College London. 2012.

[87] Treeby, B. E.; Jaros, J.; Cox, B. T.: Advanced photoacoustic image reconstruction using the k-Wave toolbox. In *SPIE Photons Plus Ultrasound: Imaging and Sensing*, vol. 9708, edited by A. A. Oraevsky; L. V. Wang. mar 2016. ISBN 9781628419429. ISSN 1605-7422. page 97082P. doi:10.1117/12.2209254.

[88] Treeby, B. E.; Jaros, J.; Cox, B. T.: Focused ultrasound waves in heterogeneous tissue. In *High Intensity Focused Ultrasound Therapy: Fundamentals through Clinical Challenges*, edited by V. Khokhlova; L. Crum; G. ter Haar; J.-F. Aubry. Springer. 2016. ISBN 978-3-319-14261-6. page 24.

[89] Treeby, B. E.; Jaros, J.; Rendell, A. P.; et al.: Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method. *The Journal of the Acoustical Society of America*. vol. 131, no. 6. 2012: pp. 4324–36. ISSN 1520-8524. doi:10.1121/1.4712021.

[90] Treeby, B. E.; Jaros, J.; Rohrbach, D.; et al.: Modelling elastic wave propagation using the k-Wave MATLAB Toolbox. In *2014 IEEE International Ultrasonics Symposium*. 5. IEEE. sep 2014. ISBN 978-1-4799-7049-0. ISSN 1948-5727. pp. 146–149. doi:10.1109/ULTSYM.2014.0037.

[91] Treeby, B. E.; Tumen, M.; Cox, B. T.: Time domain simulation of harmonic ultrasound images and beam patterns in 3D using the k-space pseudospectral method. In *Lecture Notes in Computer Science*, vol. 6891. Springer. 2011. ISBN 9783642236228. pp. 363–370. doi:10.1007/978-3-642-23623-5_46.

[92] Verweij, M. D.; Huijssen, J.: A filtered convolution method for the computation of acoustic wave fields in very large spatiotemporal domains. *The Journal of the Acoustical Society of America*. vol. 125, no. 4. 2009: pp. 1868–1878. ISSN 0001-4966. doi:10.1121/1.3077220.

[93] Vysocky, O.: *Optimization of the distributed I/O subsystem of the k-Wave project*. PhD. Thesis. Brno University of Technology. 2015.

[94] Wang, K.; Teoh, E.; Jaros, J.; et al.: Modelling nonlinear ultrasound propagation in absorbing media using the k-Wave toolbox: Experimental validation. In *IEEE International Ultrasonics Symposium, IUS*. Dresden, DE: Institute of Electrical and Electronics Engineers. 2012. ISBN 9781467345613. ISSN 1948-5719. pp. 523–526. doi:10.1109/ULTSYM.2012.0130.

[95] Wende, F.; Marsman, M.; Steinke, T.: On Enhancing 3D-FFT Performance in VASP. In *CUG Proceedings*. 2016. page 9.

[96] Westervelt, P. J.: Parametric Acoustic Array. *The Journal of the Acoustical Society of America*. vol. 35, no. 4. 1963: page 535. ISSN 0001-4966. doi:10.1121/1.1918525.

[97] Wilt, N.: *Summary for Policymakers*. Addison-Wesley Professional. 2013. ISBN 978-0-321-80946-9. 1–30 pp.. doi:10.1017/CBO9781107415324.004. arXiv:1011.1669v3.

[98] Wojcik, G.; Mould, J.; Ayter, S.; et al.: A study of second harmonic generation by focused medical transducerpulses. *IEEE Ultrasonics Symposium Proceedings*. vol. 2. 1998: pp. 1583–1588. ISSN 0300-9084. doi:10.1109/ULTSYM.1998.765247.

[99] Yamaguchi, T.; Tsuchiya, T.; Nakahara, S.; et al.: Efficacy of Left Atrial Voltage-Based Catheter Ablation of Persistent Atrial Fibrillation. *Journal of Cardiovascular Electrophysiology*. vol. 27, no. 9. sep 2016: pp. 1055–1063. ISSN 1540-8167. doi:10.1111/jce.13019. 1011.1669v3.

[100] Yan, D.; Yang, L.; Wang, Q.: Alternative thermodiffusion interface for simultaneous speciation of organic and inorganic lead and mercury species by capillary GC-ICPMS using tri-n-propyl-lead chloride as an internal standard. *Analytical Chemistry*. vol. 80, no. 15. 2008: pp. 6104–6109. ISSN 0003-2700. doi:10.1021/ac800347j.

[101] Zhou, Y.-F.; Syed Arbab, A.; Xu, R. X.: High intensity focused ultrasound in clinical tumor ablation. *World journal of clinical oncology*. vol. 2, no. 1. 2011: pp. 8–27. ISSN 2218-4333. doi:10.5306/wjco.v2.i1.8.

Appendices

Appendix A

Acoustic Model

A.1 Derivation of Fluid Model

Treeby, B. E.; **Jaros, J.**; Rendell, A. P.; Cox, B. T.: Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method. *The Journal of the Acoustical Society of America.* vol. 131, no. 6. 2012: pp. 4324–36. ISSN 1520-8524. doi:10.1121/1.4712021, **(IF 1.572)**.

Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a *k*-space pseudospectral method

Bradley E. Treeby[a)]
Research School of Engineering, College of Engineering and Computer Science, The Australian National University, Canberra ACT 0200, Australia

Jiri Jaros and Alistair P. Rendell
Research School of Computer Science, College of Engineering and Computer Science, The Australian National University, Canberra ACT 0200, Australia

B. T. Cox
Department of Medical Physics and Bioengineering, University College London, Gower Street, London WC1E 6BT, United Kingdom

(Received 6 September 2011; revised 3 April 2012; accepted 11 April 2012)

The simulation of nonlinear ultrasound propagation through tissue realistic media has a wide range of practical applications. However, this is a computationally difficult problem due to the large size of the computational domain compared to the acoustic wavelength. Here, the *k*-space pseudospectral method is used to reduce the number of grid points required per wavelength for accurate simulations. The model is based on coupled first-order acoustic equations valid for nonlinear wave propagation in heterogeneous media with power law absorption. These are derived from the equations of fluid mechanics and include a pressure-density relation that incorporates the effects of nonlinearity, power law absorption, and medium heterogeneities. The additional terms accounting for convective nonlinearity and power law absorption are expressed as spatial gradients making them efficient to numerically encode. The governing equations are then discretized using a *k*-space pseudospectral technique in which the spatial gradients are computed using the Fourier-collocation method. This increases the accuracy of the gradient calculation and thus relaxes the requirement for dense computational grids compared to conventional finite difference methods. The accuracy and utility of the developed model is demonstrated via several numerical experiments, including the 3D simulation of the beam pattern from a clinical ultrasound probe.

© *2012 Acoustical Society of America*. [http://dx.doi.org/10.1121/1.4712021]

PACS number(s): 43.25.Jh, 43.20.Bi, 43.35.Bf, 43.25.Cb [TDM] Pages: 4324–4336

I. INTRODUCTION

The simulation of ultrasound propagation through soft biological tissue has a wide range of practical applications. These include the design of transducers for diagnostic and therapeutic ultrasound, the development of new signal processing and imaging techniques, studying the aberration of ultrasound beams in heterogeneous media, ultrasonic tissue classification, training ultrasonographers to use ultrasound equipment and interpret ultrasound images, model-based medical image registration, and treatment planning and dosimetry for high-intensity focused ultrasound.[1] The most general approach for ultrasound simulation is to directly solve the equations of continuum mechanics. However, this is a computationally difficult problem due to the large size of the region of interest in relation to the size of the acoustic wavelength. For example, a typical diagnostic ultrasound image formed using a 3 MHz curvilinear transducer has a depth penetration of around 15 cm. This distance is on the order of 300 acoustic wavelengths at the fundamental frequency, and 600 wavelengths at the second harmonic. Established numerical methods such as the finite difference or finite element methods require on the order of 10 grid points per wavelength to achieve acceptable accuracy. This equates to a computational domain with thousands of grid points in each spatial dimension. Consequently, many simulations of interest are simply intractable, or require very large amounts of computer memory and can take days or weeks to run.[2]

To reduce this computational burden, simplifying assumptions are frequently made. For modeling the beam patterns from ultrasound transducers, a common approach is to only consider one-way (or forward) wave propagation (see Huijssen and Verweij[3] for a recent review). If the problem is axisymmetric, the governing equations can also be solved in 2D.[4] However, these approaches are unable to account for all aspects of nonlinear wave propagation in heterogeneous media. For the simulation of diagnostic ultrasound images, a Green's function method is also often used.[5] In this case, the scattering medium is modeled as series of point sources in a homogeneous background (the widely used FIELD II program is based on this approach). However, this does not account for more complex acoustic phenomena, for example, multiple scattering or nonlinearity. Given the

[a)] Author to whom correspondence should be addressed. Electronic mail: bradley.treeby@anu.edu.au wide range of possible applications, there is a strong motivation for the development of new ultrasound simulation tools with less restrictive assumptions and improved computational efficiency.

Here, a computationally efficient approach for the simulation of nonlinear wave propagation is derived using a k-space pseudospectral method.[6] In Sec. II, existing methods for modeling ultrasound propagation in tissue realistic media are reviewed. In this context, approaches for modeling both heterogeneous media and power law absorption are discussed. In Sec. III, governing equations suitable for modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption are then developed. In Sec. IV, these equations are discretized using a k-space pseudospectral method. The spectral calculation of the spatial derivatives is performed using the Fourier-collocation method and significantly relaxes the requirement for dense computational grids compared to finite difference methods. In Sec. V, several numerical experiments are presented to illustrate the accuracy and efficacy of the developed model. Discussion and summary are then given in Sec. VI, with further details of the computer code provided in the Appendix.

II. TISSUE REALISTIC ULTRASOUND MODELING

A. Model requirements

The development of accurate models for ultrasound propagation in soft tissue requires the consideration of three important factors. (1) In most cases the amplitude of the acoustic waves is sufficiently large that the wave propagation is nonlinear. For example, modern ultrasound scanners rely on nonlinear wave propagation for tissue harmonic imaging which gives images with improved clarity and contrast. (2) The material properties of biological tissue (e.g., the sound speed and density) are weakly heterogeneous, with variations between the different soft tissue types and water on the order of 5%.[1] (3) The tissue is absorbing at ultrasonic frequencies with the absorption following a frequency power law. In the context of nonlinear wave propagation, an accurate model of acoustic absorption is of particular importance as the generation of higher frequency harmonics via nonlinearity is delicately balanced with their absorption.

When considered individually as extensions to the standard equations of linear acoustics, each of these factors has been extensively studied. It is the unification of these effects into a consistent set of coupled first-order equations, in addition to the efficient solution of these equations, that is the subject of interest here. The use of first-order governing equations rather than the corresponding second-order wave equation has several advantages. First, it allows the pressure and particle velocity to be computed on staggered grids which improves accuracy. Second, it allows the use of an anisotropic perfectly matched layer (PML) for absorbing the acoustic waves at the edges of the computational domain.[6] Third, it provides an intuitive way to directly include both mass and force sources in the discrete equations. Finally, the explicit calculation of the particle velocity allows the vector components of acoustic intensity to be directly computed. This is of particular relevance when modeling the heat deposition from therapeutic ultrasound probes.

B. Accounting for tissue heterogeneities

Over the last half a century, a large number of researchers have contributed to an extensive body of knowledge on the nonlinear propagation of acoustic waves.[7] From a theoretical perspective, the required governing equations can be derived by including second order (and in some cases third order) terms in the conservation equations and pressure-density relation. However, despite the long history of nonlinear acoustics, most rigorous derivations are based on the assumption of a homogeneous medium with thermoviscous absorption. In particular, there have been very few attempts to consider acoustic heterogeneities at the level of the governing equations. (Two recent exceptions are the papers by Taraldsen,[8] who derived a heterogeneous Westervelt equation in Lagrangian coordinates, and Coulouvrat,[9] who considered the case of a heterogeneous and moving turbulent fluid.) While it is straightforward to consider heterogeneous medium parameters in the conservation equations, the derivation of a pressure-density relation valid for nonlinear wave propagation in heterogeneous media is more difficult to find in the literature.

The most common approach to modeling heterogeneous medium parameters is to assume that the effects of nonlinearity and heterogeneity are sufficiently small that their interactions can be neglected. In this way, an appropriate wave equation can be formed by combining the additional terms from the linear wave equation for heterogeneous media with those from the nonlinear wave equation for homogeneous media. For example, Hallaj *et al.*,[10] and Pinton *et al.*,[2] both utilized a Westervelt equation augmented with the heterogeneous density term from the linear wave equation. Jing and Cleveland[11] presented a similar wave equation including local nonlinear effects. This was then reduced to a Khokhlov–Zabolotskaya–Kuznetsov (KZK) equation suitable for heterogeneous media. Comparisons of simulations using this equation with experimental measurements of ultrasound propagation through a heterogeneous gel layer showed good agreement. An analogous approach was taken by Verweij and Huijssen[12] and Jing and Clement[13] where both the nonlinearity and heterogeneity terms were introduced as contrast source terms. Similarly, Averyanov *et al.*, supplemented a linear parabolic wave equation for heterogeneous media with additional terms describing the effects of nonlinearity and absorption.[14] While the accuracy of these wave equations for modeling nonlinear wave propagation in weakly heterogeneous media is well established, they do not provide heterogeneous forms of the conservation and pressure-density equations which can be solved as a set of coupled first-order equations.

C. Accounting for power law acoustic absorption

Classical lossy wave equations based on the inclusion of viscosity and thermal conduction into the governing equations yield an acoustic absorption term that is proportional to frequency squared. However, the absorption mechanisms in soft biological tissue are significantly more complex (including vibrational, structural, and chemical relaxations) which leads to an experimentally observed attenuation of the form

$$\alpha = \alpha_0 \omega^y, \tag{1}$$

where the power law exponent y is typically in the range $1 - 1.5$.[1] To account for this difference, the thermoviscous absorption term can be replaced with an alternate loss term. This idea was first proposed by Blackstock who replaced the thermoviscous term in the lossy Burgers equation with a general absorption operator.[15] Szabo later derived a causal form of this operator to account for power law absorption with an arbitrary frequency dependence.[16,17] This was derived in the form of a time domain convolution operator and was used to replace the classical thermoviscous absorption terms in the KZK, Burgers, and Westervelt equations.[16] Similar convolution operators for the KZK equation[18] and Kuznetsov's equation[19] have also been derived.

Although Szabo's lossy operator correctly incorporates the required power law behavior, the operator is dependent on the time history of the pressure field which makes it difficult to encode in a memory efficient manner. As an alternative, Chen and Holm[20] derived a lossy operator based on the fractional Laplacian. This was similarly used to replace the absorption terms in the KZK, Burgers, and Westervelt equations.[21] This operator was later extended to correctly account for power law dispersion as required by the Kramers–Kronig relations.[22] In comparison to Szabo's operator, the computation of the fractional Laplacian only depends on the values of the pressure field at the current time. This makes the operator efficient to compute, particularly using Fourier-based pseudospectral and k-space methods.[22,23]

An alternative approach to using a phenomenological operator to account for power law absorption is to explicitly consider the absorption as a sum of relaxation processes. This is based on a physical analogy with the different absorption mechanisms in tissue which act as relaxation processes with varying relaxation times. Models for both a continuous distribution of relaxation parameters[24] and a discrete set of relaxation parameters[25] have been proposed. However, despite the physical appeal of such models, for biological materials the individual relaxation processes and their relaxation times are not generally known. Consequently, the model parameters must instead be derived using a fitting procedure based on experimental data. In this case, the derived relaxation parameters do not necessarily have any direct connection with the physical absorption mechanisms and thus can also be considered as phenomenological terms. The discrete relaxation model has been applied to the linear,[25] KZK,[26] and Westervelt[2] equations. For the latter, two relaxation processes were found to be sufficient to approximate power law absorption over a 12 MHz frequency range.[2]

Given that only a small number of relaxation parameters are generally needed to approximate power law absorption over a given frequency range, there is not a clear argument for using a phenomenological absorption operator over a relaxation operator, or vice-versa. Indeed, Näsholm and Holm have recently shown that, under certain conditions, fractional loss operators can be derived from a continuum of relaxation processes, rendering the two approaches equivalent.[27] On one hand, the relaxation approach is more general and is able to model absorption with an arbitrary frequency dependence. On the other hand, the extraction of the relaxation parameters needed for the model requires an *a priori* fitting procedure for each value of absorption and range of frequencies under consideration. If the objective is specifically to model power law absorption (as is the case here), it is easier to directly use an operator that can account for this behavior.

III. NONLINEAR GOVERNING EQUATIONS FOR HETEROGENEOUS MEDIA

A. General nonlinear equations

The equations required to describe the nonlinear propagation of compressional acoustic waves through heterogeneous fluid media can be obtained directly from the equations of fluid mechanics. Under the assumption of a quiescent, isotropic, and inviscid medium (acoustic absorption is explicitly considered later as an energy loss term), the momentum and mass conservation equations can, respectively, be written in a Eulerian coordinate system as[7]

$$\rho_0 \frac{\partial \mathbf{u}}{\partial t} + \nabla p = -\rho \frac{\partial \mathbf{u}}{\partial t} - \frac{1}{2}\rho_0 \nabla \left(\mathrm{u}^2\right), \tag{2a}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho_0 \mathbf{u}) = -\nabla \cdot (\rho \mathbf{u}). \tag{2b}$$

Here p and ρ are the acoustic pressure and density, $\mathbf{u}$ is the acoustic particle velocity where $\mathrm{u}^2 = \mathbf{u}\cdot\mathbf{u}$, and ρ_0 is the ambient density. Note, in these and subsequent expressions, only terms up to second order in the acoustic variables are retained. This is sufficient for modeling the finite amplitude effects of interest here.

A nonlinear pressure-density relation for an arbitrary fluid medium can then be obtained by expanding the state equation $\hat{p} = \hat{p}(\hat{\rho}, \hat{s})$ in a Taylor series about the ambient density and entropy. Here the ˆ symbol is used to denote a total quantity, i.e., the sum of ambient and acoustic parts. Following Lighthill,[28] it is assumed that the effects of nonlinearity and changes in entropy (which are due to acoustic absorption) are both second order. Consequently, higher order entropy terms can be discarded. Considering the change in the total pressure of a fluid element for a small but finite time step $\delta t = t_1 - t_0$, the Taylor series expansion can then be written as

$$\hat{p}(t_1) - \hat{p}(t_0) = \left(\frac{\partial \hat{p}}{\partial \hat{\rho}}\right)_{\hat{s}} (\hat{\rho}(t_1) - \hat{\rho}(t_0)) + \frac{1}{2}\left(\frac{\partial^2 \hat{p}}{\partial \hat{\rho}^2}\right)_{\hat{s}} \times (\hat{\rho}(t_1) - \hat{\rho}(t_0))^2 + \left(\frac{\partial \hat{p}}{\partial \hat{s}}\right)_{\hat{\rho}} (\hat{s}(t_1) - \hat{s}(t_0)). \tag{3}$$

If the ambient medium parameters are heterogeneous, the change in the total density $\hat{\rho}(t_1) - \hat{\rho}(t_0)$ can arise either due to local acoustic perturbations, or due to the displacement of the fluid element under consideration to a new position in which the ambient density is different.[29] If $\hat{\rho}$ is written as a function of both position x and time t, the change in $\hat{\rho}$ between t_0 and t_1 can similarly be expanded using a Taylor series. This yields the expression

$$\hat{\rho}(t_1) - \hat{\rho}(t_0) = \left(\frac{\partial \hat{\rho}}{\partial t}\right)_x (t_1 - t_0) + \left(\frac{\partial \hat{\rho}}{\partial x}\right)_t (\zeta_1 - \zeta_0), \quad (4)$$

where ζ_1 and ζ_0 are the positions of the fluid element at t_1 and t_0. The first term is the conventional Eulerian description of the acoustic density ρ (assuming the medium is quiescent), while the second term is due to the displacement of the fluid element. This is equivalent to describing the change in density within a fixed element in a Eulerian coordinate system which has an additional component due to the displacement of the surrounding fluid. Defining the particle displacement vector as $\mathbf{d} = \zeta_1 - \zeta_0$ and writing the spatial derivative of $\hat{\rho}$ at constant time in vector notation as

$$\left(\frac{\partial \hat{\rho}}{\partial x}\right)_t \equiv \nabla \hat{\rho}(t_0) = \nabla \rho_0,$$

Eq. (4) then becomes

$$\hat{\rho}(t_1) - \hat{\rho}(t_0) = \rho + \mathbf{d} \cdot \nabla \rho_0. \quad (5)$$

Assuming the medium is initially in thermodynamic equilibrium, the equivalent spatial gradients of $\hat{s}$ and $\hat{p}$ are zero giving $\hat{s}(t_1) - \hat{s}(t_0) = s$ and $\hat{p}(t_1) - \hat{p}(t_0) = p$. Using these expressions and making the substitutions[7]

$$A \equiv \rho_0 \left(\frac{\partial \hat{p}}{\partial \hat{\rho}}\right)_{\hat{s}} = \rho_0 c_0^2, \quad B \equiv \rho_0^2 \left(\frac{\partial^2 \hat{p}}{\partial \hat{\rho}^2}\right)_{\hat{s}}$$

(where the first equation above defines the isentropic sound speed c_0), Eq. (3) can then be written in the form

$$p = c_0^2(\rho + \mathbf{d} \cdot \nabla \rho_0) + \left(\frac{\partial \hat{p}}{\partial \hat{s}}\right)_{\hat{\rho}} s + \frac{B}{2A}\frac{c_0^2}{\rho_0}\left(\rho^2 + (\mathbf{d} \cdot \nabla \rho_0)^2 + 2\rho \mathbf{d} \cdot \nabla \rho_0\right). \quad (6)$$

Here B/A is the parameter of nonlinearity which characterizes the relative contribution of finite-amplitude effects to the sound speed.[7] Note, in the linear case, an equivalent relation can be derived by considering the Lagrangian derivative of the state equation.[30]

The remaining entropy term in Eq. (6) represents an energy loss or acoustic absorption term. In the case of thermoviscous media, this can be related to the thermal conductivity and specific heat capacity of the medium by explicit consideration of the appropriate energy conservation equation.[7] More generally (by analogy with Blackstock[15]), this can be replaced by a phenomenological loss term of form

$$\left(\frac{\partial \hat{p}}{\partial \hat{s}}\right)_{\hat{\rho}} s = -\left(\frac{\partial \hat{p}}{\partial \hat{\rho}}\right)_{\hat{s}} \left(\frac{\partial \hat{\rho}}{\partial \hat{s}}\right)_{\hat{p}} s \equiv -c_0^2 L \rho,$$

where L is a general loss operator. For modeling power law absorption, it is convenient to define L as a derivative operator based on the fractional Laplacian[22]

$$L = \tau \frac{\partial}{\partial t}\left(-\nabla^2\right)^{y/2-1} + \eta\left(-\nabla^2\right)^{(y+1)/2-1}. \quad (7)$$

Here τ and η are absorption and dispersion proportionality coefficients given by $\tau = -2\alpha_0 c_0^{y-1}$ and $\eta = 2\alpha_0 c_0^y \tan(\pi y/2)$, α_0 is the power law prefactor in Np $(\mathrm{rad/s})^{-y}\mathrm{m}^{-1}$, and y is the power law exponent. The two terms in L separately account for power law absorption and dispersion for $0 < y < 3$ and $y \neq 1$ under particular smallness conditions.[22] These conditions are generally satisfied for the range of attenuation parameters observed in soft biological tissue (for very high values of absorption and frequency the accuracy of the loss operator decreases due to second-order effects[23]). The use of a fractional derivative in the pressure-density relation can also be related to a general relaxation relationship between the temperature gradient and resulting heat flux which leads to a fractional entropy equation.[31]

B. Reduced nonlinear equations

While the general first-order equations derived in the previous section could be directly solved using standard numerical techniques, it is both unwieldy and unnecessary to do so. For many applications in biomedical ultrasonics, it is sufficient to consider only cumulative nonlinear effects.[32] We also make the assumption that the effect of acoustic heterogeneities on the wave field can be considered as second-order. Any higher order heterogeneity terms or interactions between nonlinearity and heterogeneity terms can then also be discarded.

Returning to the momentum and mass conservation equations given in Eq. (2), following the approach taken by Aanonsen *et al.*, [7,33] the second-order terms which appear on the right hand side can now be re-written in terms of the acoustic Lagrangian density via the repeated substitution of the homogeneous acoustic equations in linearized form (using the premise that the substitution of first-order equations into second-order terms yields third-order errors). This gives the expressions

$$\rho_0 \frac{\partial \mathbf{u}}{\partial t} + \nabla p = -\nabla \mathcal{L}, \quad (8a)$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho_0 \mathbf{u}) = \frac{1}{c_0^2}\frac{\partial \mathcal{L}}{\partial t} + \frac{1}{\rho_0 c_0^4}\frac{\partial p^2}{\partial t}, \quad (8b)$$

where $\mathcal{L}$ is the second-order Lagrangian density given by

$$\mathcal{L} = \frac{1}{2}\rho_0 \mathrm{u}^2 - \frac{p^2}{2\rho_0 c_0^2}.$$

This characterizes the difference between the kinetic and potential energy density of the acoustic wave. If only cumulative nonlinear effects are important, the Lagrangian density can be set to zero which leaves

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{\rho_0}\nabla p = 0, \tag{9a}$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho_0 \mathbf{u}) = \frac{1}{\rho_0 c_0^4}\frac{\partial p^2}{\partial t}. \tag{9b}$$

If the governing equations are solved using spectral methods (as is the case here), it is convenient to re-write the convective nonlinear term that appears in the mass conservation equation given in Eq. (9b) in its original form as a spatial gradient. This is because spatial gradients can be computed spectrally, while temporal gradients require the use of a finite difference approximation as well as additional storage. By following the series of substitutions that yield Eq. (8b) from Eq. (2b), it can be shown that the final term is equivalent to the expression $-2\rho\nabla \cdot \mathbf{u}$ (to second order). Using this substitution gives an alternate form of the mass conservation equation valid for modeling cumulative nonlinear effects. Combined with the momentum conservation equation and the pressure-density relation, the full set of coupled equations can now be written as

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho_0}\nabla p, \tag{10a}$$

$$\frac{\partial \rho}{\partial t} = -(2\rho + \rho_0)\nabla \cdot \mathbf{u} - \mathbf{u} \cdot \nabla \rho_0, \tag{10b}$$

$$p = c_0^2\left(\rho + \mathbf{d} \cdot \nabla \rho_0 + \frac{B}{2A}\frac{\rho^2}{\rho_0} - L\rho\right), \tag{10c}$$

where the loss operator L is defined in Eq. (7). The momentum conservation equation is identical to the linear case,[34] while the mass conservation equation has an additional term which accounts for a convective nonlinearity in which the particle velocity contributes to the wave velocity. The four terms within the pressure-density relation separately account for linear wave propagation, heterogeneities in the ambient density, material nonlinearity, and power law absorption and dispersion (the sound speed c_0 can also be heterogeneous). Note, the $\mathbf{u} \cdot \nabla \rho_0$ term in the mass conservation equation and the $\mathbf{d} \cdot \nabla \rho_0$ term in the pressure-density relation cancel when these equations are combined (or solved as coupled equations). Consequently, these terms are not included in the discrete equations given in the following section to improve computational efficiency.[35]

Using the mass conservation equation in the form given in Eq. (9b) and neglecting higher order absorption terms, the coupled governing equations can also be combined to give a modified form of the Westervelt equation valid for heterogeneous media with power law absorption

$$\nabla^2 p - \frac{1}{c_0^2}\frac{\partial^2 p}{\partial t^2} - \frac{1}{\rho_0}\nabla \rho_0 \cdot \nabla p + \frac{\beta}{\rho_0 c_0^4}\frac{\partial^2 p^2}{\partial t^2} - L\nabla^2 p = 0,$$

where $\beta = 1 + B/2A$ is the coefficient of nonlinearity (an equivalent expression assimilating the Laplacian into the loss operator is given in Ref. 36). This expression can be considered as a particular case of the generalized Westervelt equation discussed by Taraldsen.[8]

IV. NUMERICAL IMPLEMENTATION

A. Overview of the *k*-space method

Closely connected with the development of accurate governing equations for describing ultrasound propagation in tissue is the issue of their efficient solution. In a standard finite difference method, spatial gradients are computed locally based on the function values at neighboring grid points. As an alternative, it is also possible to calculate spatial gradients globally using the function values across the whole domain via spectral methods. This increases the accuracy of the gradient calculation and thus reduces the number of grid points required per wavelength for a given level of accuracy. For smoothly varying fields, spatial gradients can be calculated with spectral accuracy up to the Nyquist limit (two grid points per wavelength). Often the spectral calculation of spatial gradients is combined with the finite difference calculation of temporal gradients. However, the finite difference approximation introduces unwanted numerical dispersion into the solution that can only be controlled by reducing the size of the time step or increasing the order of the approximation.

Fortunately, for the standard linear wave equation valid for homogeneous and lossless media, an exact finite difference scheme for the temporal derivative exists. This can be used to derive an exact pseudospectral discretization of both the second-order wave equation,[37] and the corresponding coupled first-order conservation equations and pressure-density relation.[6] This approach is known as the *k*-space pseudospectral method (or simply the *k*-space method), because the difference between the exact and standard finite difference approximations reduces to an operator in the spatial frequency domain (referred to herein as the *k*-space operator). In the case of heterogeneous and absorbing media, the temporal discretization is no longer exact. However, if these perturbations are small, the *k*-space operator still reduces the unwanted numerical dispersion associated with the finite difference approximation of the time derivative (see discussion in Sec. V A).[6,37] Recently, Jing and Clement[13] presented a *k*-space method based on the Westervelt equation for thermoviscous media in which the absorption and nonlinearity terms were introduced as contrast source terms. Here, the *k*-space pseudospectral method described by Tabei *et al.*, [6] is used to discretize the coupled governing equations derived in Sec. III B.

B. Discrete *k*-space equations

Solving for the particle velocity in Eq. (10a) using an explicit first-order forward difference and for the acoustic density in Eq. (10b) using an implicit first-order forward difference, the conservation equations written in discrete form using the Fourier-collocation *k*-space pseudospectral method are given by

$$\frac{\partial}{\partial\xi}p^n = \mathbb{F}^{-1}\{ik_\xi\kappa\mathbb{F}\{p^n\}\}, \tag{11a}$$

$$u_\xi^{n+1} = u_\xi^n - \frac{\Delta t}{\rho_0}\frac{\partial}{\partial\xi}p^n, \tag{11b}$$

$$\frac{\partial}{\partial\xi}u_\xi^{n+1} = \mathbb{F}^{-1}\{ik_\xi\kappa\mathbb{F}\{u_\xi^{n+1}\}\}, \tag{11c}$$

$$\rho_\xi^{n+1} = \frac{\rho_\xi^n - \Delta t\rho_0\frac{\partial}{\partial\xi}u_\xi^{n+1}}{1 + 2\Delta t\frac{\partial}{\partial\xi}u_\xi^{n+1}}. \tag{11d}$$

The acoustic density is artificially divided into Cartesian components to allow an anisotropic PML to be applied.[38] Here, $\mathbb{F}$ and $\mathbb{F}^{-1}$ denote the forward and inverse spatial Fourier transform, the superscript n and $n+1$ denote the function values at current and next time points, respectively, i is the imaginary unit, k_ξ is the wavenumber in the ξ direction, Δt is the time step, and κ is the k-space operator given by[6,39]

$$\kappa = \mathrm{sinc}(c_{\mathrm{ref}}k\Delta t/2), \tag{11e}$$

where $k^2 = \sum_\xi k_\xi^2$, and c_{ref} is a reference sound speed (see discussion in Sec. V A). Equations (11a)–(11d) are repeated for each Cartesian direction in $\mathbb{R}^N$ where $\xi = x$ in $\mathbb{R}^1$, $\xi = x, y$ in $\mathbb{R}^2$, and $\xi = x, y, z$ in $\mathbb{R}^3$. Using the Fourier transform of the negative fractional Laplacian[20]

$$\mathbb{F}\{(-\nabla^2)^a\rho\} = k^{2a}\mathbb{F}\{\rho\},$$

the corresponding pressure-density relation in discrete form can be written as

$$p^{n+1} = c_0^2\left(\rho^{n+1} + \frac{B}{2A}\frac{1}{\rho_0}\left(\rho^{n+1}\right)^2 - L_d\right), \tag{11f}$$

where the total acoustic density is given by $\rho = \sum_\xi \rho_\xi$ and the discrete loss term is

$$L_d = -\tau\mathbb{F}^{-1}\left\{k^{y-2}\mathbb{F}\left\{\rho_0\sum_\xi\frac{\partial}{\partial\xi}u_\xi^{n+1}\right\}\right\} + \eta\mathbb{F}^{-1}\left\{k^{y-1}\mathbb{F}\left\{\rho^{n+1}\right\}\right\}. \tag{11g}$$

Here the temporal derivative of the acoustic density in the absorption term has been replaced using the linearized mass conservation equation $\partial\rho/\partial t = -\rho_0\nabla\cdot\mathbf{u}$ analogous to the first-order substitutions made in Sec. III B.

The discrete equations in Eq. (11) are iteratively solved using a time step based on the Courant–Friedrichs–Lewy (CFL) number, where $\Delta t = \mathrm{CFL}\Delta x/c_{\max}$. A CFL number of 0.3 typically provides a good balance between accuracy and computational speed for weakly heterogeneous media.[6] At each time step, a mass or force source can be included by adding the source values to the appropriate grid points within the computational domain. Similarly, the output from the simulation can be obtained by recording the acoustic variables at each time step at particular grid points. For regularly spaced Cartesian grids, the gradients can be computed efficiently using the fast Fourier transform (FFT). For the simulations presented here, a split-field PML was implemented to prevent waves from wrapping around the domain. The grids were also spatially and temporally staggered to improve accuracy.[6] The discrete equations were implemented in C++ as an extension to the open source K-WAVE acoustics toolbox for MATLAB (Mathworks, Natick, MA).[40] A description of the computer code is given in the Appendix.

V. NUMERICAL ACCURACY

A. Accuracy of the *k*-space operator for wave propagation in heterogeneous media

In the limit of linear wave propagation in a lossless and homogeneous medium, the k-space pseudospectral discretization of the three coupled governing equations is exact, and the algorithm is unconditionally stable. Although the finite difference time step still introduces unwanted numerical dispersion (or phase error) as expected, this is corrected by the k-space operator κ that appears in Eqs. (11a) and (11c). Provided the scalar sound speed used in the k-space operator c_{ref} is chosen to match the sound speed in the medium c_0, this correction is exact. However, in the case of heterogeneous media, there will necessarily be regions of the medium where the local value of c_0 does not match the value of c_{ref}. Consequently, the phase correction provided by κ will no longer be exact, and unwanted numerical dispersion will still be introduced into the solution.

To illustrate the effect of a mismatched c_{ref} and c_0, the phase error as a function of c_{ref} for a homogeneous medium with $c_0 = 1500$ m/s is shown in Fig. 1. This error corresponds to the numerical dispersion (as a percentage of c_0) in the propagation of a plane wave after 50 wavelengths using four grid points per wavelength and a CFL parameter of 0.3. When $c_{\mathrm{ref}} = c_0$, the phase correction provided by the k-space operator is exact and there is no phase error. When $c_{\mathrm{ref}} \approx c_0$, the correction is no longer exact. However, the use of the k-space operator still provides a significant reduction in the phase error as compared to that introduced by a finite

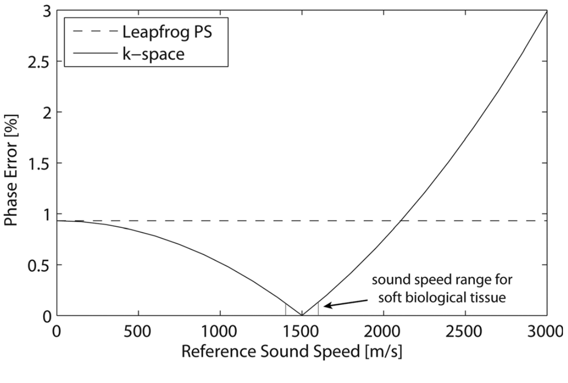


FIG. 1. Phase error in the propagation of a plane wave after 50 wavelengths against the reference sound speed c_{ref} used in the k-space operator κ for $c_0 = 1500$ m/s.

difference time step in the absence of this correction (dashed line shown in Fig. 1). Consequently, for soft biological tissue where the medium parameters are only weakly heterogeneous, the k-space method remains an apposite numerical technique.

It is useful to note, in the limit as c_{ref} approaches 0, κ approaches 1 and thus the k-space operator has no effect (see Fig. 1). Conversely, for $c_{\mathrm{ref}} \gg c_0$, this operator over compensates for the actual phase error introduced by the finite difference time step and thus increases the total phase error that is observed. Consequently, for modeling wave propagation in media with strong sound speed contrasts, care should be taken to select an appropriate reference sound speed, in addition to ensuring the appropriate stability constraints are met.[6] In the case of strongly heterogeneous media, if the maximum phase error introduced by the finite difference time step is still unacceptable after the k-space correction, a higher order finite difference scheme could alternatively be used for the temporal discretization.

B. Accuracy of the Fourier-collocation spectral method for heterogeneous media

The Fourier-collocation spectral method used for the computation of the spatial derivatives in Eqs. (11) decomposes the pressure and velocity fields into a discrete Fourier series with a finite number of coefficients. This decomposition is accurate for periodic fields that vary sufficiently smoothly throughout the computational domain such that they can be accurately represented using the band-limited set of supported frequencies (there is an explicit and well understood relationship between the smoothness of a function and the rate of decay of its Fourier coefficients[41]). However, this is not the case when there are sharp gradients in the acoustic fields. These can occur when the field variables are multiplied by heterogeneous medium parameters, for example, in Eq. (11f). In this case, the band-limited Fourier representation of the acoustic fields will exhibit oscillations (analogous to Gibbs' phenomenon) and will no longer provide an accurate representation of the discontinuities as they appear in the continuous domain.

To investigate the error introduced when the medium parameters are heterogeneous, the accuracy of the transmission and reflection coefficients for a plane wave traveling through a step change in the ambient density and sound speed was examined. The resulting coefficients for a 10% change in the material parameters are shown in Fig. 2(a) along with their theoretical values (shown as straight solid lines). At the Nyquist sampling limit (two grid points per wavelength), there is a large error in the calculated coefficients. However, this reduces quickly as the number of grid points is increased. The corresponding results using both first-order and fourth-order accurate finite difference schemes (including staggered grids) for the computation of the spatial derivative are shown in Fig. 2(b), with the relative errors in the transmission coefficient shown in Fig. 2(c). To achieve an error in the transmission coefficient of less than 1%, the Fourier-collocation spectral method requires only three grid points per wavelength, the fourth-order accurate finite difference scheme requires six points per wavelength, and the first-order accurate finite difference scheme requires 14. Similar results are observed for the reflection coefficient, although the overall errors are increased due to the smaller value of the coefficient (the error in the reflection coefficient for the Fourier-collocation and fourth-order finite difference schemes are approximately the same after six grid points per wavelength). Overall, for three-dimensional simulations, using a fourth-order accurate finite difference scheme

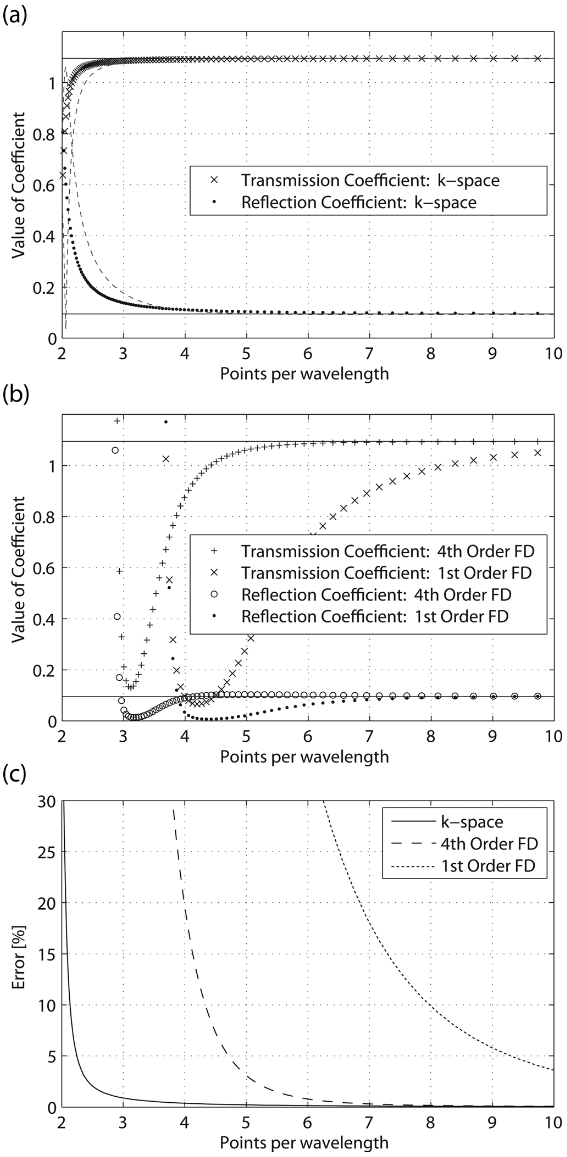


FIG. 2. (a) Transmission and reflection coefficients computed using the k-space model for a 10% step change in sound speed and density against the number of grid points per wavelength. The dashed lines show the results without the use of the staggered grid. (b) Analogous results when the spatial gradients are computed using a first-order and fourth-order accurate finite difference scheme. (c) Corresponding error in the transmission coefficient with the number of grid points used per wavelength.

requires around a 10-fold increase in the total number of grid points to achieve the same level of accuracy, while using a first-order scheme requires a 100-fold increase. This corresponds to a significant increase in the computer memory required for a given simulation. For a given CFL, this also increases the number of time steps required. Similar results have previously been shown for scattering problems in two and three dimensions.[6,42]

C. Nonlinear propagation in media with thermoviscous absorption

To investigate the accuracy of the k-space model for simulating nonlinear wave propagation in absorbing media, the propagation of a monochromatic plane wave was compared to the analytical solution of Burgers equation derived by Mendousse.[7,43] This solution accounts for thermoviscous absorption (which is proportional to frequency squared) for a source condition equal to $p = p_0 \sin(2\pi f_0 t)$. It is convenient to describe the contribution of nonlinearity to the shape of the waveform using the non-dimensional shock parameter σ. For a monochromatic plane wave this is defined as

$$\sigma = \frac{\beta p_0 2\pi f_0 x}{\rho_0 c_0^3}, \tag{12}$$

where x is the distance between the observation point and the source. A comparison between the k-space model and Mendousse's solution is shown in Fig. 3(a) for $\sigma = 1$, where $\beta = 4.8$, $p_0 = 5$ MPa, $f_0 = 1$ MHz, $\rho_0 = 1000$ kg/m^3, $c_0 = 1500$ m/s, and $\alpha_0 = 0.25$ dB MHz^{-2} cm^{-1}. The k-space discretization used 30 grid points per wavelength at f_0 (supporting at most 15 harmonics) and a CFL number of 0.3. The amplitudes of the first ten harmonics are shown in Fig. 3(b). There is a close agreement between the two models illustrating that acoustic absorption and cumulative nonlinear effects are correctly encapsulated. The corresponding waveform calculated using Eq. (11) with $\kappa = 1$ (equivalent to a leapfrog pseudospectral model) is also shown. In this case, additional phase error is noticeable near the maximum and minimum of the waveform. The corresponding errors as a function of the CFL number are shown in Fig. 3(b), where the least squares error metric is defined as

$$\text{error}[\%] = 100\frac{\sum\left(p_{k\text{-space}}(t) - p_{\text{mendousse}}(t)\right)^2}{\sum\left(p_{\text{mendousse}}(t)\right)^2}. \tag{13}$$

It is evident from this example that the k-space operator still significantly improves the accuracy of the solution, even when the governing equations include additional nonlinearity and absorption terms.

A second comparison is shown in Fig. 4 for a varying shock parameter again using 30 grid points per wavelength at f_0 and a CFL number of 0.3. For low values of the shock parameter, the range of spatial wave numbers supported by the computational grid is sufficient to accurately represent the waveform. Consequently, the least squares error is small. As the shock parameter is increased, harmonics with wavelengths smaller than that supported by the grid spacing are

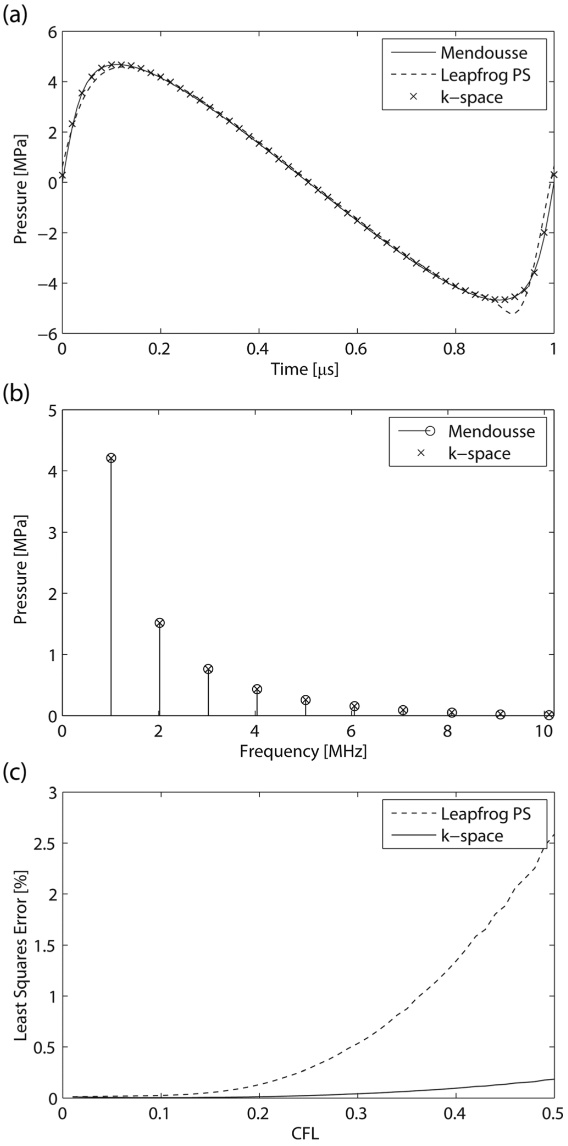


FIG. 3. (a) Comparison between Mendousse's analytical solution for the nonlinear propagation of a plane wave in a lossy medium and the nonlinear k-space model for a shock parameter of $\sigma = 1$. (b) Corresponding amplitudes of the first 10 harmonics. (c) Change in the least squares error with the size of the time step defined by the Courant–Friedrichs–Lewy (CFL) number.

generated. This creates an aliasing effect known as spectral blocking in which wavenumbers higher than the Nyquist limit are aliased to wavenumbers supported by the computational grid.[41] This effect is visible in Fig. 4(b) for $\sigma = 3$. In this case, the amplitudes of the generated harmonics no longer decay, and the energy at 15 MHz is erroneously greater than at 14 MHz due to aliasing.

For a given grid size, there are several possible strategies to overcome spectral blocking (see Ref. 41, and references therein). For systems with a quadratic nonlinearity,

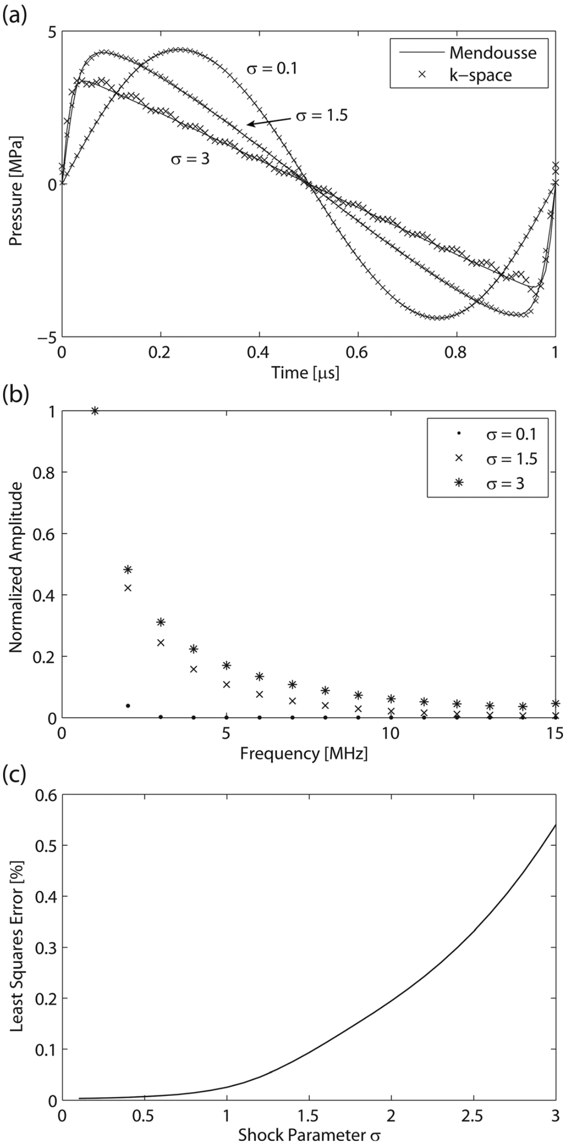


FIG. 4. (a) Comparison between Mendousse's analytical solution and the nonlinear *k*-space model for a varying shock parameter using 30 grid points per wavelength at 1 MHz. (b) Corresponding harmonic amplitudes calculated by the *k*-space model. (c) Change in the least squares error as a function of the shock parameter.

aliasing can be counteracted by increasing the grid size in each spatial dimension by a factor of $1/3$ and then filtering out the additional wavenumbers after each time step. For convective nonlinearity in the case of incompressible flow, aliasing errors can also be reduced by using the skew-symmetric form of the convective term. However, while these strategies can help minimize aliasing errors, the overall solution will still be inaccurate if there is significant energy at wavenumbers not supported by the computational grid.[41] This is not a problem specific to spectral methods, rather, it is applicable regardless of the chosen numerical method. In this case, if the accurate calculation of the total acoustic pressure field is the desired outcome, the best strategy is to monitor the computed wavenumbers for spectral blocking, and increase the number of grid points used in the simulation if significant aliasing is observed.

More generally, the accuracy of the *k*-space model is dependent on several parameters. First, the number of grid points used per wavelength will control whether the computational grid can support the propagation of the generated harmonics. In turn, the rate at which these harmonics are produced will depend on the shock parameter (for example, the source strength and the coefficient of nonlinearity), while the rate at which they are absorbed will depend on the power law absorption parameters. Finally, the CFL number will control the amount of unwanted numerical dispersion introduced by the finite difference time step, as well as the accuracy with which the nonlinearity and absorption terms in the pressure-density relation are computed.

D. Linear propagation in media with power law absorption

To investigate the accuracy of the power law absorption term for general absorption parameters, the pressure field produced by a rectangular piston in an absorbing medium was compared to the fast near-field method as implemented in the FOCUS toolbox.[44] This approach is comparable to evaluating the Rayleigh–Sommerfeld integral, but converges more rapidly by using an equivalent integral expression that removes numerical singularities. A comparison between the two models for a 6 mm × 6 mm rectangular piston driven by a 3 MHz sinusoid is shown in Fig. 5. To capture the rapid field variations close to the piston surface, the *k*-space model used four grid points per wavelength, with an overall computational grid size of 384 × 128 × 128 grid points. The medium parameters were set to $c_0 = 1510$ m/s, $\rho_0 = 1020$ kg/m^3, and $\alpha_0 = 0.25$ dB MHz^{-1} cm^{-1}. The beam pattern for the *k*-space model was produced by taking the maximum steady state pressure at each grid point. There is excellent agreement between the two models which confirms that absorption and near-field effects are correctly modeled.

E. Simulation of nonlinear ultrasound beam patterns

To illustrate the applicability of the developed nonlinear *k*-space model to ultrasound simulation more generally, the beam pattern produced by an Ultrasonix L9-4/38 linear array probe in a heterogeneous medium was investigated. This probe has 128 rectangular transducer elements with an element pitch of 304.8 μm, an elevation height of 6 mm, and an elevation focus of 19 mm. The beam pattern was computed in three dimensions using 32 active elements and an electronic focus of 15 mm. The computational grid used including the PML was 1024 × 512 × 512 grid points with a grid point spacing of 30.5 μm. This corresponds to a maximum frequency of 25.2 MHz at two grid points per wavelength, or 16.8 MHz at three grid points per wavelength (giving a computational domain size of 340 × 170 × 170 wavelengths at the maximum frequency). The transducer was driven by a five cycle tone burst with a center frequency of 5 MHz and an equivalent source pressure of 0.25 MPa per grid node of

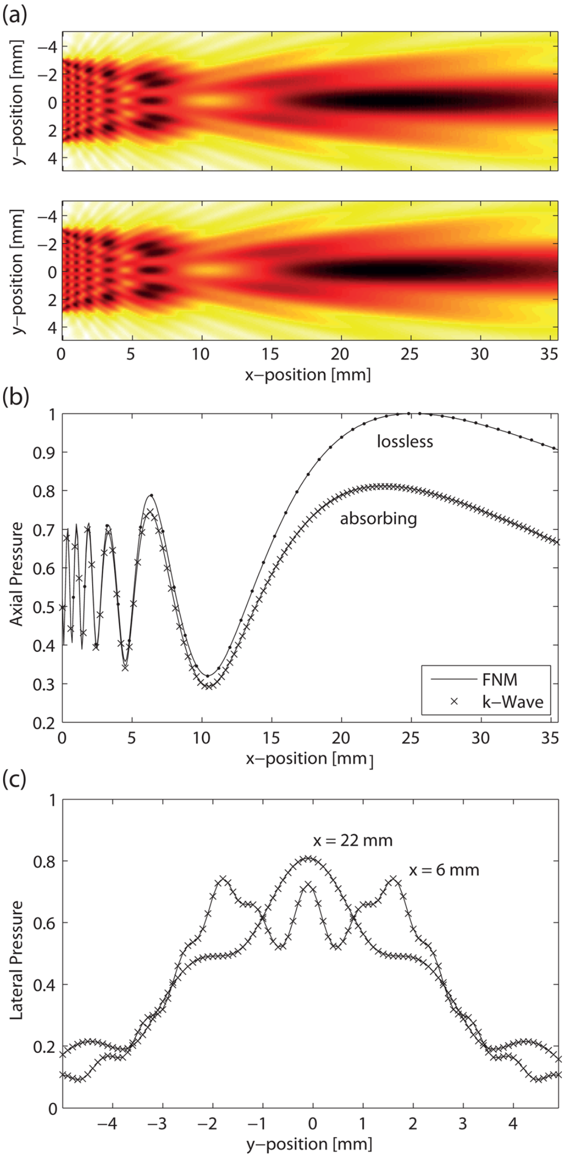


FIG. 5. (Color online) (a) Azimuth plane beam pattern generated by a rectangular piston in an homogeneous absorbing medium using the *k*-space method (top panel) and the fast near-field method (bottom panel). (b) Comparison of the pressure maximum along the transducer axis. (c) Comparison of the lateral pressure at $x = 6$ mm and $x = 22$ mm.

each transducer element. The input signal was assigned to the x-direction particle velocity (rather than the pressure) to mimic the directionality of the physical piezoelectric transducer elements. Each element was represented by 9 grid points in the azimuth direction (with a kerf width of 1 grid point) and 196 grid points in the elevation direction. The beamforming delays were calculated geometrically, and the elevation focus was modeled by applying additional beamforming delays across the grid points in the elevation direction. The CFL number was set to 0.3 giving a time step of 6 ns, and the simulation was run for 4000 time steps.

For the homogeneous medium, the properties were set to those of breast tissue, where $c_0 = 1510$ m/s, $\rho_0 = 1020$ kg/m^3, $B/A = 9.63$, $\alpha_0 = 0.75\,\mathrm{dB\,MHz^{-y}\,cm^{-1}}$, and $y = 1.5$.[1] For the heterogeneous medium, the sound speed and density maps were derived from a CT scan of a human neck. To simulate small scale heterogeneities, the values of the sound speed and density at each grid point were scaled by a Gaussian random variable with a mean of 1 and a standard deviation of 0.05. Each simulation took 7.5 h to run and used 27 GB of memory (using the Tyan server in the 48 GB configuration; details are given in the Appendix). The generated azimuth and elevation plane beam patterns are shown in Fig. 6. The total beam patterns were produced by taking the maximum value of the pressure recorded at each grid point, while the beam patterns at the second harmonic correspond to the relative spectral amplitudes at this frequency. When the medium is heterogeneous, the variations in the medium parameters alter both the shape and the position of the beam focus. By recording the acoustic signals reflected back to the active transducer elements, it is straightforward to extend the simulations to form B-mode ultrasound images.[36]

F. Comparison with other full-wave nonlinear models

The computational complexity of solving general nonlinear equations means only a limited number of three-dimensional full-wave models have previously been reported in the literature. Pinton *et al.*,[2] recently presented a solution to the heterogeneous Westervelt equation with a relaxation absorption term using a second-order-in-time, fourth-order-in-space finite difference method. Simulations using computational grid sizes on the order of 800 × 800 × 800 grid points were run on a distributed cluster with run times on the order of 32 h. Comparatively, the *k*-space pseudospectral method reduces the number of grid points and time steps required for the same level of accuracy.

Verweij and Huijssen[3,12] also recently presented an iterative method to solve the linear homogeneous wave equation with absorption, nonlinearity, and heterogeneity included as contrast source terms. This approach allows both the spatial and temporal fields to be sampled at the Nyquist limit (equivalent to a CFL number of 1). It also provides a mechanism for high frequency harmonics not supported by the computational grid to be removed via spatiotemporal filtering. However, a significant disadvantage is that the complete time history of the field data must be stored to allow the evaluation of the required convolutions. This considerably increases memory requirements. For the example discussed in Sec. V E, even after accounting for a reduction in the total number of grid points by a factor of 8 (assuming the *k*-space model requires a conservative four grid points per wavelength at the maximum frequency of interest) and using a CFL number of 1, the storage of the time history of one field variable in single precision still requires 75 GB of memory.

VI. SUMMARY

A set of coupled first-order equations valid for modeling nonlinear wave propagation in heterogeneous media with power law absorption is derived. The additional terms

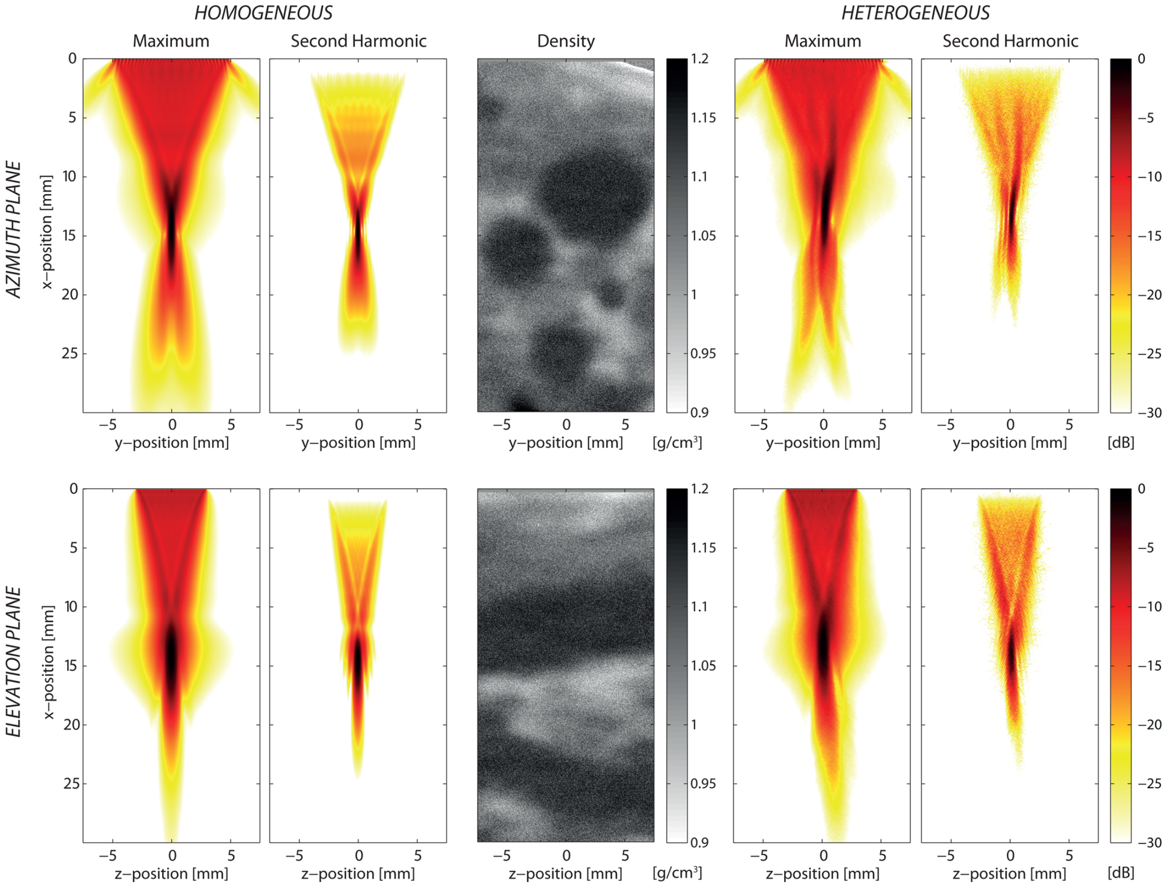


FIG. 6. (Color online) Normalized azimuth and elevation plane beam patterns generated by a clinical linear array ultrasound transducer for a homogeneous medium and a heterogeneous medium using a density map derived from a CT scan.

accounting for cumulative nonlinear effects and power law absorption are expressed as spatial gradients which makes them efficient to numerically encode. The derived governing equations are then discretized using the *k*-space pseudospectral method. The use of the *k*-space operator significantly reduces the phase error introduced by the finite difference time step, allowing larger time steps to be taken for the same level of accuracy. The use of the Fourier-collocation spectral method similarly improves the accuracy of the spatial gradient calculations which relaxes the requirement for dense computational grids compared to conventional finite difference methods. A number of numerical examples are given to illustrate the accuracy of the model. The utility of the nonlinear *k*-space model is then demonstrated via the three-dimensional simulation of the beam pattern from a clinical ultrasound transducer in both homogeneous and heterogeneous media. Compared to previous ultrasound models based on the KZK equation, the current model does not have any restrictions on the directionality or spatial variation of the sound waves. This facilitates arbitrary full-wave simulations of cumulative nonlinear effects in tissue-realistic media. The model is particularly relevant to the simulation of diagnostic and therapeutic ultrasound fields in heterogeneous media, as well as the generation of full-wave harmonic ultrasound images.

ACKNOWLEDGMENTS

This work was supported in part by the Australian Research Council/Microsoft Linkage Project LP100100588 and the Engineering and Physical Sciences Research Council, UK. The authors would like to thank Anthony Harker, Mustafa Tumen, Nimalan Nandapalan, and Trond Varslot for useful discussion.

APPENDIX: COMPUTER IMPLEMENTATION

The discrete equations described in Sec. IV B were implemented in C++ as an extension to the open source K-WAVE toolbox.[40] The codes were optimized to run using a Tyan server (MiTAC, Taipei, Taiwan) with two six-core Intel Xeon X5650 processors. To maximize computational efficiency, several stages of code optimization were performed.[45] First, the 3D FFTs were computed using the real-to-complex FFT from the FFTW library. Compared to the complex-to-complex FFT, this reduced the compute time

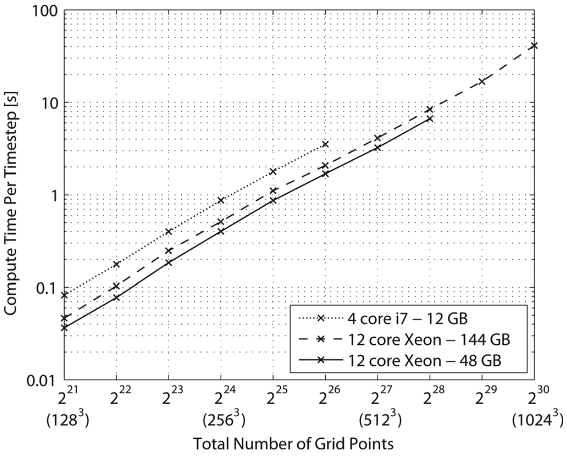


FIG. 7. Compute times per time step for the nonlinear *k*-space model for different 3D grid sizes. The time complexity is on the order of $O(n \log n)$ where $n = Nx \times Ny \times Nz$.

and memory associated with the FFT by nearly 50%. Second, to save memory bandwidth, all operations were computed in single precision. Parameters such as the PML and wavenumber operators were stored as 1D vectors and virtually replicated to 3D as needed via indexing. Third, element-wise operations were parallelized using OPENMP and then optimized using streaming SIMD extensions (SSE). Multiple mathematical operations were applied to each grid point where possible to maximize temporal data locality. Finally, as the Tyan server is based on the non-uniform memory access architecture, policies were implemented to bind threads to cores and allocate memory to nearby memory locality domains.[45]

The compute times per time step for a range of computational grid sizes are shown in Fig. 7. The three curves correspond to three different computer configurations: a desktop computer with a four-core Intel Core i7 950 processor and 12 GB of DDR3 RAM, and the Tyan server with either 144 GB (18 × 8 GB) or 48 GB (12 × 4 GB) of DDR3 RAM. The performance difference between the two memory configurations for the Tyan server is due to a reduction in the memory speed from 1333 MHz to 1066 MHz when the memory channels are fully populated. The memory usage for a given grid size can be estimated by

$$\text{memory usage[GB]} \approx \frac{21NxNyNz + 9\frac{Nx}{2}NyNz}{1024^3/4}, \qquad \text{(A1)}$$

where Nx, Ny, and Nz are the grid sizes in the x, y, and z directions. The first term accounts for 21 real matrices and the second term accounts for 3 real and 3 complex matrices in the spatial Fourier domain. The relatively large number of 3D matrices is required to store the heterogeneous material parameters, field values, and temporary results.

For a computational grid with 512 × 512 × 512 grid points, the overall performance of the *k*-space model running on the Tyan server (in the 48 GB memory configuration) was approximately 17 GFLOPS. In this case, the maximum achievable performance was limited by the available memory bandwidth as many of the element-wise operations have limited or no data re-usage so benefit little from the availability of cache. For comparison, the LINPACK benchmark from the Intel Math Kernel Library (which is used to test the theoretical peak performance of Intel processors) had a performance of 118 GFLOPS, while the 3D real-to-complex FFT in isolation had a performance of 42 GFLOPS. For a representative simulation, around 60% of the total computation time is spent performing the forward and inverse FFT.

[1] T. L. Szabo, *Diagnostic Ultrasound Imaging* (Elsevier, Burlington, 2004), pp. 4–6.

[2] G. F. Pinton, J. Dahl, S. Rosenzweig, and G. E. Trahey, "A heterogeneous nonlinear attenuating full-wave model of ultrasound," IEEE Trans. Ultrason. Ferroelectr. Freq. Control **56**, 474–488 (2009).

[3] J. Huijssen and M. D. Verweij, "An iterative method for the computation of nonlinear, wide-angle, pulsed acoustic fields of medical diagnostic transducers," J. Acoust. Soc. Am. **127**, 33–44 (2010).

[4] V. W. Sparrow and R. Raspet, "A numerical method for general finite amplitude wave propagation in two dimensions and its application to spark pulses," J. Acoust. Soc. Am. **90**, 2683–2691 (1991).

[5] J. A. Jensen, "A model for the propagation and scattering of ultrasound in tissue," J. Acoust. Soc. Am. **89**, 182–190 (1991).

[6] M. Tabei, T. D. Mast, and R. C. Waag, "A k-space method for coupled first-order acoustic propagation equations," J. Acoust. Soc. Am. **111**, 53–63 (2002).

[7] *Nonlinear Acoustics*, edited by M. F. Hamilton and D. T. Blackstock (Acoustical Society of America, Melville, 2008), pp. 1–455.

[8] G. Taraldsen, "A generalized Westervelt equation for nonlinear medical ultrasound," J. Acoust. Soc. Am. **109**, 1329–1333 (2001).

[9] F. Coulouvrat, "New equations for nonlinear acoustics in a low Mach number and weakly heterogeneous atmosphere," Wave Motion **49**, 50–63 (2012).

[10] I. M. Hallaj, R. O. Cleveland, and K. Hynynen, "Simulations of the thermo-acoustic lens effect during focused ultrasound surgery," J. Acoust. Soc. Am. **109**, 2245–2253 (2001).

[11] Y. Jing and R. O. Cleveland, "Modeling the propagation of nonlinear three-dimensional acoustic beams in inhomogeneous media," J. Acoust. Soc. Am. **122**, 1352–1364 (2007).

[12] M. D. Verweij and J. Huijssen, "A filtered convolution method for the computation of acoustic wave fields in very large spatiotemporal domains," J. Acoust. Soc. Am. **125**, 1868–1878 (2009).

[13] Y. Jing and G. T. Clement, "A k-space method for nonlinear wave propagation," arXiv:1105.2210.

[14] M. V. Averyanov, V. A. Khokhlova, O. A. Sapozhnikov, P. Blanc-Benon, and R. O. Cleveland, "Parabolic equation for nonlinear acoustic wave propagation in inhomogeneous moving media," Acoust. Phys. **52**, 623–632 (2006).

[15] D. T. Blackstock, "Generalized Burgers equation for plane waves," J. Acoust. Soc. Am. **77**, 2050–2053 (1985).

[16] T. L. Szabo, "Time domain nonlinear wave equations for lossy media," in *Advances in Nonlinear Acoustics: Proceedings of the 13th International Symposium on Nonlinear Acoustics* (World Scientific, Singapore, 1993), pp. 89–94.

[17] T. L. Szabo, "Time domain wave equations for lossy media obeying a frequency power law," J. Acoust. Soc. Am. **96**, 491–500 (1994).

[18] J. Tavakkoli, D. Cathignol, R. Souchon, and O. A. Sapozhnikov, "Modeling of pulsed finite-amplitude focused sound beams in time domain," J. Acoust. Soc. Am. **104**, 2061–2072 (1998).

[19] J. Wojcik, "Conservation of energy and absorption in acoustic fields for linear and nonlinear propagation," J. Acoust. Soc. Am. **104**, 2654 (1998).

[20] W. Chen and S. Holm, "Fractional Laplacian time-space models for linear and nonlinear lossy media exhibiting arbitrary frequency power-law dependency," J. Acoust. Soc. Am. **115**, 1424–1430 (2004).

[21] W. Chen and S. Holm, "Fractional Laplacian, Levy stable distribution, and time-space models for linear and nonlinear frequency-dependent lossy

media," Technical Report, Research Report of Simula Research Laboratory (2002).

[22] B. E. Treeby and B. T. Cox, "Modeling power law absorption and dispersion for acoustic propagation using the fractional Laplacian," J. Acoust. Soc. Am. **127**, 2741–2748 (2010).

[23] B. E. Treeby and B. T. Cox, "A k-space Greens function solution for acoustic initial value problems in homogeneous media with power law absorption," J. Acoust. Soc. Am. **129**, 3652–3660 (2011).

[24] H. A. H. Jongen, J. M. Thijssen, M. van den Aarssen, and W. A. Verhoef, "A general model for the absorption of ultrasound by biological tissues and experimental verification," J. Acoust. Soc. Am. **79**, 535–540 (1986).

[25] A. I. Nachman, J. F. Smith III, and R. C. Waag, "An equation for acoustic propagation in inhomogeneous media with relaxation losses," J. Acoust. Soc. Am. **88**, 1584–1595 (1990).

[26] R. O. Cleveland, M. F. Hamilton, and D. T. Blackstock, "Time-domain modeling of finite-amplitude sound in relaxing fluids," J. Acoust. Soc. Am. **99**, 3312–3318 (1996).

[27] S. P. Nasholm and S. Holm, "Linking multiple relaxation, power-law attenuation, and fractional wave equations," J. Acoust. Soc. Am. **130**, 3038–3045 (2011).

[28] M. J. Lighthill, "Viscosity effects in sound waves of finite amplitudes," in *Surveys in Mechanics*, edited by G. K. Batchelor and R. M. Davies (Cambridge University Press, Cambridge, 1956), pp. 250–351.

[29] In a fluid model at equilibrium, a heterogeneous ambient density physically requires a body force to support it. In soft tissue this could be provided, for example, by stresses in the extracellular matrix. As the fluid is stationary in the ambient state, this body force must be matched by a gradient in the ambient pressure, where $\nabla p_0 = \rho_0 \mathbf{f}$. Because these terms exactly cancel, they are not included in the dynamic momentum equation given in Eq. (2a).

[30] A. D. Pierce, "Mathematical theory of wave propagation," in *Handbook of Acoustics*, edited by M. J. Crocker (Wiley, New York, 1998), pp. 21–37.

[31] F. Prieur and S. Holm, "Nonlinear acoustic wave equations with fractional loss operators," J. Acoust. Soc. Am. **130**, 1125–1132 (2011).

[32] Y. Jing, D. Shen, and G. T. Clement, "Verification of the Westervelt equation for focused transducers," IEEE Trans. Ultrason. Ferroelectr. Freq. Control **58**, 1097–1101 (2011).

[33] S. I. Aanonsen, T. Barkve, J. N. Tjotta, and S. Tjotta, "Distortion and harmonic generation in the nearfield of a finite amplitude sound beam," J. Acoust. Soc. Am. **75**, 749–768 (1984).

[34] M. F. Hamilton and D. T. Blackstock, "On the linearity of the momentum equation for progressive plane waves of finite amplitude," J. Acoust. Soc. Am. **88**, 2025–2026 (1990).

[35] Subtly, this means the acoustic density calculated by the discrete equations is not exactly equal to the true acoustic density as defined in the general conservation equations. However, because the acoustic density is not generally used for output, this difference does not affect the accuracy of the simulations.

[36] B. E. Treeby, M. Tumen, and B. T. Cox, "Time domain simulation of harmonic ultrasound images and beam patterns in 3D using the k-space pseudospectral method," in *Medical Image Computing and Computer-Assisted Intervention, Part I* (Springer, Heidelberg, 2011), Vol. 6891, pp. 363–370.

[37] T. D. Mast, L. P. Souriau, D.-L. D. Liu, M. Tabei, A. I. Nachman, and R. C. Waag, "A k-space method for large-scale models of wave propagation in tissue," IEEE Trans. Ultrason. Ferroelectr. Freq. Control **48**, 341–354 (2001).

[38] J.-P. Berenger, "Three-dimensional perfectly matched layer for the absorption of electromagnetic waves," J. Comput. Phys. **127**, 363–379 (1996).

[39] B. T. Cox, S. Kara, S. R. Arridge, and P. C. Beard, "k-space propagation models for acoustically heterogeneous media: Application to biomedical photoacoustics," J. Acoust. Soc. Am. **121**, 3453–3464 (2007).

[40] B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," J. Biomed. Opt. **15**, 021314 (2010).

[41] J. P. Boyd, *Chebyshev and Fourier Spectral Methods* (Dover, Mineola, NY, 2001), pp. 202–221.

[42] Y. Jing, F. C. Meral, and G. T. Clement, "Time-reversal transcranial ultrasound beam focusing using a k-space method," Phys. Med. Biol. **57**, 901–917 (2012).

[43] J. S. Mendousse, "Nonlinear dissipative distortion of progressive sound waves at moderate amplitudes," J. Acoust. Soc. Am. **25**, 51–54 (1953).

[44] R. J. McGough, "Rapid calculations of time-harmonic nearfield pressures produced by rectangular pistons," J. Acoust. Soc. Am. **115**, 1934–1941 (2004).

[45] J. Jaros, B. E. Treeby, and A. P. Rendell, "Use of multiple GPUs on shared memory multiprocessors for ultrasound propagation simulations," in *10th Australasian Symposium on Parallel and Distributed Computing*, edited by J. Chen and R. Ranjan, ACS (2012), Vol. 127, pp. 43–52.

A.2 Validation of Fluid Model

Wang, K.; Teoh, E.; **Jaros, J.**; Treeby. B.E.: Modelling nonlinear ultrasound propagation in absorbing media using the k-Wave toolbox: Experimental validation. In *IEEE International Ultrasonics Symposium, IUS*. Dresden, DE: Institute of Electrical and Electronics Engineers. 2012. ISBN 978-1-4673-4561-3. ISSN 1948-5719. pp. 523–526. doi:10.1109/ULTSYM.2012.0130.

10.1109/ULTSYM.2012.0130

Modelling Nonlinear Ultrasound Propagation in Absorbing Media using the k-Wave Toolbox: Experimental Validation

Kejia Wang*, Emily Teoh*, Jiri Jaros†, and Bradley E. Treeby*‡
*Research School of Engineering, College of Engineering and Computer Science
†Research School of Computer Science, College of Engineering and Computer Science
The Australian National University, Canberra, ACT 0200, Australia
‡Email: bradley.treeby@anu.edu.au

***Abstract*—The simulation of nonlinear ultrasound waves in biological tissue has a number of important applications. However, this is a computationally intensive task due to the large domain sizes required for many problems of practical interest. Recently, an efficient full-wave nonlinear ultrasound model was developed and released as part of the open source k-Wave Toolbox. Here, this model is validated using a series of experimental measurements made with a linear diagnostic ultrasound probe and a membrane hydrophone. Measurements were performed in both deionised water and olive oil, the latter exhibiting power law absorption characteristics similar to human tissue. Steering angles of 0° and 20° were also tested, with propagation distances on the order of hundreds of acoustic wavelengths. The simulated and experimental results show a close agreement in both the time and frequency domains. These results demonstrate the quantitative validity of performing nonlinear ultrasound simulations using the k-Wave toolbox.**

I. Introduction

Accurately simulating the propagation of nonlinear ultrasound waves through soft biological tissue is important for a number of applications, including equipment design, dosimetry, and computer-aided diagnosis. Currently, most nonlinear ultrasound simulations are based on variations of the Khokhlov-Zabolotskaya-Kuznetsov (KZK) or Burgers equations [1], [2]. Whilst these models are accurate for many practical situations [3], they are restricted to modelling directional sound beams, and often only consider one-way wave propagation in homogeneous media [4]. Recently, a full-wave nonlinear ultrasound model based on the k-space pseudospectral method was developed and released as part of the open-source k-Wave Acoustics Toolbox [5], [6]. This model can account for the propagation of nonlinear ultrasound waves in generally heterogeneous media including power law acoustic absorption, with no restrictions on the directionality of the waves. The objective of the current work was to validate this model using experimental measurements of the ultrasound fields produced by a diagnostic ultrasound transducer. Comparisons were made with both on and off axis beam patterns and in absorbing fluids. This builds on earlier work to validate k-Wave for modelling problems in photoacoustics when the wave propagation is assumed to be linear and lossless [7].

II. Methods

A. Experimental Measurements

The experiments were performed in a 40 × 40 × 60 cm test tank using a two-axis computer controlled positioning system with an accuracy of ± 2.5 μm (Precision Acoustics, Dorchester, UK). The two motorised axes were aligned in either the x-y plane to acquire data perpendicular to the beam axis, or the x-z plane to acquire data along the beam axis (see Fig. 1). In both cases, the third axis was controlled by a manual translation stage with a positioning accuracy of ± 50 μm. The positioning arm was used to hold a calibrated PVDF membrane hydrophone with a thickness of 15 μm and a 0.4 mm active element (Precision Acoustics, Dorchester, UK). Time domain signals at each spatial position were automatically acquired using a digital storage oscilloscope with a 200 MHz bandwidth controlled by the positioning software (DSO-X 3024A, Agilent Technologies, Santa Clara, CA). The signals were acquired using a sampling frequency of 2 GHz and 512 averages.

A SONIX RP diagnostic ultrasound scanner (Ultrasonix, British Columbia, Canada) with an L9-4/38 linear probe was used to generate the ultrasound waves. The probe had 128 rectangular elements with an element height of 6 mm, element width of 0.2698 mm, kerf width of 0.035 mm, and a fixed elevation focus of 19 mm. The transmit pulse was programmed using the Texo SDK with 32 active elements, a transmit frequency of 5 MHz, a pulse shape of `'+-+-+-'`, and a transmit power of 12 V. The transducer was enclosed in a plastic probe cover (Cone Instruments, Solon, OH) with a thin covering of ultrasound gel, and held in the tank using a probe holder (see Fig. 1).

Two different experiments were performed. In the first experiment, time domain pressure signals in four planes perpendicular to the beam axis were acquired (the x-y plane shown in Fig. 1). The planes were 10 × 15 mm in size and spaced 10 mm apart in the z-direction, with the first plane 1.3 mm from the probe face (this distance was measured using the SONIX RP imaging software). The electronic transducer focus distance in the beam plane was set to 21.3 mm to coincide with the third measurement plane. Within each plane, 100 × 150 waveforms were acquired with a step size of 100 μm. The acquisition of each waveform and subsequent movement of the hydrophone (including a 250 ms settle time) took just under 1 second, with each measurement plane taking on the order of 4 hours to complete. The experiment was repeated using both deionised water and olive oil (Extra-Virgin Olive Oil, Olives Direct, Nambour, Australia) at room temperature (22^{o}C ± 1^{o}C). Olive oil was chosen as a test fluid as it exhibits power law acoustic absorption characteristics similar to those observed in human tissue (see Table I).

In the second experiment, time domain pressure signals within the beam plane were acquired. The ultrasound probe was rotated 90^o about the z-axis compared to the alignment shown in Fig. 1, and the hydrophone was moved through the x-z plane. The 2D scan plane was aligned with the beam plane by progressively moving the hydrophone to the beam centre in the y-direction (using the manual translation stage) at two different z-positions (5 mm and 30 mm from the transducer face). The experiment was performed in deionised water using the same transmit pulse as the first experiment, and repeated for electronic steering angles of 0^o and 20^o. For a steering angle of 0^o, the scan area was 10 × 30 mm, and for 20^o it was 20 × 30 mm. In both cases, the step size was 100 μm, and the electronic focus distance was 20 mm.

B. k-Wave Simulations

The simulations were performed using the nonlinear ultrasound model included the k-Wave Toolbox as described in [5]. This model is based on the iterative solution of three coupled first-order partial differential equations which describe the propagation of nonlinear ultrasound waves in heterogeneous media with power law acoustic absorption. (These equations are equivalent to a generalised form of the Westervelt equation.) The complete spatial domain in 3D is discretised, and the solution for the pressure and particle velocity fields everywhere within the domain are calculated in a time-stepping fashion. This is different to one-way models (for example the KZK Texas code [1]) which propagate the complete time trace of the pressure signals between parallel planes. The governing equations in k-Wave are solved using the k-space pseudospectral method [8]. Compared to finite-difference and finite-element methods, this allows for much coarser grid spacings and larger time steps for the same degree of accuracy [8], [5].

The experimental signals measured in the x-y plane closest to the ultrasound probe were used as the input to the simulations. The signals were temporally down-sampled (using an anti-aliasing filter) by a factor of 10 to give a time step of 5 ns, and spatially up-sampled by a factor of 4 using nearest neighbour interpolation to give a grid point spacing of 25 μm. These parameters were used to define the simulation grid, giving a maximum supported frequency of 30 MHz and a Courant-Friedrichs-Lewy (CFL) number of 0.3. The total grid size was 432 × 640 × 1250 grid points (10 × 15 × 30 mm) including a perfectly matched layer (PML) of 16 × 20 × 25 grid points on each side of the domain. The PML sizes were chosen to give grid dimensions with small prime factors to minimise compute times (the model makes heavy use of the FFT which is fastest for grid dimensions that are powers of two or have small prime factors).

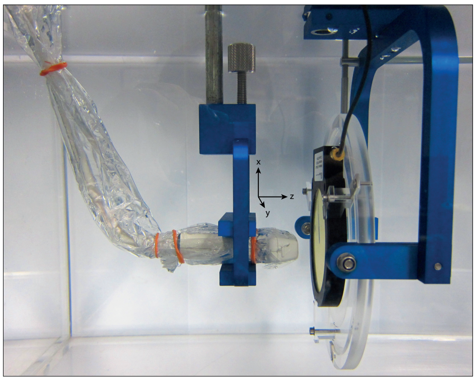


Fig. 1. Photograph of the ultrasound test tank showing the linear ultrasound probe, membrane hydrophone, and computer controlled positioning system.

TABLE I
ACOUSTIC PROPERTIES OF WATER AND OLIVE OIL USED FOR THE K-WAVE SIMULATIONS. VALUES ARE TAKEN FROM: * [9] † [10] ‡ [11] § [12]

	c_0 [m/s]	ρ [kg/m^3]	B/A	α_0 [dB/(MHz^{y}cm)]	y
water	1482*	1000*	4.96*	2.17e-3*	2*
olive oil	1446†	916‡	11.1§	0.13†	1.68†

The input pressure signals were assigned to `source.p` and enforced as a time-varying Dirichlet boundary condition over a 10 × 15 mm plane within the computational grid by setting `source.p_mode = 'dirichlet'`. The acoustic properties of the medium were defined to be homogeneous using the book values given in Table I. The sensor mask used to record the output from the simulations was defined to be the x-y and x-z planes matching the size and position of the scan planes used in the experiments.

The simulations were performed in single precision using C++ versions of `kspaceFirstOrder3D` written for shared memory and distributed memory platforms [5]. The shared memory simulations were run on a Tyan server with two six-core Intel Xeon X5650 processors [5]. The distributed memory simulations were run on the VAYU supercomputer (based on Sun X6275 blades) at the NCI national facility. Each simulation took between 2 hours (distributed memory code running on 256 cores) and 13 hours (shared memory code running on 12 cores), and used 23 GB of memory.

III. RESULTS

A comparison of the waveforms in the centre of each plane (i.e., along the beam axis) for the first experiment in deionised water is shown in Fig. 2(a). The left panels show the time domain pressure signals, and the right panels show the corresponding amplitude spectrums. The distances given are relative to the input plane, and the time domain signals have been aligned to correct for minor errors in the sound speed used in the simulations. The pressure is highest in the beam focus, and the wave steepens with distance due to cumulative nonlinear effects. There is a very close agreement between the experimental data and the simulated results in absolute units. No numerical dispersion is evident in the simulated results, even for the final measurement plane which is 30 mm from the input. This distance corresponds to 100 acoustic wavelengths at the transmit frequency, and 600 wavelengths at the highest modelled harmonic.

The equivalent results for the experiments in olive oil are shown in Fig. 2(b). For these experiments, the sensitivity of the hydrophone was significantly increased (using the standard calibration data, the pressure magnitudes measured in the plane closest to the ultrasound probe were nearly 3 times greater in olive oil compared to water for the same transmit settings). This change in sensitivity might be due to improved coupling between the hydrophone and the olive oil, or to viscous heating within the boundary layer between the oil and the membrane (further analysis is required). To account for this change in sensitivity, an additional calibration factor was introduced to normalise the maximum pressure measured in the plane closest to the probe in olive oil against the equivalent maximum pressure measured in deionised water.

There is again a good agreement between the experimental data and the simulated results in both the time and frequency domains. Compared to the measurements in deionised water, the increased acoustic absorption in olive oil results in a decrease in the pressure magnitudes. In particular, the higher frequency harmonics in the final measurement plane have been noticeably diminished. The slightly larger deviation between the results in the beam focus might be due to a weak thermal lensing effect caused by localised heating of the olive oil through acoustic absorption [13]. Other possible sources of error include uncertainties in the hydrophone calibration, uncertainties in the material properties, small alignment errors, and slight changes in temperature over the course of the experiment. This experiment provides a realistic test of the numerical model, as the acoustic absorption in olive oil is similar to that in soft biological tissue. The ability to directly and efficiently model acoustic absorption following a frequency power law is one of the strengths of the model used in k-Wave [14].

A comparison of the beam patterns recorded in the azimuth plane of the transducer is shown in Fig. 3. The total beam patterns were extracted using the integral of the amplitude spectrum at each spatial position, while the beam patterns at the fundamental and second harmonic correspond to the relative spectral amplitudes at 5 and 10 MHz. For a steering angle of 0°, there is a good agreement between the simulated and experimental results. In particular, the interference fringes at the fundamental frequency, and the spatial extent and magnitude of the second harmonic are closely matched. Similar results are obtained for a steering angle of 20°. In both cases, the small variations can be attributed to difficulties in aligning the scan plane exactly through the centre of the beam plane. In the experiment, the scan plane was aligned by progressively moving to the centre of the beam at varying distances from the transducer face. However, this approach is unable to account for angular misalignment about the beam axis. In the simulation, the scan plan was chosen to be the Cartesian plane in the x-z direction intersecting the beam focus. However, if the measured x-y plane used for the simulation input is not exactly perpendicular to this, the results close to the transducer face will be misaligned.

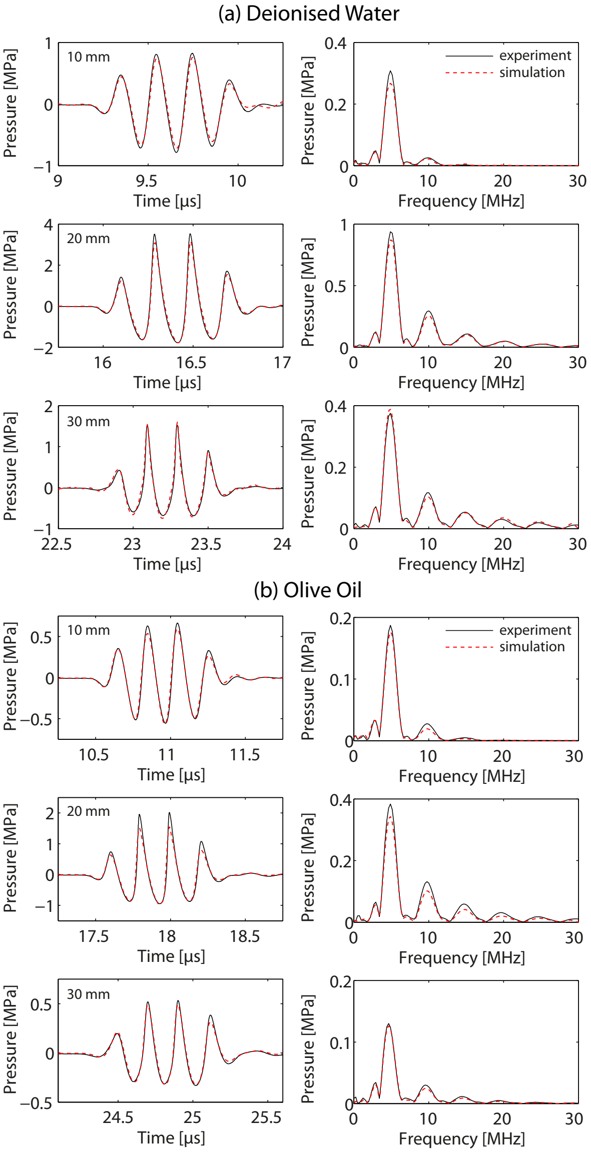


Fig. 2. Comparison of experimental (solid black line) and simulated (dashed red line) pressure signals along the axis of an ultrasound beam produced by a diagnostic ultrasound probe in (a) deionised water, and (b) olive oil at 22^{o}C.

IV. Conclusion

This paper reports the first series of experiments used to validate the accuracy of the nonlinear ultrasound model in the k-Wave Toolbox. Experiments were performed in deionised water and olive oil using a linear ultrasound probe. Beam patterns for steering angles of 0° and 20° were also compared. There was a close agreement between the simulated and experimental results in both the time and frequency domains. These results demonstrate the quantitative validity of performing ultrasound simulations using the k-Wave toolbox. Future experiments will compare results for a wider range of transducers and steering angles, and for heterogeneous media.

Acknowledgment

This work was supported in part by the Australian Research Council/Microsoft Linkage Project LP100100588, and computational resources from the NCI National Facility at the Australian National University (ANU). KW is currently at the University of Auckland and was supported by a Summer Research Scholarship at the ANU. The authors would like to thank Ben Cox for useful discussion.

References

[1] R. O. Cleveland, M. F. Hamilton, and D. T. Blackstock, "Time-domain modeling of finite-amplitude sound in relaxing fluids," *J. Acoust. Soc. Am.*, vol. 99, no. 6, pp. 3312–3318, 1996.

[2] T. Varslot and G. Taraldsen, "Computer simulation of forward wave propagation in soft tissue," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 52, no. 9, pp. 1473–1482, 2005.

[3] M. A. Averkiou and R. O. Cleveland, "Modeling of an electrohydraulic lithotripter with the KZK equation." *J. Acoust. Soc. Am.*, vol. 106, no. 1, pp. 102–112, 1999.

[4] G. F. Pinton and G. E. Trahey, "A comparison of time-domain solutions for the full-wave equation and the parabolic wave equation for a diagnostic ultrasound transducer." *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 55, no. 3, pp. 730–733, 2008.

[5] B. E. Treeby, J. Jaros, A. P. Rendell, and B. T. Cox, "Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method," *J. Acoust. Soc. Am.*, vol. 131, no. 6, pp. 4324–4336, 2012.

[6] B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *J. Biomed. Opt.*, vol. 15, no. 2, p. 021314, 2010.

[7] B. T. Cox, J. Laufer, K. Kostli, and P. C. Beard, "Experimental validation of photoacoustic k-space propagation models," in *Proc. SPIE*, vol. 5320, 2004, pp. 238–248.

[8] M. Tabei, T. D. Mast, and R. C. Waag, "A k-space method for coupled first-order acoustic propagation equations," *J. Acoust. Soc. Am.*, vol. 111, no. 1, pp. 53–63, 2002.

[9] T. L. Szabo, *Diagnostic Ultrasound Imaging*. Burlington: Elsevier, 2004.

[10] B. E. Treeby, B. T. Cox, E. Z. Zhang, S. K. Patch, and P. C. Beard, "Measurement of broadband temperature-dependent ultrasonic attenuation and dispersion using photoacoustics," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 56, no. 8, pp. 1666–1676, 2009.

[11] J. N. Coupland and D. J. McClements, "Physical properties of liquid edible oils," *J. Am. Oil Chem. Soc.*, vol. 74, no. 12, pp. 1559–1564, 1997.

[12] E. C. Everbach and R. E. Apfel, "An interferometric technique for B/A measurement," *J. Acoust. Soc. Am.*, vol. 98, no. 6, pp. 3428–34 328, 1995.

[13] I. M. Hallaj, R. O. Cleveland, and K. Hynynen, "Simulations of the thermo-acoustic lens effect during focused ultrasound surgery," *J. Acoust. Soc. Am.*, vol. 109, no. 5, pp. 2245–2253, 2001.

[14] B. E. Treeby and B. T. Cox, "Modeling power law absorption and dispersion for acoustic propagation using the fractional Laplacian," *J. Acoust. Soc. Am.*, vol. 127, no. 5, pp. 2741–2748, 2010.

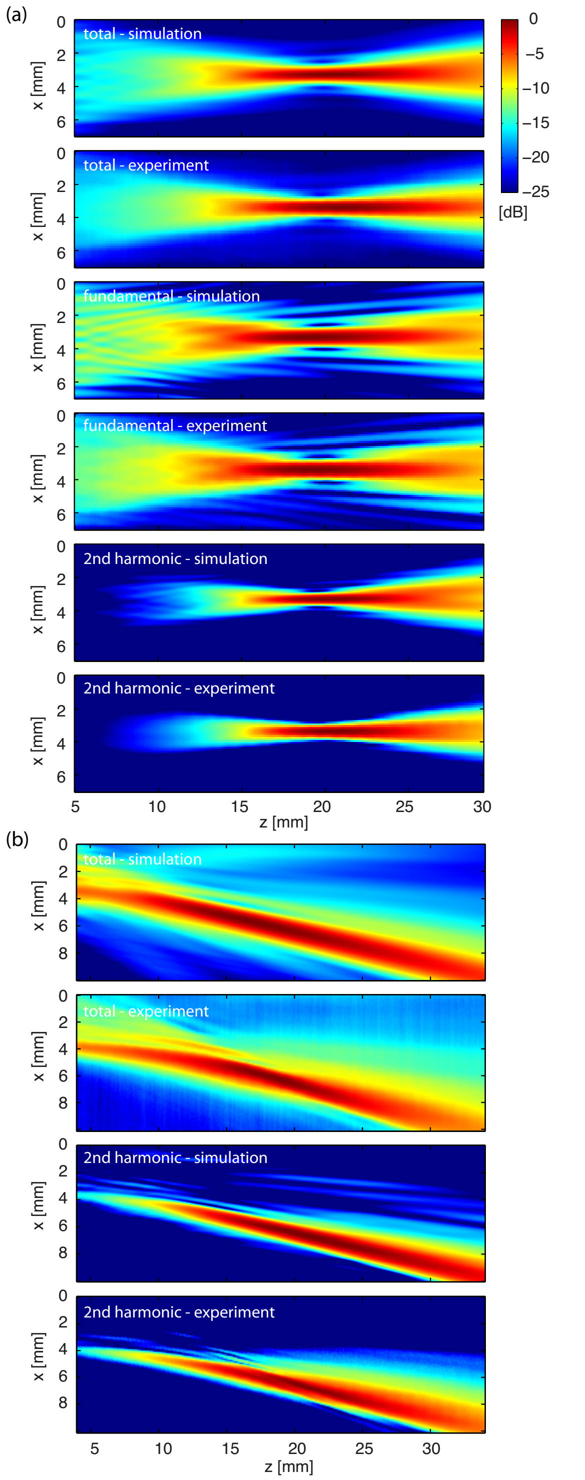


Fig. 3. Experimental and simulated beam patterns produced by a diagnostic ultrasound probe in deionised water for steering angles of (a) 0° and (b) 20°.

A.3 Derivation of Elastic Model

Treeby, B. E.; **Jaros, J.**; Rohrbach, D.; Cox. B. T: Modelling Elastic Wave Propagation Using the k-Wave MATLAB Toolbox. In *IEEE International Ultrasonics Symposium.* no. 5. 2014: pp. 146–149. ISSN 1948-5719. doi:10.1109/ULTSYM.2014.0037.

10.1109/ULTSYM.2014.0037

Modelling Elastic Wave Propagation Using the k-Wave MATLAB Toolbox

Bradley E. Treeby*†, Jiri Jaros‡, Daniel Rohrbach§, and B. T. Cox*
*Department of Medical Physics and Biomedical Engineering, University College London, London, UK
‡Faculty of Information Technology, Brno University of Technology, Brno, CZ
§Lizzi Center for Biomedical Engineering, Riverside Research, New York, NY, USA
†Email: b.treeby@ucl.ac.uk

***Abstract*—A new model for simulating elastic wave propagation using the open-source k-Wave MATLAB Toolbox is described. The model is based on two coupled first-order equations describing the stress and particle velocity within an isotropic medium. For absorbing media, the Kelvin-Voigt model of viscoelasticity is used. The equations are discretised in 2D and 3D using an efficient time-stepping pseudospectral scheme. This uses the Fourier collocation spectral method to compute spatial derivatives and a leapfrog finite-difference scheme to integrate forwards in time. A multi-axial perfectly matched layer (M-PML) is implemented to allow free-field simulations using a finite-sized computational grid. Acceleration using a graphics processing unit (GPU) is supported via the MATLAB Parallel Computing Toolbox. An overview of the simulation functions and their theoretical and numerical foundations is described.**

I. Introduction

The simulation of elastic wave propagation has many applications in ultrasonics, including the classification of bone diseases and non-destructive testing [1]. In biomedical ultrasound in particular, elastic wave models have been used to investigate the propagation of ultrasound in the skull and brain, and to optimise the delivery of therapeutic ultrasound through the thoracic cage [2]. However, many existing elastic wave models are based on low-order finite difference or finite element schemes and thus require large numbers of grid points per wavelength to avoid numerical dispersion. Here, an accurate and computationally efficient elastic wave model is introduced as part of the open-source k-Wave MATLAB toolbox (http://www.k-wave.org) [3]. An overview of the numerical model is given, and the architecture of the simulation functions is described.

II. Numerical Model

A. Kelvin-Voigt Model

In an elastic medium, the propagation of compressional and shear waves can be described using Hooke's law and an expression for the conservation of momentum. For viscoelastic materials in which damping or absorption is present, Hooke's law is extended such that the stress-strain relation exhibits time dependent behaviour. For example, the classical Kelvin-Voigt model of viscoelasticity gives a time-dependent relationship that can be understood as the response of an elastic spring and viscous damper connected in parallel [4]. This model is widely used for studying the loss behaviour of viscoelastic materials. For an isotropic medium, the Kelvin-Voigt model can be written using Einstein summation notation as

$$\sigma_{ij} = \lambda\delta_{ij}\varepsilon_{kk} + 2\mu\varepsilon_{ij} + \chi\delta_{ij}\frac{\partial}{\partial t}\varepsilon_{kk} + 2\eta\frac{\partial}{\partial t}\varepsilon_{ij} \quad . \qquad (1)$$

Here σ is the stress tensor, ε is the dimensionless strain tensor, λ and μ are the Lamè parameters where μ is the shear modulus, and χ and η are the compressional and shear viscosity coefficients. The Lamè parameters are related to the shear and compressional sound speeds by

$$\mu = c_s^2\rho_0 \quad , \qquad \lambda + 2\mu = c_p^2\rho_0 \quad , \qquad (2)$$

where ρ_0 is the mass density. Using the relationship between strain and particle displacement u_i for small deformations

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad , \qquad (3)$$

Eq. (1) can be re-written as a function of the particle velocity v_i, where $v_i = \partial u_i/\partial t$

$$\frac{\partial \sigma_{ij}}{\partial t} = \lambda\delta_{ij}\frac{\partial v_k}{\partial x_k} + \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) + \chi\delta_{ij}\frac{\partial^2 v_k}{\partial x_k \partial t} + \eta\left(\frac{\partial^2 v_i}{\partial x_j \partial t} + \frac{\partial^2 v_j}{\partial x_i \partial t}\right) \quad . \qquad (4)$$

To model the propagation of elastic waves, this is combined with an equation expressing the conservation of momentum. Written as a function of stress and particle velocity, this is given by

$$\frac{\partial v_i}{\partial t} = \frac{1}{\rho_0}\frac{\partial \sigma_{ij}}{\partial x_j} \quad . \qquad (5)$$

Equations (4) and (5) are coupled first-order partial differential equations that describe the propagation of linear compressional and shear waves in an isotropic viscoelastic solid. When the effect of the loss term is small, these equations account for absorption of the form $\alpha_p \approx \alpha_{0,p}\omega^2$ and $\alpha_s \approx \alpha_{0,s}\omega^2$ (for compressional and shear waves, respectively) [5]. Here ω is temporal frequency in rad/s and the power law absorption prefactors $\alpha_{0,p}$ and $\alpha_{0,s}$ in $\mathrm{Np}\,(\mathrm{rad/s})^{-2}\,\mathrm{m}^{-1}$ are given by

$$\alpha_{0,p} = \frac{\chi + 2\eta}{2\rho_0 c_p^3} \quad , \qquad \alpha_{0,s} = \frac{\eta}{2\rho_0 c_s^3} \quad . \qquad (6)$$

B. Pseudospectral Time Domain Solution

A computationally efficient model for elastic wave propagation in absorbing media can be constructed based on the explicit solution of the coupled equations given in Eqs. (4)-(5) using the Fourier pseudospectral method [6, 7]. This uses the Fourier collocation spectral method to compute spatial derivatives, and a leapfrog finite-difference scheme to integrate forwards in time. Using a temporally and spatially staggered grid, the field variables in 2D are updated in a time stepping fashion as follows (similarly for 3D):

978-1-4799-7049-0/14/$31.00 ©2014 IEEE

(1) Calculate the spatial gradients of the stress field using the Fourier collocation spectral method

$$
\begin{aligned}
\partial_x \sigma_{xx}^- &= \mathcal{F}_x^{-1}\left\{ik_x e^{+ik_x\Delta x/2}\mathcal{F}_x\left\{\sigma_{xx}^-\right\}\right\}\\
\partial_y \sigma_{yy}^- &= \mathcal{F}_y^{-1}\left\{ik_y e^{+ik_y\Delta y/2}\mathcal{F}_y\left\{\sigma_{yy}^-\right\}\right\}\\
\partial_x \sigma_{xy}^- &= \mathcal{F}_x^{-1}\left\{ik_x e^{-ik_x\Delta x/2}\mathcal{F}_x\left\{\sigma_{xy}^-\right\}\right\}\\
\partial_y \sigma_{xy}^- &= \mathcal{F}_y^{-1}\left\{ik_y e^{-ik_y\Delta y/2}\mathcal{F}_y\left\{\sigma_{xy}^-\right\}\right\} \quad .
\end{aligned}
\tag{7a}
$$

Here $\mathcal{F}_{x,y}\{\}$ and $\mathcal{F}_{x,y}^{-1}\{\}$ are the 1D forward and inverse Fourier transforms over the x and y dimensions, i is the imaginary unit, k_x and k_y are the discrete set of wavenumbers in each dimension, and Δx and Δy give the grid spacing assuming a uniform Cartesian mesh. The exponential terms are spatial shift operators that translate the output by half the grid point spacing (see Fig. 1). This improves the accuracy of the model.

(2) Update the particle velocity using a finite difference time step of size Δt, where the $+$ and $-$ superscripts denote the field values at the next and current time step

$$
\begin{aligned}
v_x^+ &= v_x^- + \frac{\Delta t}{\rho_0}\left(\partial_x \sigma_{xx}^- + \partial_y \sigma_{xy}^-\right)\\
v_y^+ &= v_y^- + \frac{\Delta t}{\rho_0}\left(\partial_x \sigma_{xy}^- + \partial_y \sigma_{yy}^-\right) \quad .
\end{aligned}
\tag{7b}
$$

(3) Calculate the spatial gradients of the updated particle velocity using the Fourier collocation spectral method

$$
\begin{aligned}
\partial_x v_x^+ &= \mathcal{F}_x^{-1}\left\{ik_x e^{-ik_x\Delta x/2}\mathcal{F}_x\left\{v_x^+\right\}\right\}\\
\partial_y v_x^+ &= \mathcal{F}_y^{-1}\left\{ik_y e^{+ik_y\Delta y/2}\mathcal{F}_y\left\{v_x^+\right\}\right\}\\
\partial_x v_y^+ &= \mathcal{F}_x^{-1}\left\{ik_x e^{+ik_x\Delta x/2}\mathcal{F}_x\left\{v_y^+\right\}\right\}\\
\partial_y v_y^+ &= \mathcal{F}_y^{-1}\left\{ik_y e^{-ik_y\Delta y/2}\mathcal{F}_y\left\{v_y^+\right\}\right\} \quad .
\end{aligned}
\tag{7c}
$$

(4) Calculate the spatial gradients of the time derivative of the particle velocity using Eq. (5)

$$
\begin{aligned}
\partial_x \partial_t v_x^- &= \mathcal{F}_x^{-1}\left\{ik_x e^{-ik_x\Delta x/2}\mathcal{F}_x\left\{\left(\partial_x \sigma_{xx}^- + \partial_y \sigma_{xy}^-\right)/\rho_0\right\}\right\}\\
\partial_y \partial_t v_x^- &= \mathcal{F}_y^{-1}\left\{ik_y e^{+ik_y\Delta y/2}\mathcal{F}_y\left\{\left(\partial_x \sigma_{xx}^- + \partial_y \sigma_{xy}^-\right)/\rho_0\right\}\right\}\\
\partial_x \partial_t v_y^- &= \mathcal{F}_x^{-1}\left\{ik_x e^{+ik_x\Delta x/2}\mathcal{F}_x\left\{\left(\partial_x \sigma_{xy}^- + \partial_y \sigma_{yy}^-\right)/\rho_0\right\}\right\}\\
\partial_y \partial_t v_y^- &= \mathcal{F}_y^{-1}\left\{ik_y e^{-ik_y\Delta y/2}\mathcal{F}_y\left\{\left(\partial_x \sigma_{xy}^- + \partial_y \sigma_{yy}^-\right)/\rho_0\right\}\right\} .
\end{aligned}
\tag{7d}
$$

(5) Update the stress field using a finite difference time step

$$
\begin{aligned}
\sigma_{xx}^+ = \sigma_{xx}^- &+ \lambda\Delta t\left(\partial_x v_x^+ + \partial_y v_y^+\right) + \mu\Delta t\left(2\partial_x v_x^+\right)\\
&+ \chi\Delta t\left(\partial_x\partial_t v_x^- + \partial_y\partial_t v_y^-\right) + \eta\Delta t\left(2\partial_x\partial_t v_x^-\right)\\
\sigma_{yy}^+ = \sigma_{yy}^- &+ \lambda\Delta t\left(\partial_x v_x^+ + \partial_y v_y^+\right) + \mu\Delta t\left(2\partial_y v_y^+\right)\\
&+ \chi\Delta t\left(\partial_x\partial_t v_x^- + \partial_y\partial_t v_y^-\right) + \eta\Delta t\left(2\partial_y\partial_t v_y^-\right)\\
\sigma_{xy}^+ = \sigma_{xy}^- &+ \mu\Delta t\left(\partial_y v_x^+ + \partial_x v_y^+\right)\\
&+ \eta\Delta t\left(\partial_y\partial_t v_x^- + \partial_x\partial_t v_y^-\right) \quad .
\end{aligned}
\tag{7e}
$$

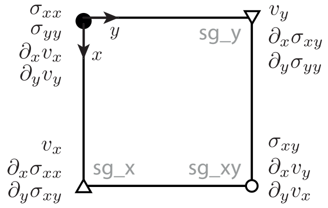


Fig. 1. Position of the field quantities and their derivatives on a spatially staggered grid in 2D. The derivatives $\partial_i\partial_t v_j$ (etc) are staggered in the same way as the $\partial_i v_j$ terms.

Here the Lamè parameters λ, μ and viscosity coefficients χ, η used in the time loop are calculated from the material properties c_p, c_s, ρ_0, $\alpha_{0,p}$, $\alpha_{0,s}$ defined by the user using Eqs. (2) and (6). For equations involving the spatially staggered grid parameters, the material properties are understood to be values defined at the staggered grid points. In addition to spatial staggering, the stress and velocity fields are also temporally staggered by $\Delta t/2$. Time varying stress and velocity sources are implemented after the update steps by adding the source terms to the relevant field values at the desired grid points within the domain. Similarly, outputs are calculated by storing the field values at the desired grid points at the end of each time step. To simulate free-field conditions, a multi-axial split-field perfectly matched layer (M-PML) is also applied to absorb the waves at the edge of computational domain [8].

III. The k-Wave Toolbox

The discrete equations given in the previous section were implemented in MATLAB as part of the open-source k-Wave toolbox (available from http://www.k-wave.org) [3]. This also contains functions for the simulation of linear and nonlinear wave fields in fluid media [9], and for the reconstruction of photoacoustic images [3]. The elastic simulation functions (named `pstdElastic2D` and `pstdElastic3D`) are called with four input structures (`kgrid`, `medium`, `source`, and `sensor`) in the same way as the other wave models in the toolbox. These structures define the properties of the computational grid, the distribution of medium properties, stress and velocity source terms, and the locations of the sensor points used to record the evolution of the wave field over time. A list of the main structure fields is given in Table I.

A simple example of a MATLAB script to perform an elastic wave simulation in a layered medium in 2D is given in Program 1. First, the computational grid is defined using the function `makeGrid`. This takes the number and spacing of the grid points in each Cartesian direction and returns an object of the `kWaveGrid` class. The time steps used in the simulation are defined by the object property `kgrid.t_array`. By default, this is set to `'auto'`, in which case the time array is automatically calculated within the simulation functions using the time taken to travel across the longest grid diagonal at the slowest sound speed, and a Courant-Friedrichs-Lewy (CFL) number of 0.1, where CFL = $c_0\Delta t/\Delta x$. The time array can also be defined by the user, either using the function `makeTime`, or explicitly in the form `kgrid.t_array = 0:dt:t_end`. The time array must be evenly spaced and monotonically increasing.

Program 1 Script for the simulation of an explosive pressure source in a layered fluid-solid half-space in 2D.

```
% create the computational grid
Nx = 128;               % [grid points]
Ny = 128;               % [grid points]
dx = 0.1e-3;            % [m]
dy = 0.1e-3;            % [m]
kgrid = makeGrid(Nx, dx, Ny, dy);

% define the compressional sound speed [m/s]
medium.sound_speed_compression = 1500*ones(Nx, Ny);
medium.sound_speed_compression(Nx/2:end, :) = 2000;

% define the shear sound speed [m/s]
medium.sound_speed_shear = zeros(Nx, Ny);
medium.sound_speed_shear(Nx/2:end, :) = 800;

% define the mass density [kg/m^3]
medium.density = 1000*ones(Nx, Ny);
medium.density(Nx/2:end, :) = 1200;

% define the absorption coefficients [dB/(MHz^2 cm)]
medium.alpha_coeff_compression = 0.1;
medium.alpha_coeff_shear = 0.5;

% define the initial pressure distribution
disc_magnitude = 5;   % [Pa]
disc_x_pos = 40;      % [grid points]
disc_y_pos = 64;      % [grid points]
disc_radius = 5;      % [grid points]
source.p0 = disc_magnitude*makeDisc(Nx, Ny,
  disc_x_pos, disc_y_pos, disc_radius);

% define a circular binary sensor mask
radius = 20;          % [grid points]
sensor.mask = makeCircle(Nx, Ny, Nx/2, Ny/2, radius);

% run the simulation
sensor_data = pstdElastic2D(kgrid, medium, source,
  sensor);
```

After the computational grid, the medium properties are defined. For a homogeneous medium, these are given as single scalar values in SI units. For a heterogeneous medium, these are defined as matrices the same size as the computational grid. There is no restriction on the distribution or values for the material properties. In this example, a heterogeneous medium is defined as a layered fluid-solid interface.

Next, any source terms are defined. Three types of source are currently supported: an initial pressure distribution (which is multiplied by -1 and assigned to the normal components of the stress), time varying velocity sources, and time varying stress sources. For time varying sources, the location of the source is specified by assigning a binary matrix (i.e., a matrix of 1's and 0's with the same dimensions as the computational grid) to `source.s_mask` or `source.u_mask`, where the 1's represent the grid points that form part of the source. The time varying input signals are then assigned to `source.sxx` (etc) or `source.ux` (etc). By default, the stress and velocity sources are added to the field variables as the injection of mass and force, respectively. The source values can also be used to replace the current values of the field variables by setting `source.s_mode` or `source.u_mode` to `'dirichlet'`. In Program 1, an initial pressure distribution within the fluid layer is defined in the shape of a small disc.

Finally, the sensor points are defined. These specify where in the computational domain the field variables are sampled at each time step. This can be defined in three ways. (1) As a binary matrix (i.e., a matrix of 1's and 0's) representing the grid points within the computational grid that will collect the data. (2) As the grid coordinates of two opposing corners of a rectangle (2D) or cuboid (3D). (3) As a series of Cartesian coordinates within the grid which specify the location of the field values stored at each time step. If the Cartesian coordinates don't exactly match the coordinates of a grid point, the output values are calculated via interpolation. In Program 1, a binary sensor mask in the shape of a circle is used. By default, only the acoustic pressure (given by the negative of the average normal stress) is stored. Other field parameters can also be returned by defining `sensor.record`.

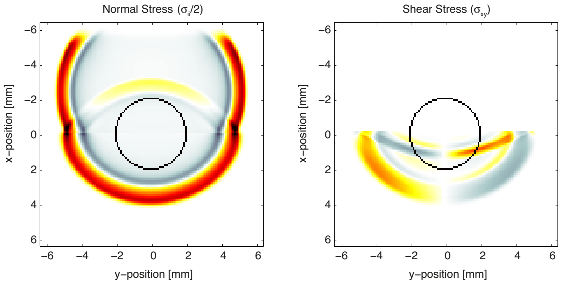


Fig. 2. Snapshot of the 2D simulation given in Program 1. The circular sensor mask is shown as a black circle. Both compression and shear waves can be seen in the lower layer. By default, the visualisation is updated every 10 time-steps. The colour map displays positive field values as yellows through reds to black, zero values as white, and negative values as light to dark blue-greys.

When the input structures have been defined, the simulation is started by passing them to `pstdElastic2D` (in 2D) or `pstdElastic3D` (in 3D). The propagation of the wave field is then computed step by step, with the field values at the sensor points stored after each iteration. By default, a visualisation of the propagating wave field and a status bar are displayed, with frame updates every ten time steps. The display is divided into two showing the normal and shear components of the stress field. In 3D, three intersecting planes through the centre of the grid are displayed. The default k-Wave colour map displays positive values as yellows through reds to black, zero values as white, and negative values as light to dark blue-greys. A snapshot of the visualisation produced by Program 1 is shown in Fig. 2. The absorption within the M-PML at the top of the domain is clearly visible.

When the time loop is complete, the function returns the field variables recorded at the sensor points defined by `sensor.mask`. If the sensor mask is given as a set of Cartesian coordinates, the computed `sensor_data` is returned in the same order. If `sensor.mask` is given as a binary matrix, `sensor_data` is returned using MATLAB's column-wise linear matrix index ordering. In both cases, the recorded data is indexed as `sensor_data(position_index, time_index)`. If `sensor.record` is defined, the output `sensor_data` is returned as a structure with the different outputs appended as structure fields. For example, if `sensor.record = {'u'}`, the output would contain the fields `sensor_data.ux` and `sensor_data.uy`.

TABLE I. SUMMARY OF THE MAIN FIELDS FOR THE FOUR INPUT STRUCTURES USED BY pstdElastic2D and pstdElastic3D.

Field	Description
kgrid.kx, kgrid.Nx, kgrid.dx, etc	Cartesian and wavenumber grid parameters returned by makeGrid
kgrid.t_array	Evenly spaced array of time points [s]
medium.sound_speed_compression	Matrix (or single value) of the compressional sound speed at each grid point within the medium [m/s]
medium.sound_speed_shear	Matrix (or single value) of the shear sound speed at each grid point within the domain [m/s]
medium.density	Matrix (or single value) of the mass density at each grid point within the domain [kg/m^3]
medium.alpha_coeff_compression	Matrix (or single value) of the power law absorption prefactor for compressional waves [dB/(MHz^2 cm)]
medium.alpha_coeff_shear	Matrix (or single value) of the power law absorption prefactor for shear waves [dB/(MHz^2 cm)]
source.p0	Matrix of the initial pressure distribution at each grid point within the domain [Pa]
source.s_mask	Binary matrix specifying the positions of the time varying stress source
source.sxx, source.sxy, etc	Matrix of time varying stress input/s at each of the source positions given by source.s_mask [Pa]
source.u_mask	Binary matrix specifying the positions of the time varying particle velocity source
source.ux, source.uy, etc	Matrix of time varying particle velocity input/s at each of the source positions given by source.u_mask [m/s]
sensor.mask	Binary matrix or a set of Cartesian points specifying the positions where the field is recorded at each time-step
sensor.record	Cell array listing the field variables to record, e.g., {'p', 'u'}
sensor.record_start_index	Time index at which the sensor should start recording

The behaviour of the simulation functions can be further controlled through the use of optional input parameters. These are given as param, value pairs following the four input structures. For example, the visualisation can be automatically recorded by setting 'RecordMovie' to true, and the plot scale can be controlled by setting 'PlotScale' in the form [sii_min, sii_max, sij_min, sij_max] (this defaults to [-1, 1, -1, 1]). Similarly, simulations can be run on an NVIDIA graphics processing unit (GPU) using the MATLAB Parallel Computing Toolbox by setting 'DataCast' to 'gpuArray-single'. The functions can also be used for time reversal image reconstruction in photoacoustics by assigning the recorded pressure values to sensor.time_reversal_boundary_data. This data is then enforced in time reversed order as a time varying Dirichlet boundary condition over the sensor surface given by sensor.mask. A full list and description of the different input options are given in the html help files and examples contained within the k-Wave toolbox.

IV. CONCLUSION

A new model for simulating the propagation of elastic waves using the k-Wave MATLAB toolbox is described. The model is based on two coupled first-order partial differential equations describing the variation of stress and particle velocity in an isotropic viscoelastic (Kelvin-Voigt) medium. These are discretised using an efficient pseudospectral time domain scheme. Spatial derivatives are computed using the Fourier collocation spectral method, while time integration is performed using a leapfrog finite difference. The new functions (named pstdElastic2D and pstdElastic3D) are called in the same way as the other wave models in k-Wave. The inputs are defined as fields to four input structures, with additional behaviour defined using optional input parameters. The medium parameters (shear and compressional sound speed, shear and compressional absorption coefficients, and mass density) can be heterogeneous and are defined as matrices the same size as the computational grid. The current code is implemented in MATLAB using a simple finite difference time scheme and assumes the medium is isotropic. In the future, this will be extended to account for orthotropic materials in which the planes of symmetry are aligned with the computational grid, and new models using a k-space corrected finite difference time scheme will also be introduced [10]. Versions of the code written in C++ based on OpenMP and MPI will also be developed and released at a later date [11].

ACKNOWLEDGMENT

This work was supported by the Dosimetry for Ultrasound Therapy project which is part of the European Metrology Research Programme (EMRP). The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union. Jiri Jaros is supported by the SoMoPro II programme, REA Grant Agreement No. 291782.

REFERENCES

[1] E. Bossy, F. Padilla, F. Peyrin, and P. Laugier. Three-dimensional simulation of ultrasound propagation through trabecular bone structures measured by synchrotron microtomography. *Phys. Med. Biol.*, 50(23):5545–56, 2005.

[2] K. Okita, R. Narumi, T. Azuma, S. Takagi, and Y. Matumoto. The role of numerical simulation for the development of an advanced HIFU system. *Comput. Mech.*, 2014.

[3] B. E. Treeby and B. T. Cox. k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *J. Biomed. Opt.*, 15(2):021314, 2010.

[4] J. J. Markham, R. T. Beyer, and R. B. Lindsay. Absorption of sound in fluids. *Reviews of Modern Physics*, 23(4):353, 1951.

[5] H. F. Pollard. *Sound Waves in Solids.* Pion Ltd, London, 1977.

[6] Q. H. Liu. Large-scale simulations of electromagnetic and acoustic measurements using the pseudospectral time-domain (PSTD) algorithm. *IEEE. T. Geosci. Remote*, 37(2):917–926, 1999.

[7] M. Caputo, J. M. Carcione, and F. Cavallini. Wave simulation in biologic media based on the Kelvin-voigt fractional-derivative stress-strain relation. *Ultrasound Med. Biol.*, 37(6):996–1004, 2011.

[8] K. C. Meza-Fajardo and A. S. Papageorgiou. On the stability of a non-convolutional perfectly matched layer for isotropic elastic media. *Soil Dyn. Earthq. Eng.*, 30(3):68–81, 2010.

[9] B. E. Treeby, J. Jaros, A. P. Rendell, and B. T. Cox. Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method. *J. Acoust. Soc. Am.*, 131(6):4324–4336, 2012.

[10] K. Firouzi, B. T. Cox, B. E. Treeby, and N. Saffari. A first-order k-space model for elastic wave propagation in heterogeneous media. *J. Acoust. Soc. Am.*, 132(3):1271–1283, 2012.

[11] J. Jaros, A. P. Rendell, and B. E. Treeby. Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound. *arXiv:1408.4675 [physics.med-ph]*, 2014.

Appendix B

Simulation Codes

B.1 Global Domain Decomposition

Jaros, J.; Rendell, A. P.; Treeby, B. E.: Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound. *International Journal of High Performance Computing Applications*. vol. 30, no. 2. 2015: pp. 137-155. ISSN 1094-3420. doi:10.1177/1094342015581024, **(IF 1.477)**.

Original Article

International Journal of
HIGH PERFORMANCE
COMPUTING APPLICATIONS

The International Journal of High Performance Computing Applications
1–19
© The Author(s) 2015
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1094342015581024
hpc.sagepub.com
SAGE

Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound

Jiri Jaros[1], Alistair P Rendell[2] and Bradley E Treeby[3]

Abstract

Model-based treatment planning and exposimetry for high-intensity focused ultrasound requires the numerical simulation of nonlinear ultrasound propagation through heterogeneous and absorbing media. This is a computationally demanding problem due to the large distances travelled by the ultrasound waves relative to the wavelength of the highest frequency harmonic. Here, the *k*-space pseudospectral method is used to solve a set of coupled partial differential equations equivalent to a generalised Westervelt equation. The model is implemented in C++ and parallelised using the message passing interface (MPI) for solving large-scale problems on distributed clusters. The domain is partitioned using a 1D slab decomposition, and global communication is performed using a sparse communication pattern. Operations in the spatial frequency domain are performed in transposed space to reduce the communication burden imposed by the 3D fast Fourier transform. The performance of the model is evaluated using grid sizes up to 4096 × 2048 × 2048 grid points, distributed over a cluster using up to 1024 compute cores. Given the global nature of the gradient calculation, the model shows good strong scaling behaviour, with a speed-up of 1.7x whenever the number of cores is doubled. This means large-scale simulations can be distributed across high numbers of cores on a cluster to minimise execution times with a relatively small overhead. The efficacy of the model is demonstrated by simulating the ultrasound beam pattern for a high-intensity focused ultrasound sonication of the kidney.

Keywords

High-intensity focused ultrasound, Fourier pseudospectral methods, Westervelt equation, FFTW, large-scale problems, distributed computing

1 Introduction

High-intensity focused ultrasound (HIFU) is a non-invasive therapy in which a tightly focused beam of ultrasound is used to rapidly heat tissue in a localised region until the cells are destroyed (Kennedy et al., 2003; ter Haar, 2007). In recent years, HIFU has been used in clinical trials for the treatment of tumours in many organs, including the prostate, kidney, liver, breast and brain (Kennedy et al., 2003; Clement, 2004; Jolesz and Hynynen, 2008; Zhang and Wang, 2010). While the number of HIFU devices on the market continues to grow, one hurdle that currently prevents wider clinical use is the difficulty in accurately predicting the region of damaged tissue given a particular patient and set of treatment conditions. In principle, this could be calculated using appropriate acoustic and thermal models (Paulides et al., 2013). However, the modelling problem is both physically complex and computationally challenging. For example, the heterogeneous material properties of human tissue can cause the ultrasound beam to become strongly distorted (Liu et al., 2005), the exact values for the material properties and their temperature dependence are normally unknown (Connor and Hynynen, 2002), and the rate

[1]Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
[2]Research School of Computer Science, Australian National University, Canberra, Australia
[3]Department of Medical Physics and Biomedical Engineering, University College London, London, United Kingdom

Corresponding author:
Jiri Jaros, Faculty of Information Technology, Brno University of Technology, Božetěchova 2, 612 66 Brno, Czech Republic.
Email: jarosjir@fit.vutbr.cz

Table 1. Examples of possible domain sizes and frequency ranges encountered in high-intensity focused ultrasound (HIFU). The grid sizes are based on using a uniform Cartesian grid at the Nyquist limit of two points per minimum wavelength (PPMW) assuming a sound speed of 1500 m/s. The memory usage is based on storing a single matrix at the specified grid size in single-precision (4 bytes per grid element).

Domain Size (cm^3)	Maximum Freq (MHz)	Domain Size (wavelengths)	Grid Size (at 2 PPMW)	Memory Per Matrix (GB)
5 × 5 × 5	5	333^3	667^3	1.1
	10	667^3	1333^3	8.8
	20	1333^3	2667^3	71
	50	3333^3	6667^3	1100
10 × 10 × 10	5	667^3	1333^3	8.8
	10	1333^3	2667^3	71
	20	2667^3	5333^3	570
	50	6667^3	13333^3	8800
20 × 20 × 20	5	1333^3	2667^3	71
	10	2667^3	5333^3	570
	20	5333^3	10667^3	4500
	50	13333^3	26667^3	71000

and mechanism for tissue damage are both temperature and cell specific (Lepock, 2003).

The question of how best to model the physical interactions between ultrasound waves and biological tissue has been widely studied, and work in this area is ongoing. However, the problem that has attracted much less attention, but which is equally important, is the issue of computational scale. This arises because of two related factors. The first is that the generated acoustic pressures are of sufficient magnitude that the wave propagation is nonlinear. This causes the ultrasound waves to steepen as they propagate, which generates high-frequency harmonics of the source frequency (this is usually between 0.5 and 2 MHz for HIFU treatments where the transducer is positioned outside the body). At low focal intensities, nonlinear effects cause energy to be generated up to at least the 10th harmonic (Wojcik et al., 1995). At very high focal intensities where strongly shocked waves are produced, as many as 600 harmonics might be required to model the focal heating accurately (Khokhlova et al., 2010). Thus, the frequency content of the propagating ultrasound waves can be very broadband.

The second factor is that the domain of interest encompassing the HIFU transducer and the treatment area is normally on the order of centimetres to tens of centimetres in each Cartesian direction. Compared to the size of the acoustic wavelength at the maximum frequency of interest, this equates to wave propagation over hundreds or thousands of wavelengths. To illustrate the scale of the problem, a list of representative domain sizes is given in Table 1. If the governing equations describing the HIFU field are solved on a uniform Cartesian grid where the grid spacing is defined to meet the Nyquist limit of two points per minimum wavelength, the resulting grid sizes can exceed 10^{12} grid points. If conventional finite difference schemes are used (which arguably is still the most common approach for time-domain modelling of broadband acoustic waves), the required grid sizes can be much greater. This is due to the large number of grid points per wavelength needed to avoid numerical dispersion over these length-scales. In many cases of practical interest, the grid sizes needed simply makes the simulations intractable.

To avoid the computational complexity of directly solving nonlinear acoustic equations in 3D, simplifying assumptions are normally made. In particular, one-way or evolution-type models have been very successful in simulating HIFU fields in homogeneous media (Averkiou and Cleveland, 1999; Curra et al., 2000; Khokhlova et al., 2001; Yuldashev and Khokhlova, 2011). In one-way models, the governing equations are formulated in retarded time, and the domain is discretised in x, y and τ (where τ is the retarded time variable) instead of x, y and z. The simulation then progresses by propagating the time history of the wave-field from plane to plane in the z-dimension (this is illustrated in Figure 1). If a small time window is used, this approach can significantly reduce the amount of memory required for broadband simulations. For example, Yuldashev and Khokhlova (2011) used grid sizes in the x-y plane with 10,000 × 10,000 grid points modelling up to 500 harmonics using a shared-memory computer with 32 GB of RAM. The main restriction of one-way models is that a heterogeneous distribution of tissue properties cannot be included (except via the use of phase layers (Yuldashev et al., 2010)), and scattered or reflected waves are not modelled. This means the significant distortion of HIFU beams that can occur in a clinical setting cannot be accounted for (Liu et al., 2005).

In addition to one-way models, several full-wave nonlinear ultrasound models in 3D have been reported

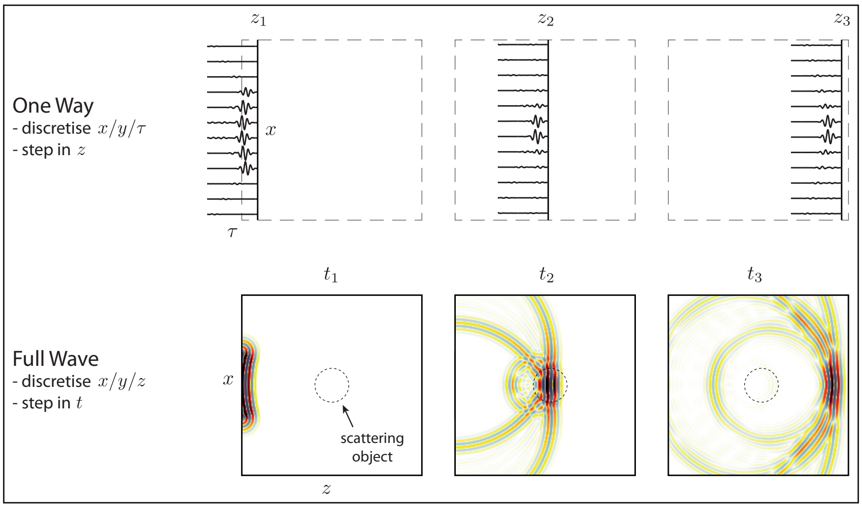


Figure 1. Schematic illustrating the difference between one-way and full-wave ultrasound models. In one-way models, the governing equations are formulated in retarded time, and the domain is discretised in *x*, *y* and τ (where τ is the retarded time variable). The simulation then progresses by propagating the time history of the wave-field from plane to plane in the z-dimension. In full-wave models, the domain is discretised in *x*, *y* and *z* and the simulation progresses by stepping through time. One-way models are typically more memory efficient, while full-wave models can account for heterogeneous material properties, reflected and scattered waves.

based on the finite difference time domain (FDTD) method (Connor and Hynynen, 2002; Pinton et al., 2009; Okita et al., 2011). For example, Pinton et al. (2009) presented a solution to the Westervelt equation (a particular nonlinear wave equation) using a second-order in time, fourth-order in space finite difference scheme. This was used to investigate the effects of nonlinear wave propagation on HIFU therapy in the brain (Pinton et al., 2011). Simulations were run on 112 cores of a distributed cluster with grid sizes and run times on the order of 800 × 800 × 800 and 32 hours. Okita et al. (2011) used a similar model to study HIFU focusing through the skull, using 128 cores of a distributed cluster with grid sizes and run times on the order of 800 × 600 × 600 and 2 hours respectively. They also demonstrated excellent weak-scaling results for a benchmark using up to 3.45×10^{11} grid points distributed across 98,304 cores on the K computer (a supercomputer installed at the RIKEN Advanced Institute for Computational Science in Japan) (Okita et al., 2014). In another study, Pulkkinen et al. (2014) used a hybrid FDTD model to study transcranial focused ultrasound using grid sizes up to 1338 × 1363 × 1120 running on 96 cores of a distributed cluster with compute times on the order of 40 hours.

Computationally, FDTD schemes have excellent weak-scaling properties and good computational performance for a given grid size. However, as mentioned above, the significant drawback is that for large domain sizes, very dense grids are needed to counteract the accumulation of numerical dispersion. One way of reducing this computational burden is to compute spatial gradients using the Fourier pseudospectral method (Hesthaven et al., 2007). This eliminates the numerical dispersion that arises from the discretisation of spatial derivatives, and significantly reduces the number of grid points needed per acoustic wavelength for accurate simulations. Computational efficiency can be further enhanced by using the *k*-space pseudospectral method. This extends the pseudospectral method by exploiting an exact solution to the linearised wave equation to improve the accuracy of computing temporal gradients via a finite-difference scheme (Mast et al., 2001; Tabei et al., 2002). This approach was first introduced in electromagnetics by Haber et al. (1973), and in acoustics by Bojarski (1982, 1985). Since then, both pseudospectral and *k*-space schemes have been applied to linear (Fornberg, 1987; Liu, 1999; Mast et al., 2001; Tabei et al., 2002; Cox et al., 2007; Daoud and Lacefield, 2009; Tillett et al., 2009) and nonlinear (Wojcik et al., 1998; Treeby et al., 2011; Albin et al., 2012; Jing et al., 2012; Treeby et al., 2012) ultrasound simulations in heterogeneous media for relatively modest grid sizes. Compared to FDTD schemes, the increase in accuracy arises due to the global nature of the basis functions (in this case complex exponentials) used to interpolate between the grid points. However, for large-scale problems where the numerical model must be implemented using a parallel computer with distributed resources, the global nature of the gradient calculation also introduces new challenges. This is due to the significant amount of global communication required between the compute cores (Daoud and Lacefield, 2009).

Another approach for minimising the grid size and number of time steps needed for accurate ultrasound simulations, is the iterative nonlinear contrast source (INCS) method (Verweij and Huijssen, 2009; Huijssen and Verweij, 2010; Demi et al., 2011). In this approach, terms describing the contributions of nonlinear effects and heterogeneous material parameters are reformulated as contrast source terms. The resulting wave equation is then solved iteratively using Green's function methods (Verweij and Huijssen, 2009). While this works well for weakly nonlinear fields, the requirement for storing the complete time history of the wavefield, and the evaluation of a 4D convolution at every time step makes it difficult to extend this approach to the large-scale problems encountered in HIFU.

Building on work by Tabei et al. (2002) and others, we recently proposed an efficient full-wave nonlinear ultrasound model based on the *k*-space pseudospectral method (Treeby et al., 2012). Here, we present an extension of this model for performing large-scale HIFU simulations on a distributed computer cluster with grid sizes up to 4096 × 2048 × 2048. A brief overview of the governing equations and the *k*-space pseudospectral model is given in Section 2. The software implementation and parallelisation strategies chosen for mapping the spectral gradient calculations onto a distributed computer cluster are then discussed in Section 3. In Section 4, performance and scaling results are presented for running simulations on up to 1024 cores. Results from a representative large-scale simulation of a HIFU treatment of the kidney are then presented in Section 5. Finally, summary and discussion are presented in Section 6.

2 Nonlinear ultrasound model

2.1 Nonlinear governing equations

For modelling the propagation of intense ultrasound waves in the human body, the governing equations must account for the combined effects of nonlinearity, acoustic absorption and heterogeneities in the material properties (sound speed, density, acoustic absorption and nonlinearity parameter). Following Treeby et al. (2012), the required governing equations can be written as three coupled first-order partial differential equations derived from the conservation laws and a Taylor series expansion for the pressure about the density and entropy

$$\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} &= -\frac{1}{\rho_0}\nabla p + \mathbf{F} && \text{(momentum conservation)}\\
\frac{\partial \rho}{\partial t} &= -\rho_0 \nabla \cdot \mathbf{u} - \mathbf{u} \cdot \nabla \rho_0 - 2\rho \nabla \cdot \mathbf{u} + \mathrm{M} && \text{(mass conservation)} \qquad (1)\\
p &= c_0^2\left(\rho + \mathbf{d} \cdot \nabla \rho_0 + \frac{B}{2A}\frac{\rho^2}{\rho_0} - \mathrm{L}\rho\right) && \text{(pressure – density relation)}
\end{aligned}$$

Here **u** is the acoustic particle velocity, **d** is the acoustic particle displacement, p is the acoustic pressure, ρ is the acoustic density, ρ_0 is the ambient (or equilibrium) density, c_0 is the isentropic sound speed and B/A is the nonlinearity parameter which characterises the relative contribution of finite-amplitude effects to the sound speed. These equations account for cumulative nonlinear effects (nonlinear effects that build up over space and time) up to second-order in the acoustic variables, equivalent to the Westervelt equation (Westervelt, 1963; Hamilton and Morfey, 2008). All the material parameters are allowed to be heterogeneous. Two linear source terms are also included, where **F** is a force source term which represents the input of body forces per unit mass in units of N kg^{-1} and M is a mass source term which represents the time rate of the input of mass per unit volume in units of kg m^{-3} s^{-1}.

The nonlinear term in the mass conservation equation accounts for a convective nonlinearity in which the particle velocity affects the wave velocity. Using the linearised form of the equations given in equation (1), this term can be written in a number of different ways. Following Aanonsen et al. (1984), the substitution of equations valid to first-order in the acoustic variables into terms that are second-order in the acoustic variables leads to third-order errors, which can be neglected. This leads to

$$-2\rho\nabla \cdot \mathbf{u} \approx \frac{2}{\rho_0}\rho\frac{\partial \rho}{\partial t} = \frac{1}{\rho_0}\frac{\partial \rho^2}{\partial t} \approx \frac{1}{\rho_0 c_0^4}\frac{\partial p^2}{\partial t} \qquad (2)$$

In equation (1), the nonlinear term is written in the first form shown in equation (2) as a spatial gradient of the particle velocity. This is significant because spatial gradients can be computed accurately using spectral methods, and don't require any additional storage. For comparison, the equivalent term in the Westervelt equation appears in the final form in equation (2), as a temporal gradient of the square of the pressure (Westervelt, 1963). However, using this expression requires the use of a finite difference scheme and storage of the pressure field at previous time steps.

The operator L in the pressure-density relation in equation (1) is an integro-differential operator that accounts for acoustic absorption that follows a frequency power law of the form $\alpha = \alpha_0\omega^y$. This type of absorption has been experimentally observed in human soft tissues, where y is typically between 1 and 2 (Duck, 1990). The operator has two terms both dependent on the fractional Laplacian and is given by (Chen and Holm, 2004; Treeby and Cox, 2010b)

$$\mathrm{L} = \tau \frac{\partial}{\partial t}\left(-\nabla^2\right)^{\frac{y}{2}-1} + \eta\left(-\nabla^2\right)^{\frac{y+1}{2}-1} \tag{3}$$

Here τ and η are absorption and dispersion proportionality coefficients given by $\tau = -2\alpha_0 c_0^{y-1}$ and $\eta = 2\alpha_0 c_0^{y}\tan(\pi y/2)$, where α_0 is the power law prefactor in Np (rad /s)$^{-y}$ m^{-1} and y is the power law exponent. The two terms in L separately account for power law absorption and dispersion for $0 < y < 3$ and $y \neq 1$ (Treeby and Cox, 2010b, 2011).

2.2 Discrete equations

Following Tabei et al. (2002) and Treeby et al. (2012), the continuous governing equations given in the previous section can be discretised using the k-space pseudospectral method. If the mass conservation equation and the pressure-density relation given in equation (1) are solved together, the $()\cdot\nabla\rho_0$ terms cancel each other, so they are not included in the discrete equations to improve computational efficiency. The mass and momentum conservation equations in equation (1) written in discrete form then become

$$\frac{\partial}{\partial \xi}p^n = \mathcal{F}^{-1}\left\{ik_\xi\,\kappa\,e^{ik_\xi\Delta\xi/2}\mathcal{F}\{p^n\}\right\} \tag{4a}$$

$$u_\xi^{n+\frac{1}{2}} = u_\xi^{n-\frac{1}{2}} - \frac{\Delta t}{\rho_0}\frac{\partial}{\partial \xi}p^n + \Delta t\,\mathrm{F}_\xi^{\mathrm{n}} \tag{4b}$$

$$\frac{\partial}{\partial \xi}u_\xi^{n+\frac{1}{2}} = \mathcal{F}^{-1}\left\{ik_\xi\,\kappa\,e^{-ik_\xi\Delta\xi/2}\mathcal{F}\left\{u_\xi^{n+\frac{1}{2}}\right\}\right\} \tag{4c}$$

$$\rho_\xi^{n+1} = \frac{\rho_\xi^n - \Delta t\rho_0\frac{\partial}{\partial\xi}u_\xi^{n+\frac{1}{2}}}{1 + 2\Delta t\frac{\partial}{\partial\xi}u_\xi^{n+\frac{1}{2}}} + \Delta t\,\mathrm{M}_\xi^{\mathrm{n}+\frac{1}{2}} \tag{4d}$$

Equations (4a) and (4c) are spatial gradient calculations based on the Fourier collocation spectral method, while equations (4b) and (4d) are update steps based on a k-space corrected finite difference scheme (Tabei et al., 2002). These equations are repeated for each Cartesian direction in $\mathbb{R}^n$ where $\xi = x$ in $\mathbb{R}^1$, $\xi = x, y$ in $\mathbb{R}^2$, and $\xi = x, y, z$ in $\mathbb{R}^3$. Here $\mathcal{F}\{\ldots\}$ and $\mathcal{F}^{-1}\{\ldots\}$ denote the forward and inverse spatial Fourier transforms over $\mathbb{R}^n$, i is the imaginary unit, Δt is the size of the time step and k_ξ represents the set of wavenumbers in the ξ-direction defined according to

$$k_\xi = \begin{cases} \left[-\frac{\mathrm{N}_\xi}{2}, -\frac{\mathrm{N}_\xi}{2}+1, \ldots, \frac{\mathrm{N}_\xi}{2}-1\right]\dfrac{2\pi}{\Delta\xi\,\mathrm{N}_\xi} & \text{if } \mathrm{N}_\xi \text{ is even} \\ \left[-\frac{(\mathrm{N}_\xi-1)}{2}, -\frac{(\mathrm{N}_\xi-1)}{2}+1, \ldots, \frac{(\mathrm{N}_\xi-1)}{2}\right]\dfrac{2\pi}{\Delta\xi\,\mathrm{N}_\xi} & \text{if } \mathrm{N}_\xi \text{ is odd} \end{cases} \tag{5}$$

Here N_ξ and $\Delta\xi$ are the number and spacing of the grid points in the ξ-direction assuming a regular Cartesian grid. The k-space operator κ in equation (4) is used to correct for the numerical dispersion introduced by the finite-difference time step, and is given by $\kappa = \mathrm{sinc}(c_{\mathrm{ref}}k\Delta t/2)$, where $k = |\mathbf{k}|$ is the scalar wavenumber, and c_{ref} is a single reference value of the sound speed (see discussion in Treeby et al. (2012) for further details).

The acoustic density and the mass source term (which are physically scalar quantities) are artificially divided into Cartesian components to allow an anisotropic perfectly matched layer (PML) to be applied. For the simulations presented here, Berenger's original split-field formulation of the PML is used (Berenger, 1996) as described in Tabei et al. (2002). The exponential terms $e^{\pm ik_\xi\Delta\xi/2}$ within equations (4a) and (4c) are spatial shift operators that translate the result of the gradient calculations by half the grid point spacing in the ξ-direction. This allows the components of the particle velocity to be evaluated on a staggered grid. Note, the ambient density ρ_0 in equation (4b) is understood to be the ambient density defined at the staggered grid points. The superscripts n and $n + 1$ denote the function values at current and next time points and $n - \frac{1}{2}$ and $n + \frac{1}{2}$ at the time-staggered points. The time-staggering arises because the update steps, equations (4b) and (4d), are interleaved with the gradient calculations, equations (4a) and (4c). An illustration of the staggered grid scheme is shown in Figure 2.

The corresponding pressure-density relation written in discrete form is given by

$$p^{n+1} = c_0^2\left(\rho^{n+1} + \frac{B}{2A}\frac{1}{\rho_0}\left(\rho^{n+1}\right)^2 - \mathrm{L_d}\right) \tag{6}$$

where the total acoustic density is given by $\rho^{n+1} = \sum_\xi \rho_\xi^{n+1}$ and $\mathrm{L_d}$ is the discrete form of the power law absorption term which is given by (Treeby and Cox, 2010b)

$$\mathrm{L_d} = \tau\mathcal{F}^{-1}\left\{k^{y-2}\,\mathcal{F}\left\{\frac{\partial\rho^n}{\partial t}\right\}\right\} + \eta\mathcal{F}^{-1}\left\{k^{y-1}\,\mathcal{F}\left\{\rho^{n+1}\right\}\right\} \tag{7}$$

To avoid needing to explicitly calculate the time derivative of the acoustic density (which would require storing a copy of at least ρ^n and ρ^{n-1} in memory), the temporal derivative of the acoustic density is replaced using the linearised mass conservation equation $d\rho/dt = -\rho_0\nabla\cdot\mathbf{u}$, which yields

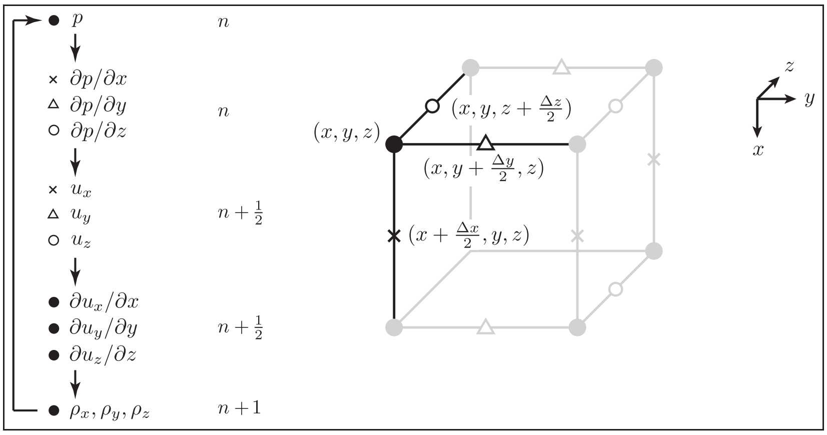


Figure 2. Schematic showing the computational steps in the solution of the coupled first-order equations using a staggered spatial and temporal grid in 3D. Here $\partial p/\partial x$ and u_x are evaluated at grid points staggered in the x-direction (crosses), $\partial p/\partial y$ and u_y are evaluated at grid points staggered in the y-direction (triangles) and $\partial p/\partial z$ and u_z are evaluated at grid points staggered in the z-direction (open circles). The remaining variables are evaluated on the regular grid points (dots). The time-staggering is denoted n, $n+\frac{1}{2}$, and $n + 1$.

$$\mathrm{L_d} = -\tau\,\mathcal{F}^{-1}\left\{k^{y-2}\,\mathcal{F}\left\{\rho_0 \sum\nolimits_{\xi} \frac{\partial}{\partial \xi} u_{\xi}^{\,n+\frac{1}{2}}\right\}\right\} + \eta\,\mathcal{F}^{-1}\left\{k^{y-1}\,\mathcal{F}\left\{\rho^{n+1}\right\}\right\} \tag{8}$$

Further details about the formulation, stability and accuracy of the k-space scheme can be found in Mast et al. (2001), Tabei et al. (2002), Cox et al. (2007), Treeby and Cox (2010b) and Treeby et al. (2012).

3 Implementation of the *k*-space pseudospectral method for distributed clusters

3.1 Overview

The k-space and pseudospectral methods gain their advantage over finite difference methods due to the global nature of the spatial gradient calculations. This permits the use of a much coarser grid for the same level of accuracy. However, even using spectral methods, the computational and memory requirements for the nonlinear HIFU problems discussed in Section 1 are still considerable, and in most cases, significantly exceed the resources of a single workstation. In this context, the development of an efficient numerical implementation that partitions the computational cost and memory usage across a large-scale parallel computer is desired. However, the global nature of the gradient calculation, in this case using the 3D fast Fourier transform (FFT), introduces additional challenges for the development of an efficient parallel code. Specifically, while the FDTD method only requires small quantities of data to be exchanged between the processes responsible for adjacent portions of the domain, performing a FFT requires a globally synchronising all-to-all data exchange. This global communication can become a significant bottleneck in the execution of spectral models.

Fortunately, considerable effort has already been devoted to the development of distributed memory FFT libraries that show reasonable scalability of up to tens of thousands of processing cores (Frigo and Johnson, 2005; Pekurovsky, 2012). These libraries have found particular utility in turbulence simulations where grid sizes of up to 4096 × 4096 × 4096 have been used (Yeung et al., 2012). They have also been considered in previous implementations of linear k-space pseudospectral models using grid sizes of up to 512 × 512 × 512 (Daoud and Lacefield, 2009; Tillett et al., 2009). The challenges for the current work were thus to determine how best to exploit these libraries within the context of the nonlinear k-space model, how to maximise the grid sizes possible for a given total memory allocation, how best to manage the generated output data and how to maximise performance and scalability.

The execution of the k-space pseudospectral model described in Section 2 can be divided into three phases: pre-processing, simulation and post-processing. During the pre-processing phase, the input data for the simulation is generated. This involves defining the domain discretisation based on the physical domain size and maximum frequency of interest, defining the spatially varying material properties (e.g. using a CT scan of the patient (Schneider et al., 1996)), defining the properties of the ultrasound transducer and drive signal and defining the desired output data (e.g. the peak positive pressure or time-averaged acoustic intensity in the region of the HIFU target (ter Haar et al., 2011)). The simulation phase involves reading the input data,

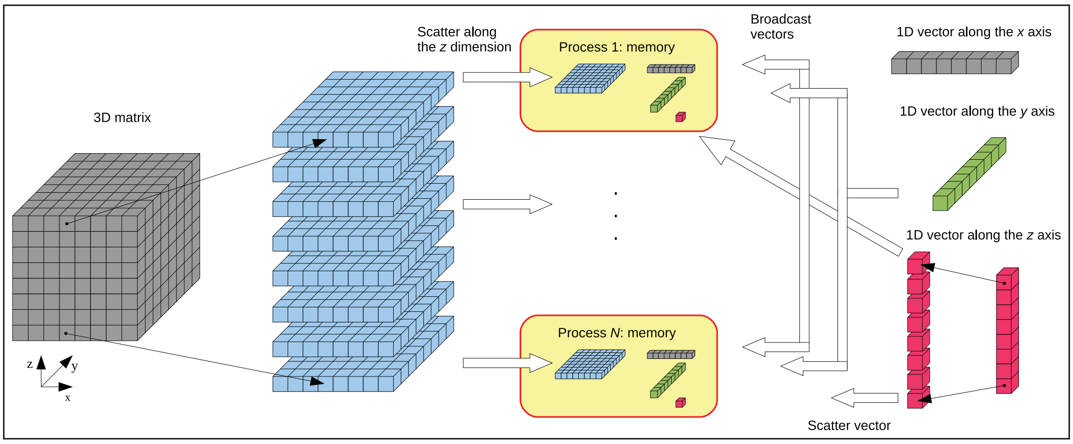

3D matrix
Scatter along the z dimension
Process 1: memory
Process N: memory
Broadcast vectors
1D vector along the x axis
1D vector along the y axis
1D vector along the z axis
Scatter vector
z
y
x

Figure 3. Illustration of the 1D slab decomposition used to partition the 3D domain within a distributed computing environment. The 3D matrices are partitioned along the z-dimension and distributed over P MPI processes. 1D vectors oriented along the x- and y-dimensions are broadcast, while the vectors along the z-dimension are scattered. All scalar variables are broadcast and replicated on each process.

running the actual simulation following the discrete equations discussed in Section 2.2, and storing the output data. The post-processing phase involves analysing the (potentially large) output files and presenting this data in a human-readable form. Here, the discussion is focused primarily on the parallel implementation of the simulation phase. Some discussion of the pre- and post-processing stages is given in Section 5.

The discrete equations solved during the simulation phase are given in equations (4a) to (4d) and equation (6). Examining these equations, the data stored in memory during the simulation phase comprises of twenty-one real 3D matrices defined in the spatial domain, and three real and three complex 3D matrices defined in the spatial Fourier domain (in addition to vector and scalar values). The 3D matrices contain the medium properties at every grid point, the time-varying acoustic quantities, the derivative and absorption operators and temporary storage. The operations performed on these datasets include 3D FFTs, element-wise matrix operations, injection of the source signal and the collection of output data.

The implementation of the discrete equations was written in C++ as an extension to the open-source k-Wave acoustics toolbox (Treeby and Cox, 2010a; Treeby et al., 2012). The standard message passing interface (MPI) was used to perform all interprocess communications, the MPI version of the FFTW library was used to perform the Fourier transforms (Frigo and Johnson, 2005) and the input/output (I/O) operations were performed using the hierarchical data format (HDF5) library. To maximise performance, the code was also written to exploit single instruction multiple data (SIMD) instructions such as streaming SIMD extensions (SSE). Further details of the implementation are given in the following sections.

3.2 Domain decomposition and the FFT

To divide the computational domain across multiple interconnected nodes in a cluster, a one-dimensional domain decomposition approach was used in which the 3D domain is partitioned along the z-dimension into 2D slabs. The slabs are then distributed over P MPI processes, where each MPI process corresponds to one physical CPU core. The total number of processes is constrained by $P \leq \mathrm{N}_z$, where N_z is the number of grid points in the z-dimension (and thus the number of slabs). This decomposition approach was used as it is directly supported by the FFTW library, while other approaches, such as 2D partitioning are not.

Figure 3 shows how the various spatial data structures are distributed to processes. For each 3D matrix there are a maximum of $\lceil \mathrm{N}_z/P \rceil$ 2D slabs stored on each process. For 1D quantities oriented along the z-axis, the data is partitioned and scattered over the processes in a similar manner. For 1D quantities oriented along either the x- or y-axis (and for scalar quantities), the data is broadcast and replicated on every process. The exception is for the source and sensor masks, which list the grid indices where the input data is defined and where the output data is collected. These are distributed such that individual processes are assigned the portion of the list that corresponds to parts of the source and/or sensor that fall within its local section of the domain. As the source and sensor masks do not usually cover the whole domain, many processes are likely to receive no source or sensor related data.

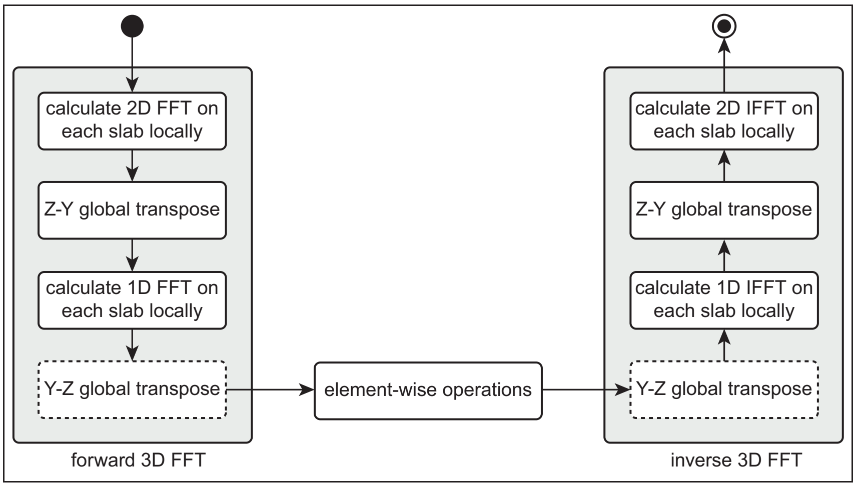

calculate 2D FFT on each slab locally
Z-Y global transpose
calculate 1D FFT on each slab locally
Y-Z global transpose
forward 3D FFT
element-wise operations
Y-Z global transpose
calculate 1D IFFT on each slab locally
Z-Y global transpose
calculate 2D IFFT on each slab locally
inverse 3D FFT

Figure 4. Chain of operations needed to calculate spatial gradients using the Fourier pseudospectral method. First, the 3D forward FFT is calculated, then element-wise operations within the spatial frequency domain are applied. Finally, the result is transformed back to the spatial domain using an inverse 3D FFT. The transposes depicted in the dashed boxes can be omitted if the element-wise matrix operations are performed in the transposed domain.

The calculation of the spatial gradients following equations (4a) and (4c) involves performing a 3D FFT, followed by one or more element-wise matrix operations, followed by an inverse 3D FFT. Using the domain decomposition scheme outlined above, each 3D FFT is executed by first performing a series of 2D FFTs in the *xy*-plane (i.e. on each slab) using local data. This is then followed by an all-to-all global communication to perform a *z*↔*y* transposition. This step is necessary as the FFT can only be performed on local data (it cannot stride across data belonging to multiple processes). The global transposition is then followed by a series of 1D FFTs performed in the transposed *z*-dimension, followed by another global transposition from *y*↔*z* to return the data to its original layout. This chain of operations is illustrated in Figure 4. In this decomposition, the main performance bottleneck is the two global transpositions required per FFT.

Examining Figure 4, it is apparent that the last global *y*↔*z* transposition of the forward FFT is paired with an identical but reverse transposition immediately after the element-wise operations. As the intervening operations are independent of the order of the individual elements, it is possible to eliminate these two transpositions such that operations in the spatial frequency domain are performed in transformed space (Frigo and Johnson, 2012). This has a significant effect on performance, with compute times reduced by 35-40% depending on the number of processes used. Note, in this case, variables defined in the spatial frequency domain must instead be partitioned in transformed space along the *y*-dimension, with the total number of MPI processes constrained by $P \leq \min(N_y, N_z)$. To avoid having idle processes during calculations involving either regular or transposed data, the number of processes P should ideally be a chosen to be a divisor of both N_y and N_z.

As the input data to the forward FFT is always real, the output of the Fourier transform in the first dimension is symmetric. Computational efficiency can thus also be improved by using the real-to-complex and complex-to-real FFTW routines which only calculate and return the unique part of the transform. To further improve performance, the element-wise matrix operations executed in between these transforms, as well as those defined in equations (4b) and (4d), were merged into compute kernels optimised to maximise temporal data locality and performance. In some cases, the latter involved manually inserting calls to SIMD intrinsic functions into loop structures that the compiler was unable to vectorise otherwise.

3.3 Input and output

The open-source HDF5 library was chosen to manipulate the input and output files because of its ability to organise complex datasets in heterogeneous computing environments. This library is available on most supercomputers and is also supported by numerical computing languages such as MATLAB and Python which is useful for pre- and post-processing. The HDF5 library provides two interfaces: serial and parallel. The serial interface targets shared memory computers and assigns a single thread to be responsible for all I/O. This interface was used to generate the input files during the preprocessing phase using a single large-memory node. The parallel interface targets clusters and essentially provides a user-friendly wrapper to the low level MPI-I/O library. This allows multiple processes to share the same file and read or write to it simultaneously. This interface was used during the simulation phase as it provides much higher I/O performance than either serialised accesses or master-slave architectures (where a single process serves all I/O requests). The parallel HDF5 interface also allows for two different I/O access patterns: independent and collective. In most cases, the collective mode is preferred as it enables the HDF5 runtime to rearrange and reshape data from different parallel processes before writing it to disk. This mode was used for all I/O operations during the simulation phase, the only exception being when writing scalar values to the output file.

Within the input and output files, all datasets were stored as three-dimensional arrays in row-major order, with dimensions defined by the tuple (N_x, N_y, N_z). For scalars and 1D and 2D arrays, the unused tuple dimensions were set to 1. For example, scalar variables were stored as arrays of dimension (1, 1, 1), 1D vectors oriented along the y-axis were stored as arrays of dimension (1, N_y, 1), etc. For datasets containing complex data, the lowest used dimension was doubled and the data stored in interleaved format. Additional information about the simulation (for example, the control flags and parameters) was stored within the file header.

To maximise I/O performance, the datasets within the input and output files were stored in a chunked layout, where each dataset is divided into multiple blocks or chunks that can be accessed individually. This is particularly useful for improving the throughput of partial I/O, where only portions of a dataset are accessed at a time. The use of chunking also provides a convenient way to overcome one of the current MPI limits, namely the 2 GB maximum size of a message (this is due to the routine headers in the MPI standard being defined using signed integers). Without chunking, this limit is easy to exceed, particularly during MPI gather operations where hundreds of small messages are aggregated. In addition to partial I/O, chunking enables the use of on-the-fly data compression. This is particularly beneficial for the input file, as there are often large areas of the domain with similar material properties, for example, the layer of water or coupling medium between the transducer and patient. These homogeneous regions give rise to good compression ratios, and thus reduce the amount of communication necessary when loading the input file. However, while both the serial and parallel HDF5 interfaces can read compressed datasets, only the serial interface can write such datasets. Thus compression was only used for the input files.

For the 3D datasets, the chunk size was chosen to be a single 2D slab matching the decomposition discussed in Section 3.2. This is the smallest portion of data read by each MPI process. Each slab is only ever accessed by one process at a time, and is usually a reasonable size in terms of I/O efficiency. For the 2D and 1D quantities, the data was divided into chunks of fixed size along the lowest used dimension. A chunk size of 8 MB was found to give reasonable I/O performance. Note, it is possible to further tune the chunk size for different size datasets to maximise I/O performance on a given platform. This can be useful on parallel cluster file systems that use hundreds of disk drives spread over many nodes, in conjunction with complex internode communication patterns.

3.4 Simulation stages

An overview of the stages within the simulation phase of the parallel implementation of the k-space pseudospectral model is shown in Figure 5. When the simulation begins, parameters such as the domain size and the number of time steps are loaded using a HDF5 broadcast read. In this operation, all processes select the same item to be read from the file. The HDF5 library then joins all I/O requests such that the data is only physically read from disk once and then broadcast over all processes using an MPI routine. Next, the 1D decomposition of the domain is calculated as discussed in Section 3.2, and memory for the various simulation and temporary quantities is allocated. Note, the use of SIMD instructions imposes several requirements on the data layout and its memory alignment. Firstly, all multidimensional matrices must be stored as linear arrays in row-major order. Secondly, complex quantities must be stored in interleaved format. Finally, data structures must be allocated using an FFTW memory allocation routine that ensures they are aligned on 16 B boundaries.

After memory allocation, the contents of the input file are loaded and distributed over all processes. The 3D datasets (e.g. the medium properties) are loaded in

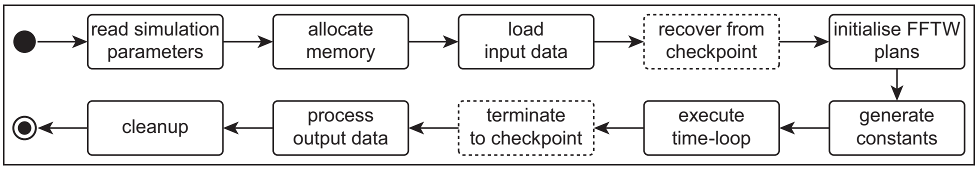


Figure 5. Overview of the stages within the simulation phase of the parallel implementation of the k-space pseudospectral model.

chunks, with each process identifying its local portion of the domain and invoking a collective HDF5 read operation. The 1D vectors that are scattered are loaded in a similar fashion, while 1D vectors and scalar values that are replicated are read using a HDF5 broadcast read. The source and sensor masks are loaded in chunks and broadcast to all processes, with each process extracting the grid indices that fall within its portion of the domain. Once the distribution of the source mask is known, the drive signal for the transducer is then sent to the relevant processes. Finally, the distribution of the sensor mask is used to allocate appropriate buffers to store the output data. These buffers are eventually flushed into the output file.

Next, execution plans for the FFT are generated. This step is required by FFTW prior to performing the first transform of a particular rank, size and direction. It involves the library trying several different FFT implementations with the objective of finding the fastest execution pathway on the current hardware (Frigo and Johnson, 2005). Consideration is given to many architectural aspects including the CPU, the memory system and the interconnection network. Depending on the domain size, this step can take a considerable amount of time. However, as it is only executed once and there are many thousands of FFTs in a typical simulation, the benefit of having a fast implementation is usually significant (this is discussed further in Section 4.2). Unfortunately, while it is possible to save plans between code executions, there is usually little benefit in doing so. This is because the plans are problem-size specific, and depend on the number and distribution of cores within the parallel system being used. On shared clusters in particular, the distribution of cores is likely to change between runs according to the cluster utilisation and management system.

After the FFT plans are generated, simulation constants such as the k-space operator (κ) and the absorption parameters (τ, η, k^{y-2}, k^{y-1}) are generated. The simulation time-loop then begins following equations (4a) to (4d) and equation (6). For each time step, there are six forward and eight inverse FFTs. There are two fewer forward FFTs as the three spatial components in equation (4a) share the same $\mathcal{F}\{p^n\}$. The source injection is implemented as an operation that updates the value of the acoustic pressure or velocity at the relevant grid points within the domain. In a similar vein, the output data is collected by storing the acoustic parameters at the grid points specified by the sensor mask. Aggregate quantities (e.g. the peak positive pressure) are kept in memory until the simulation is complete, while time-varying quantities are flushed to disk at the end of each time step using a collective HDF5 write routine.

When running larger simulations, a checkpoint-restart capability is also used. This allows large simulations to be split into several phases, which is useful for staying within wall-clock limitations imposed on many clusters, in addition to providing a degree of fault tolerance. During checkpointing, an additional HDF5 output file is created which stores the current values of the time-varying acoustic quantities as well as the aggregated output quantities. Restart is performed in the same way as a new simulation, however, after the input file has been loaded, the checkpoint file is opened, and the acoustic quantities and the aggregated values are restored.

4 Performance evaluation

4.1 Benchmark platform

To evaluate the utility of the implemented k-space pseudospectral model in the context of large-scale nonlinear ultrasound simulations for HIFU, a number of performance metrics were investigated. These included the strong and weak scaling properties of the code, the absolute simulation time and cost, and the effect of the underlying hardware architecture. The tests were performed using the VAYU supercomputer run by the National Computational Infrastructure (NCI) in Australia. This system comprises 1492 nodes, each with two quad-core 2.93 GHz Intel Nehalem architecture Xeon processors and 24 GB of memory configured in a non-uniform memory access (NUMA) design. The compute nodes are connected to the cluster via an onboard infiniband interface with a theoretical bandwidth of 40 Gb/s, while the interconnection network consists of four highly integrated 432-port infiniband switches. This relatively simple network design reduces the impact of process placement on interprocess network bandwidth and latency. The I/O disk subsystem is physically separated from the compute nodes. The Lustre parallel distributed file system is used to manage 832 TB of disk space distributed over 1040 disk drives. This configuration offers very high bandwidth, however, latency for small I/O transactions (e.g. storing less than 1 MB of data) can be relatively high. Also, as the I/O subsystem is shared amongst all users, notable performance fluctuations can occasionally be observed. VAYU, similar to many other clusters, runs a Linux based operating system controlled by the OpenPBS batch queuing systems. Compute cores granted to the job are always dedicated and simulation cost is determined in terms of service units (wall-clock time multiplied by the number of CPU cores used).

The performance metrics were evaluated using a common set of benchmarks. The benchmark set was designed to cover a wide range of domain sizes, from small simulations that can be run on desktop systems, up to large-scale simulations that approach the limits of what is currently feasible using a supercomputer. The simulations accounted for nonlinear wave propagation in an absorbing medium in which all material properties were heterogeneous. The time varying output pressure was recorded over a 2D xy-plane centred in the z-direction. The grid sizes increased from 2^{24} grid points ($256 \times 256 \times 256$) to 2^{34} grid points ($4096 \times 2048 \times 2048$), where each successive benchmark was doubled in size, first by doubling the grid size in the x-dimension, then the y-dimension, then the z-dimension, and so on. The benchmarks were executed on VAYU using different numbers of compute cores ranging from 8 (one node) to 1024 (128 nodes) in multiples of 2. The minimum and maximum number of cores used for a particular grid size were limited by the available memory per core from the bottom (24 GB per node or 3 GB per core), and the size of the simulation domain from the top ($P \leq \min(N_y, N_z)$). Compute times were obtained by averaging over 100 time-steps, excluding pre- and post-processing.

4.2 Strong scaling

Strong scaling describes how the execution time changes when using an increasing number of compute cores for a fixed problem size. Ideally, whenever the number of compute cores is doubled, the execution time should reduce by a factor of 2. However, in practice, all but embarrassingly parallel applications include some level of overhead that causes them to eventually reach a point where increasing the number of compute cores does not reduce the execution time, or can even make it worse.

Figures 6(a) and 6(b) show the strong scaling results obtained for the eleven different benchmark problem sizes using two different FFTW planing flags, FFTW_MEASURE and FFTW_EXHAUSTIVE. These flags determine the extent to which FFTW attempts to find the best FFT implementation during the plan generation phase. The flag FFTW_MEASURE is the default, while the flag FFTW_EXHAUSTIVE triggers a much more extensive search covering the full range of options implemented by the FFTW library. When using the FFTW_MEASURE flag, the results show almost ideal strong scaling for domain sizes up to 2^{31} grid points ($2048 \times 1024 \times 1024$) and core counts up to 256. Within this region, all the curves have approximately equal gradients corresponding to a speed-up of roughly 1.7x when doubling the number of cores. Beyond 256 cores, acceptable scaling is obtained for many cases, however, there is virtually no improvement in execution time for grid sizes of 2^{29} and 2^{30} grid points. Further increasing the core count to 1024 leads to significantly worse performance, with the exception of the grid size of 2^{30} grid points ($1024 \times 1024 \times 1024$) where, interestingly, the performance increases markedly.

The erratic performance of FFTW at large core counts when using the FFTW_MEASURE flag is due to differences in the communication pattern chosen during the planning phase. Specifically, using the integrated performance monitoring (IPM) profiler, it was found that in most cases, the global transposition selected by FFTW was based on a simple all-to-all MPI communication routine which exchanges $P(P-1)$ messages amongst P compute cores. This becomes increasingly inefficient when the message size drops into the tens of kB range and the number of messages rises into the millions. This is the case for the very large simulations on high core counts shown in Figure 6(a). However, FFTW also includes other communication algorithms, including one that uses a sparse communication pattern. This routine was selected by FFTW_MEASURE for the simulation with 2^{30} grid points and 1024 cores, leading to significant performance gains.

The sparse communication pattern used by FFTW combines messages local to the node into a single message before dispatch. In the case of VAYU, this means that 8 messages are combined together within each node before sending to other nodes. This decreases the number of messages by a factor of 64 and also increases the message size. When using FFTW_EXHAUSTIVE, the sparse communication pattern was always selected. As shown in Figure 6(b), the impact of this on performance is considerable. Almost all anomalies are eliminated, and the scaling remains close to ideal over the whole range of grid sizes and core counts considered. This is consistent with profiling data, which revealed that 30-40% of the execution time was associated with FFT communication, another 30-40% due to FFT operations and the remaining 15-20% due to element-wise matrix operations. These results indicate that given large enough grid sizes, reasonable scalability to even larger numbers of compute cores should be possible. Small deviations in the scaling results are likely to be caused by the degree of locality in the process distribution over the cluster, as well as by process interactions when performing I/O operations. For comparison, the time to execute the FFTW planning phase using FFTW_MEASURE for a grid size of 2^{33} grid points ($2048 \times 2048 \times 2048$) was 239 s when using 512 cores, and 164 s using 1024 cores. The corresponding times using FFTW_EXHAUSTIVE were 999 s and 834 s. This is equivalent to the time taken to compute approximately 100 time steps. As a typical simulation of this scale requires tens of thousands of time steps, the planning times represent a very small proportion of the total simulation time.

In addition to the benchmark set considering nonlinear wave propagation in a heterogeneous and absorbing medium, additional benchmarks were performed considering (1) nonlinear wave propagation in a homogeneous and absorbing medium, and (2) linear wave propagation in a homogeneous and lossless medium. The strong-scaling results for both cases were

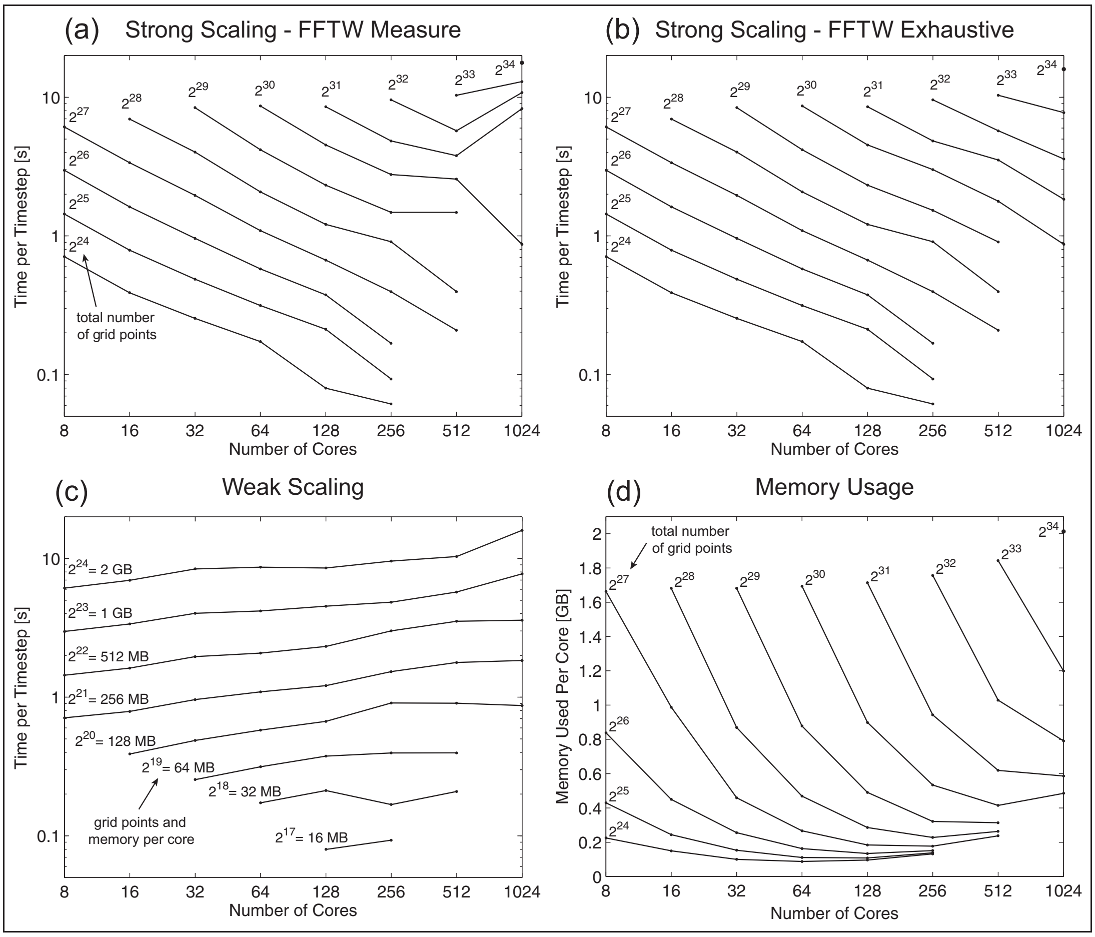


Figure 6. Performance results for nonlinear ultrasound simulations in heterogeneous and absorbing media. (a), (b) Strong scaling showing the variation of the execution time with the number of CPU cores for a fixed grid size. The domain size varies from 2^{24} (256^3) grid points to 2^{34} (4096×2048×2048) grid points. The plots clearly show the difference in strong scaling between `FFTW_MEASURE` and `FFTW_EXHAUSTIVE` flags used for planning the execution of FFTs and related global communications. (c) Weak scaling showing the potential of more cores to solve bigger problems in the same wall-clock time. The curves show the working dataset size per core growing from 16 MB (2^{17} grid points) up to 2 GB (2^{24} grid points). (d) Memory usage per core for the strong scaling results given in part (b).

similar to those given above. This is not surprising given that the FFT is the dominant operation in all test cases. In terms of absolute performance, the results for case (1) benefit from both a reduced memory footprint and a reduced memory bandwidth requirement. Specifically, because the medium is homogeneous, the medium parameters can be described using scalar variables that are easily cached, rather than large 3D matrices that are usually read from main memory each time they are used. This reduces the memory requirements by approximately 30% and the execution times by 5-10%. For case (2), an additional advantage comes from a reduction in the number of FFTs performed within each time step (ten instead of fourteen). This reduction is mirrored almost exactly in the observed execution times, which were typically 30% less than the execution times for an absorbing medium.

4.3 Weak scaling

Weak scaling describes how the execution time changes with the number of compute cores when the problem size (e.g. number of grid points) per core remains fixed. In the ideal case, doubling the number of compute cores should enable a problem twice as big to be solved in the same wall-clock time. However, for the k-space pseudospectral model, the computational work grows slightly greater than linear with $O(N \log(N))$ in the number of grid points due to the FFT, while the number of point-to-point communication events increases in the parallel implementation with $O(P^2)$ in the number of processors due to the matrix transpositions. As a consequence, the weak scaling curves will always grow (for a small number of cores, the computation time will be dominant while for large numbers of cores the communication overhead will take over). Figure 6(c) shows the weak scaling results for the benchmark set defined in Section 4.1 when using the `FFTW_EXHAUSTIVE` planning flag. Despite the fact that the cost of the underlying all-to-all communication step grows with the square of the number of processes, the graph shows relatively good weak scaling performance. The trends suggest that simulations with even larger grid sizes could be solved with reasonable efficiency using higher numbers of compute cores.

4.4 Memory usage

Figure 6(d) shows how the memory usage per core changes in-line with the strong scaling results given in Figure 6(b). Whenever the number of cores is doubled, the domain is partitioned into twice as many 2D slabs. Theoretically, the memory required by each core should be halved. However, in practice, there is an overhead associated with the domain decomposition that grows with the number of cores, and eventually becomes dominant. This behaviour is clearly observed for almost all the simulation sizes in the benchmark set. The overhead is comprised of several components, each with a different origin.

First, the MPI runtime allocates a significant amount of memory for communication buffers. When the MPI library is not restricted, it allocates as many communication buffers for each process as there are distinct messages to be received (one for each sender). Moreover, if the sparse communication pattern is used, additional scratch space must be allocated where smaller messages can be combined and separated on send and receive. For simulations using high process counts where the memory used per core for the local partition is quite low (tens of MB), the memory allocated for MPI communication buffers can become the dominant component.

Second, in addition to communication buffers, when the domain is partitioned into more parts over an increasing number of processes, the total memory consumed by locally replicating scalar variables and 1D arrays is increased. Moreover, additional memory is needed for storing the code itself. While the size of the compiled binary file is only on the order of 20 MB, a private copy is needed for every process. Thus, when 1024 processes are used, storing the code consumes more than 20 GB of memory. For small simulations, this can become a significant portion of the total memory consumption.

4.5 Simulation cost and execution time

For the user, another important metric that must be considered when running large-scale simulations is their financial cost. For this purpose, the simulation cost is defined as the product of the wall-clock execution time and the number of compute cores used. On VAYU, this cost is expressed in terms of service units (SUs). These are directly related to the number of core-hours used, scaled by a few other factors such as the priority the user assigns to the job. SUs represent an accountant's view on the parallel economy. On a large shared parallel computer, each user is typically allocated a share of the resource. On VAYU this corresponds to a certain number of SUs per quarter, and it is left to the user to determine how best to use this allocation. As scaling is never ideal, the more compute cores assigned to a job, the higher the effective cost. However, for time critical problems, using the highest possible number of cores may still be desirable.

Figure 7(a) shows the anticipated total simulation cost for the grid sizes in the benchmark set against the number of compute cores used. As the grid size increases, more time steps are also necessary to allow the ultrasound waves to propagate from one side of the domain to the other. Using the diagonal length of the domain and assuming a Courant-Friedrichs-Lewy (CFL) number of 0.2, the number of time steps grows from 2200 for a grid size of $256 \times 256 \times 256$ to 25,000 for a grid size of $4096 \times 2048 \times 2048$ grid points. The results in Figure 7(a) show that for a given grid size, the simulation cost remains fairly constant as a function of core count, with the ratio between the highest and lowest costs always less than 2 (this is to be expected given the strong-scaling results).

NCI, the owners of VAYU, charge US$ 0.1 per SU to commercial projects. Using this value, the approximate cost of each simulation in US dollars is also shown in Figure 7(a). If run to completion, the largest simulation performed would take around 4.5 days on 1024 compute cores and would cost slightly over US$10,000. At this point in time, such a large simulation is clearly not routine. However, VAYU is now a relatively old system, and with continued price performance trends it is not impossible to see the cost of such a simulation dropping to a few hundred dollars within the next few years.

4.6 Comparison of different architectures

Thus far, only the performance of the MPI implementation of the *k*-space pseudospectral model running on the VAYU cluster has been considered. For comparison, Figure 7(b) illustrates the execution times per time step for two equivalent models implemented in (a) MATLAB and (b) C++ but parallelised for shared-

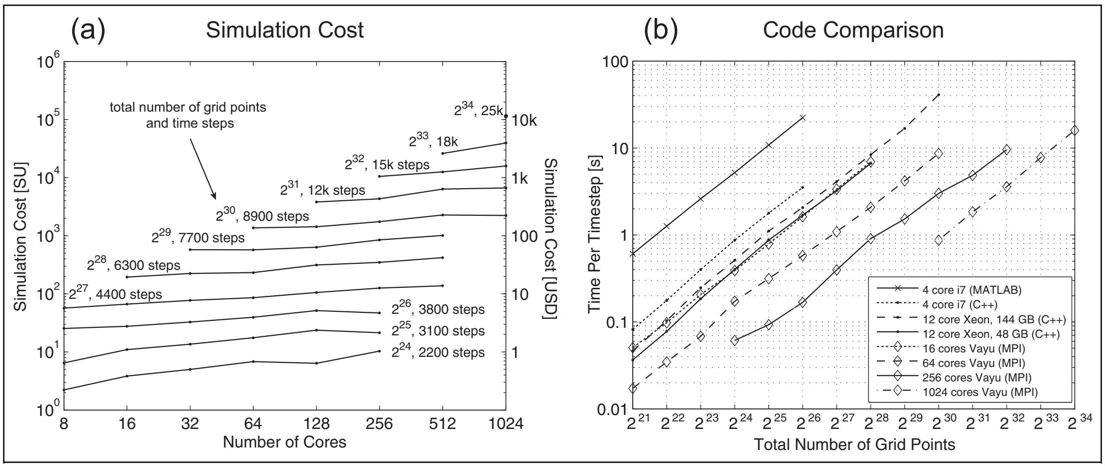


Figure 7. (a) The simulation cost in terms of service units (core-hours) displayed on the left *y*-axis and USD on the right *y*-axis. The number of time steps is derived from the time the wave needs to propagate along the diagonal length of the domain. (b) Execution time per time-step running different implementations of the *k*-space pseudospectral model on different machines.

memory architectures using OpenMP rather than MPI (some of the details of this implementation are discussed in Jaros et al. (2012)). For these comparisons, the benchmark set was extended to include smaller grid sizes starting at 2^{21} grid points ($128 \times 128 \times 128$). Three different computer configurations were considered: (1) a desktop machine with 12 GB of RAM and a single Intel Core i7 950 quad-core CPU running at 3.2 GHz, (2) a server with 48 GB of RAM and two Intel Xeon X5650 hex-core CPUs running at 2.66 GHz and (3) an identical server with 144 GB of RAM. The two server configurations differ in that the memory controller is not able to operate at 1333 MHz when serving 144 GB of RAM and drops its frequency to 800 MHz. The effect of this is to reduce the memory bandwidth by about 33%.

Each line in Figure 7(b) represents a different combination of computer configuration and implementation. The slowest performing combination is the MATLAB implementation running on the quad-core i7 system (this was the starting point for the development of the C++ codes (Treeby and Cox, 2010a)). Using the shared-memory C++ implementation on this machine reduces the execution time by a factor of about 8. Moving to the 12-core Xeon systems more than doubles this factor, although there are clear differences between the 48 GB and 144 GB configurations due to the fact that the model is memory bound. In comparison, to match the performance of the shared memory code running on the 12-core Xeon system, the MPI version of the code requires 16-cores on VAYU. The higher core-count needed by the MPI version of the code for equivalent performance reflects the fact that a non-trivial overhead is incurred since all communications between cores must pass through the MPI library (and in many cases, over the network). Of course, the benefit of the MPI implementation is the possibility of running the code over multiple nodes on a cluster, enabling both much larger domain sizes and much faster execution times. For example, for a grid size of 2^{26} grid points, the MPI code running on 256 cores is approximately two orders of magnitude faster than the MATLAB implementation on the quad-core i7 system, and one order of magnitude faster than the shared memory C++ implementation running on the 12-core Xeon systems.

5 Application example

To illustrate the utility of the developed nonlinear ultrasound model for solving real-world problems, a complete large-scale nonlinear ultrasound simulation representing a single HIFU sonication of the kidney was performed. The medium properties for the simulation were derived from an abdominal CT scan using the MECANIX dataset (OsiriX Imaging Software)

This was re-sampled using linear interpolation to give the appropriate resolution. The density of the tissue was calculated from Hounsfield units using the data from Schneider et al. (1996), and the sound speed was then estimated using the empirical relationship given by Mast (2000). The remaining material properties were assigned book values (Duck, 1990). The HIFU transducer was defined as a circular bowl with a width of 10 cm and a focal length of 11 cm. The shape of the transducer within the 3D Cartesian grid was defined using a 3D extension of the midpoint circle algorithm (Treeby and Cox, 2010a). The transducer was positioned behind the patient as shown in Figure 8(a), and

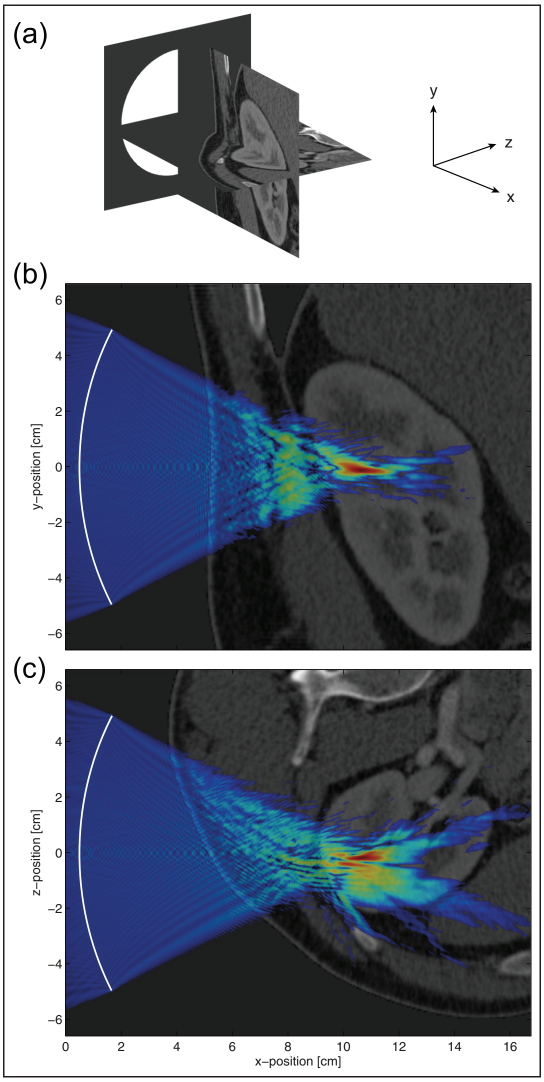


Figure 8. (a) Sagittal and transverse slices through the abdominal CT scan used to define the material properties for simulating a HIFU treatment of the kidney. The approximate position of the HIFU transducer is shown with a white circle. (b)-(c) Saggittal and transverse slices through the simulated distribution of maximum pressure overlaid onto the corresponding CT slices. The pressure distribution is displayed using a log-scale and is thresholded at -30 dB. The distortion of the HIFU focus due to the body wall and the fat layer surrounding the kidney is clearly visible.

was driven at 1 MHz by a continuous wave sinusoid. The acoustic intensity at the transducer surface was set to 2 W/cm^2 to simulate a treatment that would likely operate largely in a thermal regime (i.e. with minimal cavitation). Outside the body, the medium was assigned the properties of water (Duck, 1990). The total domain size was 17 cm × 14.3 cm × 14.3 cm and the grid spacing in each Cartesian direction was set to 93 μm, giving a total grid size of 2048 × 1536 × 1536 grid points and a maximum supported frequency of 8 MHz (i.e. eight harmonics of the source frequency). The simulation length was set to 220 μs with a CFL number of 0.18, giving a total of 19,800 time steps. Simulations of this scale and complexity have not previously been possible.

The simulation was executed using 768 cores on VAYU with seven checkpoint-restart stages. The total wall-clock time was 31 hours and 20 minutes, and the total memory used was 780 GB. The compressed input file was 45 GB, while the output file was 450 GB. This comprised 36 GB to store the peak positive and peak negative pressure across the domain, and 414 GB to store the time varying pressure and particle velocity for 6 periods in steady state over a 45 mm × 30 mm × 30 mm region surrounding the HIFU focus. Transverse and sagittal slices through the peak positive pressure overlaid onto the corresponding CT data used to define the material properties are shown in Figure 8(b)-(c). The distortion of the ultrasound focus due to the body wall and the fat layer surrounding the kidney is clearly visible. These effects have been noted clinically, and remain a barrier to the application of HIFU in the kidney (Illing et al., 2005). Thus, one possible future application of the developed *k*-space model would be a systematic investigation into the conditions necessary for viable HIFU ablation in the kidney (for example, the maximum thickness of the fat layer). In any case, the example serves to illustrate the utility of the implementation.

6 Summary and discussion

A full-wave model for simulating the propagation of nonlinear ultrasound waves through absorbing and heterogeneous media is presented in the context of model-based treatment planning for HIFU. The governing equations are discretised using the *k*-space pseudospectral method. The model is implemented in C++ using MPI to enable large-scale problems to be solved using a distributed computer cluster. The performance of the model is evaluated using grid sizes up to 4096 × 2048 × 2048 grid points and up to 1024 compute cores. This is significantly larger than most ultrasound simulations previously presented, both in terms of grid size and the number of wavelengths this accurately represents. Given the global nature of the gradient calculation, the model shows good strong scaling behaviour, with a speed-up of 1.7x whenever the number of cores is doubled. This means large-scale problems can be spread over increasing numbers of compute cores with only a small computational overhead. The overall efficiency of the parallel implementation is on the order of 60%, which corresponds to the ratio between computation and communication. The large communication overhead is due to the global all-to-all transposition that must be performed for every FFT (a second transposition is avoided by performing operations in the spatial frequency domain in transformed space). Finally, the efficacy of the model for studying real-world problems in HIFU is demonstrated using a large-scale simulation of a HIFU sonication of the kidney.

In the context of the large-scale problems outlined in Section 1 and Table 1, the model developed here allows many problems of interest to be solved using a full-wave model for the first time. This is relevant for studying the aberration of HIFU beams in the body when the focal intensities are relatively low. This has many possible applications, for example, in treatment planning, exposimetry, patient selection and equipment design. However, solving even larger problems involving high focal intensities where many 10's or 100's of harmonics may be present (Yuldashev and Khokhlova, 2011) is still currently out of reach. Looking forward, there are two numerical strategies that might allow the model to be extended further. First, the governing equations could be solved on a non-uniform grid (Treeby, 2013). In the current model, the grid spacing is globally constrained by the highest frequency harmonic that exists anywhere in the domain. However, in practice, the high-frequency harmonics are usually restricted to a small region near the ultrasound focus. Using a non-uniform grid would allow the grid points to be clustered around steep regions of the wavefield, and thus significantly reduce the total number of grid points needed for accurate simulations (Treeby, 2013). Second, a domain decomposition approach could be used in which FFTs are computed locally on each partition using local Fourier basis (Israeli et al., 1994; Albin et al., 2012). This would replace the global communication required by the 3D FFT with local communications between processes responsible for neighbouring partitions.

Considering the computer code, the main limitation is related to the 1D decomposition used to partition the domain. Although this approach exhibits relatively good scaling characteristics, it limits the maximum number of cooperating processes to be $P \leq \min(\mathrm{N}_y, \mathrm{N}_z)$. This limitation is particularly relevant looking towards the exascale era, where supercomputers integrating over 1M cores are predicted to appear before 2020 (Dongarra et al., 2011). Being prepared for this sort of compute facility requires simulation tools that can efficiently employ hundreds of thousands of compute cores. Moreover, while the trend in supercomputing is to integrate more and more compute cores, the total amount of memory is growing much more slowly (Dongarra et al., 2011). Effectively, this means the memory available per core will remain constant or even decrease in next generation systems. As an example, VAYU has 11,936 cores with 3 GB/core, while its successor RAIJIN has 57,472 cores with only 2 GB/core. This is relevant because with the current 1D decomposition, the maximum grid size that can be solved is ultimately limited by the memory per core. Both of these drawbacks could be solved by using a 2D partitioning approach where the 2D slabs are further broken into 1D pencils, with every process assigned a subset of pencils rather than a complete 2D slab. This would make higher numbers of compute cores (and thus memory) accessible to the simulation, where $P \leq \min\left(\mathrm{N}_y^2, \mathrm{N}_z^2\right)$.

Another challenge for the current implementation is the amount of output data generated by the code. When recording the time-varying acoustic pressure and particle velocity in a central region (e.g. near the HIFU focus), a single simulation can easily generate 0.5 TB of output data. Copying, post-processing, visualising and archiving such large amounts of data quickly becomes impractical. New techniques for on-the-fly post-processing are thus needed. The use of localised data sampling also introduces a work imbalance into the simulation code. If the output data is only collected from a small region of the domain, only a small subset of the processes actually store data to the disk, with the rest idle. The effective bandwidth to disk could thus be improved by redistributing the data to idle processes after it is collected, allowing more cores to be used for disk operations. Similarly, if some processes only collect a very small amount of data (e.g. from a single grid point in the local partition), the I/O subsystem can become congested by many small write requests resulting in poor performance. In this case, it would be better to collect the output data within each node before writing to disk. These improvements will be explored as part of a future work.

Funding

This work was supported in part by the Australian Research Council/Microsoft Linkage Project LP100100588, the Engineering and Physical Sciences Research Council UK, and the SoMoPro II Programme co-financed by the European Union and the South-Moravian Region. Computational resources were provided by the Australian National Computing Infrastructure (NCI) National Facility at the Australian National University (VAYU supercomputer), and iVEC at Murdoch University (EPIC supercomputer).

References

Aanonsen SI, Barkve T, Tjotta JN and Tjotta S (1984) Distortion and harmonic generation in the nearfield of a finite amplitude sound beam. *J. Acoust. Soc. Am.*, 75(3): 749–768.

Albin N, Bruno OP, Cheung TY and Cleveland RO (2012) Fourier continuation methods for high-fidelity simulation of nonlinear acoustic beams. *J. Acoust. Soc. Am.*, 132(4):2371–87.

Averkiou MA and Cleveland RO (1999) Modeling of an electrohydraulic lithotripter with the KZK equation. *J. Acoust. Soc. Am.*, 106(1):102–112.

Berenger JP (1996) Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 127(2):363–379.

Bojarski NN (1982) The k-space formulation of the scattering problem in the time domain. *J. Acoust. Soc. Am.*, 72(2): 570–584.

Bojarski NN (1985) The k-space formulation of the scattering problem in the time domain: An improved single propagator formulation. *J. Acoust. Soc. Am.*, 77(3):826–831.

Chen W and Holm S (2004) Fractional Laplacian time-space models for linear and nonlinear lossy media exhibiting arbitrary frequency power-law dependency. *J. Acoust. Soc. Am.*, 115(4):1424–1430.

Clement GT (2004) Perspectives in clinical uses of high-intensity focused ultrasound. *Ultrasonics*, 42(10):1087–93.

Connor CW and Hynynen K (2002) Bio-acoustic thermal lensing and nonlinear propagation in focused ultrasound surgery using large focal spots: A parametric study. *Phys. Med. Biol.*, 47(11):1911–28.

Cox BT, Kara S, Arridge SR and Beard PC (2007) k-space propagation models for acoustically heterogeneous media: Application to biomedical photoacoustics. *J. Acoust. Soc. Am.*, 121(6):3453–3464.

Curra FP, Mourad PD, Khokhlova VA, Cleveland RO and Crum LA (2000) Numerical simulations of heating patterns and tissue temperature response due to high-intensity focused ultrasound. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 47(4):1077–89.

Daoud MI and Lacefield JC (2009) Distributed three-dimensional simulation of B-mode ultrasound imaging using a first-order k-space method. *Phys. Med. Biol.*, 54(17):5173–5192.

Demi L, van Dongen KWA and Verweij MD (2011) A contrast source method for nonlinear acoustic wave fields in media with spatially inhomogeneous attenuation. *J. Acoust. Soc. Am.*, 129(3):1221–1230.

Dongarra J, Beckman P, Moore T, et al. (2011) The international exascale software project roadmap. *Int. J. High Perform. Comput. Appl.*, 25(1):3–60.

Duck FA (1990) *Physical Properties of Tissue: A Comprehensive Reference Book*. London: Academic Press.

Fornberg B (1987) The pseudospectral method: Comparisons with finite differences for the elastic wave equation. *Geophysics*, 52(4):483–501.

Frigo M and Johnson SG (2005) The design and implementation of FFTW3. *Proc. IEEE*, 93(2):216–231.

Frigo M and Johnson SG (2012) FFTW user manual. Technical Report, Massachusetts Institute of Technology, USA,November.

Haber I, Lee R, Klein HH and Boris JP (1973) Advances in electromagnetic plasma simulation techniques. In *Proc. Sixth Conf. Num. Sim. Plasmas*, pages 46–48.

Hamilton MF and Morfey CL (2008) Model Equations. In Hamilton MF and Blackstock DT (eds) *Nonlinear Acoustics*. New York: Acoustical Society of America, pp.41–63.

Hesthaven JS, Gottlieb S and Gottlieb D (2007) *Spectral Methods for Time-Dependent Problems*. Cambridge: Cambridge University Press.

Huijssen J and Verweij MD (2010) An iterative method for the computation of nonlinear, wide-angle, pulsed acoustic fields of medical diagnostic transducers. *J. Acoust. Soc. Am.*, 127(1):33–44.

Illing RO, Kennedy JE, Wu F, et al. (2005) The safety and feasibility of extracorporeal high-intensity focused ultrasound (HIFU) for the treatment of liver and kidney tumours in a Western population. *Brit. J. Cancer,* 93(8): 890–895.

Israeli M, Vozovoi L and Averbuch A (1994) Domain decomposition methods with local fourier basis for parabolic problems. *Contemp. Math*, 157:223–230.

Jaros J, Treeby BE and Rendell AP. Use of multiple GPUs on shared memory multiprocessors for ultrasound propagation simulations. In: *10th Australasian Symposium on Parallel and Distributed Computing* (eds Chen J and Ranjan R), Melbourne, Australia, 31 January - 3 February 2012, volume 127, pp. 43–52. Melbourne: ACS.

Jing Y, Wang T and Clement GT (2012) A k-space method for moderately nonlinear wave propagation. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 59(8): 1664–1673.

Jolesz FA and Hynynen KH (2008) *MRI-guided focused ultrasound surgery*. New York: Informa Healthcare.

Kennedy JE, ter Haar GR and Cranston D (2003) High intensity focused ultrasound: Surgery of the future? *Brit. J. Radiol.*, 76(909):590–599.

Khokhlova VA, Souchon R, Tavakkoli J, Sapozhnikov OA and Cathignol D (2001) Numerical modeling of finite-amplitude sound beams: Shock formation in the near field of a cw plane piston source. *J. Acoust. Soc. Am.,* 110(1): 95–108.

Khokhlova VA, Bessonova OV, Soneson JE, Canney MS, Bailey MR and Crum LA (2010) Bandwidth limitations in characterization of high intensity focused ultrasound fields in the presence of shocks. In: *9th International Symposium on Therapeutic Ultrasound*, Tokyo, Japan, 9–12 June 2010, pp. 363–366.

Lepock JR (2003) Cellular effects of hyperthermia: Relevance to the minimum dose for thermal damage. *Int. J. Hyperthermia*, 19(3):252–66.

Liu LH, McDannold N and Hynynen K (2005) Focal beam distortion and treatment planning in abdominal focused ultrasound surgery. *Med. Phys.*, 32 (5):1270–1280.

Liu QH (1999) Large-scale simulations of electromagnetic and acoustic measurements using the pseudospectral time-domain (PSTD) algorithm. *IEEE. T. Geosci. Remote*, 37(2):917–926.

Mast TD (2000) Empirical relationships between acoustic parameters in human soft tissues. *Acoustics Research Letters Online*, 1(2):37–42.

Mast TD, Souriau LP, Liu DLD, Tabei M, Nachman AI and Waag RC (2001) A k-space method for large-scale models of wave propagation in tissue. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 48(2):341–354.

Okita K, Ono K, Takagi S and Matsumoto Y (2011) Development of high intensity focused ultrasound simulator for large-scale computing. *Int. J. Numer. Meth. Fluids*, 65:43–66.

Okita K, Narumi R, Azuma T, Takagi S and Matumoto Y (2014) The role of numerical simulation for the development of an advanced HIFU system. *Comput. Mech.*, 54(4): 1023–1033.

OsiriX Imaging Software, DICOM sample image sets. Available from: http://www.osirix-viewer.com/datasets. [19 June 2013].

Paulides MM, Stauffer PR, Neufeld E, et al. (2013) Simulation techniques in hyperthermia treatment planning. *Int. J. Hyperthermia*, 29(4):346–357.

Pekurovsky D (2012) P3DFFT: A framework for parallel computations of Fourier transforms in three dimensions. *SIAM J. Sci. Comput.,* 34(4): C192–C209.

Pinton G, Aubry JF, Fink M and Tanter M (2011) Effects of nonlinear ultrasound propagation on high intensity brain therapy. *Med. Phys.,* 38(3): 1207–1216.

Pinton GF, Dahl J, Rosenzweig S and Trahey GE (2009) A heterogeneous nonlinear attenuating full-wave model of ultrasound. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 56(3):474–488.

Pulkkinen A, Werner B, Martin E and Hynynen K (2014) Numerical simulations of clinical focused ultrasound functional neurosurgery. *Phys. Med. Biol.*, 59(7):1679–1700.

Schneider U, Pedroni E and Lomax A (1996) The calibration of CT Hounsfield units for radiotherapy treatment planning. *Phys. Med. Biol.*, 41(1):111–124.

Tabei M, Mast TD and Waag RC (2002) A k-space method for coupled first-order acoustic propagation equations. *J. Acoust. Soc. Am.*, 111(1):53–63.

ter Haar G (2007) Therapeutic applications of ultrasound. *Prog. Biophys. Mol. Biol.*, 93(1–3):111–129.

ter Haar G, Shaw A, Pye S, et al. (2011) Guidance on reporting ultrasound exposure conditions for bio-effects studies. *Ultrasound Med. Biol.,* 37(2): 177–183.

Tillett JC, Daoud MI, Lacefield JC and Waag RC (2009) A k-space method for acoustic propagation using coupled first-order equations in three dimensions. *J. Acoust. Soc. Am.*, 126(3):1231–1244.

Treeby BE (2013) Modeling nonlinear wave propagation on nonuniform grids using a mapped k-space pseudospectral method. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 60(10):2208–2013.

Treeby BE and Cox BT (2010a) k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *J. Biomed. Opt.*, 15(2): 021314.

Treeby BE and Cox BT (2010b) Modeling power law absorption and dispersion for acoustic propagation using the fractional Laplacian. *J. Acoust. Soc. Am.*, 127(5):2741–2748.

Treeby BE and Cox BT (2011) A k-space Greens function solution for acoustic initial value problems in homogeneous media with power law absorption. *J. Acoust. Soc. Am.*, 129(6): 3652–3660.

Treeby BE, Tumen M and Cox BT (2011) Time domain simulation of harmonic ultrasound images and beam patterns in 3D using the k-space pseudospectral method. In: *Medical Image Computing and Computer-Assisted Intervention, Part I.* Heidelberg: Springer, vol 6891 pp.363–370.

Treeby BE, Jaros J, Rendell AP and Cox BT (2012) Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method. *J. Acoust. Soc. Am.,* 131(6): 4324–4336.

Verweij MD and Huijssen J (2009) A filtered convolution method for the computation of acoustic wave fields in very large spatiotemporal domains. *J. Acoust. Soc. Am.*, 125(4): 1868–1878.

Westervelt PJ (1963) Parametric acoustic array. *J. Acoust. Soc. Am.,* 35(4): 535–537.

Wojcik G, Mould J, Ayter S and Carcione L (1998) A study of second harmonic generation by focused medical transducer pulses. In: *IEEE International Ultrasonics Symposium*, Sendai, Japan, 5–8 October 1998, pp.1583–1588

Wojcik GL, Mould J, Abboud N, et al. (1995) Nonlinear modeling of therapeutic ultrasound. In: *IEEE International Ultrasonics Symposium*, Seattle, WA, 7–10 November 1995, pp. 1617–1622. IEEE.

Yeung PK, Donzis DA and Sreenivasan KR (2012) Dissipation, enstrophy and pressure statistics in turbulence simulations at high Reynolds numbers. *Journal of Fluid Mechanics*, 700:5–15.

Yuldashev PV and Khokhlova VA (2011) Simulation of three-dimensional nonlinear fields of ultrasound therapeutic arrays. *Acoust. Phys.,* 57(3): 334–343.

Yuldashev PV, Krutyansky LM, Khokhlova VA, et al. (2010) Distortion of the focused finite amplitude ultrasound beam behind the random phase layer. *Acoust. Phys.*, 56(4): 467–474.

Zhang L and Wang ZB (2010) High-intensity focused ultrasound tumor ablation: review of ten years of clinical experience. *Front. Med. China,* 4(3): 294–302.

Author biography

Jiri Jaros is currently a Marie Curie Fellow and an assistant professor at the Faculty of Information Technology, Brno University of Technology. He received his MSc and PhD in Computer Science from the Brno University of Technology in 2003 and 2010 respectively. He worked for two years as a postdoctoral researcher at the Australian National University in the Computer Systems group under the supervision of Prof Alistair Rendell, and 6 months as a postdoctoral researcher in the Centre for Computational Science, University College London under Prof Peter Coveney. His research interests include high performance computing, scientific computation, parallel programming, many-core accelerator and GPU architecture and programming. He is an active developer of the k-Wave project responsible for large-scale code development, validation and optimisation.

Alistair Rendell is a professor and currently the Director of the Research School of Computer Science at the Australian National University. He received a BSc in chemistry from Durham University, UK in 1983 and a PhD in theoretical chemistry from Sydney University, Australia in 1988. His research interests include computational science, high performance computing, parallel and distributed programming and computer architecture.

Bradley Treeby is currently an EPSRC Early Career Fellow and leads the Biomedical Ultrasound Group in the Department of Medical Physics and Biomedical Engineering at University College London, UK. He received a BE degree with first class honors in Mechatronics Engineering in 2003, and a PhD in acoustics in 2007, both from the University of Western Australia, Australia. His research interests include biomedical ultrasound, numerical methods and high performance computing. He has published more than 40 scientific papers, and is the author of an open-source acoustics toolbox for MATLAB called k-Wave.

B.2 Hybrid 3D Fast Fourier Transform

Nikl, V.; **Jaros, J.**: Parallelisation of the 3D fast Fourier transform using the hybrid OpenMP/MPI decomposition. In *Lecture Notes in Computer Science*, vol. 8934. Springer International Publishing Switzerland. 2014. ISBN 978-3-319-14895-3. ISSN 16113349. pp. 100–112. doi:10.1007/978-3-319-14896-0_9.

Parallelisation of the 3D Fast Fourier Transform Using the Hybrid OpenMP/MPI Decomposition

Vojtech Nikl( ) and Jiri Jaros

Faculty of Information Technology, Brno University of Technology,
Bozetechova 2, 612 66 Brno, Czech Republic
{inikl,jarosjir}@fit.vutbr.cz

Abstract. The 3D fast Fourier transform (FFT) is the heart of many simulation methods. Although the efficient parallelisation of the FFT has been deeply studied over last few decades, many researchers only focused on either pure message passing (MPI) or shared memory (OpenMP) implementations. Unfortunately, pure MPI approaches cannot exploit the shared memory within the cluster node and the OpenMP cannot scale over multiple nodes.

This paper proposes a 2D hybrid decomposition of the 3D FFT where the domain is decomposed over the first axis by means of MPI while over the second axis by means of OpenMP. The performance of the proposed method is thoroughly compared with the state of the art libraries (FFTW, PFFT, P3DFFT) on three supercomputer systems with up to 16k cores. The experimental results show that the hybrid implementation offers 10-20% higher performance and better scaling especially for high core counts.

1 Introduction

The fast Fourier transform (FFT)[1] is the heart of many spectral simulation methods where it is used to calculate spatial gradients of various physical quantities. This approach eliminates the numerical dispersion that arises from the discretisation of the spatial derivative operators, and significantly reduces the grid density required for accurate simulations [2].

A recent application of spectral methods, we have been working on, is the k-Wave toolbox [3] oriented on the full-wave simulation of the ultrasound waves propagation in biological materials (both soft and hard tissues) intended for ultrasound treatment planning such as cancer treatment, neurostimulation, diagnostics, and many other. In many realistic simulations with domain sizes ranging from 512^3 to 4096^3, as much as 60% of the total computational time is attributed to the 3D FFTs. Reducing the 3D FFT compute time thus remains a challenge even in the petascale era [4].

Many libraries have been developed to compute the FFT in the massively parallel distributed memory environment, such as FFTW (Fastest Fourier Transform from West)[5], PFFT (Parallel FFT)[6] and P3DFFT (Parallel Three-Dimensional Fast Fourier Transforms)[7]. All of these libraries use the

P. Hliněný et al. (Eds.): MEMICS 2014, LNCS 8934, pp. 100–112, 2014.
DOI: 10.1007/978-3-319-14896-0_9

pure-MPI message passing approach to calculate the FFT in parallel. However, modern high-performance computer architectures usually consist of a hybrid of the shared and distributed paradigms: distributed networks of multicore processors. The hybrid paradigm marries the high bandwidth low-latency interprocess communication featured by shared memory systems with the massive scalability afforded by distributed computing.

In this work, we describe recent efforts to exploit modern hybrid architectures, using the popular MPI interface to communicate among distributed nodes and the OpenMP multi-threading paradigm to communicate among the individual cores of each processor to speed up the calculation of 3D Fast Fourier Transform. Moreover, we introduce a novel hybrid 2D pencil decomposition that allows us to employ more compute cores than the standard 1D slab decomposition implemented in the FFTW while keeping the communication burden significantly lower compared to PFFT and P3DFFT also based on pencil decompositions.

2 Parallel Implementations of the 3D Fast Fourier Transform

There are two main approaches for parallelising multidimensional FFTs; the first is binary exchange algorithms, and the second is transpose algorithms. An introduction and theoretical comparison can be found in [8]. In this paper, we restrict ourselves to transpose algorithms that need much less data to be exchanged [9] and have direct support in many software libraries, e.g. FFTW [5].

Regardless of decomposition, a Fourier transform in three dimensions is comprised of three 1D FFTs in the three dimensions (X, Y, and Z) in turn. When all of the data in a given dimension of the grid resides entirely in a processors memory (i.e., it is local) the transform consists of a 1D FFT done over multiple grid lines by every processor, which can be accomplished by a serial algorithm provided by many well-known FFT libraries and is usually a fairly fast operation. The transforms proceed independently on each processor with regard to its own assigned portion of the array. When the data are divided across processor boundaries (i.e., nonlocal), the array is reorganized by a single step of global transposition so that the dimension to be transformed becomes local, and then serial 1D FFT can be applied again. These global transpositions are known to be the main bottleneck of the 3D FFT since arithmetic intensity (computational work divided by communication work) grows only as a factor of $logN$ [5], [6], [7].

A general algorithm to calculate a distributed 3D FFT of size $Z \times Y \times X$ stored in C-like row major order follows this procedure:

1. Perform $Z \times Y$ one-dimensional FFTs along the X axis.
2. Perform $X \leftrightarrow Y$ data transposition.
3. Perform $Z \times X$ one-dimensional FFTs along the Y axis.
4. Perform $Z \leftrightarrow X$ data transposition.
5. Perform $Y \times X$ one-dimensional FFTs along the Z axis.
6. Transpose data back into the original order (optional).

2.1 Decomposition of the 3D Fast Fourier Transform

Solving the 3D FFT in parallel requires the compute grid to be partitioned and distrusted over processing cores. In the case of 3D FFT, there are three possible ways how to partition the grid; one-dimensional slab decomposition, two-dimensional pencil decomposition, and three-dimensional cube decomposition (see Fig. 1).

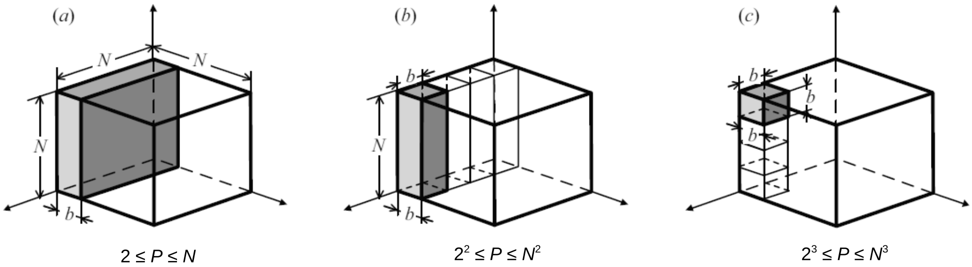


Fig. 1. Domain decompositions for three-dimensional grid over P processing cores. (a) slab decomposition, (b) pencil decomposition, (c) cube decomposition [10]. Data associated with a single processing core is shaded.

Most of the parallel 3D FFT libraries to date use the slab domain decomposition over the first dimension (Z in our case) [5], [11]. This decomposition is faster on a limited number of cores because it only needs one global transpose, minimizing communication. The main disadvantage of this approach is that the maximum parallelisation is limited by the largest size along an axis of the 3D data array used. At the age of petascale platforms more and more systems typically have numbers of processing cores far exceeding this limit. For example, cutting edge ultrasound simulations performed by the k-Wave toolbox [3] use 2048^3 grids and so with the slab decomposition would scale only to 2048 cores at most leading to the calculation time exceeding clinically acceptable time of 24 hours (here between 50 and 100 hours).

The second approach is the 2D pencil decomposition that further partitions the slabs into a set of pencils, see Fig. 1(b). This approach has recently been implemented in two novel FFT libraries PFFT[6] and P3DFFT[7]. Although this approach increases the maximum number of processor cores from N to N^2, it also requires another global communication. Nevertheless, these global transposition steps require communication only among subgroups of all compute cores. However according to Pekurovsky [7], attention must be paid to the pencil placement over the computing cores to keep good locality and efficacy.

The cube decomposition studied in [10] brings the highest scalability, however it requires one-dimensional FFTs to be calculated non-locally and thus fine-tuned FFT cores provided by FFTW cannot be used.

The parallel 3D FFT is usually implemented using a pure-MPI approach and one of the described decomposition techniques. However, many current supercomputers comprise of shared memory nodes typically integrating 16 cores. The use of shared memory significantly reduces the amount of inter-process communication and helps in exploiting local caches. The most sensible implementation of the hybrid decomposition bases on the pencil decomposition where a slab is assigned per compute node, and the cores within node take each their portion of pencils. One of the obvious advantages of exploiting hybrid parallelism is the reduction in communication since messages no longer have to be passed between threads sharing a common memory pool. Another advantage is that some algorithms can be formulated, through a combination of memory striding and vectorization, so that local transposition is not required within a single MPI node (while this is even possible for multi-dimensional FFTs, the recent availability of serial cache-oblivious in-place transposition algorithms appears to have tipped the balance in favour of doing a local transpose). The hybrid approach also allows smaller problems to be distributed over a large number of cores. This is particularly advantageous for 3D FFTs: the reduced number of MPI processes allows for a more slab-like than pencil.

Some authors object that this approach does not push the scaling significantly far [7]. However, for the grid of practical interest (1024^3 - 4096^3), the number of cores that can be employed lies between 16384 and 65536. These numbers of cores can only offer largest supercomputers in Europe accessible via the PRACE Tier-0 allocation scheme[1]. As the trend of integrating more cores within a node is going to continue, we consider the scaling to be good enough from the practical point of view. Although pure-MPI implementation may allow us to distribute the work over much more compute cores, the efficiency is then still very low anyway (less than 6% for 100k and more cores as presented in [6]).

2.2 Libraries for Distributed FFT

This section provides an overview of the most popular libraries for calculating the 3D FFT using both the slab and pencil decomposition and serves as a firm background for experimental comparison.

The Fastest Fourier Transform in the West (FFTW)[5] is probably the most popular library for calculating n-dimensional FFT over an arbitrary input size grid and still reaching the $NlogN$ time complexity. FFTW uses the so called *plan and execute* approach to select the most suitable implementation of FFT for the underlying hardware. This allows FFTW to be easily portable and still extremely fast. The FFTW supports both multi-threaded and memory distributed architectures. In case of distributed memory environment, the grid is decomposed using the slab decomposition. This feature is considered to be a significant drawback nowadays. Fortunately, FFTW allows to combine multi-threaded FFT kernels with custom grid decomposition and data exchange and is thus often used as

[1] PRACE: Partnership for Advanced Computing in Europe, http://www.prace-ri.eu

basis for advanced implementations (some of them are discussed later in this section).

The Parallel FFT library (PFFT) proposed by Michael Pippig [6] is one of a few FFT implementation using the pencil decomposition, unfortunately it is still in an alpha version. It builds on serial FFTW kernels applied on one-dimensional FFT and custom data exchange around the pure-MPI approach. PFFT has been tested on a BlueGene/P machine employing up to 256k PowerPC cores. However, the scaling with increasing number of cores becomes flat reaching only 6% for 256k cores.

The last library we took into account is the Parallel three-dimensional FFT (P3DFFT) by Dimitry Pekurovsky [7]. This library is specialised on calculating the 3D FFT using the pencil decomposition and the pure-MPI approach. The library employs one-dimensional kernels provided by FFTW or IBM ESSL[2]. This implementation allows to collapse the pencil decomposition into the slab one for low core counts preserving good efficacy. The implementation shows good performance for moderate core counts up to 65k. One of the main obstacles for us is the implementation language being Fortran and the support for only real-to-complex and complex-to-real transforms.

3 Proposed Method

The proposed implementation of the distributed hybrid OpenMP/MPI 3D FFT is called HyFFT. It is based on the modified pencil decomposition built on the top of the FFTW library. The 3D grid is first decomposed by MPI processes into slabs. The slabs are further partitioned into pencils assigned to threads to demand. This ensures the entire slab being always stored within the shared memory leading to the first transposition being local. In the corner case of small grids where the number of slabs is smaller than the number of cores, the decomposition naturally collapses into the original 1D slab decomposition and the pure-MPI implementation.

Exploiting full potential of modern clusters with multicore/multisocket nodes introduces some restrictions on the process/threads placement on nodes, sockets and cores. In the case of dual-socket x86 clusters, the best is usually to run a separate process per socket and spawn as many threads as cores per socket. This yields the advantage of the slab being stored in the socket's local memory with the fastest access. If a higher number of threads (higher scaling) is required, a single process per node can be run, instead. However, this implies the slab to be split over two memory islands leading in the non uniform memory access (NUMA) slowing down the local transposition. The situation is similar in the case of IBM PowerPC architectures, though the best is to spawn two threads per core to fully exploit all its HW resources.

The proposed HyFFT follows the diagram shown in Fig. 2. We can clearly see three series of 1D FFTs interleaved with local and global transpositions.

[2] http://www-03.ibm.com/systems/power/software/essl/

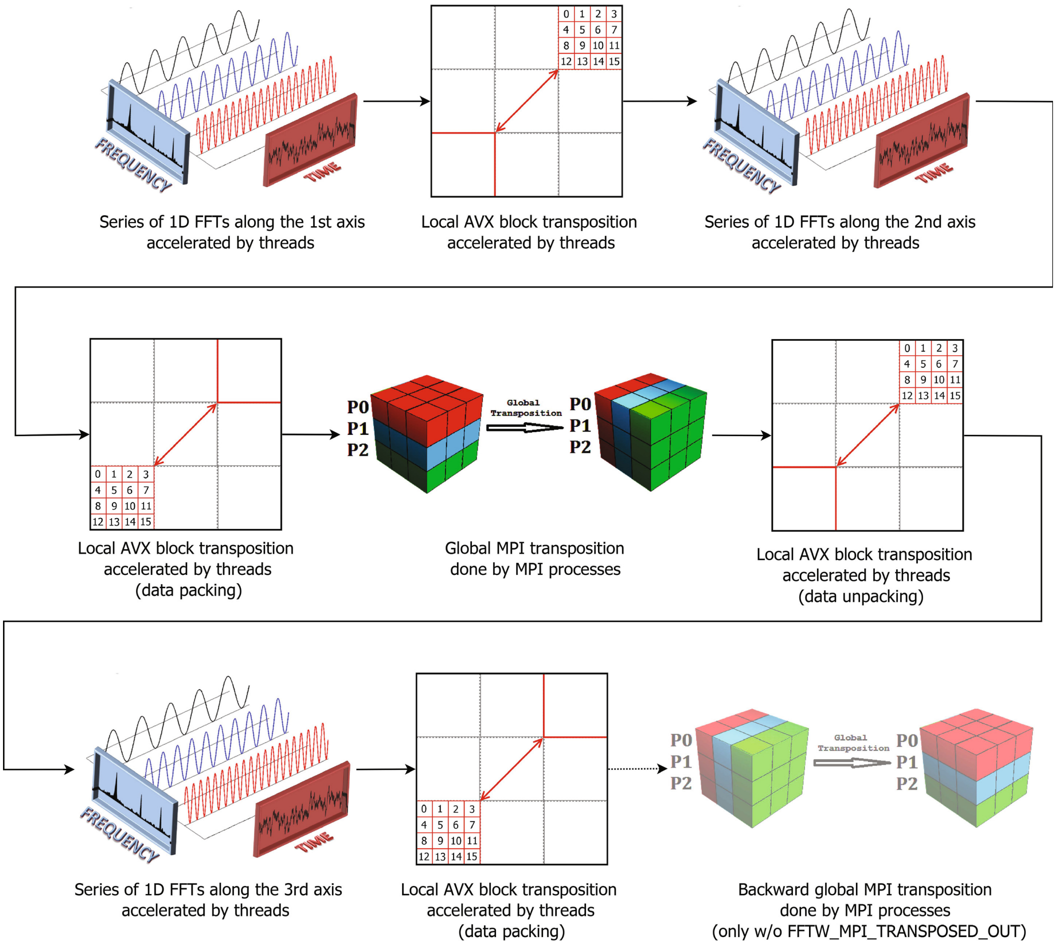
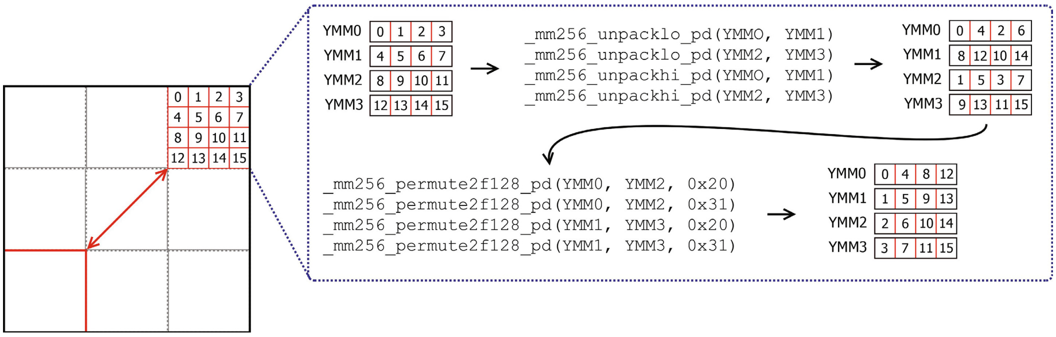

FREQUENCY
TIME
Series of 1D FFTs along the 1st axis accelerated by threads
Local AVX block transposition accelerated by threads
Series of 1D FFTs along the 2nd axis accelerated by threads
Local AVX block transposition accelerated by threads (data packing)
P0
P1
P2
Global Transposition
Global MPI transposition done by MPI processes
Local AVX block transposition accelerated by threads (data unpacking)
Series of 1D FFTs along the 3rd axis accelerated by threads
Local AVX block transposition accelerated by threads (data packing)
Backward global MPI transposition done by MPI processes (only w/o FFTW_MPI_TRANSPOSED_OUT)

Fig. 2. The steps of HyFFT to be carried out to perform a forward 3D FFT

The first local transpositions rearranges data within a slab before the second FFT transform. The global transposition is wrapped by data packing and unpacking steps carried out as local transpositions. The last FFT transform is followed by a local transposition to get the output data compatible with the FFTW library under `FFTW_MPI_TRANSPOSED_OUT` flag omitting the second global transposition for the sake of performance. However, if the same shape of the grid is required after the 3D FFT, the global transpose has to be performed.

The calculation itself comprises of three main kernels as outlined in Section 2: series of 1D FFTs, local transpositions and a global distributed transposition:

1. **FFT kernels**: There are two different ways how to calculate FFTs over the slab in the shared memory. The one primarily used in this work distributes the pencils over the threads using OpenMP pragmas, calculates 1D FFTs in parallel using 1D FFTW kernels, performs the local transposition and continues over the second axis. If there are more pencils in the slab than threads, every thread is responsible for a bunch of pencils. These can be calculated



```
_mm256_unpacklo_pd(YMM0, YMM1)
_mm256_unpacklo_pd(YMM2, YMM3)
_mm256_unpackhi_pd(YMM0, YMM1)
_mm256_unpackhi_pd(YMM2, YMM3)
```

```
_mm256_permute2f128_pd(YMM0, YMM2, 0x20)
_mm256_permute2f128_pd(YMM0, YMM2, 0x31)
_mm256_permute2f128_pd(YMM1, YMM3, 0x20)
_mm256_permute2f128_pd(YMM1, YMM3, 0x31)
```

Fig. 3. The block based local transposition using the Intel AVX vector intrinsics

one by one (our approach) or simultaneously. Calculating a bunch of pencils sequentially is preferred for larger grid sizes due to a better utilisation of L1 cache (e.g. a complex single precision pencil of 1024 grid points occupies 8KB - one half of L1 cache) and because of only having a single implementation of the FFT kernel for all three calculation phases.

The second approach to calculate the FFT over the slab is a use a multi-threaded 2D FFT provided by FFTW instead of the doing the sequence of 1D FFTs, local transposition and 1D FFTs. This can increase the performance by a few percent in specific cases although it does not support multi-threaded transposition. That is why it is always considered by HyFFT as an alternative to the previous approach.

2. **Local transposition**: The local transposition is based on a multi-threaded, cache-friendly algorithm further accelerated by vector units (see in Fig. 3). The slab is first chopped into square blocks that can fit nicely into L1 or L2 cache. Threads then take pairs of blocks sitting symmetrically over the main diagonal, transpose the data inside and finally swap each other. In the case of square slabs, this can be done in-place. However, rectangular slabs enforce an out-of-place algorithm.

   The block being transposed is further divided into tiles of size 2x2 or 4x4 complex numbers depending on whether the SSE or AVX vector instruction set is available. A fast, vector register based, kernel is used to permute the grid elements within the tile yielding the transposed order. Since we work with single precision floating point numbers only, complex single precision values can be treated as double precision real ones leading in fewer instruction needed. In the case the size of the slab is not divisible by the size of the vector registers (2 or 4 for SSE and AVX, respectively), the reminders are treated separately using scalar kernels.
3. **Global transposition**: The distributed transposition getting the grid points over the last axis (Z) contiguous is done by a composition of two local transpositions and a global one. The FFTW library offers a fine-tuned routine to exchange data amongst the processes that is supposed to be faster than simple `MPI_Alltoall`. Let us note that this operation is performed only by the master thread (a single core per socket or node).

4 Experimental Results

Experiments were performed on 3 different clusters - Zapat[3], Anselm[4] and Fermi[5]. The performance and scaling were investigated on grid sizes ranging from 256^3 to 1024^3 and the core count from 128 to 16384. For the sake of brevity and similarity of plots, we only present the performance for the grid size of 1024^3. Each test consists of running 100 complex-to-complex forward single precision 3D FFTs in a loop to make sure everything settles down properly (branch predictors, etc.). The presented times are normalised per transform. Since P3DFFT does not support complex-to-complex transforms, they were simulated by calculating real-to-complex transforms on both real and imaginary parts of the input. Our code (HyFFT) runs one MPI process per socket and one OpenMP thread per core. Other libraries run one MPI process per core. In case of PFFT and P3DFF, the MPI processes are to be placed in a virtual 2D mesh by MPI routine `MPI_Cart_Create`. We used as squared process meshes as possible to minimise communication overhead since they reached the best performance. Execution times were measured by the `MPI_Wtime` routine. When possible, more accurate `FFTW_EXHAUSTIVE` planning flag was used (Zapat, Anselm). Since the exhaustive planning consumes a significant amount of time for high core counts, we had to roll back to less accurate `FFTW_MEASURE` on Fermi.

4.1 Experimental Supercomputing Clusters

The performance investigation was carried out on machines listed bellow. The first two are based on Intel x86 CPUs connected by a fat tree infiniband while the last machine is based on IBM BlueGene/Q architecture with a 5D torus topology.

1. **Zapat Cluster**
   Hardware configuration: 112 nodes (1792 cores), each node integrates 2 × 8-core Intel E5-2670 at 2.6GHz and 128GB RAM (14.3TB total), 2 × 600GB 15k hard drives, Infiniband 40 Gbit/s interconnection.
   Software configuration: GNU gcc 4.8.1 compiler (-std=c99 -O3), Open MPI 1.6.5, FFTW 3.3.4 (FFTW_EXHAUSTIVE only), PFFT 1.0.7 alpha, P3DFFT 2.6.1.
2. **Anselm Cluster**
   Hardware configuration: 209 nodes (3344 cores), each node integrates 2 × 8-core Intel E5-2665 at 2.4GHz, 64GB RAM (15.1TB total), Infiniband 40 Gbit/s QDR, fully non-blocking fat-tree interconnection.
   Software configuration: GNU gcc 4.8.1 compiler (-std=c99 -O3), Open MPI 1.6.5, FFTW 3.3.4 (FFTW_EXHAUSTIVE only), PFFT 1.0.7 alpha, P3DFFT 2.6.1.

[3] CERIT scientific cloud, CZ, https://www.cerit-sc.cz/en/Hardware/

[4] IT4Innovation Centre of Excellence, CZ, https://docs.it4i.cz/anselm-cluster-documentation

[5] CINECA consortium, IT, http://www.hpc.cineca.it/content/ibm-fermi-user-guide

3. **Fermi Cluster**
   Hardware configuration: IBM-BlueGene/Q, 10,240 nodes (163,840 cores), each node integrates a 16-core IBM PowerA2 at 1.6 GHz, 16GB RAM (163.8TB), 5D torus interconnection.
   Software configuration: GCC 4.4.6 compiler (-std=c99 -O3), FFTW 3.3.2 (FFTW_MEASURE only), PFFT 1.0.7 alpha.

4.2 Strong Scaling Investigation

The most important comparison of the HyFFT and other libraries involves the strong scaling, where the amount of work is fixed and the number of cores progressively increased by a factor of two. In an ideal case, any time the number of cores is doubled the execution time is be halved.

Fig. 4 shows the strong scaling for HyFFT, PFFT and the original FFTW library on the Fermi cluster. The results for P3DFFT has not been obtained yet due to difficulties while compiling the library on the BlueGene machine yet is expected to be very similar to PFFT. The most exciting observation is that both HyFFT and PFFT libraries scales very well even for very high core counts (the maximum number was limited by our allocation). Taking into consideration that each of 16k thread only processes 256KB of data, this is an extremely good result. The second favourable fact is that the curves remains steep without any flattening making us optimistic about further scaling. The average scaling factor is 1.87 while 2.0 would be optimal, with some superlinear drops attributed to cache effects (the slab/pencil is small enough to fit in cache).

The FFTW shows its superiority as long as there are enough slabs to employ all cores (slab decomposition has naturally lower overhead than the pencil one). The HyFFT is about 30% slower and the PFFT about 75% slower than the FFTW for low core counts. The advantages of the hybrid decomposition is clearly visible in this measurement (roughly 20-30% time reduction). The true potential of HyFFT and PFFT emerges when scaling beyond the number of slabs. Spreading the work over 16k cores can accelerate the calculation of 3D FFT over a 1024^3 grid by a factor of 7.8.

The strong scaling obtained on Anselm and Zapat shows the same tendency, thus only the plot for Anselm is presented, see Fig. 5. The first interesting observation is that the performance for all libraries almost matches for low and moderate core counts (up to 1024). Indeed, there is only about 10% difference between the fastest and slowest library. The difference becomes significant when running on 2048 cores where FFTW is not able to scale, the performance of PFFT and P3DFFT is almost identical and the HyFFT outperforms both by a factor of 1.27. The advantage of shared memory is again clearly visible. The average scaling factor reached by HyFFT is 1.9. Unfortunately, it was not possible to run the test on more cores as Anselm does only integrate 3.3k cores.

4.3 Comparison of Different Cluster Architectures

This section mutually compares the performance of the investigated libraries reached across different cluster architecture. Fig. 6 compares the performance

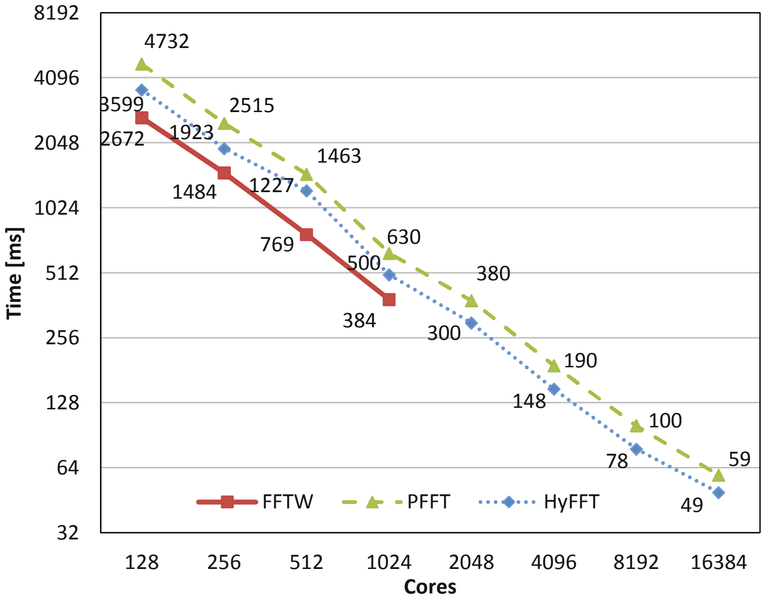


Fig. 4. Strong scaling for the grid size of 1024^3 on the Fermi cluster

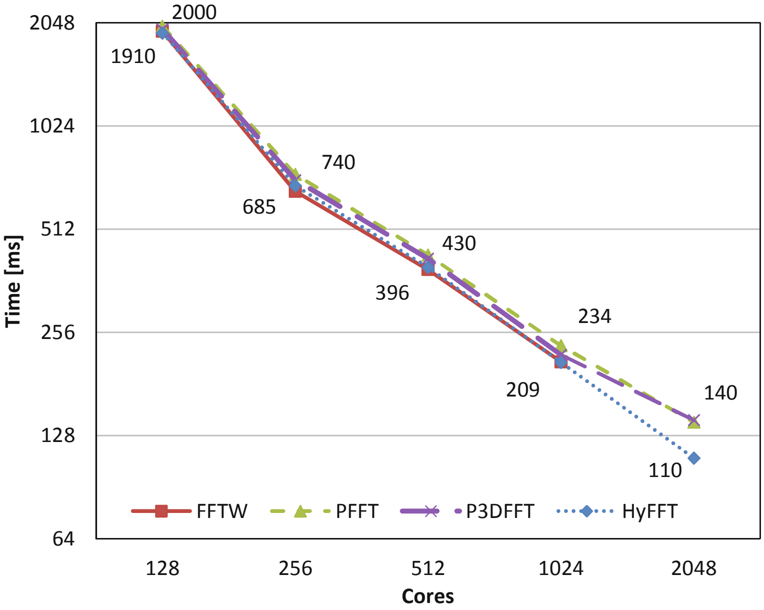


Fig. 5. Strong scaling for the grid size of 1024^3 on the Anselm cluster

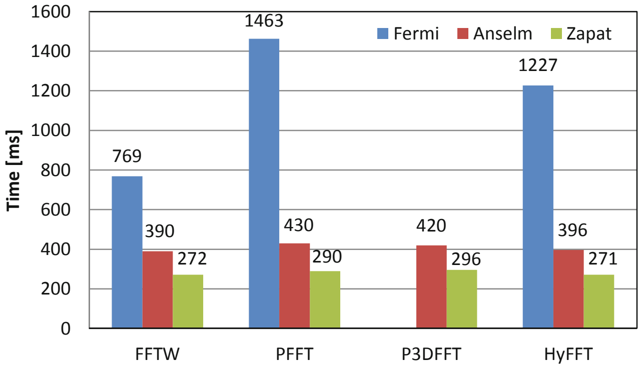


Fig. 6. The execution time of the 3D FFT over a 1024^3 grid distributed over 512 cores

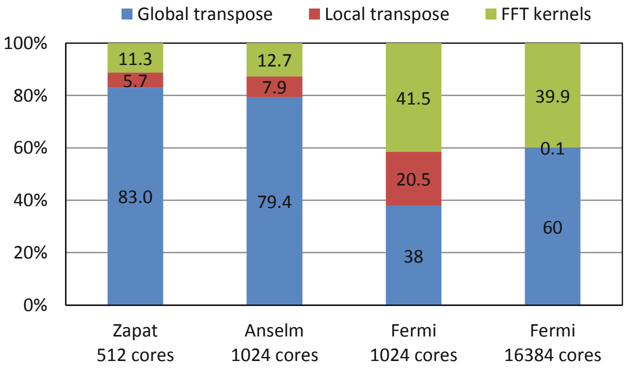


Fig. 7. Time distribution over the main components of HyFFT for a 1024^3 grid

provided by 512 cores because we did not have more cores at our disposal on Zapat.

It can be seen that the x86 based clusters (Anselm, Zapat) provide significantly higher performance than the BlueGene one (Fermi). It is caused by the joined factor of the lower raw performance per core as well as the different interconnection network. Fermi gives about 50% of FLOP/s per core compared to Anselm. The lower performance could also be caused by the less explorative FFTW planning flag. Interestingly, Zapat is approx. 1.4× faster than Anselm. Looking at the specification it is not obvious why there is such a big difference considering the interconnection is the same and the clock speed difference is less than 10%.

4.4 Time Distribution over HyFFT's Components

This section investigates the time distribution over the main components of HyFFT. Fig. 7 shows the time spent on calculating FFTs, local and global transposition for a 1024^3 grid on different clusters with different core counts. The picture demonstrates that the global transposition remains the most time consuming part of the 3D FFT. It's overhead is highest for Zapat, closely followed by Anselm reaching up to 80%. The picture is a bit different for Fermi. For moderate core count, the compute time dominates, however with increasing number of cores, the compute part becomes smaller at the expense of communication. Finally, the time per local transpose seems reasonable.

5 Conclusions

The results have shown that the hybrid OpenMPI/MPI decomposition performs very well on current supercomputers. On Intel x86 clusters, HyFFT provides comparable performance to FFTW on low numbers of cores and outperforms the pure-MPI state-of-the-art libraries PFFT and P3DFFT by 10 to 20% for high core counts. Running HyFFT on a BlueGene machine reveals the true potential of the hybrid decomposition. Although being beaten by FFTW in situation where the 1D decomposition is enough to employ available cores, it further extends FFTW's scalability, reaching 8× higher performance on 16384 cores compared to the maximum number of employable cores (1024) of FFTW using a 1024^3 grid size. HyFFT also helps reduce communication overhead for high core counts leading in better execution times than other pure-MPI libraries.

This has a huge practical impact on many spectral simulations. Speaking about the k-Wave project, deploying the hybrid decomposition has the potential to decrease the simulation time by a factor of 8, bringing the simulation time within the clinically meaningful timespan of 24 hours and allowing patient specific treatment plans to be created.

In the future, we would like to add support for AVX-512 and ALTIVEC extensions to be able to vectorize the code on as many different machines as possible. We also plan to use non-blocking MPI communication to overlap some of the communication with computation. Finally, as the communication step is often dominant, we would like to focus our attention on low-power clusters.

Acknowledgments. The work was financed from the SoMoPro II programme. The research leading to this invention has acquired a financial grant from the People Programme (Marie Curie action) of the Seventh Framework Programme of EU according to the REA Grant Agreement No. 291782. The research is further co-financed by the South-Moravian Region. This work reflects only the authors view and the European Union is not liable for any use that may be made of the information contained therein. This work was also supported by the research project "Architecture of parallel and embedded computer systems", Brno University of Technology, FIT-S-14-2297, 2014-2016.

This work was further supported by the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), funded by the European Regional Development Fund and the national budget of the Czech Republic via the Research and Development for Innovations Operational Programme, as well as Czech Ministry of Education, Youth and Sports via the project Large Research, Development and Innovations Infrastructures (LM2011033). Computational resources were also provided by the MetaCentrum under the program LM2010005 and the CERIT-SC under the program Centre CERIT Scientific Cloud, part of the Operational Program Research and Development for Innovations, Reg. no. CZ.1.05/3.2.00/08.0144. We acknowledge CINECA and PRACE Summer of HPC project for the availability of high performance computing resources.

References

1. Cooley, J., Tukey, J.: An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation, 297–301 (1965)
2. Hesthaven, J.S., Gottlieb, S., Gottlieb, D.: Spectral Methods for Time-Dependent Problems. Cambridge University Press (2007)
3. Treeby, B.E., Jaros, J., Rendell, A.P., Cox, B.T.: Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method. The Journal of the Acoustical Society of America **2012**(131), 4324–4336 (2012)
4. Jaros, J., Rendell, A.P., Treeby, B.E.: Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound. ArXiv e-prints (2014)
5. Frigo, M., Johnson, S.G.: The Design and Implementation of FFTW3. Proceedings of the IEEE **93**(2), 216–231 (2005)
6. Michael, P.: PFFT-An extension of FFTW to massively parallel architectures. Society for Industrial and Applied Mathematics **35**(3), 213–236 (2013)
7. Pekurovsky, D.: P3DFFT: A Framework for Parallel Computations of Fourier Transforms in Three Dimensions. SIAM Journal on Scientific Computing **34**(4), C192–C209 (2012)
8. Gupta, A., Kumar, V.: The scalability of FFT on parallel computers. IEEE Transactions on Parallel and Distributed Systems **4**(8), 922–932 (1993)
9. Foster, I.T., Worley, P.H.: Parallel algorithms for the spectral transform method. SIAM J. Sci. Comput. **18**(3), 806–837 (1997)
10. Sakai, T., Sedukhin, S., Tsuruga, I.: 3D Discrete Transforms with Cubical Data Decomposition on the IBM Blue Gene/Q. The University of AIZU, Fukushima, Japan, Technical report (2013)
11. Rahman, R.: The intel math kernel library and its fast fourier transform routines. Intel Corporation, Technical report (2012)

B.3 Hybrid Global Domain Decomposition

Jaros, J.; Nikl, V.; Treeby, B. E.: Large-scale Ultrasound Simulations Using the Hybrid OpenMP/MPI Decomposition. In *Exascale Applications and Software Conference.* Edinburgh: Association for Computing Machinery. 2015. ISBN 978-3-319-14895-3. pp. 115–119.

Large-scale Ultrasound Simulations Using the Hybrid OpenMP/MPI Decomposition

Jiri Jaros
Faculty of Information Technology
Brno University of Technology
Bozetechova 2
612 66 Brno, CZ
jarosjir@fit.vutbr.cz

Vojtech Nikl
Faculty of Information Technology
Brno University of Technology
Bozetechova 2
612 66 Brno, CZ
inikl@fit.vutbr.cz

Bradley E. Treeby
Dept. of Medical Physics and Biomedical Engineering
University College London
Malet Place Eng Bldg
London WC1E 6BT, UK
b.treeby@ucl.ac.uk

ABSTRACT

The simulation of ultrasound wave propagation through biological tissue has a wide range of practical applications including planning therapeutic ultrasound treatments of various brain disorders such as brain tumours, essential tremor, and Parkinson's disease. The major challenge is to ensure the ultrasound focus is accurately placed at the desired target within the brain because the skull can significantly distort it. Performing accurate ultrasound simulations, however, requires the simulation code to be able to exploit several thousands of processor cores and work with datasets on the order of tens of TB. We have recently developed an efficient full-wave ultrasound model based on the pseudospectral method using pure-MPI with 1D slab domain decomposition that allows simulations to be performed using up to 1024 compute cores. However, the slab decomposition limits the number of compute cores to be less or equal to the size of the longest dimension, which is usually below 1024.

This paper presents an improved implementation that exploits 2D hybrid OpenMP/MPI decomposition. The 3D grid is first decomposed by MPI processes into slabs. The slabs are further partitioned into pencils assigned to threads on demand. This allows 8 to 16 times more compute cores to be employed compared to the pure-MPI code, while also reducing the amount of communication among processes due to the efficient use of shared memory within compute nodes.

The hybrid code was tested on the Anselm Supercomputer (IT4-Innovations, Czech Republic) with up to 2048 compute cores and the SuperMUC supercomputer (LRZ, Germany) with up to 8192 compute cores. The simulation domain sizes ranged from 256^3 to 1024^3 grid points. The experimental results show that the hybrid decomposition can significantly outperform the pure-MPI one for large simulation domains and high core counts, where the efficiency remains slightly below 50%. For a domain size of 1024^3, the hybrid code using 8192 cores enables the simulations to be accelerated by a factor of 4 compared to the pure-MPI code. Deployment of the hybrid code has the potential to eventually bring the simulation times within clinically meaningful timespans, and allow detailed patient specific treatment plans to be created.

Keywords

Ultrasound simulations; 2D domain decomposition; OpenMP/MPI Hybrid programming; Performance evaluation; Supercomputing, k-Wave toolbox.

1. INTRODUCTION

The simulation of ultrasound wave propagation through biological tissue has a wide range of practical applications. Recently, high intensity focused ultrasound has been applied to functional neurosurgery as an alternative, non-invasive treatment of various brain disorders such as brain tumours, essential tremor, and Parkinson's disease. The technique works by sending a focused beam of ultrasound into the tissue, typically using a large transducer. At the focus, the acoustic energy is sufficient to cause cell death in a localised region while the surrounding tissue is left unharmed. The major challenge is to ensure the focus is accurately placed at the desired target within the brain because the skull can significantly distort it.

Performing accurate ultrasound simulations, however, requires the simulation code to be able to operate on large domains and deliver the results in a clinically meaningful time. Apart from the physical complexity, the main obstacle in implementing new ultrasound treatment planning procedures in clinical practice is the computational complexity. Considering the domain of interest encompassing the ultrasound transducer and the treatment area (normally on the order of centimetres in each Cartesian direction), and the size of the acoustic wavelength (on the order of hundreds of micrometers at the maximum frequency of interest), we have to simulate the wave propagation over hundreds or thousands of wavelengths. A sufficiently fine discretisation of the simulation domain which avoids numerical dispersion and instability can easily lead to grid sizes exceeding 10^{12} elements. Storing all the necessary acoustic quantities for such a large simulation domain in computer memory requires petabytes of memory and its processing reaches the order of exascale.

We have recently developed a pure-MPI pseudospectral simulation code using 1D domain decomposition that has allowed us to run reasonable sized simulations using up to 1024 compute cores [3]. However, this implementation suffers from the maximum parallelism being limited by the largest size of the 3D grid used. At the age of exascale, more and more systems will have numbers of processing cores far exceeding this limit. For example, a realistic ultrasound simulation performed by the k-Wave toolbox might use a grid size of 1024^3. Here, the 1D pure-MPI decomposition would only scale up to 1024 cores at most leading to calculation times exceeding clinically acceptable times (in this case between 30 and 72 hours). In contrast, top supercomputer facilities dispose with several hundred thousand compute cores and could provide the simulation result within an hour, if efficiently employed.

The second problem arising from limited parallelism is the total amount of memory that can be used to store simulation data. Not scaling the code to larger core counts holds the simulation domain size below 4096^3, which is not enough for some clinical applications (e.g., the use of shocked waves to vaporise a piece of tissue which can produce hundreds of harmonics).

This paper presents an improved implementation that exploits a 2D hybrid OpenMP/MPI decomposition. The 3D grid is first decomposed by MPI processes into slabs. The slabs are further partitioned into pencils assigned to threads on demand. This is supposed to (i) exploit shared memory within nodes and limit inter-process communication, (ii) employ 8 to 16 times more compute cores, (iii) increase the overall memory capacity while reducing the communication time.

2. DISTRIBUTED IMPLEMENTATION OF ULTRASOUND SIMULATIONS

The k-Wave toolbox [8] is designed to simulate ultrasound wave propagation in soft-tissues and bone, modelled as fluid and elastic media, respectively. In the k-Wave toolbox, the k-space pseudospectral method is used to solve the system of governing equations described in detail by Treeby in [9]. These equations are derived from the mass conservation law, momentum conservation law, and an empirically derived acoustic pressure-density relation that accounts for acoustic nonlinearity, absorption, and heterogeneity in the material properties [9].

The k-space and pseudospectral methods gain their advantage over finite difference methods due to the global nature of the spatial gradient calculations [4]. This permits the use of a much coarser grid for the same level of accuracy. However, the global nature of the gradient calculation, in this case using the 3D fast Fourier transform (FFT), introduces additional challenges for the development of an efficient parallel code. Specifically, the FFT requires a globally synchronising all-to-all data exchange. This global communication can become a significant bottleneck in the execution of spectral models. Fortunately, considerable effort has already been devoted to the development of distributed memory FFT libraries that show reasonable scalability of up to tens of thousands of processing cores [2], [5], [7].

The distributed implementation was written in C++ as an extension to the open-source k-Wave acoustics toolbox [8]. The standard message passing interface (MPI) was used to perform all interprocess communications, the MPI version of the FFTW library was used to perform the Fourier transforms [2], and the input/output (I/O) operations were performed using the HDF5 library [1]. To maximise performance, the code was also written to exploit single instruction multiple data (SIMD) instructions such as SSE or AVX. A detailed description can be found in [3]. The simulation time loop can be broken down into several phases:

1. The gradient of acoustic pressure is calculated by the Fourier collocation spectral method. This operation requires one forward 3D FFT and a few element-wise operations.
2. The acoustic particle velocity (a 3D vector) is calculated based on the acoustic pressure gradient using three inverse 3D FFTs and a few element-wise operations.
3. The gradients of particle velocity for each spatial dimension are calculated using three forward and three inverse 3D FFTs interleaved by several element wise operations.
4. The acoustic density is updated based on the particle velocity gradients using several element-wise operations.
5. The acoustic pressure field is updated based on the particle velocity gradients, acoustic density, and the non-linearity and absorption operators. This step includes two forward and two inverse 3D FFTs, and several elementary element-wise operations such as multiplication, addition, division, etc.
6. The desired acoustic quantities are sampled in regions of interest and either stored on the disk as time-varying series or further processed to calculate e.g. maximum, average, RMS, etc.

There are two important features of the time loop that should be highlighted. First, there are only two places where communication among MPI processes is required. It is within the 3D FFT while performing the distributed matrix transposition, and while the data is being sampled, collected, and stored using the parallel HDF5 library. To reduce the communication burden, pairs of forward and inverse FFTs do not bring the data into the original shape in between, instead a transposed shape is used to reduce the amount of communication to one half [3]. Moreover, the output data is collected and stored using chunks enabling buffering and staging of I/O operations. The second observation is that the simulation time loop is dominated by the FFT calculation. This accounts for nearly 60-80% (the higher number of processes, the higher proportion) of the execution time while the rest of the element-wise operations and the I/O only contribute by 40-20% [3]. Moreover, the FFT itself spends the vast majority of its time waiting for data being transmitted and transposed over the network.

The following subsections describe two different decompositions of the 3D simulation space we have developed: the 1D pure-MPI decomposition and the 2D Hybrid OpenMP/MPI decomposition.

2.1 Pure-MPI Decomposition

The pure-MPI decomposition is based on the 1D slab decomposition natively supported by the FFTW library. In this case, the 3D domain is partitioned along the z axis and every MPI process receives a given number of 2D slabs. In practice, all 3D matrices (acoustic pressure, velocity, density, etc.) are partitioned and distributed this way while several other support data structures are either partitioned and scattered or simply replicated [3]. The communication phase consists of one `MPI_Alltoall` communication performed as a part of the FFT, see Fig 1.

It has to be noted, that this decomposition provides reasonable scaling as long as the number of MPI processes is smaller than the z dimension size of the simulation domain. It also allows easy deployment on many supercomputing systems and eliminates problems with proper thread pinning, memory affinity, and so on. However, the disadvantage, apart from the limited number of processes to be used, is the communication overhead. With a growing number of MPI processes, the messages get smaller and smaller, while the number of messages grows with P^2. This eventually leads to network congestion and bandwidth decrease caused by the high latency of routing small messages.

2.2 Hybrid OpenMP/MPI Decomposition

The hybrid OpenMP/MPI decomposition tries to alleviate the disadvantages of the pure MPI decomposition by introducing a second level of decomposition and further breaking the 1D slabs up into pencils. In contrast to pure-MPI 2D decompositions, the smallest chunk an MPI process can receive still remains a 1D slab. Thus, the total number of MPI processes inherits the same limit as the 1D decomposition presented above. However, in this case, MPI processes are not mapped and bound to all compute cores, but only to one core per socket or node. Once a process is mapped on a socket/node, it spawns several OpenMP threads to process a given number of pencils from the allocated slab/slabs. Considering that

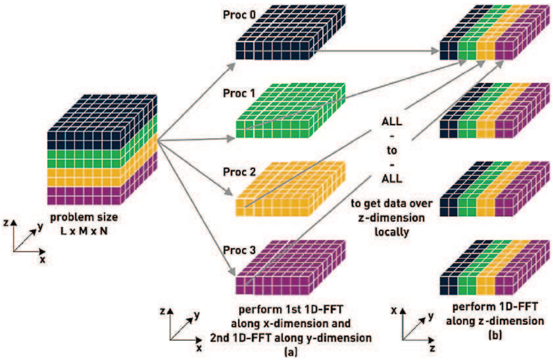


Figure 1: 1D domain decomposition and communication patterns within a 3D FFT.

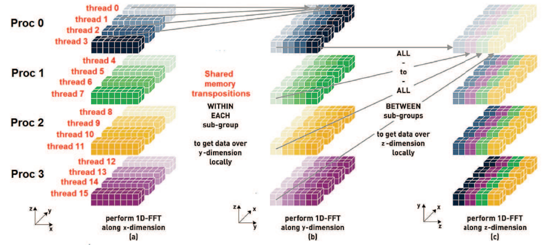


Figure 2: 2D domain decomposition and communication patterns within a 3D FFT.

many current supercomputers comprise of shared memory nodes typically integrating two sockets of 8 cores, we are able to scale the simulation up by a factor of 8 or 16. Moreover, the OpenMP threads can employ shared memory to significantly reduce the amount of inter-process communication and help in exploiting local caches.

It should be noted, that the 2D decomposition requires two communication phases to be carried out (one transpose along the y axis followed by another one along the z axis). Pure-MPI approaches typically implement this by a sequence of `MPI_Alltoall` communication over the y and z axis [7], [5]. Since the whole 1D slab is always placed on one socket/node, the hybrid implementation can efficiently employ the shared memory to perform the first transposition. The second transposition is carried out the same way as the 1D decomposition (see Fig 2), however, with a fewer number of processes (fewer and bigger messages, higher bandwidth, etc.).

The hybrid OpenMP/MPI simulation code was implemented in a very similar way to the pure-MPI one. The FFT calculation is based on the FFTW library tuned to be able to work with the 2D decomposition. We used our custom implementation presented in [6]. In a nutshell, it uses OpenMP FFTW kernels to perform series of 1D FFTs, a multi-threaded local transposition accelerated by SIMD instructions, and a distributed transposition offered by the FFTW library to carry out the communication part. This implementation has proved its superiority over pure-MPI approaches and enables better scaling than the original FFTW library (see [6] for more detail).

The element wise operations implemented in various steps of the simulation time loop were merged into a small number of kernels to maximize the temporal locality, written to utilise SIMD extensions, and run in parallel using the OpenMP library. To ensure correct thread and memory affinity, the First Touch Strategy was used.

3. EXPERIMENTAL RESULTS

The experimental evaluation of the hybrid decomposition was performed on two supercomputing systems, Anselm and SuperMUC. Anselm is a Czech supercomputer operated by the IT4Innovations National Supercomputing Center in Ostrava, Czech Republic. Anselm is an Intel-infiniband cluster based on Sandy Bridge processors (2x8 core Intel E5-2665 at 2.4GHz and 64GB RAM per node) interconnected by a 40Gb Fat-tree infiniband interconnection. The maximum number of cores we could use was 2048.

SuperMUC is a German supercomputer operated by Gauss Centre for Supercomputing and Leibniz Supercomputer Centre in Munich, Germany. SuperMUC is also an Intel-infiniband cluster based on similar Sandy Bridge CPUs (2x8 core Intel Xeon E5-2680 at 2.7 GHz and 32GB RAM per node) interconnected by a 40Gb Fat-tree infiniband network. The maximum number of cores we could use was 8192.

Comparing the hardware configuration, both systems are very similar and should produce very close results. The software stack on the other hand is different and allows us to check different compilers and MPI libraries. On Anselm, we used a GNU software stack comprising of a GNU C++ compiler (g++-4.8), the OpenMPI library in version 1.8.4, FFTW 3.3.3, and HDF5 1.8.13. The schedule manager is based on the OpenPBS software. SuperMUC on the other hand is based on an Intel software stack including an Intel Compiler 2015, Intel MPI in version 5.0, FFTW 3.3.3 and HDF5 1.8.12. The schedule manager is based on LoadLeveler.

3.1 Test configurations

One of the most important issues rising when working with a hybrid OpenMP/MPI code is the proper mapping of MPI processes and threads to cores, sockets and nodes. Improper setting can significantly deteriorate performance by allowing the threads to migrate among cores/sockets and losing the memory affinity. Since the default behaviour of MPI is to bind one process per core, spawning new threads by this process often leads to the threads being bound to the same core. As a consequence, one core is heavily overloaded while others are kept idle. The setting for three test configurations was as follows:

1. **Pure-C** (pure-MPI code, core level mapping) - This configuration uses the pure-MPI code implementing the 1D decomposition compiled without the OpenMP extension. This code is the reference for comparison. No special care has to be taken to run this code.

2. **Hybrid-S** (hybrid code, socket level mapping) - This configuration uses the hybrid OpenMP/MPI code implementing the 2D decomposition compiled with the OpenMP library. The code starts one MPI process per socket and then spawns 8 threads per process. On Anselm, the code was launched with `mpirun -map-by socket -bind-to socket ./executable`, the number of threads was set by environmental variable `OMP_NUM_THREADS=8` pinned by `GOMP_CPU_AFFINITY="0-15"`. On SuperMUC, the LoadLeveler automatically sets all necessary environmental variables when specifying task per nodes equal to 2.

3. **Hybrid-N** (hybrid code, node level mapping) - This configuration uses the hybrid OpenMP/MPI code implementing the 2D decomposition. The code starts one MPI process per node and then spawns 16 threads per process. On

Anselm, the code was launched with `mpirun -map-by node -bind-to none ./executable`, the number of threads was set by `OMP_NUM_THREADS=16` and thread binding by `GOMP_CPU_AFFINITY="0-15"`. On SuperMUC, the LoadLeveler automatically sets all necessary environmental variables when specifying task per nodes equal to 1.

The performance was investigated by a few simulation cases calculating the propagation of nonlinear waves in heterogeneous and absorbing media with a source driven by a sine wave. The domain sizes were chosen to equal 256^3, 512^3, and 1024^3 grid points. We did not test larger domains due to extensive simulation cost and the allocation limits. However, we expect better scaling with large simulation domains. The number of simulation timesteps varied from 100 to 1000 in order to get stable results and run the simulation for a reasonable timespan. The overall simulation run was, however, much longer due to the necessity of FFTW plan creation, which could take up to 30 minutes [3].

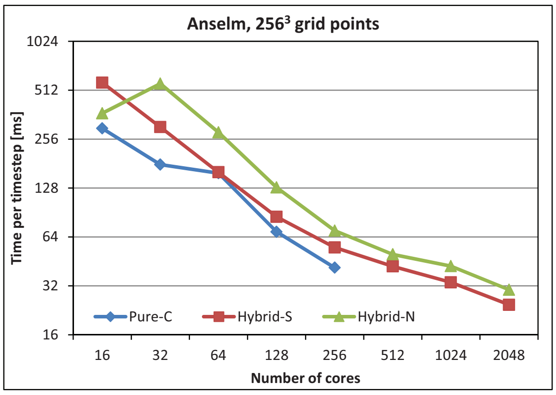


Figure 3: Strong scaling on Anselm, simulation grid of 256^3.

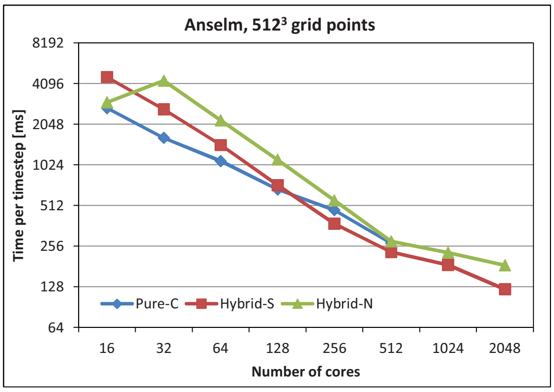


Figure 4: Strong scaling on Anselm, simulation grid of 512^3.

3.2 Strong Scaling

The strong scaling plots describe how the execution time decreases with increasing number of compute resources. The size of the problem is fixed. Fig. 3 and Fig. 4 show strong scaling for simulation domains of 256^3 and 512^3 grid points, respectively, and the number of compute cores growing from 16 (1 node) up to 2048 cores (128 nodes) on the Anselm supercomputer. The curves show the average execution time per one time step of the pure-MPI and two hybrid versions.

It can be seen that the simulation time decreases linearly, slowly reaching a plateau at the end (2048 cores). This is given by the size of the simulation grid, which is simply too small to keep all cores busy; one core only has 8k or 65k grid points to calculate. We can also conclude that the hybrid implementation is not so efficient for small core counts and the Pure-C code beats the hybrid ones almost twice. The clue is hidden in the communication part (the amount of computation is the same in all cases). In the Pure-C code, all cores participate in the communication transposing its part of the grid. However, the hybrid codes only use the master thread to communicate while the others are sleeping. Since the messages are quite big at low core counts, the loss in concurrency affects the performance by a great deal. For the smallest simulation domain size of 256^3, the hybrid decomposition seems to be inefficient. The Hybrid-S code offers a factor of two in performance, however, when using 8 times more resources. The efficiency is thus very low. For a bigger domain of 512^3, the hybrid codes scale much better and catches up with the Pure-C code at 128 cores (Hybrid-S version) or 512 cores (Hybrid-N version). The real strength of the hybrid code becomes evident beyond the scaling capability of the Pure-C code (512 cores). The Hybrid-S configuration offers more than 2.3 times higher performance when running on 2048 cores (efficiency of 57% compares to 512 cores).

The same test was also performed on SuperMUC, see Fig. 5. Since having a much bigger allocation here, we used a grid size of 1024^3 and executed the simulation with core counts ranging from 64 to 8192. Again, the Pure-C code is faster for lower core counts while the hybrid implementations win at the other side of the range. An interesting peak occurs for 2048 cores (Hybrid-S) and 4096 cores (Hybrid-N) where the performance is much lower than expected. This peak was also observed on other grid sizes always at the position where the number of cores is twice as high as the size of z dimension for Hybrid-S version, and four times higher for the Hybrid-N version. When investigating of this phenomenon, we tried different FFTW planning flags (patient and exhaustive), various compiler flags, MPI versions, and pinning strategies, however, we did not succeed in eliminating this behaviour. We suspect that it has something to do with the critical message size where MPI changes the policy of transmitting messages (sync. vs buffered), or that FFTW is unable to find a good communication plan.

To support this hypothesis we took a simulation flat profile, see Table 1. The peaks in execution time directly correspond to the communication share. In a typical run, the communication share is about 50%, while in those exceptional cases the communication share springs up to 75%. The profile confirmed our hypothesis that the distributed transposition is not done optimally and a custom routine needs to be implemented to ensure the correct behaviour. This table also reveals that the hybrid OpenMP/MPI decomposition bounds the communication at a reasonable level of 50%, even for high core counts.

Fortunately, at least one of the hybrid versions works correctly

Table 1: Communication share for various core counts and hybrid implementations on SuperMUC (grid 1024^3).

core count	Hybrid-S (MPI share)	Hybrid-N (MPI share)
1024	51.60%	46.24%
2048	71.48%	48.95%
4096	52.84%	74.38%

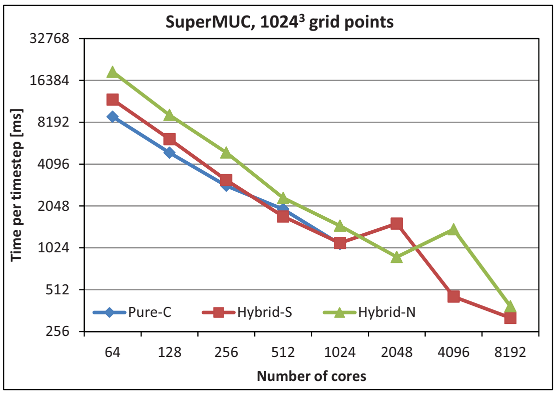


Figure 5: Strong scaling on SuperMUC, simulation grid of 1024^3.

at a given core count and the user has the ultimate choice. Finally, we would like to note that Hybrid-S version offers almost 4 times higher performance over Pure-C, which yields efficacy of almost 50%, which is not so bad considering the code is proven to be communication and memory bound.

4. CONCLUSIONS

This paper has presented our first attempt to improve scaling of large-scale ultrasound simulations using the hybrid OpenMP/MPI decomposition. The main goal was to enable the code to employ a number of compute cores exceeding the limit imposed by the standard 1D decomposition (the size of the z dimension). By introducing a second level of decomposition and breaking the 1D slabs assigned to MPI processes into pencils computed by OpenMP threads, as well as eliminating the need for another inter-process transposition by the shared memory, we have been able to accelerate the simulation by a factor of 4. This was achieved on SuperMUC when using 8192 compute cores to compute ultrasound wave propagation over a simulation domain discretised into 1024^3 grid points. We also managed to keep the communication overhead at an acceptable 50%.

We also observed curious behaviour for some configurations (number of processes and threads) where the simulation time abruptly increased. This may be attributed to the inability of the FFTW to find an optimal communication plan at this configuration. We can also conclude, that the scaling gets better for bigger simulation domains. While for domain sizes of 256^3 grid points, the hybrid decomposition does not bring much improvement due to the small amount of work, large domains of 1024^3 and bigger appear to benefit from the additional compute resources very well.

In our future work, we would like to test the code for bigger grid sizes, introduce custom communication plans, and further optimise the simulation code.

5. ACKNOWLEDGEMENTS

The project is financed from the SoMoPro II programme. The research leading to this invention has acquired a financial grant from the People Programme (Marie Curie action) of the Seventh Framework Programme of EU according to the REA Grant Agreement No. 291782. The research is further co-financed by the South-Moravian Region. This work reflects only the author's view and the European Union is not liable for any use that may be made of the information contained therein. This work was also supported by the research project "Architecture of parallel and embedded computer systems", Brno University of Technology, FIT-S-14-2297, 2014-2016.

This work was supported by the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), funded by the European Regional Development Fund and the national budget of the Czech Republic via the Research and Development for Innovations Operational Programme, as well as Czech Ministry of Education, Youth and Sports via the project Large Research, Development and Innovations Infrastructures (LM2011033).

The authors gratefully acknowledge the assistance of Sebastian Lehrack from the LMU Faculty of Physics, and the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time on the GCS Supercomputer SuperMUC at Leibniz Supercomputing Centre (LRZ, www.lrz.de).

6. REFERENCES

[1] M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson. An Overview of the HDF5 Technology Suite and Its Applications. In *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*, AD '11, pages 36–47, New York, NY, USA, 2011. ACM.

[2] M. Frigo and S. G. Johnson. The Design and Implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005.

[3] J. Jaros, A. P. Rendell, and B. E. Treeby. Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound. *The International Journal of High Performance Computing Applications*, 2015(2):1–19, 2015.

[4] T. D. Mast, L. P. Souriau, D.-L. D. Liu, M. Tabei, A. I. Nachman, and R. C. Waag. A k-space method for large-scale models of wave propagation in tissue. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 48(2):341–354, 2001.

[5] P. Michael. PFFT-An extension of FFTW to massively parallel architectures. *Society for Industrial and Applied Mathematics*, 35(3):213–236, 2013.

[6] V. Nikl and J. Jaros. Parallelisation of the 3D Fast Fourier Transform Using the Hybrid OpenMP/MPI Decomposition. In *Mathematical and Engineering Methods in Computer Science*, LNCS 8934, pages 100–112. Springer International Publishing, 2014.

[7] D. Pekurovsky. P3DFFT: A Framework for Parallel Computations of Fourier Transforms in Three Dimensions. *SIAM Journal on Scientific Computing*, 34(4):C192–C209, Jan. 2012.

[8] B. E. Treeby and B. T. Cox. k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *Journal of Biomedical Optics*, 15(2):021314, 2010.

[9] B. E. Treeby, J. Jaros, A. P. Rendell, and B. T. Cox. Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method. *The Journal of the Acoustical Society of America*, 2012(131):4324–4336, 2012.

B.4 Analysis of Multi-GPU Systems

Jaros, J.; Treeby, B. E.; Rendell, A. P.: Use of multiple GPUs on shared memory multiprocessors for ultrasound propagation simulations. In *Conferences in Research and Practice in Information Technology Series*, vol. 127. Melbourne, Australia: ACS. 2012. ISBN 978-1-921770-08-1. ISSN 1445-1336. pp. 43–52.

Use of Multiple GPUs on Shared Memory Multiprocessors for Ultrasound Propagation Simulations

Jiri Jaros[1], Bradley E. Treeby[2] and Alistair P. Rendell[1]

[1]Research School of Computer Science, College of Engineering and Computer Science
The Australian National University
Canberra, ACT 0200, Australia

jiri.jaros@anu.edu.au, alistair.rendell@anu.edu.au

[2]Research School of Engineering, College of Engineering and Computer Science
The Australian National University
Canberra, ACT 0200, Australia

bradley.treeby@anu.edu.au

Abstract

This paper outlines our effort to migrate a compute intensive application of ultrasound propagation being developed in Matlab to a cluster computer where each node has seven GPUs. Our goal is to perform realistic simulations in hours and minutes instead of weeks and days. In order to reach this goal we investigate architecture characteristics of the target system focusing on the PCI-Express subsystem and new features proposed in CUDA version 4.0, especially simultaneous host to device, device to host and peer-to-peer transfers that the application is going to highly benefit from. We also present the results from a CPU based implementation and discuss future directions to exploit multiple GPUs.

Keywords: Ultrasound simulation, 7-GPU system, CUDA, Matlab, FFT, PCI-Express, bandwidth, multi-core.

1 Introduction

In 1994 Becker and Sterling (1995) proposed the construction of supercomputer systems through the use of off-the-shelf commodity parts and open source software. Over the ensuing year, the so called Beowulf cluster computer systems came to dominate the top 500 list of most powerful systems in the world. The advantages of such systems are many, including ease of creation, administration and monitoring, and full support of many advanced programming techniques and high performance computing libraries such as OpenMPI. Interestingly, however, what was originally a major advantage of these systems, namely price and running costs, is now much less so. This is because for even a small to moderately sized cluster it is necessary to house the system in specially air-conditioned machine rooms.

Recently, developments in Graphics Processing Units (GPUs) have prompted another revolution in high-end computing, equivalent to that of the original Beowulf cluster concept. Although these chips were designed to accelerate rasterisation of graphic primitives such as lines and polygons, their raw computing performance has attracted a lot of researchers to utilize them as acceleration units for special kind of mathematical operations in many scientific applications (Kirk and Hwu 2010). Compared to a CPU, the latest GPUs are about 15 times faster than six-core Intel Xeon processors in single-precision calculations. Stated another way, a cluster with a single GPU per node offers the equivalent performance of a 15 node CPU only cluster. Even more interestingly, the availability of multiple PCI-Express buses even on very low cost commodity computers means that it is possible to construct cluster nodes with multiple GPUs. Under this scenario, a single node with multiple GPUs offers the possibility of replacing fifty or more nodes of a CPU only cluster.

On the other hand, the development tools for debugging and profiling of GPU-based applications are in their infancy. Obtaining the peak performance is very difficult and sometimes impossible for a lot of real-world problems. Moreover, only a few basic GPU libraries such as LAPACK and BLAS have so far been developed, and these are only able to utilize one GPU in a node (CUDA Math Libraries 2011). GPU-based applications are also limited by the GPU architecture and memory model making general-purpose computing much more difficult to implement than a CPU-based application.

The purpose of this paper is to outline our efforts to migrate a compute intensive application for ultrasound simulation being developed in Matlab to a cluster computer where each node has seven GPUs. The utilised numerical methods are very memory efficient compared to conventional finite-difference approaches, and the Matlab implementation already outperforms many of the other codes in the literature (Treeby 2011). However, for large scale simulations, the computation times are still prohibitively long. Our overall goal is to perform realistic simulations in hours or minutes instead of weeks or days. This paper provides an overview of the ultrasound propagation application, the development of an optimised C++ version of the original Matlab code for the CPU that exploits streaming extensions, our attempts to characterise the multi-GPU target system, and a preliminary plan for the GPU code to run on that system.

Section 2 provides background on ultrasound simulation, the simulation method used here, and the time consuming operations. Section 3 introduces the architecture of our 7-GPU Tyan servers that will be used for testing and benchmarking our implementations written in C++ and CUDA. Section 4 gives preliminary results of the first C++ implementation using only CPUs and investigates the bottlenecks. Section 5 focuses on the GPU side of the Tyan servers and measures the basics parameters of them in order to acquire necessary experience and investigate the potential architecture limitations. The last section summarizes open questions and issues that will be dealt with in the future.

2 Ultrasound Propagation Simulations

The simulation of ultrasound propagation through biological tissue has a wide range of practical applications. These include the design of ultrasound probes, the development of image processing techniques, studying how ultrasound beams interact with heterogeneous media, training ultrasonographers to use ultrasound equipment, and treatment planning and dosimetry for therapeutic ultrasound applications. Here, ultrasound simulation can mean either predicting the distribution of pressure and energy produced by an ultrasound probe, or the simulation of diagnostic ultrasound images. The general requirements are that the models correctly describe the different acoustic effects whilst remaining computationally tractable.

In our work, the *k*-space pseudospectral method is used to reduce the number of grid points required per wavelength for accurate simulations (Tabei 2002). The system of governing equations used is described in detail by Treeby (2011). These are derived from general conservation laws, discretised using the *k*-space pseudospectral method, and then implemented in Matlab (Treeby 2010). In order to be able to simulate real-world systems, both huge amounts of memory and computation power are required.

Let us calculate a hypothetical execution time requested for simulating a realistic ultrasound image using Matlab on a dual six-core Intel Xeon processor. The ultrasound image is created by steering the ultrasound beam through the tissue and recording the echoes received from that particular direction. The recorded signal from each direction is called an A-line, and a typical image is constructed from at least 128 of these. This means we need 128 independent simulations with slightly modified input parameters. Using a single computer, these must be computed sequentially. Every simulation is done over the 3D domain with grid sizes starting at 768x768x256 grid points and 3000 time steps. From preliminary experiments performed using the Matlab code, each simulation takes about 27 hours of execution time and consumes about 17 GB of memory. Thus to compute one ultrasound image would require roughly 145 days. The objective of this work is to reduce this time to hours or even minutes by exploiting the parallelism inherent in the algorithm.

2.1 *k*-space Pseudospectral Simulation Method Implemented in Matlab

The Matlab code simulating non-linear ultrasound propagation using the *k*-space pseudospectral method is based on the forward and inverse 3-dimensional fast Fourier transformation (FFT) supported by a few 3D matrix operations such as element-wise multiplication, addition, subtraction, division, and a special `bsxfun` operation. This function replicates a vector in particular dimensions to create a 3D matrix on the fly and then performs a defined element-wise operation with another 3D matrix (such as multiplication denoted by `@times`). Most operations work over the real domain, however, some of them are done over the complex one.

The time step loop in a simplified form is shown in Figure 1. This listing identifies all the necessary mathematical operations and presents all matrices, vectors, and scalar values necessary for computation. For the computation, it is necessary to maintain the complete dataset in main memory. This data set is composed of 14 real matrices, 3 complex matrices, 6 real and 6 complex vectors.

An iteration of the loop represents one time step in the simulation of ultrasound propagation over time. The computation can be divided into a few phases corresponding to the particular code statements:

(1) A 3D FFT is computed on a 3D real matrix representing the acoustic pressure at each point within the computational domain. Despite the fact the matrix `p` is purely real, a 3D complex-to-complex FFT is executed in Matlab.

(2) - (4) New values for the local particle velocities in each Cartesian dimension *x*, *y*, *z* are computed. These velocities describe the local vibrations due to the acoustic waves. The result of `fftn(p)` is element-wise multiplied by a complex matrix `kappa` and then multiplied by a vector expanded into a 3D matrix in the given directions using `bsxfun`. After that, the 3D inverse FFT is computed. As we are only interested in real signals, the complex part of the inverse FFT is neglected. Other element-wise multiplications and subtractions are further applied. Note that the old values of the particle velocities are necessary for determining the new ones.

(5) The particle velocities in the x-direction at particular positions are modified due to the output of the ultrasound probe. (Note, additional source conditions are also possible, only one is shown here for brevity). The matrix `ux_sgx` is transformed to a vector and mask-based element-wise addition is executed.

(6) - (8) The gradient of the local particle velocities in each Cartesian direction is computed. First, the 3D FFT of the particle velocity is computed, then, the result is multiplied by `kappa` and a vector in the complex domain. After that, the inverse 3D FFT is calculated. Only the real part of the FFT is used in the difference matrix.

(9) - (11) The mass conservation equations are used to calculate the `rhox`, `rhoy` and `rhoz` matrices (acoustic density at each point within the computational domain). All operations are done over the real domain on 3D matrices. If an operand is a scalar or a vector, it is expanded to a 3D matrix on the fly.

(12) The new value of pressure matrix is computed here using data from all three dimensions. Two forward and inverse 3D FFTs are necessary for intermediate results. All other operations are done over the real domain.

(13) The pressure matrix is sampled and the samples are stored as the final result.

In summary, at a high level we need to calculate 6 forward and 8 inverse 3D FFTs, and about 50 other element-wise operations, mainly multiplications.

```
% start time step loop
for t_index = 2:Nt

   % compute 3D fft of the acoustic pressure
1  p_k = fftn(p);

   % calculate the local particle velocities in
   % each Cartesian direction
2  ux_sgx = bsxfun(@times, pml_x_sgx,
              bsxfun(@times, pml_x_sgx, ux_sgx)
               - dt./rho0_sgx .* real(ifftn(
                 bsxfun(@times, ddx_k_shift_pos,
                   kappa .* p_k) ))
             );
3  uy_sgy =  bsxfun(@times, pml_y_sgy,
                bsxfun(@times, pml_y_sgy, uy_sgy)
                 - dt./rho0_sgy .* real(ifftn(
                  bsxfun(@times, ddy_k_shift_pos,
                    kappa .* p_k) ))
               );
4  uz_sgz = bsxfun(@times, pml_z_sgz,
                bsxfun(@times, pml_z_sgz, uz_sgz)
                 - dt./rho0_sgz .* real(ifftn(
                  bsxfun(@times, ddz_k_shift_pos,
                    kappa .* p_k) ))
             );

   % add in the transducer source term
5  if transducer_source >= t_index
     ux_sgx(us_index) = ux_sgx(us_index) +
            transducer_input_signal(delay_mask);
     delay_mask = delay_mask + 1;
   end

   % calculate spatial gradient of the particle
   % velocities
6  duxdx = real(ifftn( bsxfun(@times,
     ddx_k_shift_neg, kappa .* fftn(ux_sgx)) ));
7  duydy = real(ifftn( bsxfun(@times,
     ddy_k_shift_neg, kappa .* fftn(uy_sgy)) ));
8  duzdz = real(ifftn( bsxfun(@times,
     ddz_k_shift_neg, kappa .* fftn(uz_sgz)) ));

   % calculate acoustic density rhox, rhoy and
   % rhoz at the next time step using a
   % nonlinear mass conservation equation
9  rhox = bsxfun(@times, pml_x, (rhox -
     dt.*rho0 .* duxdx) ./ (1 + 2*dt.*duxdx));
10 rhoy = bsxfun(@times, pml_y, (rhoy -
     dt.*rho0 .* duydy) ./ (1 + 2*dt.*duydy));
11 rhoz = bsxfun(@times, pml_z, (rhoz -
     dt.*rho0 .* duzdz) ./ (1 + 2*dt.*duzdz));

   % calculate the new pressure field using a
   % nonlinear absorbing equation of state
12 p = c.^2.*( ...
      (rhox + rhoy + rhoz)
      + absorb_tau.*real(ifftn(
         absorb_nabla1 .*
         fftn(rho0.*(duxdx+duydy+duzdz)) ))
      - absorb_eta.*real(ifftn(
          absorb_nabla2 .*
          fftn(rhox + rhoy + rhoz) ))
      + BonA.*(rhox + rhoy + rhoz).^2
           ./(2*rho0)
     );

   % extract and save the required storage data
13 sensor_data(:, t_index)= p(sensor_mask_ind);

end
```

Figure 1: Matlab code for the *k*-space pseudospectral method showing the necessary operations.

3 Architecture of Tyan 7-GPU Servers

This section describes the architecture of the Tyan servers targeted for use in the ultrasound propagation simulations. The Tyan servers are 7-GPU servers based on the Tyan barebones TYAN FT72B7015 (Tyan 2011). The barebones consist of a standard 4U rack case and three independent hot-swap 1kW power supplies.

A schematic of the Tyan 7-GPU server configuration can be seen in Figure 2. The motherboard of the servers offers two LGA 1366 sockets for processors based on the Core i7 architecture in a NUMA configuration. The server is populated with two six-core Intel Xeon X5650 processors offering 12 physical cores in total (24 with HyperThreading technology). As each processor contains three DDR3 memory channels, the server is equipped with six 4GB modules (24 GB RAM). The memory capacity can be expanded up to 144GB using 12 additional memory slots.

Communication among CPUs and attached memories is supported by the Intel QuickPath Interconnection (QPI) with a theoretical bandwidth of 12 GB/s. This interconnection also serves as a bridge between CPUs and two Intel IOH chips that offer various I/O connections including four PCI-Express links.

By themselves, the four PCI-Express x16 links are insufficient to connect 7 GPUs and an Infiniband card at full speed. (We would have needed 128 PCI-E links, but unfortunately, had only 64.) Therefore, intermediate PEX bridges were placed between the IOH chips and other devices to double the number of PCI-E links. One PEX bridge is shared between two GPUs (or a GPU and an Infiniband card). The PEX bridges allocate PCI-Express links to the GPUs based on their actual requirements. If one GPU is idle the other one can use all 16 links.

As the servers are designed as a cutting edge GPGPU platform, the most powerful NVIDIA GTX 580 cards with 512 CUDA cores and 1.5GB of main memory have been used. These cards, based on the Fermi architecture, support the latest NVIDIA CUDA 4.0 developer kit and represent the fastest cards that can currently be acquired.

The operating system and user data are stored on two 500GB hard disks, one of which serves as a system disk and the other one as temporary disk space for users. The servers are interconnected using the Infiniband links and a 48 port QLogic Infiniband switch, and to the internet using one of four Gb Ethernet cards.

The operating system the servers are running is Ubuntu 10.04 LTS server edition. For our implementation we have decided to use standard GNU C++ compiler and the latest CUDA version 4.0. This introduces a lot of new features mainly targeted to multi-GPU systems, such as peer-to-peer communication among GPUs, zero-copy main memory accesses from GPUs, etc. OpenMPI is used to communicate between servers and OpenIB layer to directly access the infiniband network card.

4 CPU-based C++ Implementation

In order to accelerate the execution of the Matlab code, the time critical simulation loop has been re-implemented in C++ while paying attention to the underlying architecture to exploit all available performance. A good CPU implementation will serve as a starting point for a GPU

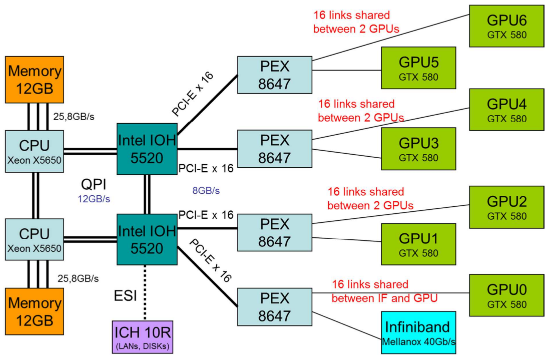


Figure 2: Architecture of 7-GPU server used for the acceleration of ultrasound simulations.

implementation, revealing all the hidden difficulties and ineffectiveness in the Matlab code while also providing ideas on how to improve the Matlab code.

First of all, the import and export of data structures from Matlab to C++ and back has to be designed. Fortunately, all Matlab matrices can be transformed into linear arrays (solving the problem with column-first ordering of multidimensional arrays in Matlab) and saved into separated files using an ASCII or binary format.

All imported matrices as well as six temporary matrices are maintained in main memory during the computation. In the C++ code, the matrices are treated as linear vectors and allocated using the `malloc` function. This organisation simplifies the computation because there is no need to use three indices in element-wise operations. The complex matrices are stored in an interleaved form (even indices correspond to real parts and odd indices the imaginary part of the elements). Another advantage of this data storage format is compatibility with FFTW and CUDA routines when implementing the GPU version.

The C++ code benefits from using an object oriented programming pattern. Each matrix is implemented as a class inheriting basic operations from base classes (real matrix class, complex matrix class) and introducing new methods reflecting the simulation method.

The C++ code does not follow the Matlab code in a verbatim way. Some intermediate results have been precomputed and several temporary matrices have been introduced and reused to save computational effort.

4.1 Complex-to-Complex FFT

Apart from easy to implement element-wise operations, the multidimensional FFT is computed many times in the code. Instead of creating a new implementation, the well-known FFTW library has been employed (FFTW 2011). This library is optimized for a huge number of CPU architectures including multi-core systems with shared memory and clusters with message passing and their streaming extensions such as MMX, SSE, AVX, etc.

A special class encapsulating the FFTW library has been designed in the C++ code. As Matlab uses complex-to-complex 3D FFTs even for real input matrices, the first version of the C++ code also employed the complex-to-complex in-place version of the 3D FFT. First, the input matrix is copied into the FFTW object and transformed into a complex matrix. Then, the forward FFT is computed. As the FFTW class is compatible with other matrix classes, it serves as a temporary storage. Having computed the FFT, a few element-wise operations are performed on this complex matrix, and finally, the inverse FFT is computed. As FFTW does not use normalization, each element has to be divided by the product of the matrix dimension sizes.

4.2 Operation Fusion

The naïve C++ implementation, created at first, encodes each mathematical operation as a separate method parallelized using OpenMP directives. It allows us to understand the algorithm and validate the code. On the other hand, this implementation is extremely ineffective. It is caused by a very poor calculation to memory access ratio while processing very large matrices in the order of hundreds of MBs, and high thread management overhead.

The operation fusion reduces the memory accesses by performing multiple mathematical operations on corresponding elements at once and saving the temporary results in cache memories. As a result, memory bandwidth is saved enabling better scalability at the expense of more complicated code.

4.3 Real-to-Complex FFT

As all the forward FFTs take only real 3D matrices as an input, the results of the forward FFTs are symmetrical. Analogously, as we are only interested in real signals, the imaginary parts of the inverse FFTs are of no use.

Substituting complex-to-complex FFTs with real-to-complex ones saves nearly 50% of the memory and computation time related to FFTs. Moreover, as other operations and matrices are applied to the result of the FFT, we save additional computation effort and memory because of not having to store the symmetrical parts of auxiliary matrices such as `kappa`.

4.4 SSE Optimization and NUMA Support

The final version of the C++ code benefits from a careful optimization of all element-wise operations in order to utilize streaming extensions such as SSE and AVX. Some of the routines were revised so that the C++ compiler could utilize automatic vectorization to produce a highly optimized code. In the cases it was not possible to do so, the compiler intrinsic functions had to be used for rewriting the particular routines from scratch.

Finally, as the Tyan servers are based on the Non-Uniform Memory Access (NUMA) architecture, some policies preventing threads and memory blocks to migrate among cores and local memories have been incorporated into the code. First, all the threads are locked on CPU cores using an OS affinity property. Secondly, the shared memory blocks for all the matrices are allocated by the master thread and immediately initialized and distributed into local memories using a parallel first touch policy (Terboven, C., Mey, D., et.al. 2008). As the access pattern remains unchanged for element-wise routines, the static OpenMP scheduling guarantees all the matrices remain in the local memories. The only exception is the FFT computation, fortunately handed by FFTW library.

4.5 Execution Time Comparison

This section presents the first results of the C++ implementation and compares the execution time with the Matlab version on a dual Intel Xeon system with 12 physical cores and 24GB RAM memory.

Figure 3 shows the relative speed-ups of four different C++ implementations against Matlab and their dependency on the number of CPU threads. All the C++ versions utilize the FFTW library compiled with OpenMP and SSE extensions under single precision. Matlab could use all CPU cores (12) and worked also with single precision in all cases. It can also be noted the server is equipped with the Intel Turbo technology raising the core frequency up to 3.2GHz under one thread workload and decreasing the frequency to 2.66GHz under full 12 thread load.

The C2C, naïve implementation represents the simplest implementation of the problem. Although very simple, it is able to outperform Matlab by about 26%. Operation fusion brings an additional significant improvement. Utilizing all 12 cores, the results are produced in 2.7 times shorter execution time. Replacing Complex-to-Complex (C2C) FFTs with the Real-to-Complex (R2C) ones and reducing some matrices sizes led to an additional reduction in execution time. This version of C++ code is up to 5.2 times faster than Matlab. Finally, revising all element-wise operations to exploit vector extensions of the CPUs and implementing basic NUMA policy, we reached speed-ups of 8.4 times.

Analysing and profiling the C++ code, we learn that nearly 58% of execution time is consumed by FFTs (see Table 1). The other operations take only a fraction of the time. Unfortunately, they cannot be optimized as one, because of intermediate FFTs.

For larger problems, the memory requirements of the complex-to-complex C++ and Matlab codes are very close. The reduction of memory requirements in the real-to-complex version is about 20% considering that most of matrices remained unchanged.

A real-word example has also been examined. The domain size was set to 768x768x256 grid points and 3000 time steps simulated. Matlab needed 27 hours and 11 minutes to compute the result and consumed about 17GB of RAM memory. C2C version with operation fusion took 8 hours and 16 minutes to complete the task and 16.8GB of RAM memory. R2C version finished after 4 hours and 55 minutes using 13.3GB of RAM. The final version of the code reduced the execution time to 3 hours and 22 minutes. Recalling our hypothetical simulation example mentioned earlier, this would decrease the computational time from 145 days to 17 days.

Another important observation is the execution time necessary to perform an iteration of the loop. Assuming the real-world simulation space size of 768x768x256, and 3000 time steps, every iteration takes about 4.1s. As it is not possible to execute multiple iterations at a time, this is the granularity of parallelisation. Moreover, during this time the entire 13GBs of memory will be touched at least once.

Naturally, the outputs from the C++ version and Matlab version have been cross-validated with relative error lower than 10^{-6} for the domain sizes up to 256^3, and 10^{-4} for domain sizes up to 768x768x256 grid points.

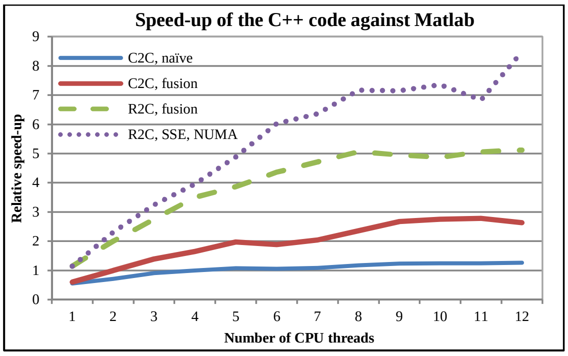


Figure 3: Relative speed-up of C++ against Matlab using a domain size of 256^3 and 1000 time steps.

% of time	Routine
30.84	Inverse FFT
26.73	Forward FFT
3.61	BonA.*(rhox + rhoy + rhoz).^2./(2*rho0)
3.46	Sum_subterms_on_line_12
3.10	rho0.*(duxdx+duydy+duzdz)
2.81	rhox + rhoy + rhoz
2.67	Compute_rhox
2.52	Compute_rhoy
2.42	Compute_rhoz
2.30	Compute_uy_sgy
2.16	Compute_uy_sgx
2.02	Compute_uy_sgz
15.36	Other operations

Table 1: Execution time composition of the C++ code.

5 Towards the Utilization of Multiple GPUs

In order to be able to solve real-world ultrasound propagation simulations in reasonable time, we need to reduce the execution time by an order of magnitude at least. For this reason we would like to utilize up to 7 GPUs placed in the Tyan server to provide the necessary computational power as well as very high memory bandwidth.

First, we would like to start with one GPU and create a CUDA implementation of the simulation code. The most time consuming operations are the fast Fourier transformations. On the CUDA platform, the cuFFT library can be used. This library is provided directly by NVIDIA and runs on a single GPU (CUDA Math Libraries 2011). All other element-wise operations can be directly implemented as simple kernels, as the element-wise operations are embarrassingly parallel. On the other hand, these operations cannot benefit from the on-chip shared memory exploitation of which is often the key factor in reaching peak performance (Sanders and Kandrot 2010). This limitation can be partially alleviated by employing CUDA texture memory and its automatic caching.

There are a few strategies how to split the work among multiple GPUs. The obvious way is to calculate each dimension independently on a different GPU with the final pressure calculation performed on a single one. Looking at the listing shown in Figure 1, we can notice that nearly entire loop can be dimensionally decomposed. That is within each time step, the calculations for the x, y and z dimensions can be done independently. The only exception is line 12, where all three dimensions are necessary to compute the new pressure matrix. This could potentially utilize 3 GPUs for dimension independent calculations while only a single GPU for the final calculation.

Another strategy is to divide the computation of each operation among multiple GPUs. There is another reason to go this way. Utilizing only a single GPU or dimension partition scheme we are strictly limited by the GPU onboard main memory size, which is 1.5GB per GPU in our situation. This value is pretty small compared with 24GB of server main memory and does not allow us to treat larger simulation spaces. If we cut the loop into the smallest meaningful operations we would need two source 3D matrices and a destination one to reside in onboard GPU memory. This would allow us to solve problems with dimensions sizes up to 512^3 grid points in single precision. Our hypothetical example would be intractable because total memory required would be 1.7GB.

Dividing element-wise operations among multiple GPUs is straightforward. We can employ a farmer-workers strategy where a farmer (CPU) divides chunks of work to do. We can imagine a chunk as several rows of multiple 3D matrices that are necessary to compute several rows of a temporary result.

Currently, cuFFT does not run over multiple GPUs. Fortunately, the 3D FFT can be decomposed into a series of 1D FFTs calculated in the x, y and z dimensions and interleaved by matrix transpositions. Considering this, one possible scenario is that the CPU distributes batches of 1D FFTs over all 7 GPUs to compute the 1D FFT in the x dimension. Then a data migration is performed via CPU main memory or using the newly introduced CUDA peer-to-peer transfers followed by calculation of 1D FFTs in the y dimension etc. (An alternative strategy would be to use 2D FFTs on each GPU, with a transpose at the end of the 2D FFTs.)

As in many other distributed schemes, the overall performance will be highly limited by memory traffic, and in this case, also by the PCI-Express bandwidth. We must not forget that we will need to force tens of GBs through the PCI-Express which has a theoretical peak bandwidth of 8GB/s.

In order to gain necessary experience with our Tyan servers with 7-GPUs, we have designed several benchmarks to verify the key parameters of the servers such as PCI-Express bandwidth, zero-copy memory scheme, and peer-to-peer transfers among multiple GPUs. All these operations are going to be utilized in our future ultrasound code.

5.1 Peak PCI-Express Bandwidth with Respect to CPU Memory Allocation Type.

Having a good knowledge about PCI-Express characteristics, behaviour and performance is a key issue when designing and implementing GPGPU applications. As all data processed on the GPU (device) has to be transported from CPU (host) memory to device memory and the results back to the host memory to interpret on the CPU, PCI-Express can easily become a bottleneck debasing any acceleration gained using this massively parallel hardware. Considering the peak CPU-host memory bandwidth is 25GB/s and the peak GPU-device memory bandwidth is 160GB/s, the theoretical throughput of PCI-Express x16 of 8GB/s is likely to be a place of congestion.

Any data structure (3D matrix or 1D vector in our case) designated for host-device data exchange has to be allocated on the host and device separately. Allocating memory on the device (GPU) side is easy as there is only one CUDA routine for this purpose. On the other hand, we need to distinguish between three different types of host memory allocation each intended for a different purpose:

- C/C++ memory allocation routines
- Pinned memory allocation with a CUDA routine
- Zero-copy memory allocation with a CUDA routine

C/C++ memory allocation routines such as `malloc` or `new` serve well for simple CUDA (GPGPU) applications. Their advantages are compatibility with non-CUDA applications and simple porting of C/C++ code onto the CUDA platform. However, using C/C++ memory allocation leads to PCI-Express throughput degradation caused by a temporary buffer for DMA introducing a redundant data movement in host memory. Moreover, only synchronous data transfers can be employed preventing communication-computation overlapping and sharing of host structures by multiple GPU and CPU cores.

A pinned memory allocation routine provided by CUDA marks an allocated region in host memory as non-pageable. This region is thus permanently presented in host memory and cannot be swapped onto disk. This enables Direct Memory Access (DMA) to this buffer, preventing any redundant data movement and allowing the buffer to be shared between multiple CPU cores and GPUs.

Zero-copy memory is a special kind of host memory that can be directly accessed by a GPU. No GPU memory allocations and explicit data transfers are needed any more. Data is streamed from host memory on demand. This is useful for GPU applications only reading input data or writing results once. However, this kind of memory allocation is extremely unsuitable for iteration-based kernels. It is important to note this has an impact on the ability of the CPU to cache this data and thus repeated accesses to the same data locations tend to be very slow. A possible scenario is that a CPU thread fills an input data structure for a GPU and never touches it again; the GPU reads it only once using zero-copy memory allowing a good level of computation and communication overlap.

Figure 4 shows the influence of host memory allocation type on the execution time needed to compute an element wise multiplication of 128M elements (512^3). First, three matrices are allocated on the host using a particular allocation type. After that, the matrices are uploaded into device memory (not in the case of zero-copy). Now, an element-wise multiplication kernel is run. The result is written into device memory and then transferred to host memory (not in case of zero-copy). The figure clearly shows the overhead of standard C memory allocation routines over the CUDA ones.

Zero-copy memory seems to be very suitable for our purposes. Although, the *k*-space method is iterative by nature, we are limited by the device memory size that does not allow us to store all global data (13GB) in

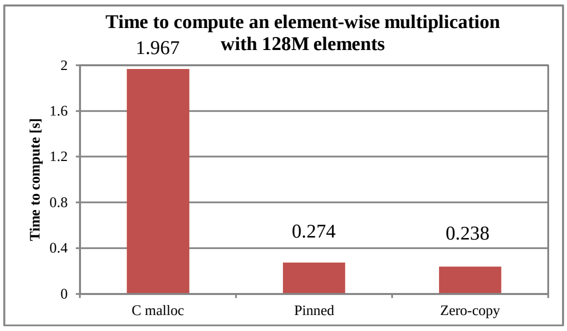


Figure 4: Time necessary to transfer two vectors of 128M elements to the GPU, perform element-wise multiplications, and transfer the resulting vector back to the CPU.

device memory even if we distribute the data over all 7 GPUs. Instead, we can leave some constant matrices in host memory and stream them to particular GPUs on demand. This perfectly suits line no. 6 in Figure 1, see below:

```
duxdx = real(ifftn( bsxfun(@times
   ddx_k_shift_neg, kappa .* fftn(ux_sgx))));
```

First, the 3D FFT of the matrix ux_sgx is calculated using a distributed version of cuFFT. The result is left in the device memory. Now we need to multiply the result of the forward FFT by the matrix kappa. As we need any element of kappa exactly once, there is no benefit in transferring the kappa matrix to the GPU. Instead, we could stream it from host memory using zero-copy memory. After that, we upload ddx_k_shift_neg vector into texture memory, because each element is read many times while expanding it to a 3D matrix on the fly and multiplying with the temporary result of the previous operation. Finally, the inverse 3D FFT is started using the data placed in the device memory.

5.2 Peak Single PCI-Express Transfer Bandwidth

Having chosen an appropriate memory allocation on the host side, we focused on measuring PCI-Express bandwidth between CPU and GPU taking into account different data block sizes starting at 1KB and finishing at 65MB. As the CPU has to serve multiple GPUs simultaneously, it is crucial to know the speed at which the CPU could feed the GPUs.

As the Tyan servers are special-purpose servers with a unique architecture using two IOH north bridges and PEX bridges, the peak bandwidth between the host and single devices was investigated in order to verify the throughput of different PCI-Express slots in both directions.

The experimental results are summarized in Figure 5. The measurements were repeated 100 times and averaged values were plotted. It can be seen that for small data blocks the PCI-Express bandwidth is degraded and reaches only a small fraction of the theoretical value of 8GB/s in one direction. In order to utilize the full potential of PCI-Express, data blocks with sizes of 500KB and larger have to be transferred. The smallest chunk of data we can possibly upload to the GPU is one row of a 3D matrix, which for the size of interest represents 3KB. This is obviously too fine-grained a decomposition and we will have to send hundreds of lines in one PCI-Express transaction. This does not pose a problem, because a typical number of rows to process is in order of hundreds of thousands. The figure also reveals that device to host transfers are slightly faster than host to device ones.

A surprising variation in the peak bandwidth when communicating with different devices was observed. On one Tyan server, the first three GPUs are 2GB/s slower than the other four when transferring data from GPU to CPU memory. Although there are small oscillations from experimental run to experimental run, the results did not change significantly. We tried to physically shuffle the GPUs between slots but the results remained virtually unchanged. One explanation is that first three GPUs are connected to the Intel IOH chipset that is also responsible for HDDs, LANs, VGA, etc. On the other hand this does not explain the situation on the second Tyan server where GPUs 3, 4, 5 and 6 are significantly slower. Considering that both motherboards are the same, other peripheries should be connected to the same Intel IOH chipset. These results have also been cross validated with well-known SHOC benchmark proposed by Danalis at al. (2010).

5.3 Peak PCI-Express Bandwidth under Multiple Simultaneous Transfers

The second set of benchmarks investigates the PCI-Express bandwidth when communicating with multiple GPUs that is essential for work distribution over multiple GPUs. In all instances, pinned memory was used and four different transfer controlling (farmer) patterns considered:

- A single CPU thread distributes the data over multiple devices using synchronous transfer.
- Multiple CPU threads distribute the data over multiple devices using synchronous transfers. Each device is served by a private CPU thread.
- A single CPU thread distributes the data over multiple devices employing asynchronous transfers.
- Multiple CPU threads distribute the data over multiple devices by asynchronous transfers.

As each pair of GPUs share 16 PCI-Express links via a PEX bridge and different pairs are connected to different chipsets with NUMA architecture, we have investigated communication throughput in these configurations:

(1) A pair of devices communicating with the host.
(2) Two devices belonging to different pairs communicating with the host.
(3) All even devices communicating with the host.
(4) Two pairs of devices communicating with the host.
(5) All seven devices communicating with the host.

The experimental measurements shown in Figure 6 demonstrate that a single CPU thread with synchronous transfers cannot saturate the PCI-Express subsystem of the Tyan servers; the peak bandwidth always freezes at the level of a single transfer. On the other hand, all remaining approaches are comparable, so there is no need to use multiple CPU threads to feed multiple GPU with data. We can employ the remaining CPU cores to work on tasks that are not worth processing on a GPU.

The second observation that can be made reveals the difference between the host to device and the device to host peak bandwidth. Whereas device to host transfers are limited by the 5.8GB/s, transfers managed by host scale up to 10.2GB/s (see Figure 6). We can conclude the device to host transfers are limited by the throughput of a single PCI-Express 16 cannel while host to device by the QPI interconnection.

Table 2 presents the peak bandwidth in different configurations using one CPU thread and asynchronous transfers with respect to the numbering above. In all cases, device to host transfers cannot exploit the potential of the underlying architecture. In case (4), two different values were observed depending on the location of the pair. As we have mentioned before, the first three PCI-Express slots are slower than the other four. This leads to the fact that the first two pairs are slower than the other ones. The upper limit for host to device transfers lies around the 10GB/s level possibly limited by the QPI interconnection.

5.4 Peak Peer-to-Peer Transfers Bandwidth

One of the new features introduced in CUDA 4.0 is a peer-to-peer transfer. This feature enables Fermi based GPUs to directly access memory of another device via PCI-Express bypassing host memory. Data can be remotely read, written or copied. As peer-to-peer (p2p) transfers could serve the data exchange phase of distributed FFTs, we have investigated the performance of this technique and compared the results with user implemented device-host-device (d-h-d) transfers.

The Fermi GPU cards are only equipped with one copy engine, this device cannot act as source and destination of a peer-to-peer (p2p) transfer at the same time. Nevertheless, having seven GPUs we can create several scenarios where multiple devices are performing p2p transfers simultaneously. Also, we can use synchronous and asynchronous p2p transfers.

Figure 7 shows a comparison of p2p and d-h-d transfers running on two devices in different pairs, namely GPU 0 and GPU 1. We can see that the new p2p technique brings a significant improvement over the d-h-d transfer where the data has to be downloaded from the source device and, after that, uploaded on the destination device. The situation rapidly changes when performing multiple p2p transfers. The synchronous transfers become a bottleneck and asynchronous ones exploit more bandwidth. Figure 8 presents the performance of three simultaneous pairwise transfers (GPU 1 -> GPU 2, GPU 3 -> GPU 4, and GPU 5 -> GPU 6) where each device is either source or destination and all sources and destination are connected to different PEXs.

A d-h-d transfer in its asynchronous form consists of two phases. In the first phase, data packages are downloaded from all source devices and placed in host memory in asynchronous way. After synchronization the data packages are distributed over destination devices also in asynchronous way (more transfers at a time).

From the figure, CUDA 4.0 does not seem to be optimized for multiple simultaneous p2p transfers and user managed device-host-device transfers win. The difference is about 800MB/s. Taking into account this finding, it appears that it is better to implement highly optimized device-host-device transfers that also involve CPU cores in data rearrangement and migration.

Pattern	Host to Device	Device to Host
(1)	6GB/s	6.5GB/s
(2)	10GB/s	6.8GB/s
(3)	10GB/s	5.2GB/s
(4)	10GB/s	5.2 / 6.8GBs
(5)	10GB/s	5.4GB/s

Table 2: Peak bandwidth of multiple simultaneous transfers in different configurations.

6 Discussion and Conclusion

This paper outlines our effort to migrate a compute intensive application of ultrasound propagation simulation to a cluster computer where each node has seven NVIDIA GPUs. The preliminary results from the CPU implementations have shown a speed-up of up to 8.4 compared to the original Matlab implementation. Given the computational benefits of using the *k*-space method compared to other approaches, this is a significant step towards creating an efficient model for large scale ultrasound simulation.

As the architecture of the Tyan 7-GPU server is not very common, we have examined a number of its specifications. We have designed several benchmarks that have revealed the behaviour of the PCI-Express subsystem.

In order to achieve the highest possible performance, we have to distribute the work over all seven GPUs. The CPU implementation of the code has revealed a low computation-memory access ratio. The asymptotic time complexity is only $O(n) = n \log n$. From the realistic experiments we found the CPU time for a single iteration is about 4.1s while global data of almost 13GBs has to be touched at least once.

Considering we could rework the code to access any element exactly once, and taking into account reachable CPU-GPU bandwidth, a naïve GPU based implementation would spend 2.1s or 1.3s distributing the data over one or multiple GPUs, respectively. Assuming all communication can be overlapped by computation using zero copy memory and the presence only one copy engine on a GPU, the realistic speed-up of a naïve implementation over a CPU one would be limited by 1.5 or 3.2 for one or multiple GPUs respectively.

On the other hand, if we accommodated all data in the on-board GPU memory we could reach much higher speed-up. Such an experiment has been carried out using a Matlab CUDA extension and a single NVIDIA Tesla GPU with 6 GB of memory and 448 CUDA cores. Using a domain size of 256^3 we have reached a speed-up of about 8.5 (compared to Matlab code), which is close to our CPU C++ implementation. Assuming we can optimize the GPU implementation in a similar way as in the CPU case, we may be able to improve on the Matlab CUDA code significantly.

The appropriate data distribution is going to play a key role in the application design. One way to reduce the data set is to calculate some matrices on the fly, exchanging spatial complexity for time complexity. Another possibility is to employ fast real-time compression and decompression of the data making the chunks smaller to transfer through PCI-Express and between GPU on-board and on-chip memory. As many of the matrices are constant, the compression would have to be done only once. As long as we know that using asynchronous transfer one CPU core is sufficient to feed all seven GPUs, the remaining cores could execute other tasks that are not worth migrating to GPUs.

Data migration between GPUs will play another key role. Provided that we also need to perform data migration as a part of distributed FFT, we have revealed that the present CUDA 4.0 is not optimized for multiple simultaneous peer-to-peer transfers bypassing the host memory and thus, this communication pattern will have to be implemented as a composition of common device to host and host to device transfers.

7 Acknowledgments

This work was supported by the Australian Research Council/Microsoft Linkage Project LP100100588.

8 References

Becker, D., Sterling, T., et al. (1995): Beowulf: A Parallel Workstation for Scientific Computation, *Proc. International Conference on Parallel Processing*, Oconomowoc, Wisconsin, 11-14.

Kirk, D., and Hwu, W. (2010): *Programming Massively Parallel Processors: A Hands-on Approach*, Morgan Kaufmann.

Danalis, A., Marin, G., McCurdy, C., Meredith, J., Roth, P., Spafford, K., Tipparaju, V., Vetter, J (2010). The Scalable HeterOgeneous Computing (SHOC) Benchmark Suite. *Proceedings of the Third Workshop on General-Purpose Computation on Graphics Processors (GPGPU 2010).*

Sanders, J. and Kandrot E. (2010): *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley Professional.

Treeby, B. E. and Cox, B. T. (2010): k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *Journal of Biomedical Optics*. **15**(2):021214.

Treeby, B. E., Tumen, M. and Cox, B. T. (2011): Time Domain Simulation of Harmonic Ultrasound Images and Beam Patternsin 3D using the k-space Pseudospectral Method. *Medical Image Computing and Computer-Assisted Intervention*, **6891**(1): 369-376, Springer, Heidelberg.

Tabei M., Mast T. D. and Waag, R. C. (2002): A k-space method for coupled first-order acoustic propagation equations. *Journal of Acoustical Society of America*. **111**(1):53-63.

Terboven, C., Mey, D., et.al. (2008): Data and Thread Affinity in OpenMP Programs. *Proceedings of the 2008 workshop on Memory access on future processors (MAW '08)*, New York, NY, ACM, 377–384.

CUDA: Parallel computing architecture, NVIDI http://www.nvidia.com/object/cuda_home_new.html, Accessed 15 Sep 2011.

CUDA Math Libraries Performance 6.14, NVIDIA, http://developer.nvidia.com/content/cuda-40-library-performance-overview, Accessed 15 Sep 2011

FFTW: Free FFT library, http://www.fftw.org/. Accessed 13 Sep 2011.

Matlab: The Language of technical computing, MathWorks, http://www.mathworks.com.au/products/matlab/index.html, Accessed 15 Sep 2011.

OpenMPI: Open Source High Performance Computing, The Open MPI project, http://www.open-mpi.org/, Accessed 15 Sep 2011.

TYAN Computer: Tyan FT72B7015 server barebone, http://www.tyan.com/product_SKU_spec.aspx?ProductType=BB&pid=439&SKU=600000195, Accessed 15 Sep 2011.

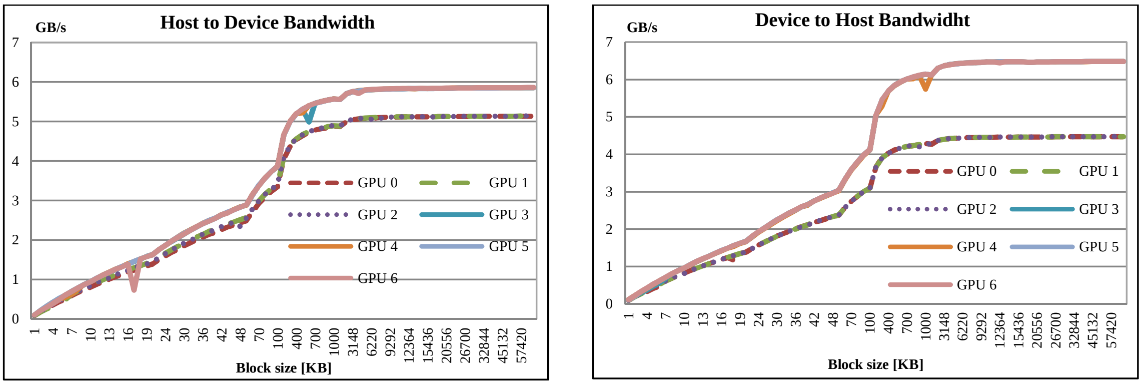


Figure 5: Peak bandwidth between host and a single device in both directions influenced by transported block size.

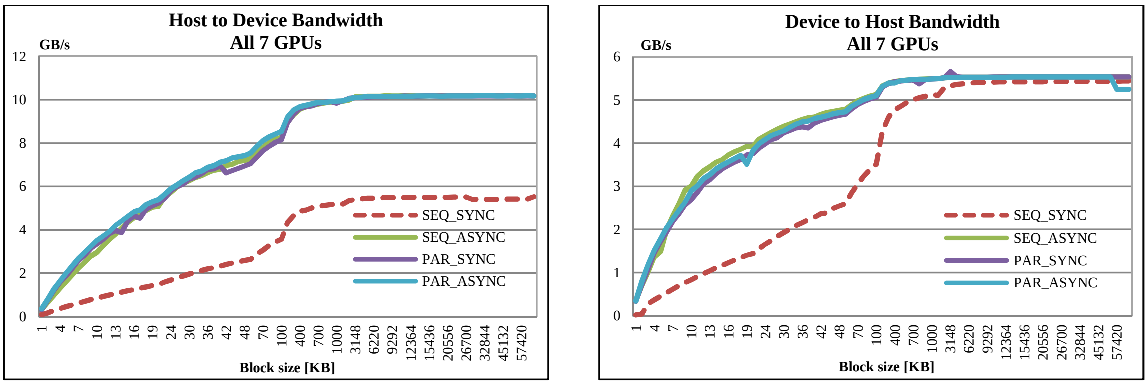


Figure 6: Peak bandwidth when host is communicating with all 7 GPUs in both directions.

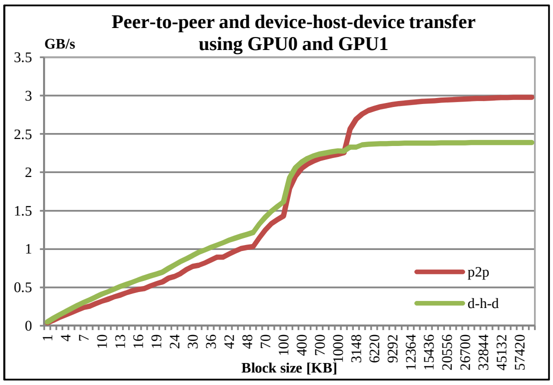


Figure 7: Peak bandwidth of a single peer-to-peer transfer and device-host-device transfer.

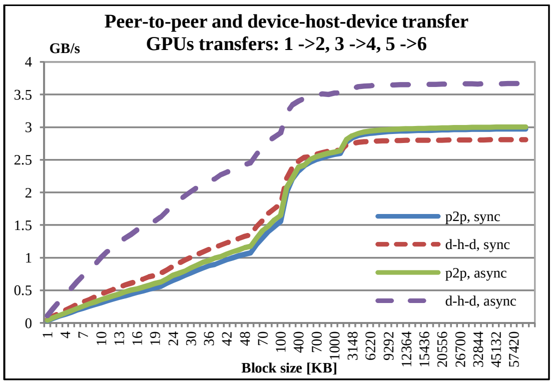


Figure 8: Peak bandwidth of multiple p2p and d-h-d transfers using three disjoint source-destination pairs.

B.5 Implementation of 3D FFT across Multiple GPUs

Nandapalan, N.; **Jaros, J.**; Treeby, B. E.; Rendell, A. P.: Implementation of 3D FFTs across multiple GPUs in shared memory environments. In *Parallel and Distributed Computing, Applications and Technologies, PDCAT Proceedings*. Beijing: IEEE. 2012. ISBN 978-0-7695-4879-1. pp. 167–172. doi:10.1109/PDCAT.2012.79.

Implementation of 3D FFTs Across Multiple GPUs in Shared Memory Environments

Nimalan Nandapalan, Jiri Jaros, and Alistair P Rendell
Research School of Computer Science
ANU College of Engineering and Computer Science
Australian National University, ACT 0200, AUSTRALIA
{*Nimalan.Nandapalan, Jiri.Jaros, Alistair.Rendell*}*@anu.edu.au*

Bradley Treeby
Research School of Engineering
ANU College of Engineering and Computer Science
Australian National University, ACT 0200, AUSTRALIA
Bradley.Treeby@anu.edu.au

***Abstract*—In this paper, a novel implementation of the distributed 3D Fast Fourier Transform (FFT) on a multi-GPU platform using CUDA is presented. The 3D FFT is the core of many simulation methods, thus its fast calculation is critical. The main bottleneck of the distributed 3D FFT is the global data exchange which must be performed. The latest version of CUDA introduces direct GPU-to-GPU transfers using a Unified Virtual Address space (UVA) that provides new possibilities for optimising the communication part of the FFT. Here, we propose different implementations of the distributed 3D FFT, investigate their behaviour, and compare their performance with the single GPU CUFFT and CPU-based FFTW libraries. In particular, we demonstrate the advantage of direct GPU-to-GPU transfers over data exchanges via host main memory. Our preliminary results show that running the distributed 3D FFT with four GPUs can bring a 12% speedup over the single node (CUFFT) while also enabling the calculation of 3D FFTs of larger datasets. Replacing the global data exchange via shared memory with direct GPU-to-GPU transfers reduces the execution time by up to 49%. This clearly shows that direct GPU-to-GPU transfers are the key factor in obtaining good performance on multi-GPU systems.**

***Keywords*-GPU; UVA; unified-virtual-address; multi-GPU; FFT; distributed; shared-memory;**

I. Introduction

The use of graphics processing units (GPUs) as general-purpose massively-parallel processors is now common place in high performance computing systems. Initially this involved augmenting each node of a distributed memory system with a single GPU. Typical node hardware can, however, support multiple GPUs and, as node CPUs become increasingly multicore, the trend would suggest that each node will become populated with multiple GPUs.

To date, relatively little work has been reported on optimising algorithms to run on multiple GPUs attached to a single shared memory host. This paper considers this issue within the context of the Fast Fourier Transform (FFT) algorithm [1]. The FFT is a core component for many computational techniques, including signal processing, fluid dynamics, molecular dynamics, medical imaging, etc.

Our particular interest in the use of multiple GPU systems is driven by the desire to perform large-scale ultrasound simulations using the k-space pseudo-spectral method in time-frames that are clinically meaningful [2]. The k-space method makes extensive use of large 3D FFTs (dimensions of 1024^3 or greater), which constitute over half of the total computation time.

This paper is structured as follows. In section II we consider the hardware and software environment in detail. Section III outlines related work. Sections IV and V describe our algorithm and the use of direct device-to-device transfers respectively. Section VI presents our performance results, while section VII contains conclusions and discussion.

II. Multi-GPU Hardware and Software

The multi-GPU system used in this work is based on the Tyan barebone TYAN FT72B7015 [3]. The motherboard has two LGA 1366 sockets for Intel Core i7 processors in a NUMA configuration (see schematic in Figure 1). Each socket is populated with a six-core Intel Xeon X5650 processor giving a total of twelve physical cores. The server is equipped with twelve 4 GB memory modules (48 GB RAM) and has an aggregated memory bandwidth of 2×25 GB/s. Communication between the CPUs is supported by the Intel QuickPath Interconnection (QPI) with a theoretical bandwidth of 12 GB/s.

The QPI also connects each CPU with an Intel IOH chip that offers various I/O connections including a total of four PCIe x16 links. Each PCIe link is branched by a PLX PEX 8647 switch to give a total of eight PCIe slots. As this system is designed as a node in a cluster, one PCIe slot is reserved for a high-bandwidth interconnect (Infiniband). The remaining seven slots are populated with GPUs.

Each GPU is an NVIDIA GeForce GTX 580 with 512 CUDA cores and 1.5 GB of memory. Access to the CPU by the GPUs or vice versa is provided by the PEX bridge multiplexing the PCIe x16 links as required on demand. All GPUs use NVIDIA CUDA 4.1 [4].

A typical CUDA workflow involves creating a task on the host (CPU) side, allocating memory for task data on the device (GPU), copying that data to the device, and then executing the task "kernel" on the device. When the kernel completes, data is retrieved from the device before the device memory is freed.

When exploiting multiple devices, multiple tasks are executed simultaneously. If the tasks are mutually independent,

978-0-7695-4879-1/12 $26.00 © 2012 Crown Copyright
DOI 10.1109/PDCAT.2012.79

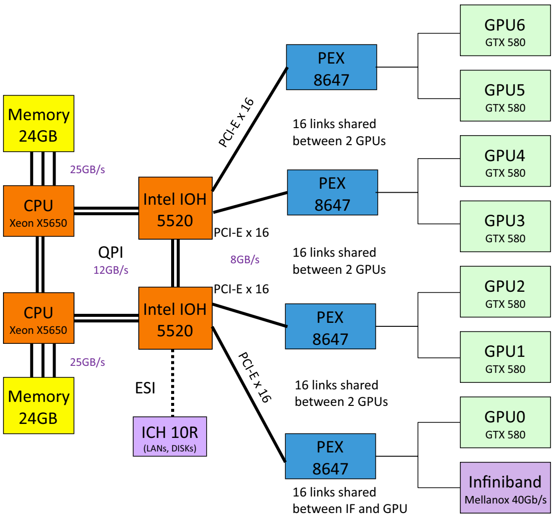


Figure 1. Schematic of the core components on the motherboard of a multi-GPU shared-memory system.

no communication among devices is necessary – if dependent, communication is required. Historically, inter-device communication in CUDA was performed by the host. That is, the host would collect pieces of data from each device into its memory space, and then send the relevant data to the relevant destination device memory.

The latest versions of CUDA introduce the Unified Virtual Address space (UVA). This enables direct access to remote device memory by CUDA kernels and CUDA memory copy routines. However, on the platform used here CUDA device-to-device communication is limited to devices that are under the same Intel IOH bridge (e.g. GPU0, GPU1 and GPU2 in Figure 1). This is due to Intel's implementation of the PCIe 2.0 protocol in the 5520 chipset and its incompatibility with the Intel QPI [5].

Recognizing the time required to transfer data to and from the GPU devices, NVIDIA included in CUDA the concept of streams. A CUDA stream represents a queue of GPU operations (kernels, memory transfers) that get executed in a specific order. Effectively the GPU devices can be thought of as being made of two parts: a copy engine (the device DMA controller); and a compute engine (the CUDA cores). With multiple streams the copy engine can execute a memory transfer from one stream while the CUDA cores are busy processing a kernel from another stream.

III. Related Work

The de facto standard for CPU FFT implementations is FFTW [1], [6], now in version 3.3. It is available for shared and distributed memory systems. For NVIDIA GPUs there is CUFFT [7] (we use CUFFT version 4.1), however, this provides no interface for utilizing multiple GPU devices.

P3DFFT is an open-source off-the-shelf framework for distributing 3D FFT [8]. It does not compute the transform itself, but handles all of the decomposition and communication tasks required for performing a distributed 3D FFT. It makes use of a 2D (pencil) decomposition, using a localized library, such as FFTW, for the component transform.

Csechowski et al. [9] extended P3DFFT to create DiGPUFFT for use on their GPU cluster. They observed significant performance bottlenecks which they attribute to the cost of communication over the PCIe bus between CPU and GPU. This was estimated to represent approximately 27% of the total time taken by the 3D FFT.

PKUFFT [10] is similar to DiGPUFFT in that it uses a pencil data decomposition with GPUs to perform the actual computation. Whereas P3DFFT appears to be limited to real-to-complex (R2C, forward) and complex-to-real (C2R, backward) transforms, PKUFFT includes complex-to-complex (C2C) transforms. It differs from P3DFFT in the data manipulation it performs at the various stages, and the factoring of architectural elements of GPU clusters into the decomposition and underlying computations.

ULSFFT [11] look at a recursive composition, essentially a restructuring of the butterfly graph to allow a scatter-gather processing model. Gu et al. [12] propose a number of techniques for performing large FFTs when the data is maintained out of a single devices memory space.

None of the above explicitly consider the case of multiple GPUs hosted by a single node; so direct communication between devices was not considered. Also aspects of all these systems are now obsolete, e.g. PKUFFT is presented for version 2.3 of CUDA, and the PCIe controller versions used in the various systems are not immediately apparent.

IV. 3D FFT Implementation

Our implementation for processing the 3D FFT on multiple GPUs can be described in three phases as illustrated in Figure 2. In the 3D domain we refer to X as the depth, Y as the width, and Z as the height with the upper-left-forward corner as the origin or zero$^{\text{th}}$ element. Consecutive elements in X are stored in consecutive memory locations (i.e. the matrix is stored in row-major order).

The first phase begins by partitioning the matrix in the Z dimension into *#batches* contiguous batches ($\#batches{=}\frac{Z}{batch\_size}$) where the dimension of each batch is: *batch_size* $\times Y \times X$. Each batch is also divided in the Y dimension to give $batch\_size \times batch\_size \times X$ pencils, where the number of pencils, $\#pencils{=}\frac{Y}{batch\_size}$. This is shown as Phase 1 in Figure 2, where the cubic matrix is divided into three batches, each with three pencils (left panel, Phase 1, Figure 2). These batches are distributed among the GPUs (center panel, Phase 1, Figure 2), and *batch_size* 2D FFTs of size XY are performed (right panel, Phase 1, Figure 2)

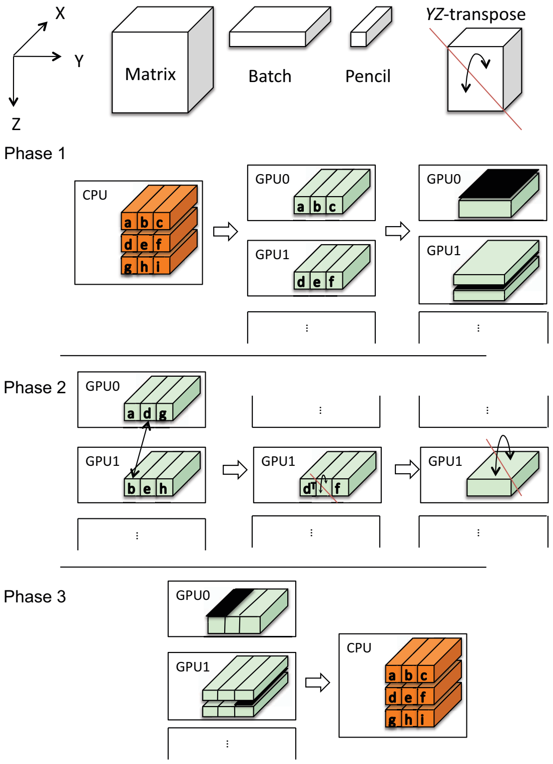


Figure 2. Decomposition and Movements of Pencils: Phase 1) matrix distributed in batches, 2D FFTs performed; Phase 2) communication between devices (D2D) with pencils and memory manipulations (transposes); Phase 3) 1D FFTs and transformed data returned in original format.

for each batch using the CUDA provided CUFFT library. At the end of Phase 1 the device memory contains the XY component of the result.

To complete the 3D FFT each Z component, which is presently distributed across multiple devices, is rearranged to be contiguous within a device. This is referred to as Phase 2 in Figure 2 and requires an all-to-all communication. This communication is managed by traversing the upper-right triangular set of pencils, calculating the destination buffer for the pencil, and performing a data swap (left panel, Phase 2, Figure 2). The destination is uniquely identified by the ID number of the device, a buffer identified by the ID of the batch stored in it, and an offset into the buffer for the number of pencils before it. These values are derived as follows:

- *device ID* = $\frac{\mathit{pencilId}}{\mathit{\#pencils/\#GPUs}}$ (where *pencilId* is the unique number of the pencil within the batch with range $[0\text{–}\mathit{\#pencils})$).
- *batch ID* = pencilId $\pmod{\frac{\mathit{\#pencils}}{\mathit{\#GPUs}}}$.
- *pencil offset* = *batchId* (the unique number of the batch in the range $[0\text{–}\mathit{\#batches})$).

After the pencil movement in Phase 2, all Z data is on the correct device for the final 1D FFT, although it is not contiguous. This is addressed by performing a YZ transposition on each pencil (center panel, Phase 2, Figure 2), followed by an XY transposition on the entire batch (right panel, Phase 2, Figure 2). The net effect of Phase 2 is a global YZ transpose plus a global XY transpose. With this complete the final stage consists of $\mathit{batch\_size} \times Y$ 1D FFTs each of size Z (left panel, Phase 3, Figure 2).

The 3D FFT of the data now exists in the device memory space, albeit in a slight permutation. Performing the inverse of the memory operations/communications will place the transformed data in the same orientation as the original data back into main memory (right panel, Phase 3, Figure 2).

V. Device-To-Device Communication

The challenging phase of the distributed 3D FFT is the distributed 3D matrix transposition in the second phase. All the elements of the 3D matrix have to be rearranged and redistributed which leads to at least $\mathit{\#GPUs} \times (\mathit{\#GPUs} - 1)$ device-to-device (D2D) transfers and at most $Z \times (Z - 1)$. These two cases occur as the number of transfers (*#transfers*) and the size of each transfer (*transfer_size*) is dependent on the *batch_size*. In a simplified analysis, $\mathit{\#transfers} = \frac{Z}{\mathit{batch\_size}} - \frac{\mathit{\#batches}}{\mathit{\#GPUs}}$ and $\mathit{transfer\_size} = \mathit{batch\_size} \times Y \times X$. Poorer performance is expected in the second case, when the *batch_size* is small, as there is a greater number of smaller transfers which are less likely to saturate the PCIe bus.

To understand the communication characteristics and bottlenecks of the given server architecture, we designed programs that perform data exchanges (swaps) between pairs of GPUs under different scenarios. These programs take a list of device IDs as pairs and swap the data, i.e. given a list (1, 2, 3, 4), two swaps will be performed: GPU1 with GPU2, and GPU3 with GPU4. Four different swap methods were designed: 1) a swap via host memory; 2) a staged CPU swap, where one transfer is performed via CPU and the other by direct D2D copy; 3) a direct D2D memory copy; and 4) a swap via kernel, where a GPU kernel uses registers to carry out the swap. In order to compare the methods we measured the effective bandwidth over the PCIe, defined as the rate at which the net data movement is achieved (in this case $\frac{\mathit{\#GPUs} \times \mathit{transfer\_size}}{\text{time}} \frac{\text{GB}}{\text{s}}$, where *#GPUs* is the number of devices in the list).

In method 1 the communication is performed via the host's memory (main memory) in two steps. First the data packages from the devices are gathered and stored in main memory in an asynchronous manner. After synchronising, the data packages are scattered to the particular devices.

Method 2 uses the CPU to stage part of the swap which takes three steps. In the first step the data of the first device

Table I
EFFECTIVE BANDWIDTH FOR INTER-DEVICE DATA SWAP (GB/S)

Method	2 GPUS		4 GPUS		6 GPUS	
	Min.	Max.	Min.	Max.	Min.	Max.
1) via CPU	2.95	5.38	3.83	7.55	4.54	5.71
2) staged CPU	2.82	4.71	2.83	9.41	4.18	9.88
3) ptr swap	2.91	8.11	2.97	16.20	4.39	24.29
4) kernel	5.94	8.24	5.96	16.46	-	-

is copied to host main memory. In the second step, a D2D memory transfer is used to move data from the second device to the first one overriding memory freed in the previous phase. Finally, the host transfers the first device's data to the second one from the temporary host buffer.

Method 3 achieves the communication using built-in CUDA functions to perform two asynchronous D2D transfers followed by pointer swapping. For this method, two distinct buffers have to be allocated on each device (the source and destination). One buffer acts as an input/output buffer, while the other contains the current data. The pointer swapping between the buffers marks the new transferred data as current, and the old current buffer as reusable.

Method 4 uses a kernel to handle the communication in place. The kernel is executed only on one device in the pair. As the UVA enables direct accesses to the remote memory, the kernel first loads the data on the executing device into local registers. Then it accesses the remote memory to perform the transfer from the other device to this. The swap is finalised by storing the value in the local registers to the remote memory.

VI. DISTRIBUTED 3D FFT PERFORMANCE RESULTS

The first set of results we present are for the effective bandwidths of the four methods of swapping data between devices. A range of bandwidths were recorded by varying the device pairs used by the methods and are displayed in Table I. Both bandwidths are important as an all-to-all communication is required.

When any communication involving the CPU occurs (methods 1 and 2), the observed effective bandwidth is significantly limited. These methods have the smallest bandwidth when devices on the same IOH chip or PEX switch are used, which limits the bandwidth in and out of the CPU. Conversely, methods 3 and 4 achieved their maximum bandwidths in these conditions due to the aggregated bandwidth of the PEX switches. These two methods, not involving the CPU, only move data when necessary. In contrast, method 1 has to move twice the minimum amount of data and method 2 one-and-a-half times the minimum amount over the PCIe network which lowers the effective bandwidth. All methods except method 4 require some amount of additional memory on the host or device. However, method 4 is limited to using devices on the same IOH chip. Although method 3 works with devices on different IOH chips, it requires data to be staged by the CPU, which is managed differently to method 1 as the effective bandwidths vary significantly depending on device selection.

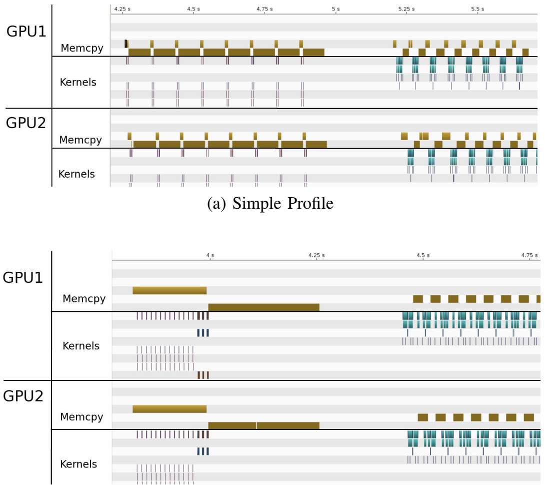
Table II
PERFORMANCE FOR ASSORTED *batch_sizes* AND SIMPLE DISTRIBUTED 1024^3 COMPLEX SINGLE-PRECISION FFT ACROSS 6 GPUS

batch_size	Time (seconds)		Size of Pencils
	1 Stream	2 Streams	
1	8.96	8.72	8 KB
2	7.27	7.30	32 KB
4	6.35	6.42	128 KB
8	6.33	6.33	512 KB
16	6.17	6.15	2 MB
32	6.77	6.57	8 MB

Considering these communication characteristics, we created three implementations for the distributed 3D FFT. The first implementation, simple, is a point of reference differing to the details in Section IV only in that the matrix is maintained in main memory, and batches and pencils are transferred on demand via the CPU to the GPUs for processing. The second implementation, D2D via ptr. swap (pointer swap), and third implementation, D2D via kernel, follow the technique in Section IV and perform the D2D communication directly between devices managing them via pointer swapping and a kernel respectively.

For all three of these implementations the *batch_size* is one parameter which can affect the performance in two ways. Table II demonstrates the effect of varying the *batch_size* for the simple implementation of the distributed 3D FFT. The first way this impacts performance is because this value directly changes the size of the largest contiguous set of data that is accessed and transferred, i.e. the size of the transfers on the PCIe. In order to saturate the PCIe bus and operate at maximum bandwidth, sufficiently large messages are required. This can be seen in the speed up as the *batch_size* increases, and the 25% improvement from a size of 1 to 32.

The second way in which the *batch_size* can affect performance is in the application of streams. This factor defines the size and consequently the time of the computations. A shorter time results in a higher granularity of tasks (kernels, or memory transfers) to be scheduled between the copy and compute engines of the device. If the computational component is significant, then the *batch_size* should be balanced according to the bandwidth to efficiently overlap computation and communication. Table II suggests that this component is insignificant, accounting for approximately 4% of the total time. We also observe some loss of performance



(a) Simple Profile

(b) D2D via Pointer Swapping Profile

Figure 3. Timeline of GPU activity showing memory transfers (brown), in-device transposes (green), and FFT kernels (blue).

for select *batch_sizes* when streamed, which we attribute to the dynamic nature of the device scheduler and the alignment of data in intermediary caches.

In order to confirm this, the implementations were profiled using the NVIDIA Visual Profiler [4]. Figure 3 provides a timeline of GPU activity for the simple and D2D via kernel implementations. This confirms that the computational component is not significant, and is hidden by communication.

In Table III we present results for the three implementations on multiple GPUs alongside results for FFTW on 6 and 12 CPU cores and CUFFT (where applicable). Each FFT was applied to complex 3D single-precision cubic matrices with dimensions 512^3, 768^3, and 1024^3. The performance was measured as the average of the times taken to perform the FFT. Times were recorded for *batch_sizes* 1, 2, 4, 8, 16, and 32, and for one and two streams per device. The fastest average times were used. The results for the simple implementation provides a baseline for the D2D versions. The performance of this version is slower in all cases to the reference CPU (FFTW) and GPU (CUFFT) versions.

The results for the D2D via ptr. swap implementation are not available for 1024^3, and 768^3 for two and four GPUs, as the total device memory is not enough for the swap method which requires 7 GB for 768^3 and 16 GB for 1024^3. Similarly, for the D2D via kernel implementation for 1024^3 and 768^3 with two GPUS. When using six GPUs, the D2D via ptr swap implementation stages some communications through main memory, and the D2D via kernel implementation can not be run due to the Intel IOH chips preventing the UVA accessing more than four devices.

Table III
TIME (SECONDS) TO COMPUTE 3D COMPLEX FFT

			Size	
Implementation	Cores/GPUs	512^3	768^3	1024^3
simple	1	1.74	6.32	14.05
	2	1.21	4.26	9.81
	4	0.81	2.78	6.52
	6	0.77	2.64	6.16
D2D via ptr. swap	2	0.61	-	-
	4	0.50	-	-
	6	0.64	2.07	-
D2D via kernel	2	0.80	-	-
	4	0.74	2.46	-
CUFFT	1	0.56	-	-
FFTW	6	0.61	2.01	5.64
	12	0.30	1.07	2.82

- missing value, not enough memory on GPUs.

In all cases the implementations using direct D2D transfers were faster than the simple version communicating via host memory. The greatest speed up was observed with the D2D via ptr. swap implementation on two GPUs over the 512^3 matrix where the performance improved by 49%.

Despite the kernel swapping approach performing better in the effective bandwidth tests, the performance of the D2D via kernel implementation was in general worse than the D2D via ptr. swap version. However, for smaller *batch_sizes* this method was faster. An explanation for this performance for large *batch_sizes* and matrices is the limit in the resource configuration of kernels. Another, is that the kernel both involves transfers and computation and consequently does not schedule well. It should be noted that the D2D via kernel implementation requires half the memory of the other implementation which allows it to compute the 768^3 matrix with only four GPUs. Although using fewer GPUs, this was slower than the pointer swapping method with six GPUs by 16%. This suggests that the slower transfers staged through main memory with pointer swapping may have been masked by the other D2D communications.

Compared to CUFFT and FFTW, all three implementations were generally slower. However, at 512^3 the D2D via ptr. swap implementation over four GPUs was found to outperform CUFFT (a single GPU) by 12%, and FFTW when six cores on a single CPU were used by 18%. FFTW over 12 cores (both CPUs) was still faster in these conditions by at least 40%.

VII. Conclusions and Future Work

The goal of this paper was to investigate the acceleration of the 3D FFT using multiple GPUs within a shared memory environment. There are two reasons to do this: to extend the available memory beyond the limits of a single device, and to reduce computation time. The distributed 3D FFT consists of three phases. First, the data is distributed in batches, and the 2D FFT on each slice is performed. Second, the data is redistributed using pencils amongst the devices to make it contiguous in the third dimension. Third, the 1D FFTs are performed over the remaining dimension, and the data gathered back to the host. The most time consuming component is the data redistribution between devices. In this work we propose several different methods for this, including via host memory, and directly between devices. The most efficient means of data exchange is by swapping pointers between multiple buffers using asynchronous D2D CUDA UVA memory copies. Compared to global data exchange via shared memory, this reduces the execution time of the 3D FFT by up to 49%. This clearly shows that direct D2D are a key factor in obtaining high performance on multi-GPU systems.

Even using the most efficient distributed 3D FFT implementation (D2D via ptr. swap), the performance is strongly limited by the PCIe throughput. Compared to the bandwidth between a single GPU's compute cores and memory (200 GB/s), or the CPU cores and host memory (25 GB/s), the limit of 8 GB/s using PCIe 2.0 is a significant bottleneck. For example, using two GPUs to compute a 512^3 3D FFT is 12% slower than using a single GPU. However, using four GPUs allows access to a greater aggregate PCIe bandwidth resulting in a 12% speedup. This has significant implications for both the hardware and software development for multi-GPU systems. It is possible that the hardware constraints may be partially alleviated by the release of systems based on PCIe 3.0 which doubles the throughput of PCIe 2.0. An additional limitation is that the GTX 580 used in our multi-GPU system has only a single copy engine (unlike the Tesla series of NVIDIA hardware [13]), and thus can not exploit the duplex capability of PCIe.

Compared to using FFTW on a modern twin CPU socket setup, the performance of the distributed 3D FFT is worse, even when using six GPUs. This is again due to the bandwidth limitations imposed by PCIe. However, there are several scenarios where using GPUs may be advantageous. For example, if the data is already distributed in GPU memory space (as part of a larger GPU based simulation), or if the availability of data is bottlenecked by another factor such as an external network (Infiniband).

Due to the hardware limitations of the Intel IOH chips, the D2D implementations are limited to four GPUs. This places a ceiling on the aggregate GPU memory (size of the UVA space) and size of the largest FFT that can be performed. To use more devices we are forced to stage data in host memory, and transfer data on demand to GPUs for processing. This is less efficient but allows for larger problem sizes.

A possible extension of our implementations would be to incorporate some of the memory transfer operations into a custom FFT kernel in each phase. The effect of this would be to perform these operations as soon as possible, within the same kernel launch, without the need for additional synchronisations. This would improve the balance between computational and communication tasks. Moreover, the CUFFT library requires additional memory for every batch of 2D and 1D FFTs processed. This could be avoided by implementing a custom FFT kernel.

Acknowledgment

This work was supported in part by the Australian Research Council/Microsoft Linkage Project LP100100588.

References

[1] M. Frigo and S. Johnson, "FFTW: An adaptive software architecture for the FFT," in *ASSP*, vol. 3. IEEE, 1998, pp. 1381–1384.

[2] B. E. Treeby, J. Jaros, A. P. Rendell, and B. T. Cox, "Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method," *J. Acoust. Soc. Am.*, vol. 131, no. 6, pp. 4324–4336, 2012.

[3] MiTAC International Corporation. (2011, Sep.) Tyan ft72b7015 server barebone.

[4] NVIDIA Corp., "NVIDIA CUDA Programming Guide Version 4.1," NVIDIA, Tech. Rep., Nov. 2011.

[5] P. Micikevicius. (2011, Nov.) M07: High performance computing with cuda. SC11 Tutorial. [Online]. Available: http://sc11.supercomputing.org

[6] M. Frigo and S. Johnson, "The Design and Implementation of FFTW3," *Proc. IEEE*, vol. 93, no. 2, pp. 216–231, 2005.

[7] NVIDIA Corp., "CUDA Toolkit 4.1 CUFFT Library," NVIDIA, Tech. Rep., Jan. 2012.

[8] D. Pekurovsky and J. Goebbert, "P3dfft-highly scalable parallel 3d fast fourier transforms library," University of California, Tech. Rep., Nov. 2011. [Online]. Available: http://code.google.com/p/p3dfft/

[9] K. Czechowski, C. McClanahan, C. Battaglino, K. Iyer, P.-K. Yeung, and R. Vuduc, "On the communication complexity of 3D FFTs and its implications for exascale," in *ICS*, San Servolo Island, Venice, Italy, Jun. 2012.

[10] Y. Chen, X. Cui, and H. Mei, "Large-scale fft on gpu clusters," *ICS*, pp. 315–324, 2010.

[11] J. Glenn-Anderson, "Ultra large-scale fft processing on graphics processor arrays," *enparallelcom*, 2009. [Online]. Available: http://enparallel.com/ULSFFT.pdf

[12] L. Gu, J. Siegel, and X. Li, *Using GPUs to Compute Large Out-of-card FFTs*, 2011, pp. 255–264.

[13] NVIDIA Corp. (2012, Jul.) Tesla. [Online]. Available: http://www.nvidia.com/object/tesla-supercomputing-solutions.html

B.6 Local Domain Decomposition

Jaros, J.; Vaverka, F.; Treeby, B. E.: Spectral Domain Decomposition Using Local Fourier Basis: Application to Ultrasound Simulation on a Cluster of GPUs. *Journal of Supercomputing Frontiers and Innovations*. vol. 3, no. 3. 2016: pp. 39–54. ISSN 2313-8734. doi:10.14529/jsfi160305.

DOI: 10.14529/jsfi160305

Spectral Domain Decomposition Using Local Fourier Basis: Application to Ultrasound Simulation on a Cluster of GPUs

J. Jaros[1], *F. Vaverka*[1], *B.E. Treeby*[2]

The simulation of ultrasound wave propagation through biological tissue has a wide range of practical applications. However, large grid sizes are generally needed to capture the phenomena of interest. Here, a novel approach to reduce the computational complexity is presented. The model uses an accelerated k-space pseudospectral method which enables more than one hundred GPUs to be exploited to solve problems with more than 3×10^9 grid points. The classic communication bottleneck of Fourier spectral methods, all-to-all global data exchange, is overcome by the application of domain decomposition using local Fourier basis. Compared to global domain decomposition, for a grid size of 1536 × 1024 × 2048, this reduces the simulation time by a factor of 7.5 and the simulation cost by a factor of 3.8.

Keywords: domain decomposition, ultrasound simulation, spectral methods, GPU, FFT, local Fourier basis.

Introduction

Accurately simulating the propagation of ultrasound waves through biological tissue has a large number of practical applications, including the physics-based simulation of diagnostic ultrasound images [1] and treatment planning for ultrasound therapy [2] (a more comprehensive list is provided in [3]). However, ultrasound simulation for these applications is computationally demanding due to the length scales involved, where the propagation length can be hundreds or thousands of times longer than the acoustic wavelength. In turn, this leads to very large domain sizes, in some cases with more than 100 billion grid points [2]. This puts the simulation times beyond clinically useful time-limits, even when using significant computational resources [4]. The overarching goal of this work is to maximise computational efficiency to minimise the wall-clock time needed to solve such large scale problems.

One of the biggest challenges in performing large-scale ultrasound simulations is the accumulation of numerical dispersion. In general, this can be overcome through the application of spectral methods, which can be considered memory minimising due to their exponential error convergence with grid density [5]. For wave problems, the k-space pseudospectral method is particularly efficient. This combines the spectral calculation of spatial gradients (using the Fourier collocation spectral method) with a dispersion-corrected finite difference scheme to integrate forward in time. This approach was first proposed in [6] and further developed in [7–9]. The parallel implementation of the k-space pseudospectral method has previously been described by several groups [2, 10–13]. Aside from a number of element-wise matrix operations, the most significant operations performed at each time step are multiple real-to-complex and complex-to-real 3D fast Fourier transforms (FFTs). The parallel efficiency of the method therefore depends primarily on the parallel efficiency of the FFT. Note, in the conventional pseudospectral time domain method, the FFTs are 1D. However, for the k-space method, the FFTs are 3D, as the dispersion correction step is applied in the spatial Fourier domain.

The biggest challenge in the calculation of the 3D FFT is a globally synchronising all-to-all data exchange. This is required to transpose the 3D matrix data, as the FFT cannot

[1] Faculty of Information Technology, Brno University of Technology, Brno, CZ

[2] Department of Medical Physics and Biomedical Engineering, University College London, London, UK

stride across data belonging to multiple processes. Despite the large amount of progress on optimizing the implementation of distributed FFTs, the inherent communication bottleneck still limits scaling efficiency. The distributed FFT implementations deployed on CPU clusters usually achieve scaling factors between 1.5 and 1.8 when the number of processing elements is doubled. Pippig [14] reported a comparative study of FFTW [15], PFFT [14], and P3DFFT [16] using an IBM Blue Gene machine. Similar investigations using Intel-Infiniband clusters were reported in [13, 17, 18]. In general, the majority of the execution time is spent in communication. For typical ultrasound simulations with grid sizes ranging from 512^3 to 2048^3 grid points, when distributed over more than 512 CPU cores, 50-90% of the execution time is wasted waiting for data exchanges [2].

The imbalance between communication and computation is even more striking when graphics processing units (GPUs) are used, as the raw performance of GPUs is an order of magnitude above current central processing units (CPUs). In addition, transfers over the peripheral component interconnect express (PCI-E) bus have to be considered as another source of communication overhead. The implementation proposed by Gholami [17], which is currently one of the most efficient, reveals the fundamental communication problem of distributed GPU FFTs. For an 1024^3 FFT calculated using 128 GPUs, the communication overhead accounts for 99% of the total execution time. Although the execution time reduces by 8.6× for a 32× increase in the number of GPUs (giving a parallel efficiency of 27%), this overhead may be acceptable in many applications.

One way to overcome the global communication imposed by the Fourier spectral method is to use a *local* Fourier basis as proposed by Israeli, *et al* [19]. This allows the evaluation of derivatives to be splitted into multiple coupled subdomains, where the Fourier transforms for each subdomain are computed independently, followed by the exchange of data in an overlap or halo region. The spectral accuracy is maintained by forcing the local domains to be periodic through multiplication of the local data by a bell function. The bell function is equal to one within the physical domain, and tapers to zero within the overlap region [20]. Using local, Founer basis rather than global ones, FFTs can have a significant impact on the computational performance of Fourier spectral methods. For example, Ding & Chen implemented a solution to Maxwell's equations using local Fourier basis [21]. For the simulation with 1024^3 grid points running on 32 CPUs, they reported reduction in communication time from 9.49 seconds per time step when using global FFTs, to 1.46 seconds per time step when using local Fourier basis with 32 subdomains. Similarly, Garbey, *et al* reported close to ideal weak scaling when using local Fourier basis to solve a combustion problem using up to 16 processors [22].

In the current work, domain decomposition using local Fourier basis is combined with the k-space pseudospectral method to allow the highly efficient simulation of ultrasound propagation using a cluster of 128 GPUs with grid sizes up to 1024 × 1536 × 2048. The governing equations and their discretisation are discussed in Sec. 2, with the local decomposition introduced in Sec. 3. Details of the parallel implementation are given in Sec. 4, with numerical experiments presented in Sec. 5. Summary and discussion are then given in Sec. 6.

1. Pseudospectral Ultrasound Model

The physical problem considered here is the propagation of small amplitude acoustic waves through a homogeneous and lossless fluid medium. In this case, the governing equations are given by a set of coupled first-order partial differential equations [23]

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho_0}\nabla p \ , \quad \frac{\partial \rho}{\partial t} = -\rho_0 \nabla \cdot \mathbf{u} \ , \quad p = c_0^2 \rho \ . \tag{1}$$

Here $\mathbf{u}$ is the acoustic particle velocity, p is the acoustic pressure, c_0 is the sound speed, and ρ_0 and ρ are the ambient and acoustic density, respectively. The governing equations are solved using the k-space pseudospectral method, where spatial gradients are computed using the Fourier collocation spectral method, and time integration is performed using a dispersion-corrected finite difference scheme [8]. Written in discrete form, the governing equations in Eq. (1) become [2, 9]

$$\begin{aligned}
\frac{\partial}{\partial \xi} p^n &= \mathcal{F}^{-1}\left\{ ik_\xi \, \kappa \, e^{ik_\xi \Delta\xi/2} \mathcal{F}\left\{ p^n \right\} \right\} \ , \\
u_\xi^{n+\frac{1}{2}} &= u_\xi^{n-\frac{1}{2}} - \frac{\Delta t}{\rho_0} \frac{\partial}{\partial \xi} p^n \\
\frac{\partial}{\partial \xi} u_\xi^{n+\frac{1}{2}} &= \mathcal{F}^{-1}\left\{ ik_\xi \, \kappa \, e^{-ik_\xi \Delta\xi/2} \mathcal{F}\left\{ u_\xi^{n+\frac{1}{2}} \right\} \right\} \ , \\
\rho_\xi^{n+1} &= \rho_\xi^n - \Delta t \rho_0 \frac{\partial}{\partial \xi} u_\xi^{n+\frac{1}{2}} \ , \\
p^{n+1} &= c_0^2 \sum_\xi \rho_\xi^{n+1} \ .
\end{aligned} \tag{2}$$

The first four equations are repeated for each Cartesian direction, where $\xi = x, y, z$. Here, $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the forward and inverse FFT over all three spatial dimensions, i is the imaginary unit, k_ξ is the wavenumber vector in the ξ direction, $\Delta\xi$ is the grid spacing in the ξ direction, Δt is the time step, and κ is the so-called k-space operator used to correct for numerical dispersion introduced by the finite difference time step [8]. The acoustic density (which is physically a scalar quantity) is artificially divided into Cartesian components to allow an anisotropic perfectly matched layer (PML) to be applied to model free-field conditions [24]. The exponential terms are spatial shift operators that allow the particle velocity to be evaluated on a staggered grid [8]. The superscripts n and $n + 1$ denote the function values at the current and next time points, and $n - \frac{1}{2}$ and $n + \frac{1}{2}$ at time staggered points.

The implementation of the discrete equations requires the storage of thirteen real 3D matrices defined in the spatial domain and three real and one complex 3D matrix defined in the Fourier domain. These are used to store the current values of the acoustic variables, their derivatives, and three temporary matrices. For a single precision shared memory implementation, the memory usage can be estimated as

$$\text{memory usage [GB]} \approx \frac{16.5 \times \text{Nx} \times \text{Ny} \times \text{Nz}}{1024^3/4} \ , \tag{3}$$

(neglecting scalars, 1D arrays, and the code itself). At each time step, the operations performed consist of four forward and six inverse 3D FFTs, and around 100 element-wise matrix operations.

Previously, this type of model has been implemented using C++ and parallelised using either OpenMP, with simulations reported up to 1024 × 1024 × 1024 grid points [9], or MPI, with simulations reported up to 4096 × 2048 × 2048 grid points [2, 10, 11]. In both cases, the 3D FFTs are calculated over the entire 3D domain, which requires an all-to-all global communication for each FFT. Although reasonable scaling is observed, particularly when using hybrid OpenMP/MPI decomposition [13], ten all-to-all communications are still required per time step, which is a major bottleneck in performance [2].

2. Local Fourier Basis decomposition

2.1. Formulation

As described in Sec. , the motivation behind domain decomposition using local Fourier basis is to replace the global FFTs by local FFTs computed independently on a series of subdomains. The general approach as applied to the k-space pseudospectral method can be described as follows. First, the field variables and material parameters are divided across the subdomains, including an overlap region (halo) with a specified width (see fig. 1). Here, the so-called non-overlapped decomposition is used [25], and the subdomains are assumed to always be of an equal size. Second, for each subdomain, an independent version of the complete k-space pseudospectral model is run. The spatial gradients are calculated as normal using local Fourier basis, but before taking the FFT, the halo is exchanged and function values are multiplied by a bell function. This tapers to zero within the overlap region to enforce periodicity. Here, the erf-like bell function defined by Boyd is used [25]. This is equal to 1 within the physical domain and given by $H(x) = \frac{1}{2}(1 + \mathrm{erf}(Lx/\sqrt{1-x^2}))$ within the overlap region, where L is a scaling parameter, which in this case is set to 2. The discrete values for x within the overlap region are given by $x = -1, -1 + 2/(N-1), \ldots, 1$, where N is the size of the overlap in grid points.

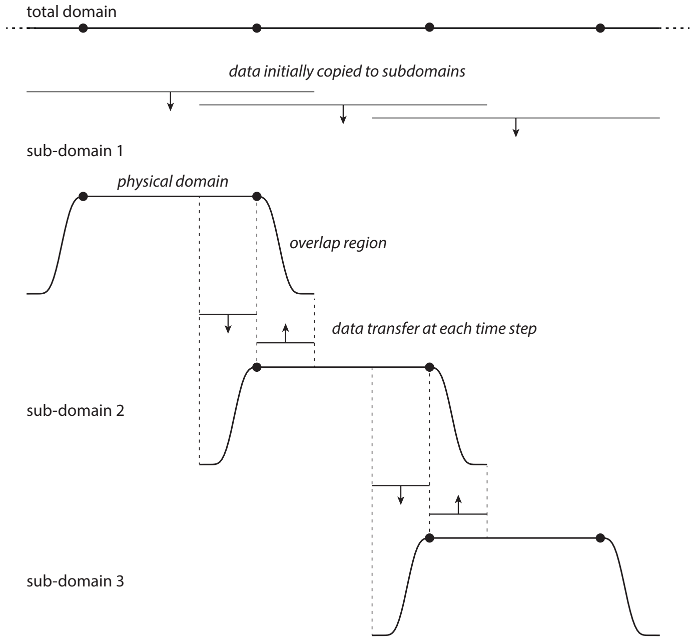


Figure 1. Schematic showing domain decomposition using local Fourier basis

2.2. Multidimensional decomposition

In 3D, there are several approaches to the decomposition of the global domain into subdomains, including 1D slab decomposition, 2D pencil decomposition, and 3D cube decomposition. The main differences are the number of interfaces a wave must travel through when traversing the grid in a given direction (this affects accuracy as discussed in Sec 2.3), the ratio between the halo and local subdomain size, and the number of data transfers that have to be performed [26]. The ratio between the halo and the local subdomain size improves with the dimensionality of the decomposition. For a fixed number of subdomains, higher dimensionality implies a smaller amount of data that must be exchanged and consequently lower computational overhead. On the other hand, the number of direct neighbours grows with the dimensionality of the decomposition, i.e., there are 2, 8, and 26 direct neighbours for uniform 1D, 2D, and 3D decompositions, respectively. Since the interconnection bandwidth is not constant for all message sizes [27], in some cases it may be better to use 1D decomposition instead of 2D or 3D, even if a larger amount of data must be exchanged among a smaller number of neighbours. The impact of different decompositions on performance is discussed in Sec 4.3.

2.3. Accuracy

To test the accuracy of domain decomposition using local Fourier basis, a series of numerical experiments were conducted using a prototype CPU code. The tests consisted of propagating a plane wave along the grid axis (which reduces to a 1D problem) with the global domain divided into a given number of subdomains with a specified overlap width. The initial particle velocity was set to zero, and the initial pressure was set to be an impulsive pressure source. The spatial distribution of the pressure source was defined as a delta function (filtered by a Blackman window) positioned within the first subdomain. This generates a wave with broadband frequency content that smoothly decays up to the Nyquist limit [28]. For each test, a reference simulation using a global spectral method was also performed.

First, the dependence of the error on the width of the overlap region was examined. The total domain size was fixed at 512 grid points, and the overlap size varied from 8 to 32 grid points. The variation in L_∞ error compared to the reference simulation is shown in fig. 2(a). For an overlap width of 32 grid points, the error has not reached machine precision. However, the equivalent accuracy of the PML is only on the order of 10^{-3} to 10^{-4}, even with optimized parameters [29]. Given accuracy of the global solution is limited by the accuracy of the PML. It is sufficient to maintain a similar level of accuracy for the domain decomposition. Thus an overlap of 16 grid points was chosen, which gives an error less than 10^{-4} when using two subdomains. The change in the error for a fixed overlap size of 16 grid points with the total size of the local subdomain is shown in fig. 2(b). There is almost no change in the error for subdomain sizes from 32 to 1024 grid points, which means the size of the subdomains can be chosen to maximise computational efficiency.

Next, the dependence of the error on the number of domain cuts the wave must traverse was examined (for 1D decomposition, the number of domain cuts is one less than the number of subdomains). The total domain size was fixed at 2048 grid points with an overlap size of 16 grid points, and the number of subdomains was increased from 2 to 32 (in powers of 2). The variation in L_∞ error compared to the reference simulation is shown in fig. 2(c), and the error growth relative to using 2 subdomains is shown in fig. 2(d). The error increases linearly with the number of domain cuts the wave traverses, with a slope of $\sim$0.5. Thus, for typical sized problems (on the order of 2048 grid points in each dimension), up to 31 domain cuts (i.e., 32 subdomains if using 1D decomposition) can be used in each dimension with an overlap size of 16 grid points, and the error is still on the order of 10^{-3}. For 3D decomposition, this corresponds to 32,768 total subdomains (and in this case, GPUs). This means in practice, the level of achievable parallelism is not limited by the reduction in accuracy due to the use of local Fourier basis.

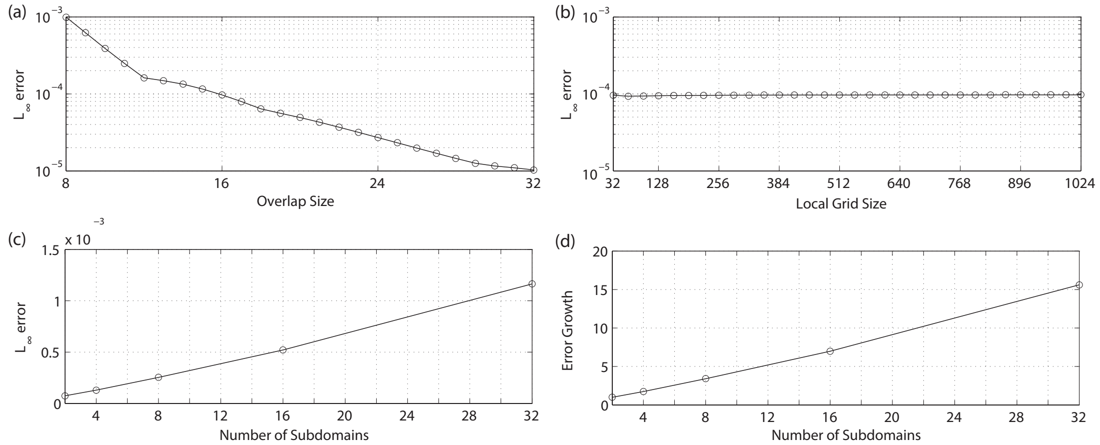


Figure 2. (a) Change in the L_∞ error with the overlap (halo) size for a fixed local domain size of 512 grid points. (b) Change in the L_∞ error with the local domain size for a fixed overlap (halo) size of 16 grid points. (c) Change in the L_∞ error with the number of domain cuts the wave must traverse for a total domain size of 2048 grid points and a fixed overlap size of 16 grid points. For 1D decomposition, the number of domain cuts is one less than the number of subdomains, where 2 subdomains corresponds to a local domain size of 1024 grid points, and 32 subdomains corresponds to a local domain size of 64 grid points. (d) Growth in the error with the number of subdomains relative to 2 subdomains (i.e., 1 domain cut)

3. Implementation

3.1. Communication framework

The numerical model described in Secs. 1 and 2.1 was implemented for multiple NVIDIA GPUs using MPI, C++, and CUDA. The implementation was divided into two components, the first responsible for the initial domain decomposition and periodic halo exchanges, and the second for performing calculations on each local subdomain. Starting with the communication framework, after the simulation is started, the number of subdomains and their organisation in 3D space is determined by parsing command line arguments. The framework supports any 1D, 2D, or 3D partition that fits into on-board GPU memory and meets the minimum size requirements. Each subdomain is assigned to an MPI process. In the case of 1D decomposition, the processes are grouped into an MPI communicator. If 2D or 3D decomposition is chosen, a virtual Cartesian topology is created and MPI is allowed to reorder ranks to preserve spatial locality between neighbouring subdomains. This is particularly useful while working on clusters with multiple GPUs per node.

The next step is GPU acquisition. Every MPI process (rank) inspects the configuration of the node being executed on, and chooses the first free GPU. This allows the framework to run on both slim and fat nodes with multiple GPUs, even in the case of non-uniform clusters (i.e., a mixture of nodes with a different number of integrated GPUs such as the Emerald cluster discussed in Sec. 4.1). The user (cluster batch scheduler) is responsible for assigning the correct number of ranks to individual nodes matching the number of integrated GPUs. The batch scheduler can also assign GPUs directly to ranks. If this feature is not supported by the scheduler and the GPUs are switched into exclusive process compute mode [30], the framework calculates the best assignment automatically and ensures mutual exclusion between ranks.

The execution proceed with the simulation setup. First, the simulation input file is opened and the simulation parameters are loaded. When the simulation domain size is determined, the framework calculates the size and position of the local subdomains, quantifies the size of the halo regions, and allocates all necessary data structures on both the CPU and GPU. The simulation data is then loaded from the input HDF5 file using parallel I/O and transferred into GPU memory.

The simulation time loop is divided into computation and communication phases. In total, there are four data exchanges per time step. These precede the gradient calculation for the acoustic particle velocity $\mathbf{u}$ in each Cartesian direction, and the gradient calculation for the acoustic pressure p (see Eq. (2)). The derivatives of the three spatial components of acoustic particle velocity can be calculated independently, which allows the MPI communications to be partially overlapped with the calculation (this is not possible for the calculation of the pressure gradient). During the data exchange, the halos are extracted from all three 3D matrices of velocity, packed into line-up buffers and downloaded to the CPU. This is done by repeated invocations of simple packing CUDA kernels followed by PCI-E transfers. This way the traffic over PCI-E is minimized. Next, the corresponding `MPI_Isends` and `MPI_Irecvs` are launched. The number of transfers depends on the decomposition chosen and varies between 4 (in the case of 1D decomposition) and 52 (in the case of full 3D decomposition). The execution only waits for the u_x halo to be delivered, uploads the halo back to the GPU, and replaces the appropriate data values. This is done by a PCI-E transfer followed by a CUDA unpack kernel. The calculation of $\frac{\partial}{\partial x}u_x$ is then started while the other four transfers proceed in the background. Thus this implementation partially hides two of three MPI communications.

3.2. Computation framework

The computation framework orchestrates all the necessary calculations in a simulation. It is divided into pre-processing, simulation time loop, data collection, and post-processing phases. The calculations are performed either as calls to the cuFFT library [31], or to custom CUDA kernels. Note, all calculations are performed by the GPU and the CPU only assists with the halo exchange and I/O operations. Since the size of the local subdomains has not been known in advance, several auxiliary variables are calculated during pre-processing, including the local wavenumber vectors, bell function, FFT shifts for staggered grids, etc [2]. The cuFFT library is then initialised and the FFT execution plans are created.

The simulation time loop then follows Eq. (2). The gradient calculations are performed by CUDA kernels which compute the required element-wise operations in both the spatial and Fourier domains. The kernels are organised into 1D CUDA grids composed of 1D CUDA blocks. The grid size is based on the actual number of CUDA multiprocessors (16 blocks per multiprocessor), and the block size is fixed to 256 threads. Every thread is responsible for processing multiple grid elements. The benefit of this solution is high occupancy and memory bandwidth. The same type of CUDA kernel is also used for halo packing and unpacking, as well as for sampled data aggregation and collection. The output data aggregation and post processing steps are described in more detail in [2].

4. Experimental results

4.1. Hardware description

The proposed implementation was deployed and evaluated on two GPU supercomputers, Emerald (e-Infrastructure South, UK) and Anselm (IT4Innovations national supercomputing center, CZ). Emerald is a heterogeneous GPU cluster consisting of several types of nodes equipped with different numbers and models of GPUs. Our allocation was limited to 128 NVIDIA Tesla M2090 cards with 6 GB of on-board memory. As error correction code (ECC) is switched on, the on-board memory capacity is reduced to approximately 5.4 GB. The GPUs are grouped in configurations of 3 or 8 per node, connected by PCI-Express 2.0. In the case of the 8-GPU configuration, pairs of GPUs share 16 PCI-E links. The CPU side is always comprised of two 6-core Westmere processors and 48 or 96 GB of RAM. The interconnection is provided by a 40 Gb/s half-duplex InfiniBand interconnect arranged into a fat-tree topology. The aggregated theoretical GPU performance is 170 TFLOPS in single precision, and the aggregated on-board memory is 768 GB.

Anselm consists of 209 compute nodes with a 40 Gb/s full-duplex InfiniBand interconnect arranged into a fat-tree topology. Each node integrates two 8-core Sandy Bridge CPUs and 64 GB of RAM. Twenty three nodes are equipped with one NVIDIA Kepler K20m GPU card with 5 GB of on-board memory and with ECC switched off. The GPUs are connected by PCI-Express 2.0. Our allocation was limited to 16 GPU cards. The aggregated GPU performance in single precision is 56 TFLOPS, and the aggregated on-board memory is 80 GB. For both systems, the GPU simulation code was compiled with the Intel compiler 2015, Intel MPI 5.0, NVIDIA CUDA 7.5, and HDF5 1.8.16. For comparison, a CPU implementation using global domain decomposition was used [2]. This code was compiled with the same tool chain, in addition to the FFTW 3.3.4 library.

4.2. Strong scaling

Several numerical experiments were performed to assess the performance of the multi-GPU implementation. First, strong scaling was assessed using domain sizes from 256^3 to $1024 \times 1024 \times 2048$ grid points with an overlap size (halo width) of 16 grid points. The problem sizes were limited by the aggregated amount of on-board memory and restrictions imposed by the smallest subdomain size. For Emerald, the scaling was investigated using up to 128 GPUs on the full range of simulation sizes (fig. 3(a)). Since our allocation on Anselm was limited to 16 GPUs, the largest domain size tested was $512 \times 512 \times 1024$ grid points (fig. 3(b)). The domain was partitioned over all three axes into a uniform number of subdomains starting from $1 \times 1 \times 1$ (i.e., running on a single GPU) up to $4 \times 4 \times 8$ and $2 \times 2 \times 4$ for the largest domain sizes on Emerald and Anselm, respectively.

Overall, the code achieves a reasonable scalability. On Emerald, for larger domain sizes the best parallel efficiency is approximately 27% when increasing from 8 to 128 GPUs, 34% from 16 to 128 GPUs, and 55% from 32 to 128 GPUs. In comparison, the strong scaling on Anselm reaches better values of parallel efficiency, reaching approximately 47% when increasing from 2 to 16 GPUs, 56% from 4 to 16 GPUs, and 85% from 8 to 16 GPUs. These levels of parallel efficiency are caused by a combination of multiple factors:

1. As the domain size grows, there is an increase in the number of neighbours the halo must be exchanged with. Since the number of GPUs is initially small, the decomposition is first done

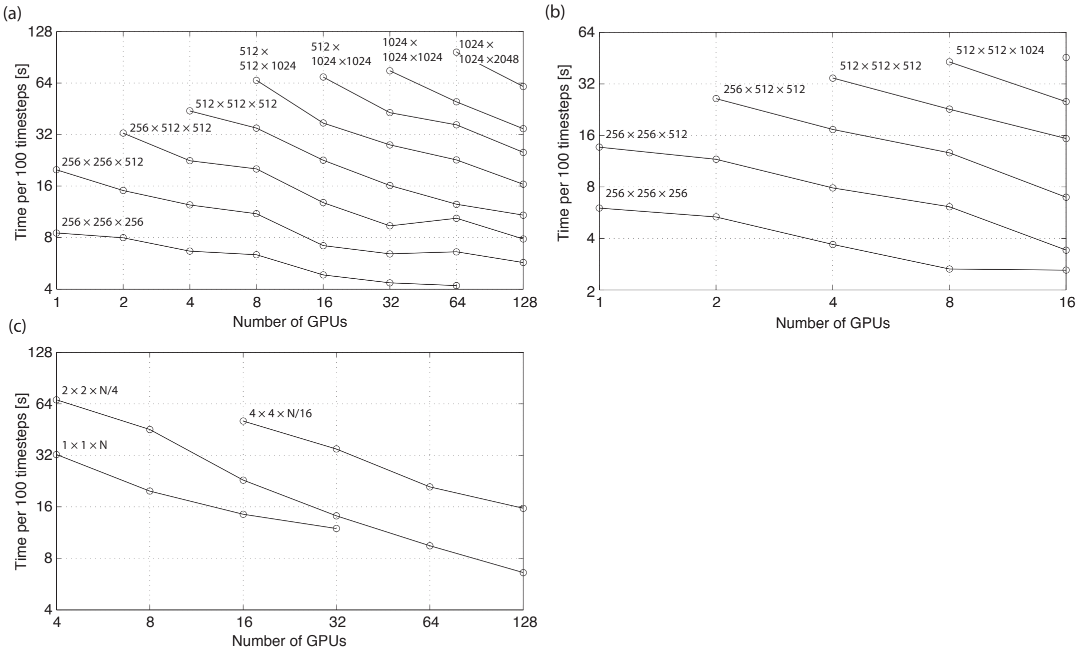


Figure 3. Strong scaling plots for (a) Emerald with 1-128 GPUs, and (b) Anselm with 1-16 GPUs. (c) The influence of different decomposition of the simulation domain of $256 \times 256 \times 2048$ on the strong scaling observed on Emerald. One-dimensional decomposition is compared with half and full 3D decomposition with 2, 11, and 26 neighbours, respectively

in 1D (2 GPUs), followed by 2D (4 GPUs) and 3D (8 GPUs) with only a single neighbour in each dimension. This situation will be referred to as half decomposition. For 16 and more GPUs, the decomposition turns into the full version with neighbours on both sides. This is done first in the x direction (16 GPUs), followed by the y direction (32 GPUs). The first complete full decomposition is employed for 64 GPUs, where the decomposition is $4 \times 4 \times 4$. The growing number of neighbours has a direct impact on the number of extraction/injection CUDA kernels, as well as the number of MPI communications to be performed.

2. On Emerald, the GPUs are packed into fat nodes sharing PCI-E links and the network adapter. The situation is worse in the case of 8-GPU nodes, where pairs of GPU cards share 16 links and the PCI-E bandwidth is halved. Moreover, 4 GPUs are connected to a single CPU socket creating contention in RAM.
3. Since the amount of on-board memory is limited, this also limits the total domain sizes. In the finest decomposition, the local subdomain only contains $64 \times 64 \times 64$ grid points, which makes the calculation time very small compared to the communication time. Moreover, for such a small subdomain, the halo accounts for 70% of the grid points in the local subdomain. The situation improves as the size of the local subdomains is increased. Unfortunately, the biggest subdomain that fits into on-board memory is approximately $256 \times 256 \times 512$. In this case, the halo accounts for around 25% of the local grid points.

Finally, comparing both systems, Anselm reaches almost twice the performance of Emerald. This is due to the combined effects of newer GPUs with higher performance and on-board memory bandwidth, ECC being switched off, and only a single GPU per node.

4.3. Decomposition comparison

Next, the influence of the domain decomposition shape on strong scaling was investigated on Emerald, using a global domain size of 256 × 256 × 2048, a halo width of 16 grid points, and 4 to 128 GPUs. A 1D decomposition was used, where the domain is cut over the longest dimension into N partitions, with N corresponding to the number of GPUs. For comparison, a half 3D decomposition with 11 neighbours cut into $2 \times 2 \times N/4$, and a full 3D decomposition with 26 neighbours cut into $4 \times 4 \times N/16$ were also tested. As shown in fig. 3(c), the impact on performance is significant. The additional communication overhead arising from using a higher dimensionality decomposition with an increased number of neighbours directly translates into a performance drop. An obvious conclusion is that wherever possible, a 1D decomposition is preferred. When this is not possible due to an unacceptable level of numerical accuracy (the numerical error scales with the number of cuts along an axis as discussed in Sec. 2.3) or the subdomains being excessively small, a half 3D decomposition is preferred. Compared to the full 3D decomposition, the half decomposition reduces the simulation time by a factor of at least 2, which correlates with the reduced number of neighbours.

4.4. Simulation time breakdown

Next, the composition of the simulation time was investigated. Only the simulation loop was considered, as the initialisation, pre-processing, and post-processing phases usually take on the order of minutes, while realistic simulations may run for many hours. Figure 4 shows the simulation time breakdown for a domain size of 256 × 256 × 1024, a halo width of 16 grid points, and 1D domain decomposition executed on Emerald and Anselm with 2 to 16 GPUs. Apart from faster execution on Anselm (due to faster GPUs), a difference in the PCI-E and MPI overhead may be observed. Although not clearly visible, the PCI-E latency on Emerald is almost 25% higher due to main memory congestion and shared PCI-E links. In addition, a higher MPI latency on Emerald is clearly noticeable. This is because Emerald only supports half-duplex, which decreases the MPI bandwidth by a factor of two. Furthermore, all inter-node communications are done via the main memory, which becomes the ultimate bottleneck. For the highest number of GPUs, where the local domain size is 256 × 256 × 64, the percentage of time spent performing calculations, communications using PCI-E, and communications using MPI is 32%, 10%, and 58% for Emerald, and 40%, 17%, and 43% for Anselm, respectively. In comparison, previous implementations using global domain decomposition have reported the time spent performing calculations is below 1% on a GPU cluster [17], and 30% on a CPU cluster [13].

4.5. Influence of halo width

The influence of halo width (overlap size) on the communication overhead was investigated on Emerald using a simulation size of 512^3 grid points partitioned over all three dimensions with a halo width of 8, 16, or 32 grid points. The simulations were executed on 4 to 128 GPUs, with the time break down shown in fig. 5. The increase in the PCI-E and MPI overhead when the halo becomes larger is evident. When the halo size is 8 grid points, there is only a small overhead. However, when the overlap is increased to 32 grid points, the communication overhead prevents the code from scaling beyond 16 GPUs. Although the PCI-E latency remains at promising levels and scales with the number of GPUs, the MPI latency is the ultimate bottleneck. The

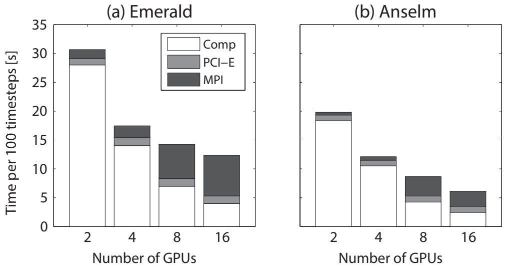


Figure 4. Breakdown of the execution time for a simulation domain size of 256 × 256 × 1024 comprising of the computation part, MPI transfers between nodes, and PCI-E transfers between CPU and GPU

slight increase in the computation time across the three overlap sizes is due to the increase in local subdomain size with the halo size. The rise in MPI overhead as the number of GPUs is increased between 4 and 16 can be attributed to the transition from half 3D decomposition to full 3D decomposition. A halo width of 16 grid points was ultimately chosen as an acceptable compromise between performance and accuracy.

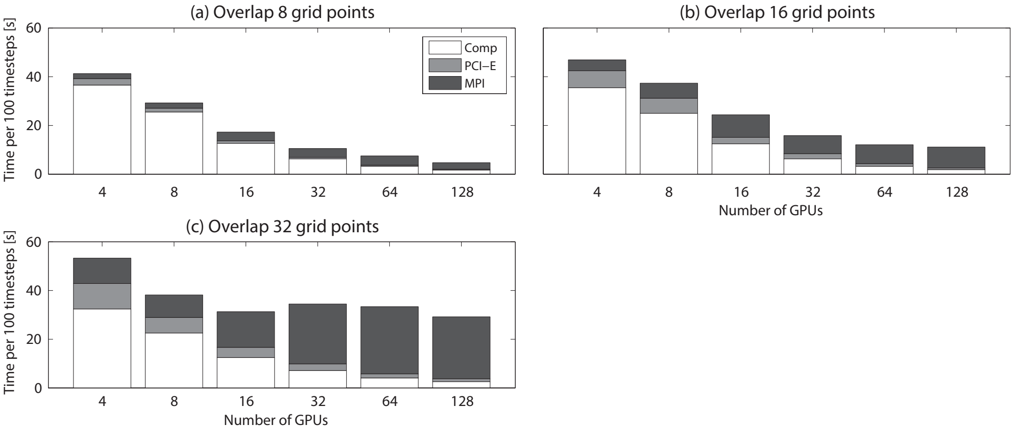


Figure 5. The influence of the overlap (halo) width on the overhead comprising of MPI and PCI-E transfers. The investigation was conducted on Emerald using a simulation size of 512^3, a halo width of 8, 16 and 32, and between 4 and 128 GPUs

4.6. Comparison of GPU and CPU

Finally, the GPU implementation using local Fourier decomposition was compared with an existing CPU implementation using global decomposition [13]. Several benchmark simulations were performed on Emerald and Anselm using three domain sizes (256^3, 512^3 and 1024^3) as shown in fig. 6. Here the horizontal axis corresponds to either the number of GPUs, or the number of CPU nodes (each of which integrates 16 processor cores). This grouping is used to estimate the simulation cost, which is charged per node on Anselm, regardless of whether the GPU is used. In comparison, Emerald has a different charge policy, with higher prices charged per GPU.

Figure 6(a) reveals that for small domain sizes, Anselm's GPUs are much faster than the CPU implementation when the number of nodes employed is small. Here, the simulation cost

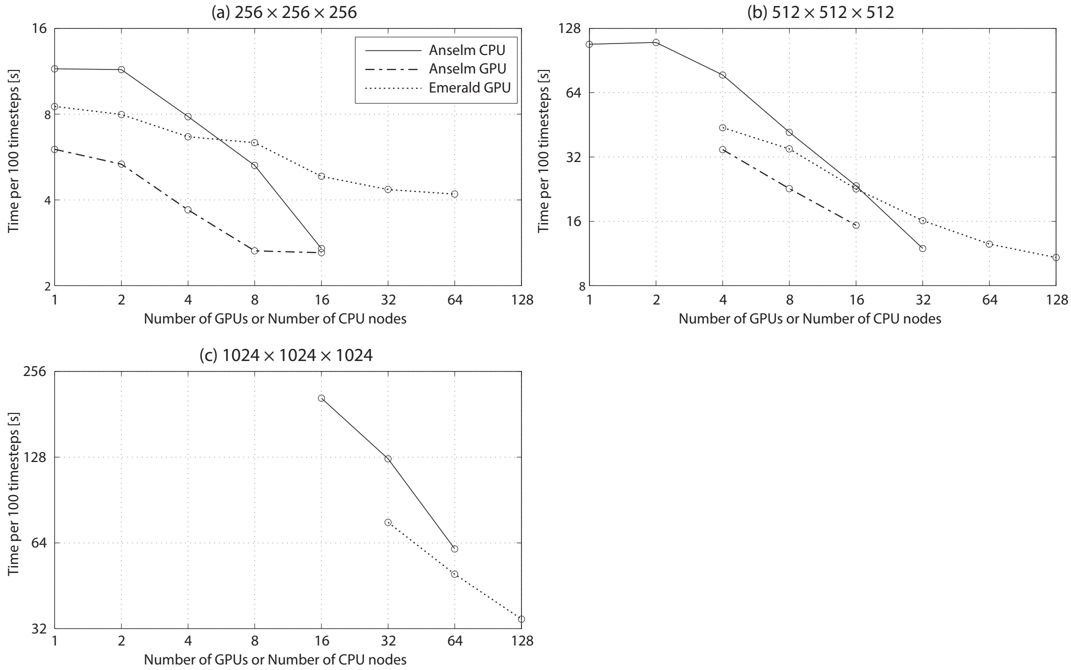


Figure 6. Performance comparison between local Fourier basis decomposition running on GPUs (Emerald and Anselm) and global domain decomposition running on CPUs (Anselm). In the case of the CPU implementation, the number of GPUs translates to the number of nodes, each of which contains 16 CPU cores

can be significantly reduced by utilising GPUs. The compute times for Anselm's GPUs and CPUs meet at 16 GPUs/nodes, where the local subdomains are extremely small. Emerald's GPUs seem to be too slow for such a small domain, where the communication is dominant. Figure 6(b) shows the same results for a larger domain size. Here, Anselm's GPUs outperform the comparable number of CPU cores. When the number of nodes is doubled (32 nodes or 512 CPU cores against 16 GPUs), the CPU cluster is faster by a factor of 1.28, however, for a doubled price. Emerald's GPUs beat Anselm when all 128 GPUs are used in the computation. The last benchmark shown in fig. 6(c) illustrates the benefits of the GPU implementation for larger domain sizes. Here 128 GPUs are faster than a cluster of 1024 CPU cores in 64 nodes by a factor of 1.76. Considering these 128 GPUs could be packed in 16 GPU nodes, this is a significant result. Note, the CPU code is limited by 1D slab decomposition so it is not possible to employ more cores than 256, 512 and 1024 for the domain sizes tested.

4.7. Production simulation

To show the impact of the proposed multi-GPU implementation on the type of ultrasound simulations used for treatment planning in focused ultrasound surgery, a comparison is given of the execution time and the financial aspects of running a single simulation using a grid size of 1536 × 1024 × 2048 with 48,000 time steps performed as a part of the characterisation of a high-intensity focused ultrasound (HIFU) transducer used to treat prostate cancer. The output of such a simulation is given in fig. 7, which shows the maximum steady state acoustic pressure in a 2D plane in front of the transducer. Table 1 illustrates that a cluster of 128 GPUs is able to deliver the simulation result in 9 hours and 29 minutes for the price of 426 USD (calculated based on the Emerald charge rate of 35.13c per GPU hour). Comparing this with the best price performance on the CPU cluster, the simulation can be completed in 3 days for 1,623 USD, or 2 days and 5 hours for 2,395 USD (calculated based on the Anselm charge rate of 8.8c per core hour). If price is the primary concern, as many tens of simulations are usually performed during any particular study, the GPU cluster can reduce the simulation time by a factor of 7.5 and the simulation cost by a factor of 3.8. The ultimate conclusion is that a GPU cluster is much a better solution for this type of simulation.

Table 1. Simulation time and cost when running a production simulation on Emerald with 96 and 128 GPUs, or Anselm with 128, 256, 512 CPU cores.

	Simulation Time	Simulation Cost
96 GPUs	14h 9m	\$475
128 GPUs	9h 29m	\$426
128 CPU cores	6d 18h	\$1,826
256 CPU cores	3d 0h	\$1,623
512 CPU cores	2d 5h	\$2,395

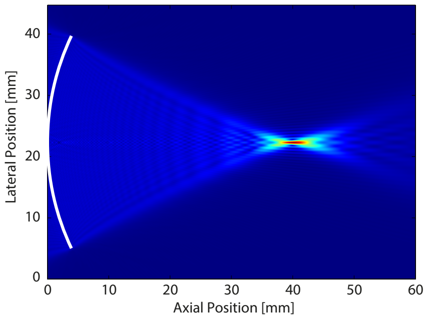


Figure 7. Pressure field from a prostate ultrasound transducer simulated using a domain size of $1536 \times 1024 \times 2048$ grid points ($45 \times 30 \times 60$ mm) with 48,000 time steps (60 μs) calculated in 9 hours and 29 minutes on 128 GPUs

Conclusion

This paper has presented a novel multi-GPU implementation of the Fourier spectral method using domain decomposition based on local Fourier basis [19]. The fundamental idea behind this work is the replacement of the global all-to-all communications introduced by the FFT (used to calculate spatial derivatives) by direct neighbour exchanges. By doing so, the communication burden can be significantly reduced at the expense of a slight reduction in numerical accuracy. The accuracy is shown to be dependent on the overlap (halo) size and independent on the local domain size. And to increase linearly with the number of domain cuts an acoustic wave must traverse. For an overlap (halo) size of 16 grid points, the error is on the order of 10^{-3}, which is comparable to the error introduced by the PML. Consequently, the level of parallelism achievable in practice is not limited by the reduction in accuracy due to the use of local Fourier basis.

Strong scaling results demonstrate that the code scales with reasonable parallel efficiency, reaching 50% for large simulation domain sizes. However, the small amount of on-board memory ultimately limits the global domain size for a given number of GPUs. 1D decomposition is shown to be the most efficient unless the local subdomain becomes too thin. Beyond, it is useful to exploit 2D or half 3D decomposition with only a single neighbour in a given direction to limit the number of MPI transfers. An overlap size of 16 grid points is shown to be a good trade off between speed and accuracy, with larger overlaps becoming impractical due to the overhead imposed by large MPI transfers. Compared to the CPU implementation using global domain decomposition, the GPU version is always faster for an equivalent number of nodes. For production simulations executed as part of ultrasound treatment planning, the GPU implementation reduces the simulation time by a factor of 7.5 and the simulation cost by a factor of 3.8. This is a promising result, given the GPUs utilized are now almost decommissioned.

In future, the code will be extended to model nonlinear wave propagation in heterogeneous media, as considered in [2]. The implementation could also be further improved by exploiting additional opportunities for overlapping communication and computation. First, the PCI-E and MPI communication could be overlapped. Second, the possibility of peer-to-peer communication among GPUs within the same node could be explored. This feature has the potential to eliminate expensive intra-node MPI communications. Third, the CPU could be utilized for additional executions, for example, assigning a subdomain to the idle CPU cores. Finally, multiple subdomains of different sizes could be executed on a single GPU, which might allow the communication on one subdomain to be overlapped while performing calculations on the others.

The project is financed from the SoMoPro II programme. The research leading to this invention has acquired a financial grant from the People Programme (Marie Curie action) of the Seventh Framework Programme of EU according to the REA Grant Agreement No. 291782. The research is further co-financed by the South-Moravian Region. This work reflects only the author's view and the European Union is not liable for any use that may be made of the information contained therein. This work was also supported by the research project "Architecture of parallel and embedded computer systems" Brno University of Technology (FIT-S-14-2297, 2014-2016), the Engineering and Physical Sciences Research Council, United Kingdom (grant numbers EP/L020262/1 and EP/M011119/1), and the Ministry of Education, Youth and Sports, Czech Republic (Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center - LM2015070"). The work presented here made use of Emerald, a GPU-accelerated High Performance Computer, made available by the Science & Engineering South Consortium operated in partnership with the STFC Rutherford-Appleton Laboratory. The authors would like to thank Alistair Rendell and Beau Johnston for useful contributions to early versions of the k-Wave GPU kernels.

References

1. G. F. Pinton, J. Dahl, S. Rosenzweig, and G. E. Trahey, "A heterogeneous nonlinear attenuating full-wave model of ultrasound," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 56, no. 3, pp. 474–488, 2009. DOI: 10.1109/TUFFC.2009.1066.

2. J. Jaros, A. P. Rendell, and B. E. Treeby, "Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound," *Int. J. High Perf. Comput. Appl.*, vol. 30, no. 2, pp. 137–155, 2016.

3. J. Gu and Y. Jing, "Modeling of wave propagation for medical ultrasound: a review," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 62, no. 11, pp. 1979–1992, 2015.

4. K. Okita, R. Narumi, T. Azuma, S. Takagi, and Y. Matumoto, "The role of numerical simulation for the development of an advanced HIFU system," *Comput. Mech.*, vol. 54, no. 4, pp. 1023–1033, 2014.

5. J. P. Boyd, *Chebyshev and Fourier Spectral Methods.* Mineola, New York: Dover Publications, 2001.

6. N. N. Bojarski, "The k-space formulation of the scattering problem in the time domain," *J. Acoust. Soc. Am.*, vol. 72, no. 2, pp. 570–584, 1982.

7. T. D. Mast, L. P. Souriau, D. L. Liu, M. Tabei, A. I. Nachman, and R. C. Waag, "A k-space method for large-scale models of wave propagation in tissue.," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 48, no. 2, pp. 341–354, 2001.

8. M. Tabei, T. D. Mast, and R. C. Waag, "A k-space method for coupled first-order acoustic propagation equations," *J. Acoust. Soc. Am.*, vol. 111, no. 1, pp. 53–63, 2002.

9. B. E. Treeby, J. Jaros, A. P. Rendell, and B. T. Cox, "Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method," *J. Acoust. Soc. Am.*, vol. 131, no. 6, pp. 4324–4336, 2012.

10. M. I. Daoud and J. C. Lacefield, "Distributed three-dimensional simulation of B-mode ultrasound imaging using a first-order k-space method.," *Phys. Med. Biol.*, vol. 54, no. 17, pp. 5173–5192, 2009.

11. J. C. Tillett, M. I. Daoud, J. C. Lacefield, and R. C. Waag, "A k-space method for acoustic propagation using coupled first-order equations in three dimensions," *J. Acoust. Soc. Am.*, vol. 126, no. 3, pp. 1231–1244, 2009.

12. J.-L. Vay, I. Haber, and B. B. Godfrey, "A domain decomposition method for pseudo-spectral electromagnetic simulations of plasmas," *J. Comput. Phys.*, vol. 243, pp. 260–268, 2013.

13. J. Jaros, V. Nikl, and B. E. Treeby, "Large-scale Ultrasound Simulations Using the Hybrid OpenMP/MPI Decomposition," in *Proceedings of the 3rd International Conference on Exascale Applications and Software*, pp. 115–119, Association for Computing Machinery, 2015.

14. M. Pippig, "PFFT-An extension of FFTW to massively parallel architectures," *SIAM J. Sci. Comput.*, vol. 35, no. 3, pp. C213–C236, 2013.

15. M. Frigo and S. G. Johnson, “The Design and Implementation of FFTW3,” *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005.

16. D. Pekurovsky, “P3DFFT: A Framework for Parallel Computations of Fourier Transforms in Three Dimensions,” *SIAM J. Sci. Comput.*, vol. 34, no. 4, pp. C192–C209, 2012.

17. A. Gholami, J. Hill, D. Malhotra, and G. Biros, “AccFFT: A library for distributed-memory FFT on CPU and GPU architectures,” *arXiv*, p. arXiv:1506.07933, 2015.

18. K. Czechowski, C. Battaglino, C. McClanahan, and K. Iyer, “On the Communication Complexity of 3D FFTs and its Implications for Exascale,” in *Proceedings of International Supercomputing Conference*, ACM, 2012. DOI: 10.1145/2304576.2304604.

19. M. Israeli, L. Vozovoi, and A. Averbuch, “Spectral multidomain technique with local Fourier basis,” *J. Sci. Comput.*, vol. 8, no. 2, pp. 135–149, 1993.

20. J. P. Boyd, “Asymptotic fourier coefficients for a C^{∞} bell (smoothed-”top-hat”) & the Fourier extension problem,” *J. Sci. Comput.*, vol. 29, no. 1, pp. 1–24, 2005. DOI: 10.1007/s10915-005-9010-7.

21. M. Ding and K. Chen, “Staggered-grid PSTD on local Fourier basis and its applications to surface tissue modeling.,” *Optics Exp.*, vol. 18, no. 9, pp. 9236–9250, 2010.

22. M. Garbey and D. Tromeur-Dervout, “Parallel Algorithms with Local Fourier Basis,” *J. Comp. Phys.*, vol. 173, pp. 575–599, 2001.

23. A. D. Pierce, *Acoustics: An Introduction to its Physical Principles and Applications.* New York: Acoustical Society of America, 1989.

24. J.-P. Berenger, “Three-dimensional perfectly matched layer for the absorption of electromagnetic waves,” *J. Comput. Phys.*, vol. 127, no. 2, pp. 363–379, 1996.

25. J. P. Boyd, “A Comparison of Numerical Algorithms for Fourier Extension of the First, Second, and Third Kinds,” *J. Comput. Phys.*, vol. 178, no. 1, pp. 118–160, 2002.

26. M. J. Quinn, *Parallel Programming in C with MPI and OpenMP.* McGraw-Hill Education Group, 2003.

27. HPC Advisory Council, “Interconnect Analysis: 10GigE and InfiniBand in High Performance Computing,” tech. rep., HPC Advisory Council, 2009.

28. B. E. Treeby and B. T. Cox, “A k-space Greens function solution for acoustic initial value problems in homogeneous media with power law absorption,” *J. Acoust. Soc. Am.*, vol. 129, no. 6, pp. 3652–3660, 2011.

29. J. L. Robertson, B. T. Cox, and B. E. Treeby, “Quantifying numerical errors in the simulation of transcranial ultrasound using pseudospectral methods,” in *IEEE Int. Ultrason. Symp.*, pp. 2000–2003, 2014.

30. NVIDIA, “CUDA Toolkit Documentation v7.5,” tech. rep., NVIDIA, 2015.

31. NVIDIA, “cuFFT Library User’s Guide,” tech. rep., NVIDIA, 2015.

Appendix C

Applications in Cancer Treatment

C.1 Focused Ultrasound Waves in Heterogeneous Tissue

Treeby B. E., **Jaros, J.**, Cox, B. T.: Focused ultrasound waves in heterogeneous tissue. In *High Intensity Focused Ultrasound Therapy: Fundamentals through Clinical Challenges*. 2016. ISBN 978-3-319-14261-6. page 24, in print.

Focused ultrasound waves in heterogeneous tissue

Bradley Treeby[1*], *Jiri Jaros*[2,1], *Ben Cox*[1]

[1] Biomedical Ultrasound Group, Department of Medical Physics and Biomedical Engineering, University College London, Gower Street, London WC1E 6BT, United Kingdom

[2] Supercomputing Group, Faculty of Information Technology, Brno University of Technology, Bozetechova 2, 612 66 Brno, Czech Republic

*corresponding author: b.treeby@ucl.ac.uk

ABSTRACT

When focused ultrasound waves propagate through the human body, the combined effects of nonlinearity, absorption, refraction, and scattering can significantly alter the position and shape of the focal volume compared to the response in water. This can affect the desired clinical outcome, and lead to adverse events such as near-field heating and skin burns. In this chapter, these wave phenomena are discussed and their respective importance is demonstrated through numerical simulations based on a generalized Westervelt equation. Simple analytical formulae for the response in reference media are also provided. Both homogeneous and heterogeneous media are considered, including sonications of the kidney and liver through the muscles surrounding the abdominal cavity. The effect of frequency selection on the focal intensity, volume, and heating rate is also discussed.

Index terms: frequency selection; beam distortion; treatment planning; simulation; scattering; absorption; refraction

4.1. INTRODUCTION

In simple reference media such as deionized water, the behavior of focused ultrasound waves is very well understood. In this case, the governing equations introduced in Chapter 1 accurately describe the propagation and nonlinear distortion of ultrasound waves, and analytical and numerical predictions based on these equations have been extensively validated by experimental measurements (see Chapter 2). However, when focused ultrasound is used clinically, the propagation medium is the human body, which is considerably more complex. In particular, biological tissue is heterogeneous, and material properties such as the sound speed and mass density vary with position according to the local tissue constituents. This gives rise to scattering and refraction phenomena that distort the shape of the ultrasound beam. Tissue is also strongly absorbing compared to water, with absorption over the megahertz range increasing with frequency according to a power law. Thus, while a given transducer can be carefully characterized in the laboratory in line with established international standards (see Chapter 16), the equivalent response *in vivo* can be very difficult to predict. This is compounded by the difficulty in performing measurements of the acoustic field parameters inside the body.

One notable advantage of therapies that induce a thermal response, such as high-intensity focused ultrasound (HIFU), is that changes in temperature inside the body can be monitored in real-time using thermometry (see Chapters 13 and 14). However, while temperature rise and thermal dose are directly relevant to the clinical outcome (see Chapter 5), temperature measurements provide only a weak surrogate for the corresponding acoustic field parameters and the antecedent wave phenomena. For example, if there is distortion or displacement of the focal spot observed on thermometry, it is difficult to infer what aspect of the patient anatomy is the underlying cause, or how it might be rectified. Similarly, if there is no radiological evidence of tissue ablation after treatment, it is very challenging to relate this to a particular cause, for example, beam distortion, excessive absorption along the beam path, lack of absorption in the focal region, changes in the transducer output, or any number of other possibilities.

In lieu of more direct methods to experimentally characterize ultrasound fields inside the body, *in silico* investigations of wave propagation using computational models can be used. With recent advances in numerical methods and computer hardware, detailed simulations that accurately capture the relevant physical behavior of focused ultrasound waves in heterogeneous tissue are now possible. This provides a very powerful tool that can be leveraged for a range of tasks, including patient selection (determining whether a patient is a good candidate for a particular procedure based on their individual anatomy), treatment verification (determining the cause of adverse events or treatment failures), and model-based treatment planning (determining the best transducer position and sonication parameters to deliver the ultrasound energy to the planning target volume).

In this chapter, numerical simulations are used to demonstrate the effect of nonlinearity, absorption, transmit frequency, refraction, and scattering on the propagation of focused ultrasound waves in the human body. Due to the strong influence of the transducer characteristics and patient anatomy, it is somewhat difficult to make statements that are both quantitative and generalizable. Notwithstanding this challenge, the overall impact and relative importance of the underlying wave phenomena are discussed. The simulations are performed in 3D using the open-source k-Wave toolbox. This solves a generalized version of the Westervelt equation discussed in Chapter 1 that accounts for nonlinearity, heterogeneous material parameters, and acoustic absorption following a frequency power law [1]. Further details on the numerical model and simulation parameters are given in Appendix A.

4.2. FOCUSED ULTRASOUND FIELDS IN HOMOGENEOUS MEDIA

4.2.1 Transducer Model and Ideal Response

To examine the effects of tissue on the propagation of focused ultrasound waves, a representative single element bowl transducer is used as shown in Figure 4.1(a). The transducer is taken to have a radius of curvature r = 140 mm, circular aperture diameter d = 120 mm, and driving frequency f = 1 MHz. These values are within the range of nominal parameters for commercially available focused ultrasound systems used for abdominal treatments (see Chapters 11 and 12), for example, the Philips Sonalleve (r = 120 mm, d = 127 mm, f = 1.2–1.45 MHz), the Insightec Exablate (r = 160 mm, d = 120 mm, f = 0.95–1.35 MHz), and the HAIFU JC-200 (r = 145 mm, d = 200 mm, f = 0.95 MHz). It is assumed the transducer is electrically driven by a single frequency continuous wave sinusoid giving rise to an acoustic surface intensity of 1 W/cm^2 and an acoustic power of 119 W. This corresponds to a surface pressure of $p_0 = \sqrt{2I_0\rho_0c_0} = 175$ kPa and normal surface velocity of $u_0 = 115 \text{ mm/s}$, assuming the transducer is immersed in water (see Appendix B for a list of material parameters). These transducer parameters are used throughout this chapter unless otherwise noted, and only steady state conditions are considered. The pressure field generated by this transducer under linear conditions in water as predicted by numerical simulation is shown in Figure 4.1(b).

In the ideal case of a homogeneous medium under linear and lossless conditions (i.e., propagation in free-field with no nonlinearity or acoustic absorption), the steady state acoustic response of a bowl transducer driven by a single frequency sinusoid can be predicted using several simple formulae. It will be helpful to briefly revisit these formulae, both for the insight they provide into the underlying wave physics, and for reference with later more complex scenarios. Considering first the axial pressure, the amplitude of the acoustic pressure P along the beam axis z of the transducer is given by the O'Neil formula [2]

$$P(z) = 2\rho_0c_0u_0\left|\frac{\sin\left[k(B-z)/2\right]}{1-z/r}\right| . \tag{4.1}$$

Here ρ_0 is the ambient mass density, c_0 is the sound speed, $k = 2\pi f/c_0$ is the wavenumber, $B = \sqrt{(z-h)^2+(d/2)^2}$ is the distance from the edge of the bowl to the position z on the transducer axis, and $h = r - \sqrt{r^2-(d/2)^2}$ is the height of the bowl (the depth of the concave surface) as shown in Figure 4.1(a). This expression is derived under the assumptions of the Rayleigh integral, which are valid when the transducer diameter is large compared to both the transducer height and the acoustic wavelength [3]. For the transducer shown in Figure 4.1(a), $d/h \approx 9$ and $d/\lambda \approx 80$ so these assumptions hold. Equation 4.1 is plotted in Figure 4.1(c), along with the corresponding values from simulation, which show close agreement. The significant increase in pressure (and thus heating rate) in the focal region

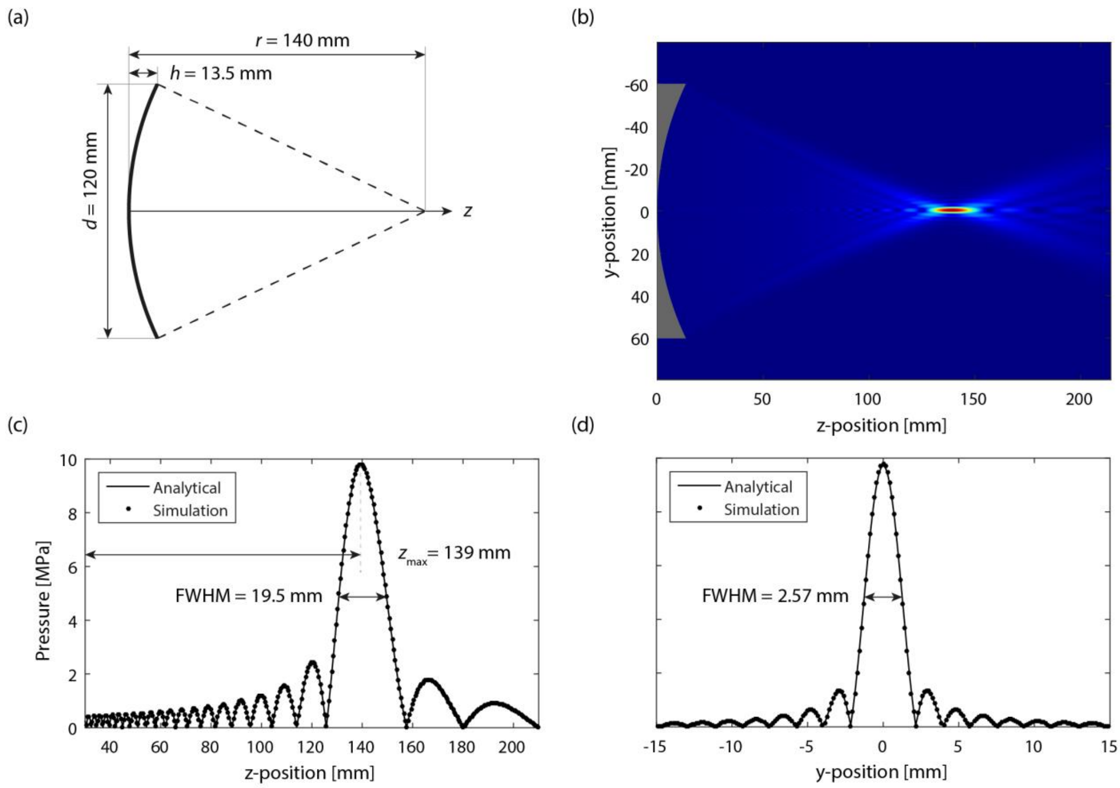


Figure 4.1: (a) Schematic of the concave spherical transducer geometry used to study the propagation of ultrasound waves in tissue. (b) 2D slice through the 3D simulated pressure field in a homogeneous and lossless medium assuming linear wave propagation for a surface intensity of 1 W/cm^2. (c) Axial pressure amplitude along the transducer beam axis as predicted by the O'Neil formula given in Equation 4.1 (solid line) and numerical simulation (dots). The position of the maximum pressure and the full-width at half maximum (FWHM) are also shown. (d) Lateral pressure amplitude through the geometric focus of the transducer as predicted by Equation 4.4 (solid line) and numerical simulation (dots).

compared to the surrounding regions is what allows HIFU to ablate tissue deep inside the body without harming intermediate tissue.

In the plane of the geometric focus (where $z = r$), the ratio of the on-axis pressure to the pressure at the transducer surface, called the *focusing gain*, is given by $G = kh$. This means the focal pressure can be increased by increasing either the transmit frequency or the curvature of the transducer (or equivalently, the transducer height relative to the wavelength). In practice, the axial position of the maximum pressure $z_{\max}$ occurs slightly before the geometric focus of the transducer. For $kh > \sim 4$, the position of maximum pressure and the corresponding focusing gain can be estimated using [2]

$$z_{\max} \approx r - \frac{12r}{12 + k^2h^2} \,, \qquad G_{\max} \approx kh + \frac{12}{kh} \,. \tag{4.2}$$

4

Using a paraxial approximation, where the diameter of the transducer is assumed to be small compared to the observation distance, the length of the focal region (sometimes called the *depth of field*) can also be estimated using

$$L \approx 9.68\,\lambda\left(\frac{r}{d}\right)^2 \ . \tag{4.3}$$

Here $\lambda = c_0/f$ is the wavelength, r/d is the *f*-number of the transducer, and the length is defined as the full-width at half maximum (FWHM) of the main lobe of axial pressure (which corresponds to a -6 dB drop in the maximum pressure). From Equation 4.3, the depth of field is proportional to the square of the *f*-number, and inversely proportional to frequency. Thus increasing the focusing gain by increasing the frequency or transducer height will also reduce the depth of field. To three significant figures, the focal position, focusing gain, and depth of field for the transducer in Figure 4.1(a) calculated directly from Equation 4.1 are 139 mm, 55.8, and 19.0 mm, respectively. In comparison, Equations 4.2 and 4.3 predict values of 139 mm, 55.9, and 20.1 mm. The small error (~6%) in the depth of field given by Equation 4.3 is due to the validity of the paraxial approximation for this transducer.

In the lateral direction y through the geometric focus (where $z = r$), the acoustic pressure amplitude P is given by [3]

$$P(y) = 2\rho_0 c_0 u_0 kh \left[\frac{J_1(kdy/2r)}{kdy/2r}\right] , \tag{4.4}$$

where $J_1(\ldots)$ is a first-order Bessel function of the first kind. This is plotted in Figure 4.1(d) along with the corresponding values from simulation, which show very good agreement. The corresponding lateral width of the focal region (sometimes called the lateral resolution or *beam width*), again defined as the FWHM or -6 dB width of the main lobe of the pressure amplitude, can be estimated using

$$\mathrm{BW} \approx 1.41\,\lambda\left(\frac{r}{d}\right) \ . \tag{4.5}$$

Consequently, increasing the frequency or transducer height will also reduce the beam width, analogous to the depth of field. For the transducer in Figure 4.1(a), the beam width calculated from both Equations 4.4 and 4.5 is 2.51 mm. Assuming the -6 dB focal volume is an ellipse, the volume can also be estimated as

$$V_{-6\,dB} \approx 10.1\,\lambda^3\left(\frac{r}{d}\right)^4 \ . \tag{4.6}$$

For the transducer in Figure 4.1(a), this gives a -6 dB focal volume of 66.2 mm^3. In comparison, the volume estimated using ellipse semi-axis lengths calculated from Equations 4.1 and 4.3 is 62.5 mm^3, and the volume calculated from simulation by counting the number of voxels with a pressure amplitude above 50% of the maximum pressure is 66.4 mm^3 (see Table 4.1).

4.2.2 Effect of Nonlinearity and Absorption

The first departure from the ideal conditions considered in Section 4.2.1 is the harmonic distortion of propagating ultrasound waves due to acoustic nonlinearity. As discussed in Chapter 1, at the acoustic pressure amplitudes used in high-intensity focused ultrasound, convective and material nonlinearities combine to cause the sound speed to depend on the local acoustic particle velocity. This means that during the compressional phase of the wave where the particle velocity is positive (i.e., the medium is being displaced in the same direction as the wave is travelling), the effective sound speed increases. Similarly, during the rarefactive phase of the wave, the particle velocity is negative and the sound speed decreases. This causes a cumulative distortion in the time domain waveform, which corresponds to the generation of higher frequency harmonics in the frequency domain. For the transducer and drive conditions used here (which are representative of those used clinically), the acoustic power is sufficient to generate significant nonlinear effects when the propagation medium is water.

The peak positive and negative pressure along the beam axis of the transducer in deionized water at body temperature when nonlinearity and absorption are included are shown in the left panel of Figure 4.2(a). The O'Neil solution from Equation 4.1 is also shown for reference. At the focus, the peak positive pressure under nonlinear conditions is more than double that under linear conditions. This occurs because some of the energy at the fundamental frequency is transferred to higher frequency harmonics which have a narrower axial and lateral spatial distribution. This is also evident in the evolution of the harmonics along the beam axis as shown in the left panel of Figure 4.2(b). The harmonic content is primarily restricted to the main focal lobe due to the strong focusing gain of the transducer. Consequently, outside this region, the response under linear and nonlinear conditions is very similar. The time domain pressure signal at the position of the maximum peak positive pressure is shown in the left panel of Figure 4.2(c). There is a considerable asymmetry in the waveform due to nonlinearity, with peak positive and negative pressures of 22.8 MPa and -6.32 MPa, respectively. There is also significant energy in the higher frequency harmonics (only the first ten harmonics are shown). The corresponding spatial peak temporal average (SPTA) intensity is 3260 W/cm^2.

The response shown in the left panel of Figure 4.2 is representative of what would be measured in a laboratory environment with the transducer driven at clinical levels in water (see Chapter 2). However, this differs considerably from the response that would be observed *in vivo* due to acoustic absorption. While values for the sound speed, density, and nonlinearity parameter in water are close to nominal values in soft biological tissue, the acoustic absorption in water is more than two orders of magnitude smaller. This means the effects of nonlinearity are significantly overestimated. The exact mechanisms for the additional absorption in tissue are complex, and occur at both the cellular level (e.g., viscous relative motion and thermal conduction) and the molecular level (e.g., molecular and chemical relaxation). Acoustic energy is also lost from the beam due to scattering. This is generally negligible at low megahertz frequencies but can become significant as the wavelength decreases. As discussed in Chapter 1, the overall acoustic attenuation in soft tissue (which includes both absorption and scattering) has been experimentally observed to follow a frequency power law. This means the acoustic energy transferred to

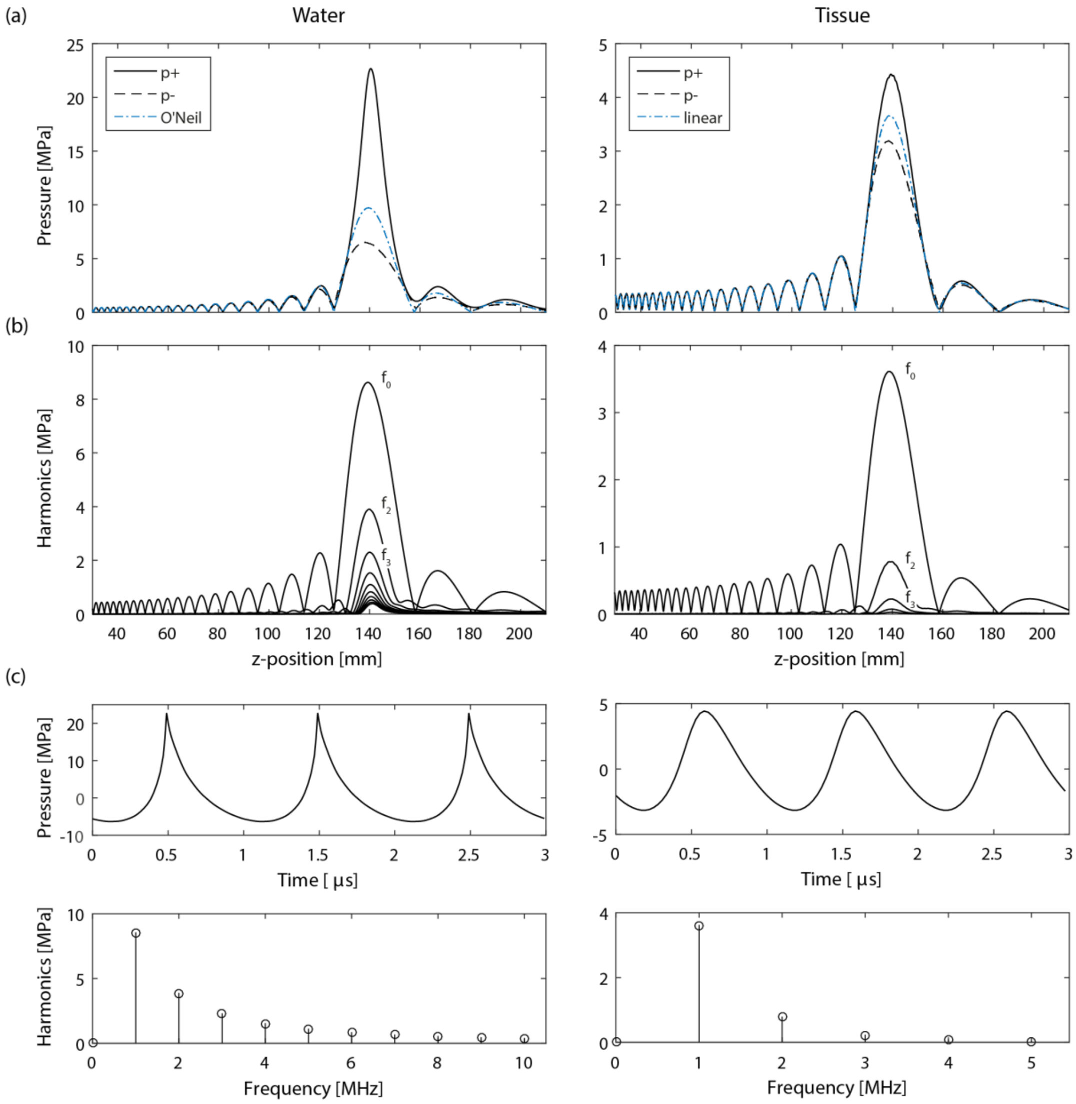


Figure 4.2: (a) Peak positive (p+) and peak negative (p-) pressure amplitude along the transducer beam axis for water (left panels) and tissue (right panels) when nonlinear effects and acoustic absorption are considered. (b) Evolution of the nonlinear harmonics along the beam axis. (c) Three cycles of the time domain acoustic pressure signal at the position of maximum peak positive pressure, and the corresponding amplitude spectrum showing the relative contribution of nonlinear harmonics. In water, only the first 10 harmonics are shown.

higher frequency harmonics is absorbed (i.e., converted to heat) more rapidly than energy at the fundamental frequency.

For comparison, the right panels of Figure 4.2 show the axial response of the transducer assuming a homogeneous medium with the acoustic properties of kidney (see Appendix B). At 1 MHz, the absorption in kidney is nearly 500 times greater than the absorption in water at body temperature. This has a significant effect on both the maximum pressure and the evolution of nonlinear harmonics. Compared to water, the maximum peak positive pressure is reduced by 80%. The response is still weakly nonlinear, with peak positive and negative pressures of 4.43 MPa and -3.16 MPa, respectively (see Table 4.1). However, there is now only significant energy up to the third or fourth harmonic. This is reflected in the peak positive pressure at the focus under nonlinear conditions, which is only 20% higher than under linear conditions (also accounting for absorption).

The effect of nonlinearity on heating can be quantified by examining the volume rate of heat deposition due to acoustic absorption. Near the focus of a bowl transducer, the wave field is approximately planar. In this case, the volume rate of heat deposition Q for a continuous wave excitation at driving frequency f is given by [4]

$$Q = 2\alpha I = \sum_{n=1}^{\infty} \frac{\alpha_n p_n^2}{\rho_0 c_0} = \sum_{n=1}^{\infty} \frac{\alpha_0 (n 2\pi f)^y P_n^2}{\rho_0 c_0} \ . \qquad (4.7)$$

Here I is the time-averaged acoustic intensity, $\alpha = \alpha_0 \omega^y$ is the frequency dependent acoustic absorption coefficient in Np m^{-1}, $\omega = 2\pi f$ is the radial frequency in rad/s, α_0 is the power law absorption pre-factor in Np (rad/s)$^{-y}$ m^{-1}, and P_n and α_n are the amplitude of the acoustic pressure and the absorption coefficient at the n^{th} harmonic, respectively. Examining Equation 4.7, for a power law exponent of $y = 1$, transferring acoustic energy to the n^{th} harmonic will give an n-fold increase in the volume rate of heat deposition compared to the same acoustic pressure at the fundamental frequency. Thus strongly nonlinear fields can generate a significant increase in the heating rate (see Chapter 7). For the weakly nonlinear field shown in the right panel of Figure 4.2, the nonlinearity enhances the volume rate of heat deposition at the focus by approximately 10%.

In clinical use, water or another coupling medium with low absorption is generally present between the ultrasound transducer and the patient (see Section 4.3). This means the total absorption along the beam path will be reduced compared to the results shown in the right panels of Figure 4.2. For typical clinical conditions used for the extracorporeal treatment of abdominal organs under MR-guidance (see Chapter 12), the wave propagation is moderately nonlinear, and the generation and absorption of higher frequency harmonics gives rise to a 15-30% increase in the volume rate of heat deposition at the focus compared to linear conditions [5], [6]. In the strongly nonlinear case (discussed in Chapter 7), the generation of shock waves can give a ten-fold enhancement in the volume rate of heat deposition [7], [8].

4.2.3 Effect of Frequency

In addition to the transducer geometry and tissue properties, the characteristics of the acoustic field and the resultant thermal energy deposition also depend strongly on frequency [9]. For HIFU therapy, the choice of optimal driving frequency is influenced by several factors [10], including:

1. **Focusing gain:** In a lossless medium, the focusing gain for a bowl transducer increases approximately linearly with frequency (see Equation 4.2). Thus for the same bowl geometry and surface pressure, using a higher frequency will increase the focal pressure.
2. **Focal size:** In a lossless medium, the focal depth of field and beam width are both inversely proportional to frequency, with the focal volume proportional to f^{-3} (see Equations 4.3, 4.5, and 4.6). Considering the trade-off between focusing gain and focal volume, using higher frequencies will increase the focusing gain and focal heating rate (neglecting pre-focal absorption), which will reduce the sonication time needed to achieve a given peak temperature. However, the ablated lesion volume will be smaller, and thus more sonications must be used to cover the same treatment volume [11].
3. **Nonlinearity:** The energy transferred to higher frequency harmonics due to acoustic nonlinearity increases with both amplitude and frequency. For a progressive plane wave, the shock formation distance is inversely proportional to frequency (see Chapter 1). Thus using higher driving frequencies can enhance nonlinear effects which provides an effective increase in focusing gain and reduction in focal size. The acoustic energy at higher harmonics is also absorbed more rapidly, which leads to nonlinear enhanced heating.
4. **Focal and pre-focal absorption:** The increased absorption at higher frequencies means that for a given focal intensity, the heating rate will increase with driving frequency. However, increased absorption along the beam path will also act to reduce the focal intensity, and thus a balance must be struck. Using higher frequencies also increases the risk of pre-focal heating, particularly in the skin and subcutaneous fat layers which have a high absorption coefficient. Avoiding the accumulation of pre-focal heating while minimizing treatment times remains a key challenge for clinical HIFU procedures.
5. **Cavitation:** The acoustic pressure required to induce cavitation activity increases with frequency (see Chapter 6). Thus for thermal therapies where it is desirable to avoid cavitation, using higher driving frequencies will reduce the risk of cavitation.
6. **Scattering:** The contribution of scattering to attenuation increases with frequency, although in soft tissue it generally only becomes significant when frequencies reach the tens of megahertz (for Rayleigh scatterers, scattering scales with f^4). Increased scattering means that additional energy is lost along the beam path with no associated increase in focal heating. The transmission through layered media also depends on frequency. For example, in transcranial therapies, transmission through the skull is generally increased at lower driving frequencies.
7. **Refraction:** While refraction is a frequency independent phenomenon, at higher frequencies where the wavelength and focal size are small, even slight beam distortions can disrupt the constructive interference that occurs at the focus, and significantly reduce the focal pressure.
8. **Transducer output:** For a given transducer design, changing the driving frequency can have a significant effect on the transduction efficiency and acoustic output power. For multi-element arrays, changing the driving frequency can also affect the formation of grating lobes due to the change in wavelength relative to the inter-element spacing.

Given the complexity and interdependence of these factors, experimental measurements or numerical simulations are typically required to devise the precise optimal frequency for a given HIFU application. However, if only focusing gain, focus size, and absorption are considered, it is possible to derive simple analytical formulae to provide useful estimates of the optimal frequency. For a bowl-shaped transducer in a homogeneous medium, the pressure focusing gain is proportional to radial frequency $\omega = 2\pi f$ (Equation 4.2), and the intensity focusing gain is proportional to ω^2. The corresponding intensity absorption along the beam path is given by $e^{-2\alpha_0\omega^y r}$, where r is the transducer radius of curvature (see Figure 4.1). Consequently, the intensity at the focus is proportional to

$$I_{spta} \propto \omega^2 e^{-2\alpha_0\omega^y r} \ . \tag{4.8}$$

The relationship between focal intensity and heating rate is given by Equation 4.7. In the linear case, this leads to [10], [12]

$$Q_{sp} \propto \omega^y I_{spta} \propto \omega^{y+2} e^{-2\alpha_0\omega^y r} \ , \tag{4.9}$$

where Q_{sp} is the spatial peak (i.e., focal) volume rate of heat deposition, and ω^y is the frequency dependence of the acoustic absorption. Considering the total power absorbed within the beam, the size of the -6 dB focal volume is proportional to ω^{-3} (see Equation 4.6), and the acoustic intensity averaged over the focal volume is directly proportional to the spatial peak intensity [13], which gives

$$W_{-6dB,volume} \propto \omega^{-3} Q_{sp} \propto \omega^{y-1} e^{-2\alpha_0\omega^y r} \ . \tag{4.10}$$

Similarly the size of the -6 dB (half-pressure maximum) focal area in the focal plane is proportional to ω^{-2}, which leads to [12]

$$W_{-6dB,area} \propto \omega^{-2} Q_{sp} \propto \omega^{y} e^{-2\alpha_0\omega^y r} \ . \tag{4.11}$$

For each of Equations 4.8 to 4.11, the optimum frequency (i.e., the frequency that maximizes the corresponding parameter) can be found by taking the derivative with respect to ω, equating to zero, and then solving for ω. This yields

$$\omega_{\text{optimal}} = \left(\frac{m}{2\alpha_0 y r}\right)^{\frac{1}{y}}, \quad \text{where } m = \begin{cases} 2 & \text{to optimise } I_{spta} \\ y+2 & \text{to optimise } Q_{sp} \\ y-1 & \text{to optimise } W_{-6dB,\text{ volume}} \\ y & \text{to optimise } W_{-6dB,\text{ area}} \end{cases} \tag{4.12}$$

For heterogeneous tissue, α_0 in Equations 4.8 to 4.12 can be replaced with the average value along the beam path [12].

Figure 4.3(a)-(c) shows the normalized change in intensity, heating rate, and power as a function of frequency calculated using Equations 4.8 to 4.11. In this case, the material properties are set to those of kidney ($\alpha_0 = 2.46 \times 10^{-7}$ Np (rad/s)$^{-y}$ m^{-1} and $y = 1.1$; see Appendix B). Normalized values calculated from numerical simulation including the effects of nonlinearity are also shown for reference, and show

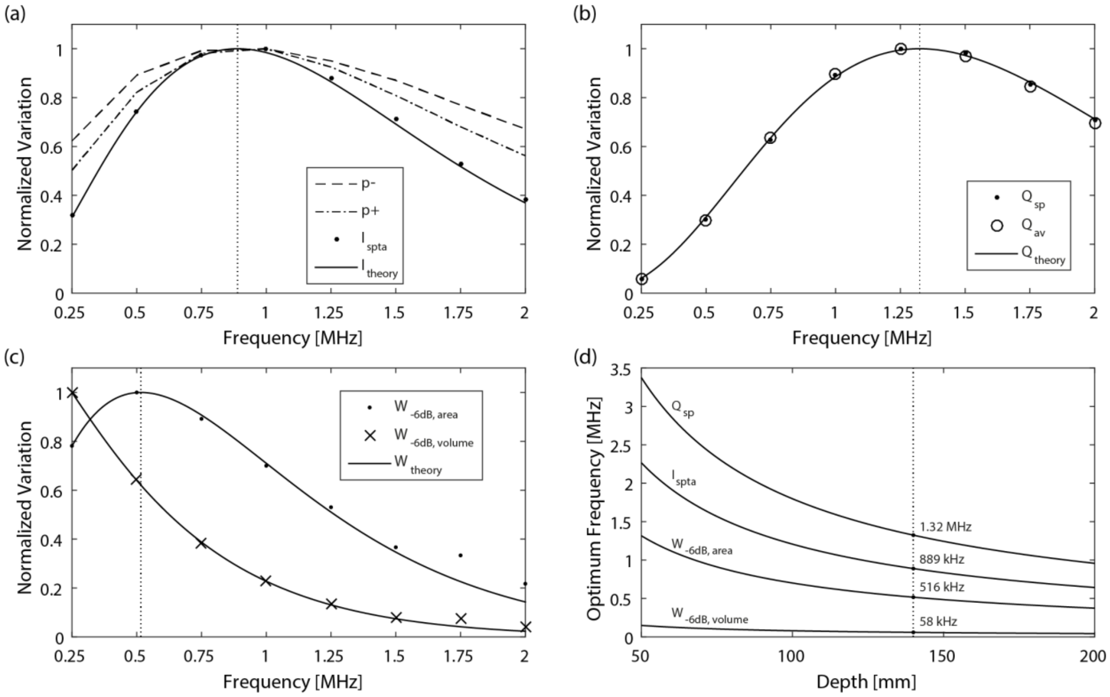


Figure 4.3: (a) Normalized variation in the peak negative pressure, peak positive pressure, and spatial peak temporal average (SPTA) intensity with frequency for the focused bowl shown in Figure 4.1 calculated from simulation. The theoretical variation in intensity from Equation 4.8 is also shown for reference (solid line). The optimal frequency which maximises the intensity is denoted with a vertical dashed line. (b) Normalized variation in the spatial peak and average volume rate of heat deposition with frequency calculated from simulation. The theoretical variation from Equation 4.9 and the optimal frequency are also shown. (c) Normalized variation in the total power deposited over the -6 dB focal volume and focal plane calculated from simulation. The theoretical variation from Equations 4.10 and 4.11 and the optimal frequency are also shown. (d) Optimal frequency as a function of focal depth calculated using Equation 4.12. The optimal frequencies for a focal depth of 140 mm are also shown.

good agreement. The optimal frequencies for each parameter calculated using Equation 4.12 are shown with vertical dotted lines. Figure 4.3(d) also shows the change in the optimal frequencies with depth (i.e., transducer radius of curvature), along with the optimal values for $r = 140$ mm.

Each of the four field parameters shown in Figure 4.3 are maximized at different optimal frequencies. The focal intensity is maximized at 889 kHz, while the focal heating rate is maximized at a slightly higher frequency of 1.32 MHz. The difference in the latter is due to the additional dependence of the heat deposition on absorption (which increases with frequency). The subsequent decline in heating rate at higher frequencies is because a further increase in focusing gain and absorption coefficient are not sufficient to counteract the reduction in focal intensity due to additional absorption along the beam path. In addition to the focal heating, which plays an important role in the initiation of a thermal lesion, the total power absorbed within the beam is also of interest. In contrast to the focal parameters, the total power absorbed over the -6 dB focal volume is maximized at 58 kHz. This parameter is strongly influenced by the large size of the focal volume at lower frequencies (the focal volume at 58 kHz is ~5000 times larger than at 1 MHz). However, large focal volumes can have a significant impact on treatment specificity. Moreover, at very low frequencies, the focusing gain and focal heating rate are also very low, which can result in unacceptably long treatment times to achieve thermal ablation (see Chapter 5). A better correlate for lesioning effectiveness is the total power deposited over the -6 dB focal area [13], which is maximized at 516 kHz.

Discounting $W_{-6dB,\,\mathrm{volume}}$, the optimal frequencies derived using Equation 4.12 span the range of frequencies used by the clinical HIFU devices discussed in Section 4.2.1. In addition, the relatively smooth variation of the curves shown in Figure 4.3 illustrates that there is some flexibility in the choice of driving frequency. However, it is useful to note the strong dependence of the calculated optimal frequencies on the power law absorption exponent y. For example, for the parameters shown in Figure 4.3, if y is increased from 1.1 to 1.2 with all other parameters remaining constant, the frequency that maximizes $W_{-6dB,\,\mathrm{area}}$ decreases from 516 kHz to 148 kHz. Given the relative uncertainty in obtaining this value from experimental measurements [14], and the range of values reported for different soft tissue types (see Appendix B), some care should be taken when assigning the absorption parameters and interpreting the results.

4.3. FOCUSED ULTRASOUND FIELDS IN HETEROGENEOUS TISSUE MEDIA

4.3.1 Anatomical Atlas

In a homogeneous medium, the overall shape of the ultrasound beam produced by a focused transducer is qualitatively very similar to that predicted by simple analytical formulae, even under moderately nonlinear conditions and including the effects of absorption. However, in biological tissue, refraction and scattering can cause noticeable distortions to the beam shape. Recall refraction is a change in the direction of wave propagation caused by transmission into a medium with a different sound speed. Similarly, scattering is a redirection of waves due to localized variations in the material properties, in particular, the sound speed and density. This can include diffusive scattering from sub-wavelength structures where the scattered field acts like point monopole or dipole sources, diffraction around wavelength sized inclusions, and specular reflection from large tissue interfaces such as organ boundaries.

To examine the effects of refraction and scattering using numerical simulation, a representative map of the material properties in the human body is required. Here, the open-source AustinWoman voxel model is used [15]. This is a segmentation of a digital image dataset acquired as part of the Visible Human Project run by the U.S. National Library of Medicine. The original dataset consists of digital cryosection images with 0.33 mm pixel spacing taken at 0.33 mm intervals through the axial direction of a 59-year-old female cadaver. Radiological images (CT and MRI) are also available. The AustinWoman segmentation divides the digital image dataset into 58 material labels that cover the major tissue types present in the body. A list of the corresponding acoustic material properties (sound speed, density, absorption coefficient, and nonlinearity parameter) is given in Appendix B.

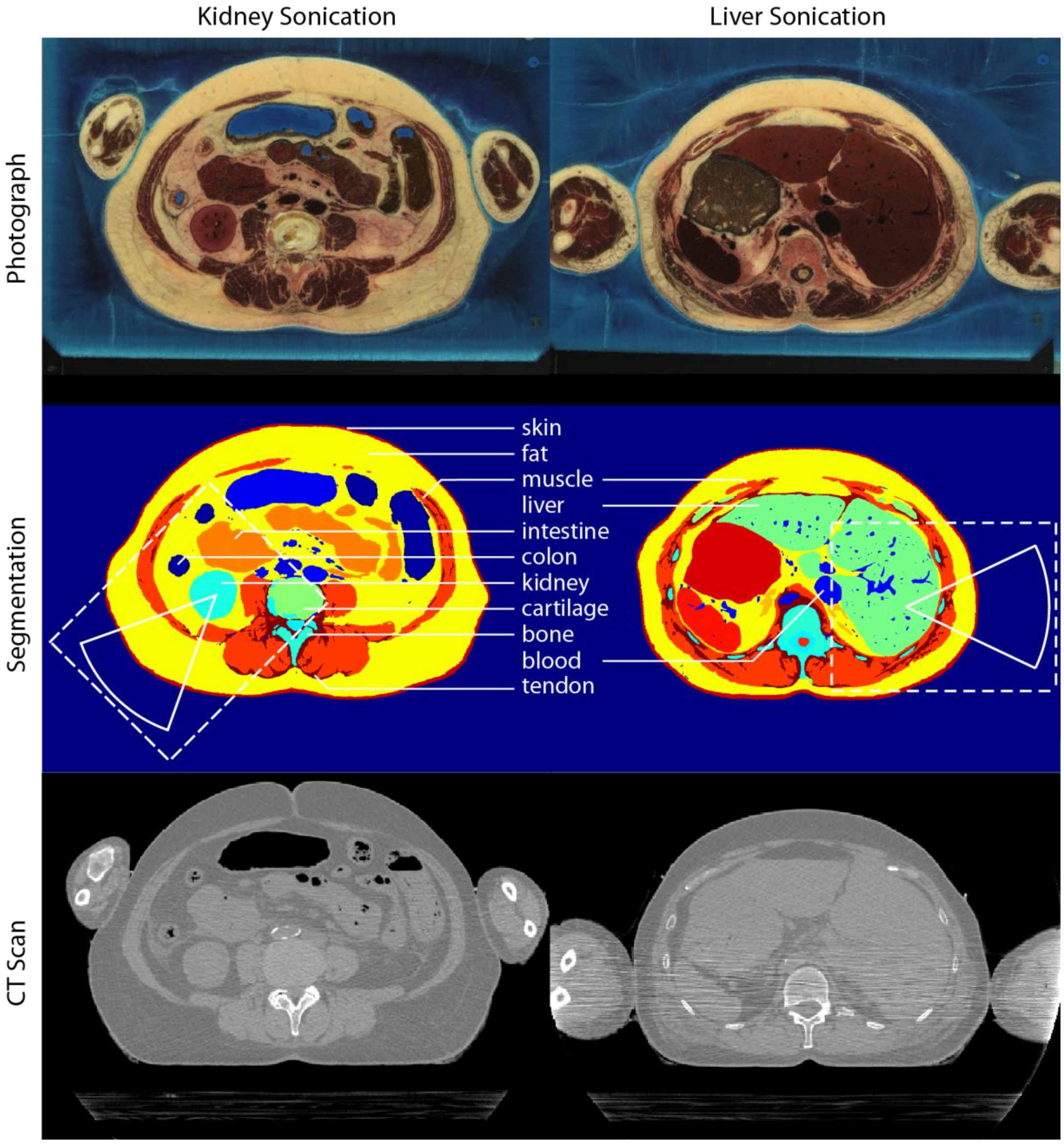


Figure 4.4: Anatomical data used to specify the acoustic material properties for numerical simulations in the human body. The top panels show cryosection images from the Visible Human Project, the middle panels show the AustinWoman segmentation of these images, and the bottom panels show the corresponding CT slice. The left and right columns correspond to kidney and liver targets, respectively. The position of the transducer and size of the computational domain are shown in the middle panels with white lines. Labels for the tissues types within the field of view are also shown.

To demonstrate the effects of tissue heterogeneities in a clinical HIFU setting, two abdominal targets are used (see Figure 4.4). The first is slightly above the inferior pole of the left kidney, and the second is within the right lobe of the liver. HIFU ablations of solid tumor deposits in the kidney and liver have been demonstrated successfully in clinical trials [16], and targeting these organs remains an active area of research [17]. Transverse slices from the cryosection images, segmentation, and CT data are shown in Figure 4.4. The in-plane position of the transducer and corresponding size of the computational domain used for the numerical simulations are shown with white lines. Labels for the major tissue types within the field of view are also shown. Both targets require sonication through the muscles surrounding the abdominal cavity, which are known to distort the propagation of ultrasound waves [18]. The liver target is also directly occluded by the body of the tenth rib. The kidney target is approximately 90 mm from the skin surface, while the liver target is approximately 65 mm deep.

Simulations of a single sonication for each abdominal target were performed using the same transducer geometry and drive conditions considered in the previous sections (see Figure 4.1 and 4.2 for reference). Simulations were repeated under both linear and nonlinear conditions, and for several artificial configurations. These included using a constant sound speed (by replacing the heterogeneous sound speed with that of the background medium), constant impedance (by artificially adjusting the density to give a uniform distribution of the characteristic acoustic impedance), and without particular anatomical features (e.g., the muscle, skin, and ribs). In total, ten simulations were performed for each target. Simulation and output parameters are summarized in Table 4.1, with grid parameters given in Table 4.2 in Appendix A.

4.3.2 Beam Distortion by the Body Wall

The simulation results for four of the kidney sonications are illustrated in Figure 4.5. The figures show the peak positive pressure (with the color bar scaled between 100 kPa and 70% of the maximum pressure) overlaid on the sound speed map derived from the AustinWoman voxel model. The three columns correspond to three orthogonal slices along the beam axis through the geometric focus of the transducer. The position of the transducer is also shown. The first row illustrates the response when all wave effects are considered (simulation reference K4 in Table 4.1). Compared to the response in water shown in Figure 4.1 (R1 in Table 4.1), the beam has been noticeably distorted. In particular, the focal pressure in the vertical plane projects downwards as the beam travels into the body. This asymmetry arises because the transducer is not positioned at normal incidence to the skin surface in this plane. This causes the beam to be refracted when it reaches the body. In addition, it creates a disparity in the arrival time from the superior and inferior surfaces of the transducer due to the difference in average sound speed along the respective ray paths (the sound speed in fat, shown in black, is less than that in the background medium).

The effect of heterogeneities can be quantified in several ways. First, when heterogeneities are introduced but only linear and lossless propagation is considered (K1), the spatial maximum of the peak positive pressure is reduced by 40% and the corresponding time averaged intensity by 67% compared to the response in water under the same conditions (R1). Second, comparing the simulation with constant sound speed (K8) with the simulation considering all wave effects (K4), the pressure and intensity are reduced by 51% and 68%, respectively. Thus for this treatment geometry, the focal intensity is reduced by approximately two thirds due to beam distortion from refraction effects. It is useful to consider the potential clinical significance of this distortion, especially given the patient geometry isn't particularly unusual. When absorption is included, the pressure and intensity are reduced by an additional 44% and 69%, respectively (K3 versus K1). Thus the reduction in focal intensity caused by absorption and refraction are of the same order. In comparison, the effects of reflection are negligible, with almost no change in the impedance matched simulations under both linear and nonlinear conditions compared to the corresponding simulations with unmodified material parameters (K2 versus K1 and K7 versus K4).

To quantify exactly which parts of the anatomy are responsible for the beam distortion, the response when all heterogeneities and wave effects are considered (K4) can be compared with the response with particular structures removed (K5 and K6). When only the skin, fat, and kidney are considered with all remaining tissue labelled as fat (K5), the peak positive pressure and intensity are increased by 28% and 44%, respectively. In this case, the main structures removed along the beam path are the muscles surrounding the abdominal cavity. These can have a significant effect on the focal waveform. If the skin layer is also removed (K6), the focal pressure and intensity are increased by a further 9% and 23%, respectively. The response when only water, fat, and kidney are considered is shown in the second row of Figure 4.5. Although the beam in the vertical plane is still asymmetric due to refraction effects, most of the distortion visible in the horizontal plane is removed. For this geometry, the skin and muscle layers cause the focal pressure to be reduced by 28%, and the focal volume rate of heat deposition to be almost halved.

In addition to studying the contribution of particular anatomical structures, the effect of frequency when the medium parameters are heterogeneous can also be examined. The simulation results when considering all wave effects using a driving frequency of 0.5 MHz (K9) and 2 MHz (K10) are shown in the bottom two rows of Figure 4.5. In general, using a lower frequency reduces the beam distortion, as the constructive interference responsible for generating high focal pressures is less sensitive to small aberrations when the wavelength is increased. This is also noticeable in the -6 dB focal volumes of the peak positive pressure displayed in Figure 4.6. In particular, the focal volume at 0.5 MHz is relatively uniform, while the focal volume at 2 MHz is almost completely destroyed, with the focus divided into many small lobes of higher pressure. Despite the reduction in aberration, for the same transducer surface pressure, the volume rate of heat deposition at 0.5 MHz is still lower than at 1 MHz. This is due to the decreased absorption and focusing gain, both of which scale approximately inversely with frequency (see Section 4.2.3).

Table 4.1: Simulation configurations and output parameters: $\boldsymbol{f}$ transducer driving frequency; $\boldsymbol{p_+}$ spatial maximum of the peak positive pressure; $\boldsymbol{Y}$ yes; $\boldsymbol{N}$ no; $\boldsymbol{p_-}$ peak negative pressure at the position of $\boldsymbol{p_+}$; $\boldsymbol{I_{spta}}$ time averaged intensity at the position of $\boldsymbol{p_+}$; $\boldsymbol{Q_{sp}}$ volume rate of heat deposition at the position of $\boldsymbol{p_+}$; $\boldsymbol{V_{-6dB}}$ -6 dB focal volume of the peak positive pressure computed by counting the number of voxels with a pressure amplitude above 50% of the maximum pressure; **Shift** change in position of $\boldsymbol{p_+}$ relative to R1. Configurations labelled with a * or $^{\#}$ are displayed in Figures 4.5 and 4.7, respectively.

Ref	$\boldsymbol{f}$	**Nonlinear /**	**Medium**	$\boldsymbol{p_+}$	$\boldsymbol{p_-}$	$\boldsymbol{I_{spta}}$	$\boldsymbol{Q_{sp}}$	$\boldsymbol{V_{-6dB}}$	**Shift**
	MHz	**Absorbing**	**Properties**	**MPa**	**MPa**	**W/cm^2**	**W/cm^3**	**mm^3**	**mm**
R1	1	N / N	Water	9.79	-9.79	3170	-	66.4	0
R2	1	Y / Y	Water	22.8	-6.32	3260	3.99	16.9	1.02
R3	1	N / Y	Tissue	3.66	-3.66	407	60.1	72.2	1.15
R4	1	Y / Y	Tissue	4.43	-3.16	415	65.2	53.9	0.286
K1	1	N / N	All tissues	5.88	-5.88	1050	-	184	2.64
K2	1	N / N	Constant Z	5.85	-5.85	1040	-	186	2.27
K3	1	N / Y	All tissues	3.27	-3.27	324	47.9	184	2.76
K4*	1	Y / Y	All tissues	3.95	-2.71	317	49.3	125	2.19
K5	1	Y / Y	Kidney, fat, skin	5.06	-3.19	455	74.8	110	2.07
K6*	1	Y / Y	Kidney, fat	5.50	-3.57	560	90.5	114	1.92
K7	1	Y/ Y	Constant Z	3.98	-2.68	315	49.3	122	2.13
K8	1	Y / Y	Constant c_0	8.08	-4.37	992	170	32.5	0.453
K9*	0.5	Y / Y	All tissues	3.51	-2.63	274	20.7	578	3.48
K10*	2	Y / Y	All tissues	2.13	-1.66	106	32.8	66.5	3.28
L1	1	N / N	All tissues	3.07	-3.07	276	-	18.3	4.09
L2	1	N / Y	All tissues	1.87	-1.87	102	14.1	30.4	4.32
L3$^{\#}$	1	Y / Y	All tissues	2.00	-1.72	101	14.1	38.8	5.21
L4	1	N / N	No ribs	5.82	-5.82	990	-	170	4.25
L5	1	N / Y	No ribs	3.48	-3.48	355	49.0	155	4.49
L6$^{\#}$	1	Y / Y	No ribs	4.12	-3.11	368	52.4	105	4.64
L7	1	Y / Y	Constant c_0	7.34	-4.40	899	137	41.2	0.184
L8$^{\#}$	1	Y / Y	Focus shifted	5.07	-3.50	505	72.9	93.5	3.08
L9$^{\#}$	0.5	Y / Y	All tissues	1.91	-1.67	93.0	6.46	325	9.75
L10	2	Y / Y	All tissues	1.39	-1.17	47.5	13.3	25.3	11.4

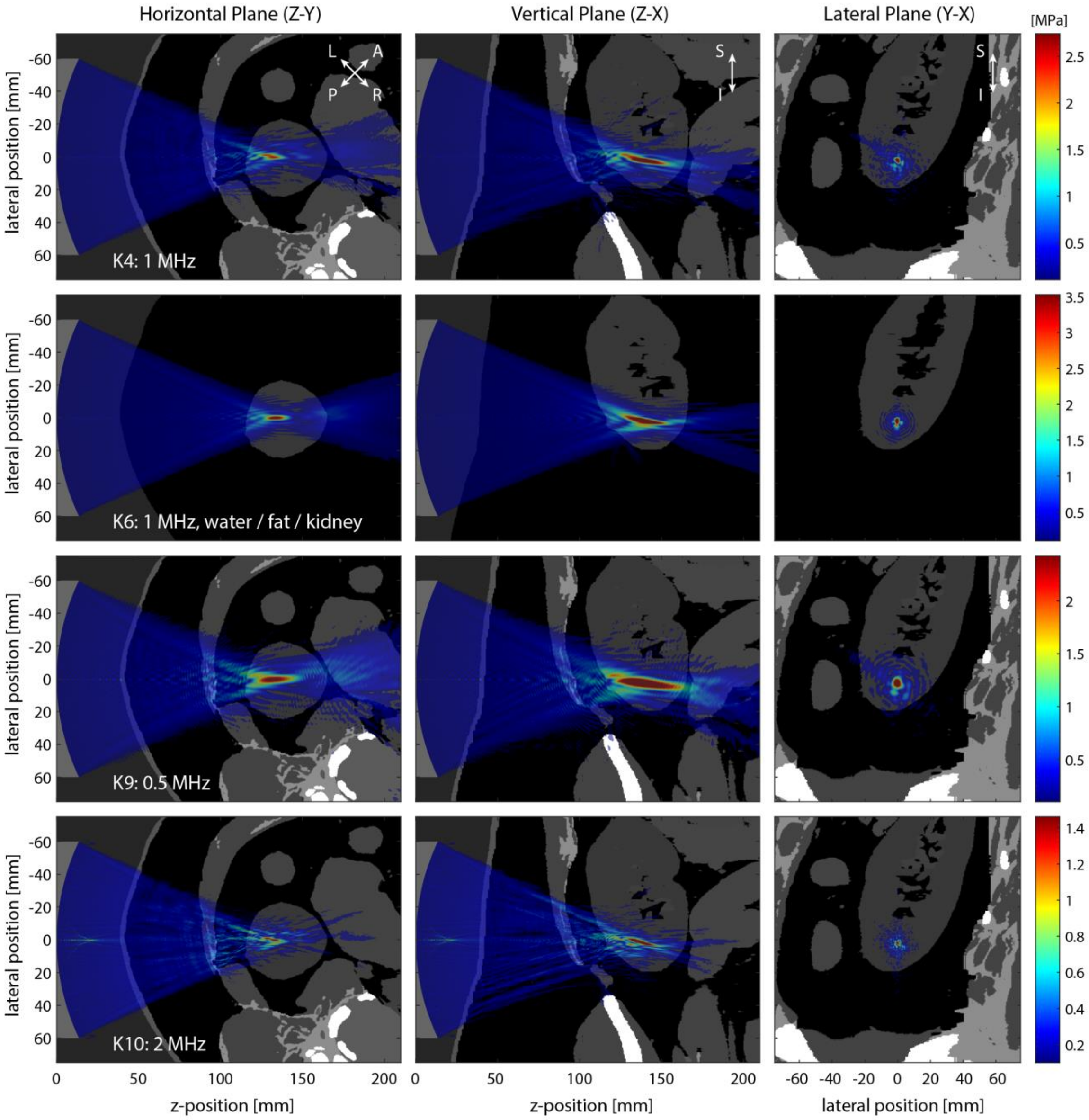


Figure 4.5: Beam plots showing the steady state distribution of peak positive pressure for a HIFU sonication of the kidney under four different conditions. The background image is the sound speed map derived from the segmented voxel data (dark: low, light: high). The position of the transducer is also illustrated. The three panels show orthogonal slices through the beam axis and geometric focus of the transducer. The color bars are truncated to 70% of the spatial peak positive pressure. The corresponding -6 dB focal volumes are shown in Figure 4.6. (A: anterior; P: posterior; L: left; R: right; S: superior; I: inferior.)

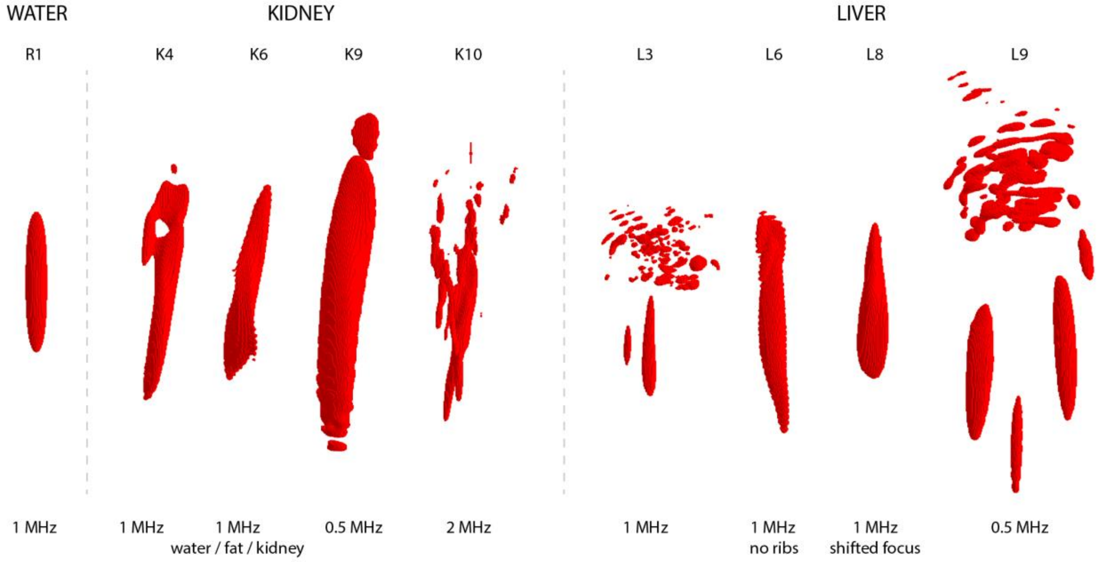


Figure 4.6: Voxel plots showing the -6 dB focal volume of the peak positive pressure for the sonications shown in Figures 4.5 and 4.7. The focal volume for a linear and lossless simulation in water is also shown for reference. Labels above each focal volume correspond to the simulation reference numbers given in Table 4.1.

4.3.4 Scattering by the Ribs

The simulation results for the liver sonications are illustrated in Figure 4.7, analogous to the kidney sonications discussed in the previous section. The first row shows the response when all wave effects are considered (L3 in Table 4.1). In this case, the beam path is directly obstructed by the body of the tenth rib. While such a treatment geometry is unlikely to be used in practice, it is useful to demonstrate the significant effect that bone can have on the propagation of focused ultrasound waves [19]. Compared to the response in water shown in Figure 4.1 (R1), the beam is strongly distorted, with additional regions of high pressure within and adjacent to the rib cage. This is also visible in the corresponding -6 dB focal volume given in Figure 4.6, which is split into a large number of small lobes. The considerable distortion to the focal volume arises because the acoustic impedance of bone is approximately 5 times higher than the surrounding soft tissue. This means the incident ultrasound waves are strongly scattered when they reach the bone surface. The acoustic absorption in bone is also an order of magnitude higher than in soft tissue. Combined, these effects lead to a large transmission loss along the beam path, which significantly reduces the pressure and intensity at the intended focus of the transducer.

The effect of the rib obstruction can be quantified in several ways. First, under linear and lossless conditions, when the thoracic cage and other heterogeneities are introduced in the beam path (L1), the spatial maximum of the peak positive pressure is reduced by 69% and the corresponding time averaged intensity by 91% compared to the response in water (R1). Second, considering the response when all

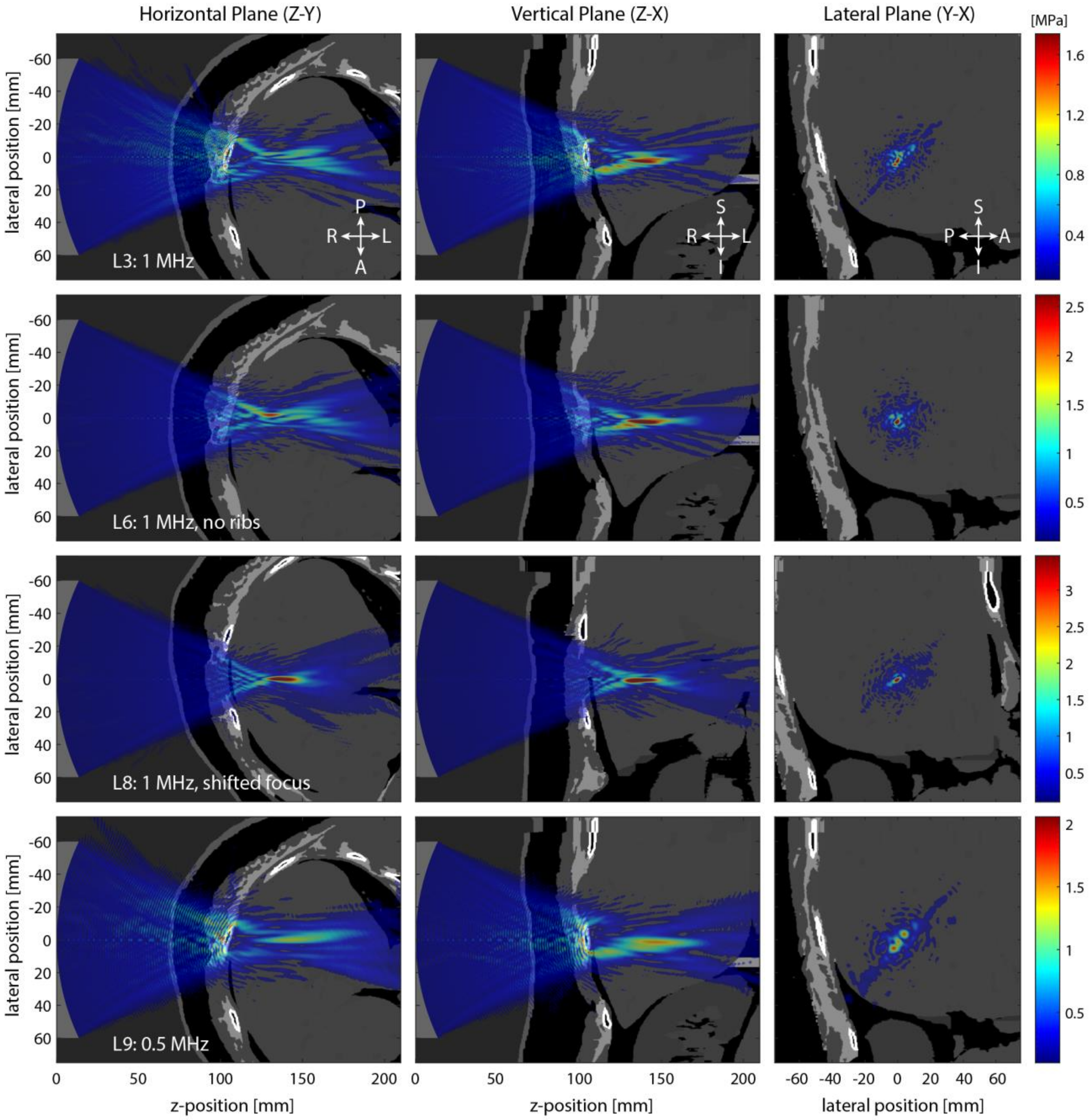


Figure 4.7: Beam plots showing the steady state distribution of peak positive pressure for a HIFU sonication of the liver under four different conditions. The background image is the sound speed map derived from the segmented voxel data (dark: low, light: high). The position of the transducer is also illustrated. The three panels show orthogonal slices through the beam axis and geometric focus of the transducer. The color bars are truncated to 70% of the spatial peak positive pressure. The corresponding -6 dB focal volumes are shown in Figure 4.6. (A: anterior; P: posterior; L: left; R: right; S: superior; I: inferior.)

tissue structures are present compared to when the ribs are removed (in this case replaced with the properties of tendon), the focal pressure and intensity are reduced by approximately 50% and 70%, respectively (L4 versus L1, L5 versus L2, and L6 versus L3). Thus, while the occluding rib is responsible for a significant proportion of the reduction in focal intensity, the skin and intercostal tissue also cause non-negligible distortions to the ultrasound beam. This is illustrated in the second row of Figure 4.7, where the focus in the horizontal plane is noticeably asymmetric. Finally, if the focus of the transducer is shifted in both the anterior and superior directions to avoid the ribs and instead sonicate through the intercostal space (L8), the focal pressure and intensity are significantly increased, in this case by a factor of 2.5 and 5, respectively. This is illustrated in the third row of Figure 4.7, where the beam shape is largely restored. Considering the effects of frequency, if the beam path is occluded by the ribs, the focal volume is still strongly distorted even when the driving frequency is reduced (Figures 4.6 and 4.7).

In addition to the beam distortion caused by tissue heterogeneities, there are several other factors to consider during the clinical ablation of tumors in the kidney and liver. First, for transcostal treatments, the strong absorption within bone can lead to local heating and thermal injury to the ribs and surrounding intercostal tissue [20]. This is particularly challenging to avoid if the transducer power must be increased to compensate for the large transmission loss encountered through the thoracic cage. Second, for kidney targets, the dense layer of peri-nephric fat that surrounds the organ has a particularly high absorption coefficient, which can lead to additional pre-focal heating and beam distortion (only a single value for the absorption of fat was considered here) [21]. Finally, the abdominal organs are mobile, and respiratory motion can cause centimeter scale movements relative to the ribs and the skin surface (see Chapter 17). It is useful to note that significant progress has been made in recent years on developing model-based aberration and motion correction algorithms for transcostal targets to address these challenges [17].

4.4. CONCLUSION

The propagation of focused ultrasound waves from bowl-shaped transducers under idealized conditions is well understood. However, under clinical conditions when the transducers are driven at high intensities to ablate tissue deep within the human body, the response is considerably more complex. In particular, the combined effects of nonlinearity, absorption, driving frequency, refraction, and scattering can significantly alter the position and shape of the focal volume, in addition to the focal pressure and intensity. This can affect the desired clinical outcome, and lead to adverse events such as near-field heating and skin burns. In many situations, the use of numerical simulations can provide detailed insight into the physical mechanisms affecting the propagating ultrasound waves, and provide useful predictions of the possible treatment outcomes.

Considering the transducer geometry, driving parameters, and the model of patient anatomy used here, several conclusions can be drawn. First, the strong acoustic absorption in biological tissue means that for the typical intensities currently used for MR-guided HIFU, the focal waveform is only weakly nonlinear. For the results given in Table 4.1, the increase in the volume rate of heat deposition due to nonlinear effects is always less than 10%. However, the contribution of nonlinear enhanced heating can become more significant when the transducer output power is increased. This is due to the rapid absorption of energy transferred to higher frequency harmonics. Second, the optimal driving frequency for abdominal targets is in the range from 500 kHz to 1.3 MHz, depending on the depth of the target and the output parameter that is maximized. Third, the reduction in focal intensity due to absorption and refraction are of the same order, and together can reduce the focal intensity by a factor of ten. In particular, the skin and muscle layers can cause significant aberrations to the ultrasound beam compared to water. Finally, the effects of reflection due to soft-tissue interfaces are negligible. However, the inclusion of bone within the beam path can reduce the focal intensity by more than 70%.

4.5. APPENDIX A: SIMULATION PARAMETERS

Simulations were performed using the open-source k-Wave Toolbox V1.1. Details of the governing equations and numerical methods can be found in Refs. [1], [22]. Input files were generated in MATLAB running on the Legion supercomputer at University College London using a Dell R820 large memory node with 32 cores and 1.5 TB of RAM. Simulations were executed using the MPI version of k-Wave [22] running on the Salomon supercomputer at the IT4Innovations national supercomputing center in The Czech Republic. For all simulations, the domain size (which determines the region of interest) was set to 165 × 165 × 220 mm, where the z dimension corresponds to the axial direction as shown in Figure 4.1. Three grid resolutions were used, depending on the required maximum frequency supported by the spatial grid (see Table 4.2). The homogeneous simulations were all performed at S1, except for those with driving frequencies at 1.75 MHz or above (Figure 4.3) which were performed at S3, and the nonlinear simulation in water (Figure 4.2) which was performed at S5. The heterogeneous simulations were all performed at S2, except for those with a driving frequency of 2 MHz or a medium with constant sound speed, which were performed at S4. For all configurations, a perfectly matched layer with a thickness of 20 grid points was used on each side of the domain to simulate free-field conditions.

Table 4.2: Compute parameters for the simulations given in Table 4.1. The memory usage and wall-clock time are representative values. The high memory usage is due to the input source being replicated by every MPI process.

Ref	**Grid Size**	**Medium**	**Δx**	**Δt**	**f_{max}**	**Time**	**Mem**	**Compute**	**Time**
		Properties	**µm**	**ns**	**MHz**	**Steps**	**TB**	**Cores**	**d:hh:mm**
S1	1152 × 1152 × 1536	homogeneous	143	27.7	5.32	10,134	1.4	1024	0:09:14
S2	1152 × 1152 × 1536	heterogeneous	143	9.43	5.32	30,604	4.1	1536	0:20:42
S3	1344 × 1344 × 1792	homogeneous	123	23.8	6.21	11,823	2.3	768	0:18:10
S4	1344 × 1344 × 1792	heterogeneous	123	8.06	6.21	35,801	6.6	1536	2:01:33
S5	3072 × 3072 × 4096	homogeneous	53.7	10.3	14.2	27,139	3.1	1024	8:21:32

For the heterogeneous simulations, medium parameters were assigned using the labelled AustinWoman V2.3 segmentation [15], and then resampled to the correct grid resolution using linear interpolation. The medium parameters are shown in Table 4.3. These were largely derived from the average values in the IT'IS tissue property database V3.0 [23]. As k-Wave only allows a single value for the power law absorption exponent y, this was set to 1.5 and values for the absorption coefficient pre-factor α_0 were re-calculated to give the equivalent absorption coefficient at the driving frequency.

Material properties for gaseous tissues not close to the beam path (e.g., air within the small intestine) were assigned the properties of muscle. The reference sound speed c_{ref} used in the k-space operator was set to the sound speed in the background medium [1]. The time step was chosen to give an integer number of points per period. Homogeneous simulations were performed with $\Delta t \approx 0.3\Delta x/c_{\mathrm{ref}}$ and heterogeneous simulations with $\Delta t \approx 0.1\Delta x/c_{\mathrm{ref}}$.

The source geometry was defined using a discrete bowl with a simply-connected surface, and the source was modelled as the injection of mass in free space including a correction for staircasing [24]. Three periods of the driving frequency were recorded in steady state within a cuboid-shaped region that enclosed the -6 dB focal volume. The maximum and minimum pressure in steady state across the complete grid were also recorded. Output files were processed in MATLAB running on Legion to generate figures and calculate the output parameters given in Table 4.1. For the liver simulations, the spatial maximum values were calculated from the focal region, excluding the area around the ribs. For the nonlinear simulation shown in the left panel of Figure 4.2(c), modelling fourteen harmonics was not sufficient to completely capture the time domain waveform, and larger simulations were not feasible. Following the observation that the energy at higher frequency harmonics decays as a power law, the time domain waveform was corrected by extrapolating the amplitude of additional harmonics using a power law fit. Typical compute times and run-time memory usage are given in Table 4.2.

4.6. APPENDIX B: TISSUE PROPERTIES

Table 4.3: Acoustic properties of abdominal tissues derived from the IT'IS tissue property database V3.0 [23]. Properties of water taken from [25].

	c_0	ρ_0	B/A	α_0	y
	m s^{-1}	kg m^{-3}		dB MHz^{-y} cm^{-1}	MHz
Blood	1578	1050	6.11	0.206	1.05
Bone	3515	1908	0	4.74	1
Cartilage	1640	1099	0	3.82e-2	1
CSF	1505	1007	0	8.68e-3	1
Fat	1440	911	10.1	0.378	1.09
Gall bladder	1584	1070	6.22	0.131	1
Kidney	1563	1053	7.44	0.642	1.1
Liver	1586	1079	7.28	0.6	1
Marrow	1450	1028	0	1.085	1
Muscle	1588	1090	7.17	0.617	1.08
Nerve	1630	1075	6.7	1.15	1
Pancreas	1591	1086	6.7	0.829	1
Plasma	1549	1020	5.74	9.33e-2	1.13
Skin	1624	1109	6.7	1.84	1
Spleen	1568	1089	7.62	0.380	1.38
Tendon	1750	1142	0	1.26	1.17
Water at 37°C	1524	993	5.54	1.35e-3	2

ACKNOWLEDGMENTS

B. T. acknowledges the support of the Engineering and Physical Sciences Research Council (EPSRC), grant numbers EP/L020262/1 and EP/M011119/1. J. J. is financed from the SoMoPro II Programme, co-financed by the European Union and the South-Moravian Region. This work reflects only the author's view and the European Union is not liable for any use that may be made of the information contained therein.

REFERENCES

[1] B. E. Treeby, J. Jaros, A. P. Rendell, and B. T. Cox, "Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method.," *J. Acoust. Soc. Am.*, vol. 131, no. 6, pp. 4324–4336, 2012.

[2] H. T. O'Neil, "Theory of focusing radiators," *J. Acoust. Soc. Am.*, vol. 21, no. 5, pp. 516–526, 1949.

[3] R. S. C. Cobbold, *Foundations of Biomedical Ultrasound*. Oxford: Oxford University Press, 2007.

[4] F. P. Curra, P. D. Mourad, V. A. Khokhlova, R. O. Cleveland, and L. A. Crum, "Numerical simulations of heating patterns and tissue temperature response due to high-intensity focused ultrasound," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 47, no. 4, pp. 1077–1089, 2000.

[5] J. E. Soneson and M. R. Myers, "Thresholds for nonlinear effects in high- intensity focused ultrasound propagation and tissue heating.," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 57, no. 11, pp. 2450–2459, 2010.

[6] M. Solovchuk, T. W. H. Sheu, and M. Thiriet, "Simulation of nonlinear Westervelt equation for the investigation of acoustic streaming and nonlinear propagation effects.," *J. Acoust. Soc. Am.*, vol. 134, no. 5, pp. 3931–3942, 2013.

[7] D. . Bacon and E. L. Carstensen, "Measurement of enhanced heating due to ultrasound absorption in the presence of nonlinear propagation," in *IEEE International Ultrasonics Symposium*, 1989, vol. 2, pp. 1057–1060.

[8] E. A. Filonenko and V. A. Khokhlova, "Effect of acoustic nonlinearity on heating of biological tissue by high-intensity focused ultrasound," *Acoust. Phys.*, vol. 47, no. 4, pp. 468–475, 2001.

[9] K. Hynynen, D. J. Watmough, and J. R. Mallard, "Design of ultrasonic transducers for local hyperthermia," *Ultrasound Med. Biol.*, vol. 7, no. 4, pp. 397–402, 1981.

[10] C. R. Hill, "Optimum acoustic frequency for focused ultrasound surgery," *Ultrasound Med. Biol.*, vol. 20, no. 3, pp. 271–277, 1994.

[11] N. Ellens and K. Hynynen, "Frequency considerations for deep ablation with high-intensity focused ultrasound: A simulation study," *Med. Phys.*, vol. 42, no. 8, pp. 4896–4910, 2015.

[12] A. S. Ergün, "Analytical and numerical calculations of optimum design frequency for focused ultrasound therapy and acoustic radiation force.," *Ultrasonics*, vol. 51, no. 7, pp. 786–794, 2011.

[13] C. R. Hill, I. Rivens, M. G. Vaughan, and G. R. ter Haar, "Lesion development in focused ultrasound surgery: a general model.," *Ultrasound Med. Biol.*, vol. 20, no. 3, pp. 259–269, 1994.

[14] J. C. Bamber, “Attenuation and Absorption,” in *Physical Principles of Medical Ultrasonics*, C. R. Hill, J. C. Bamber, and G. R. ter Haar, Eds. Chichester: John Wiley & Sons, 2004, pp. 93–166.

[15] J. Massey, C. Geyik, N. Techachainiran, C. Hsu, R. Nguyen, T. Latson, M. Ball, and A. Yilmaz, “AustinMan and AustinWoman: High fidelity, reproducible, and open-source electromagnetic voxel models,” in *Bioelectromagnetics Society 34th Annual Meeting*, 2012.

[16] R. O. Illing, J. E. Kennedy, F. Wu, G. R. ter Haar, a S. Protheroe, P. J. Friend, F. V Gleeson, D. W. Cranston, R. R. Phillips, and M. R. Middleton, “The safety and feasibility of extracorporeal high-intensity focused ultrasound (HIFU) for the treatment of liver and kidney tumours in a Western population.,” *Brit. J. Cancer*, vol. 93, no. 8, pp. 890–895, 2005.

[17] B. D. de Senneville, C. Moonen, and M. Ries, “MRI-Guided HIFU Methods for the Ablation of Liver and Renal Cancers,” in *Therapeutic Ultrasound*, J.-M. Escoffre and A. Bouakaz, Eds. Heidelberg: Springer, 2016, pp. 43–81.

[18] L. M. Hinkelman, T. D. Mast, M. J. Orr, and R. C. Waag, “Effects of abdominal wall morphology on ultrasonic pulse distortion,” *J. Acoust. Soc. Am.*, vol. 104, no. 6, pp. 3635–3649, 1998.

[19] P. Gélat, G. ter Haar, and N. Saffari, “The optimization of acoustic fields for ablative therapies of tumours in the upper abdomen,” *Phys. Med. Biol.*, vol. 57, no. 24, pp. 8471–8497, 2012.

[20] H.-L. Liu, H. Chang, W.-S. Chen, T.-C. Shih, J.-K. Hsiao, and W.-L. Lin, “Feasibility of transrib focused ultrasound thermal ablation for liver tumors using a spherically curved 2D array: a numerical study.,” *Med. Phys.*, vol. 34, no. 9, pp. 3436–3448, 2007.

[21] R. Ritchie, J. Collin, C. Coussios, and T. Leslie, “Attenuation and de-focusing during high-intensity focused ultrasound therapy through peri-nephric fat,” *Ultrasound Med. Biol.*, vol. 39, no. 10, pp. 1785–1793, 2013.

[22] J. Jaros, A. P. Rendell, and B. E. Treeby, “Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound,” *Int. J. High Perform. C.*, vol. 30, no. 2, pp. 137–155, 2016.

[23] P. Hasgall, F. Di Gennaro, C. Baumgartner, E. Neufeld, M. Gosselin, D. Payne, A. Klingenböck, and N. Kuster, “IT’IS Database for thermal and electromagnetic parameters of biological tissues (Version 3.0).” DOI: 10.13099/VIP21000-03-0, 2015.

[24] E. Martin, Y. T. Ling, and B. E. Treeby, “A Discrete Bowl Model for Simulating Focused Ultrasound Transducers on Regular Cartesian Grids,” *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. X, no. X, pp. 1–7, 2016.

[25] K. W. Van Dongen and M. D. Verweij, “Sensitivity study of the acoustic nonlinearity parameter for measuring temperatures during high intensity focused ultrasound treatment,” *J. Acoust. Soc. Am.*, vol. 123, p. 3225, 2008.

C.2 Simulations in the Prostate

Georgiou, P. S.; **Jaros, J.**; Payne, H.; Allen, C.; Shah, T. T.; Ahmed, H. U.; Gibson, E.; Barratt, D.; Treeby, B. E.: Beam distortion due to gold fiducial markers during salvage high-intensity focused ultrasound in the prostate. *Medical Physics*. vol. 44, no. 2. 2017: pp. 679–693. ISSN 0094-2405. doi:10.1002/mp.12044. **(IF 2.496)**.

Beam distortion due to gold fiducial markers during salvage high-intensity focused ultrasound in the prostate

P. S. Georgiou[a)]
Department of Medical Physics and Biomedical Engineering, University College London, London, UK

J. Jaros
Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic

H. Payne
Department of Medical Physics and Biomedical Engineering, University College London, London, UK
Department of Oncology, University College London Hospitals, London, UK

C. Allen
Department of Oncology, University College London Hospitals, London, UK

T. T. Shah and H. U. Ahmed
Division of Surgery and Interventional Science, University College London, London, UK

E. Gibson, D. Barratt, and B. E. Treeby
Department of Medical Physics and Biomedical Engineering, University College London, London, UK

(Received 23 August 2016; revised 21 October 2016; accepted for publication 17 November 2016; published 13 February 2017)

Purpose: High intensity focused ultrasound (HIFU) provides a non-invasive salvage treatment option for patients with recurrence after external beam radiation therapy (EBRT). As part of EBRT the prostate is frequently implanted with permanent fiducial markers. To date, the impact of these markers on subsequent HIFU treatment is unknown. The objective of this work was to systematically investigate, using computational simulations, how these fiducial markers affect the delivery of HIFU treatment.

Methods: A series of simulations was performed modelling the propagation of ultrasound pressure waves in the prostate with a single spherical or cylindrical gold marker at different positions and orientations. For each marker configuration, a set of metrics (spatial-peak temporal-average intensity, focus shift, focal volume) was evaluated to quantify the distortion introduced at the focus. An analytical model was also developed describing the marker effect on the intensity at the focus. The model was used to examine the marker's impact in a clinical setting through case studies.

Results: The simulations show that the presence of the marker in the pre-focal region causes reflections which induce a decrease in the focal intensity and focal volume, and a shift of the maximum pressure point away from the transducer's focus. These effects depend on the shape and orientation of the marker and become more pronounced as its distance from the transducer's focus decreases, with the distortion introduced by the marker greatly increasing when placed within 5 mm of the focus. The analytical model approximates the marker's effect and can be used as an alternative method to the computationally intensive and time consuming simulations for quickly estimating the intensity at the focus. A retrospective review of a small patient cohort selected for focal HIFU after failed EBRT indicates that the presence of the marker may affect HIFU treatment delivery.

Conclusions: The distortion introduced by the marker to the HIFU beam when positioned close to the focus may result in an undertreated region beyond the marker due to less energy arriving at the focus, and an overtreated region due to reflections. Further work is necessary to investigate whether the results presented here justify the revision of the patient selection criteria or the markers' placement protocol. © *2016 American Association of Physicists in Medicine* [https://doi.org/10.1002/mp.12044]

Key words: EBRT, fiducial marker, prostate cancer, salvage HIFU

1. INTRODUCTION

Prostate cancer is the most commonly occurring male cancer and the second leading cause of cancer-related death in men in the European Union (EU) and the United States of America (USA).[1,2] More specifically, in 2012 of all reported cancer cases in men, prostate cancer accounted for 24.1% in the EU, with a 10% mortality rate, and 28.3% in the USA, with a 9.4% mortality.[1] According to the American Cancer Society, these rates are estimated to continue in 2016.[3] These figures differ when viewed based on worldwide data, where prostate cancer appears as the second most diagnosed cancer in men with 1.1 million diagnosed cases in 2012 (14.8%) and as the fifth cause of cancer-related death with 307 thousand deaths (6.6%).[1,4] Despite the lower incidence and mortality rates from worldwide data, prostate cancer is still one of the major malignancies affecting hundreds of thousands of men each year and improving its diagnosis and treatment is of great importance.

There is a range of available options for treating prostate cancer with some of them having a curative intent and others palliative. Selecting an appropriate therapy depends on several factors, such as the stage of the tumor, biochemical indicators (e.g. prostate specific antigen value), Gleason score, other associated diseases, the patient's age and life expectancy, as well as the patient's personal preference.[5] For localized or locally advanced prostate cancer, recommended treatments for primary therapy include active surveillance, radical prostatectomy (RP), external beam radiation therapy (EBRT) and temporary (high-dose rate) or permanent (low-dose rate) brachytherapy (BT) with or without additional EBRT. These treatments may be offered independently or in combination with hormonal therapy (androgen deprivation therapy). In recent years, new minimally invasive modalities have emerged and provide alternative treatment options with the most notable being high intensity focused ultrasound (HIFU) and cryosurgery.[5–7]

EBRT is an effective primary therapy option with good survival rates reported.[8–10] It is estimated that 12–24% of patients diagnosed with localized prostate cancer receive EBRT as a primary treatment.[11,12] Although numbers may differ depending on the definition of failure used, in approximately 30% of these patients their cancer will recur[8,13–16] with some studies reporting even higher rates.[14,17–20] For patients with local recurrence after EBRT, depending on life expectancy and tumor progression, an alternative (salvage) therapy may still be appropriate in order to limit further progression of the disease and metastasis.[14,20] The four major options for re-treatment available after EBRT failure are salvage RP, salvage BT, salvage cryosurgery and salvage HIFU.[15,16,19,21] Amongst these methods, salvage RP is the most established treatment with good oncological outcomes.[15] However, it is associated with high morbidity,[14–16,18,19,22,23] thus, doctors may be reluctant to recommend it, especially for patients with a short life expectancy.[14–16,19] The other three modalities provide a less invasive alternative, with HIFU offering the least invasive approach.[20]

HIFU has been the subject of many studies indicating its potential as a primary treatment for locally confined prostate cancer.[24] Accurately determining the efficacy of this modality is not easy, especially due to the inconsistency in reporting biochemical failure and due to the absence of long-term oncological outcomes.[25,26] As a result and despite already being in use in many centers across the world, HIFU is still classified as an experimental treatment, for example, by the European Association of Urologists (EAU).[6] Nonetheless, some studies report encouraging results for primary HIFU treatment with low mortality rate, high metastasis-free survival rate, and acceptable side-effects comparable to other minimally invasive modalities.[24,26]

Currently, only a limited number of studies report on the efficacy and safety of HIFU as a salvage therapy after failure of EBRT.[5,15,27] Additionally, no prospective randomized trials have been reported.[27] Consequently, comparison of HIFU with other conventional salvage modalities is difficult. Most of the published investigations are retrospective studies,[14,17,20,28,29] with only a few prospective series reported.[5,19,23] The majority of these studies report good local cancer control, indicating the potential of HIFU as an effective salvage therapy for low- and intermediate-risk patients. In some studies, the rate of complications reported is high, with some adverse effects comparable to the other salvage therapies. This presents a limitation for the use of HIFU as a salvage therapy.[23] However, it is interesting to observe that in those studies where new refined treatment parameters were introduced, dedicated to post-radiation salvage-HIFU treatments, the side-effects were significantly reduced.[14,19,23]

Salvage treatment in a previously irradiated prostate is technically challenging, and higher rates and more severe side effects are expected. However, the positive effect of the new treatment parameters introduced in some of the studies for post-EBRT salvage-HIFU demonstrates that there is significant opportunity for improving both the delivery of salvage-HIFU treatment as well as patient selection. An aspect of salvage HIFU that has been overlooked so far, and which may affect the treatment's safety and efficacy as well as patient selection, is the presence of fiducial markers that are increasingly introduced in the prostate as part of modern image-guided radiotherapy (IGRT).[30,31] To the best of our knowledge, there are no studies investigating the effect of these markers on the delivery of salvage-HIFU after EBRT, with the exception of a small number of studies reporting on the effect of permanent BT seeds on HIFU.[32,33] The purpose of this work was to perform an extensive quantitative investigation of the effect of fiducial markers on the propagation and focusing of the ultrasound (US) waves when the beam path is obstructed by an EBRT fiducial marker.

The fiducial markers are introduced in the prostate in order to improve the accuracy of EBRT. They facilitate the localization of the prostate, enable motion and deformation tracking and act as reference points for distance measurements as well as for registering images obtained from different imaging modalities.[30,31,34,35] Typically 3 radio-opaque markers are implanted in the prostate before the patient undergoes EBRT planning and remain permanently in the prostate after the completion of the treatment.[36,37] The markers are placed within the prostate gland using a transperineal or transrectal approach with a needle holding one or two markers, under the guidance of transrectal ultrasound (TRUS) in a procedure similar to that of a biopsy.[36,38] Although this may differ between hospitals, the markers are typically placed in the prostate base, mid-gland, and apex at a distance of approximately 2 cm from each other and at least a 15° angle between any fiducial triplets.[36,39,40] Consequently, only a single marker is likely to be encountered by the HIFU beam for any individual sonication.

There is a large range of commercially available markers made from a variety of materials in different shapes and dimensions.[30,34,35,41] The most commonly used markers are made of gold, making them visible in a variety of imaging modalities, and have a cylindrical shape with their surface appropriately shaped to minimize migration.[30] Although less frequent, spherical gold markers are also utilized.[42,43] These two shapes facilitate their insertion using a needle. More recently, new types of markers have emerged that offer some advantages but have yet to gain wide popularity. There are three notable examples. First, carbon fiducials, which offer better visibility and produce less artefacts on computed tomography (CT) images. Second, metallic coils and strings of markers on an absorbable strand, which may offer better stability and localization since they stretch across the whole gland. Finally, a new family of markers with a transponder built-in to allow wireless tracking of their position in real-time without the need for additional imaging of the prostate.[30]

This work investigates the effect of a single gold marker on the HIFU beam using numerical simulations based on a model of the prostate containing a spherical or cylindrical fiducial. For each marker shape, a series of simulations was performed on a high performance computer (HPC) cluster to evaluate the propagation of the HIFU beam in the prostate when its path is obstructed by a single gold marker. The simulations used the open-source k-Wave[44,45] toolbox developed by our group for accurate modelling of the propagation of US waves. The simulation results were also used to verify an analytical model developed for approximating the effect of the marker on the intensity at the focus and for identifying a region within which the marker has a significant impact on the focusing. The study was performed in silico for several reasons. First, computer simulations provide an effective and low-cost method for investigating the key factors at play in HIFU therapy delivery in a way that would be impractical, costly, and possibly unethical in patients. Second, simulations give access to a complete characterization of the acoustic field parameters that are not accessible from an experiment. Knowledge of these parameters is critical to understand and quantify the effect of the markers.

2. METHODS

2.A. Clinical HIFU systems for prostate cancer

There are three approved transrectal HIFU treatment systems dedicated to prostate cancer and one transurethral device currently in clinical trials. The transrectal systems are the Sonablate 500 (SonaCare Medical LLC, Charlotte, NC, USA), the Ablatherm II (EDAP TMS, Vaulx-en-Velin, France), and the Focal One also developed by EDAP TMS. The operation of these systems is based on similar principles. They all deliver the treatment using a probe with an integrated HIFU and imaging transducer, which is used transrectally under US guidance to induce ablation of the targeted region.[24] On the other hand, the TULSA-PRO (PROFOUND MEDICAL Corp., Toronto, Canada) delivers transurethral ultrasound ablation of prostate cancer under magnetic resonance imaging (MRI) guidance.[46]

The transducer model used in this work nominally followed the specifications of the Sonablate 500 (SonaCare Medical). This system is currently in use at University College London Hospital (UCLH).[47] The Sonablate 500 consists of a console, a transrectal probe, and a cooling and degasing module. The transducer is held at the tip of the probe by a motorized system that allows it to move in the longitudinal and transverse direction with a 90 degree treatment window.[48] The transducer module itself consists of two dual-mode (splitbeam) transducers stacked back-to-back capable of both imaging at 6.3 MHz and treatment at 4 MHz. To achieve this dual-mode operation, each side of the transducer consists of two elements: a circular element at its center, dedicated to imaging, surrounded by an annular element, used for treatment. Each side is manufactured with a different curvature resulting in two fixed focal lengths (30 mm and 40 mm) by means of geometric focusing. This allows the whole prostate gland to be treated using a single probe and without the need for electronic beam steering. The current dimensions of the transducer are 22 mm by 30 mm, noting that earlier revisions of the system used a 22 mm by 35 mm transducer. The dimensions of the earlier revision are followed in this work.

During treatment, the ablation is given in blocks and can be applied to the whole gland or focally to only the cancerous lesion within the prostate. The ablated volume is pseudoellipsoidal and its precise location is determined by the focal length of the transducer. For each sonication, the ablated volume is on the order of $3 \times 3 \times 10$ mm^3. Multiple sonications with slight overlap move sequentially through the prostate with 3 s ‘on’ time exposures and 6 s ‘off’ time exposures. The prostate is divided into six blocks, left and right with corresponding anterior, middle, and posterior blocks. The 40 mm focal length is used for anterior and middle block treatment and the 30 mm probe for posterior block treatment. Tissue destruction is produced by thermal, mechanical, and cavitation effects to produce a clearly demarcated region of coagulative necrosis.

2.B. Simulation setup

The simulations were performed using the open-source k-Wave Toolbox.[44] This solves a generalized version of the Westervelt equation accounting for the combined effects of nonlinearity, heterogeneous material properties, and acoustic absorption following a frequency power law. The transducer geometry was assumed to be a spherical section with width $W_t = 22$ mm, length $L_t = 35$ mm, focal length $R_t = 40$ mm and without an imaging element included. The simulations were performed using a regular Cartesian mesh and the transducer was defined in the grid as a simply-connected sphere with a single grid-point thickness truncated

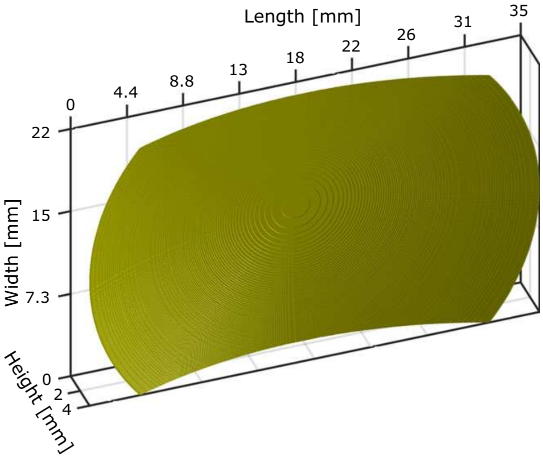


FIG. 1. The discretized transducer model used in the simulations shown in 3D. [Color figure can be viewed at wileyonlinelibrary.com]

to the appropriate width and height. The discretized transducer model is shown in Fig. 1. The transducer was driven by a $f_0 = 1/T = 4$ MHz sinusoidal input signal, with surface pressure p_0 given by

$$p_0 = \sqrt{2I_{\text{av}}\rho_0 c_0}. \quad (1)$$

Here ρ_0 and c_0 are, respectively, the density and sound speed of the background medium and $I_{\text{av}} = 4$ W/cm^2 is the time-averaged source surface intensity. The value of I_{av} was selected such that the focal intensity is of the order of magnitude reported for the Sonablate 500 (1000–2000 W/cm^2).[49,50]

The total duration of the input signal was 60 μs, which was equal to the total simulation time and long enough to ensure the pressure had reached steady-state. To ensure stability, a smaller time step dt was used for the simulations in which a marker was introduced in the grid.[45] The time step for the homogeneous case (without marker) was given by $dt = 0.33dx/c_0$, and for the heterogeneous case (with marker) by $dt = 0.066dx/c_0$, where dx is the spatial grid-spacing. As a result, the total number of time steps was 5 times higher for the heterogeneous simulations.

The physical dimensions of the simulation volume were $(L_x, L_y, L_z) = (44.7, 29.4, 60.0)$ mm. This was discretized to a regular Cartesian grid with dimensions (N_x, N_y, N_z), which included a $L_{\text{PML}} = 20$ grid-points (pt) perfectly matched layer (PML) at either end of each coordinate axis.[51] The grid spacing was uniform along all three coordinate axes and was defined according to $dx = dy = dz = L_z/(N_z - 2L_{\text{PML}})$. The background medium was assigned the material properties of the prostate (density: $\rho_0 = 1050$ kg/m^3 and sound-speed: $c_0 = 1578$ m/s), and the spherical or cylindrical volume occupied by the marker was assigned the properties of gold (density: $\rho_m = 19300$ kg/m^3 and sound-speed: $c_m = 3240$ m/s). Reference simulations were also performed without the inclusion of a marker in order to record the characteristics of an uninterrupted HIFU beam. Both sets of simulations were nonlinear (nonlinearity parameter: $B/A = 6.75$) and accounted for absorption following a frequency power law of the form $\alpha_0 f^{y_0}$ where $\alpha_0 = 0.5$ dB MHz$^{-y_0}$cm^{-1} and $y_0 = 1.1$.

2.C. Marker placement

To investigate the effect of fiducials on the propagation and focusing of the HIFU beam, each simulation included a single spherical or cylindrical gold marker positioned at different coordinates, with the position of the transducer kept fixed across all simulations. The spherical marker had a 3 mm diameter, whereas the cylindrical had a 3 mm height and 1 mm diameter. The simulated positions for each marker shape are shown in Fig. 2.

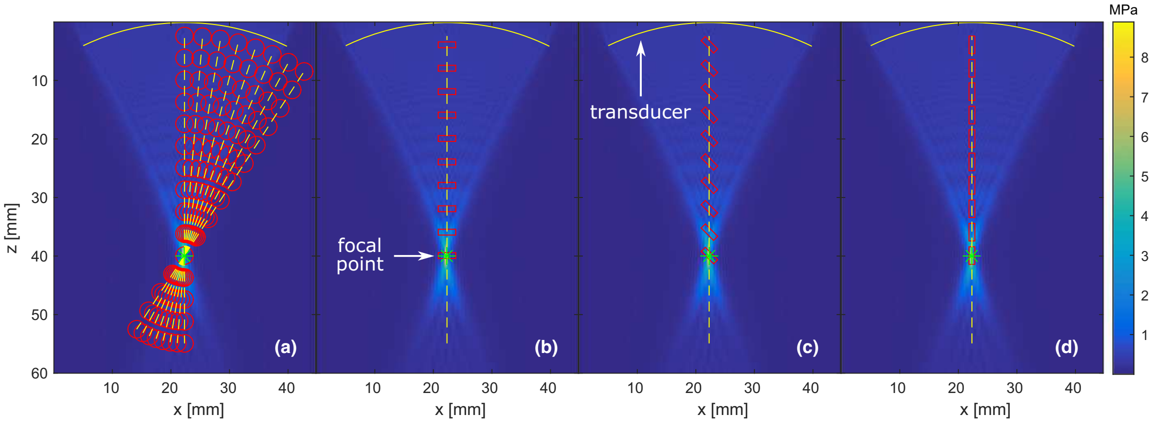


FIG. 2. The positions and orientations simulated for (a) the spherical markers with 3 mm diameter and (b)–(d) the cylindrical markers with 1 mm diameter and 3 mm height. The simulated positions are shown superimposed on the maximum pressure field of a homogeneous medium. The markers were placed on the zx plane which includes the transducer's focal-point $\mathbf{m}_f$, indicated by the star-like marker. The cross-section of the transducer is also indicated at the top of each figure. [Color figure can be viewed at wileyonlinelibrary.com]

In order to reduce the number of simulations performed, the center positions of the markers were limited to the xz plane at $y = N_y/2$. This plane includes the center of the focal region which was expected at approximately $\mathbf{m}_f = (N_x/2, N_y/2, \lfloor R_t/dz \rfloor)$, where $\mathbf{m}$ is a 3-dimensional (3D) vector denoting the coordinates of any point in the grid. If the plane is further divided into four quadrants with the origin at the focal point, the simulated positions were limited to the top-right and bottom-left quadrant as shown in Fig. 2(a). The choice of limiting the tested marker positions into two quadrants on a single plane was based on the assumption that any observed effect will be approximately symmetric about the beam axis. Both types of markers were positioned radially along the axes connecting the focal point to the transducer. The distance between markers along each axis, as well as the angle between successive axes, were kept constant. Hence, the markers can be grouped either with respect to the angle of the radial axis from the beam axis (see Fig. 5(f) inset), or with respect to their distance from the focal-point (see Fig. 5(c) inset).

2.D. Quantifying marker effect

To systematically evaluate the effect of a single marker on the focusing of the HIFU beam, four metrics were selected and evaluated using the simulation results for each marker position. These metrics were later compared to the corresponding quantities obtained from a homogeneous simulation without a marker.

The first quantity evaluated was defined to measure how much the focus has shifted from its intended position. Let $\mathbf{m}_{\text{ref}}$ denote the coordinates of the maximum-pressure point extracted from a homogeneous simulation without a marker. This point will be referred as the focal-point, or simply the focus. Let also $\mathbf{m}_{\max}$ denote the coordinates of the maximum-pressure point extracted from a heterogeneous simulation with a marker. Then, the focus-shift was calculated using

$$d_{\text{shift}} = \|\mathbf{m}_{\max} - \mathbf{m}_{\text{ref}}\| \tag{2}$$

which is the Euclidean distance between $\mathbf{m}_{\max}$ and $\mathbf{m}_{\text{ref}}$.

The next set of metrics were based on the spatial-peak temporal-average (SPTA) intensity (I_{spta}). For each simulated marker position, this quantity was evaluated both at the coordinates of the homogeneous focus ($\mathbf{m}_{\text{ref}}$) and the coordinates of the maximum-pressure point ($\mathbf{m}_{\max}$). The two scalar values ($I_{\text{spta}}(\mathbf{m}_{\text{ref}}) \equiv I_{\text{focus}}$ and $I_{\text{spta}}(\mathbf{m}_{\max}) \equiv I_{\max}$) were obtained using

$$I_{\text{spta}}(\mathbf{m}) = \frac{1}{nT}\int_0^{nT} \frac{p^2(\mathbf{m}, t)}{\rho_0 c_0} dt \tag{3}$$

where $p(\mathbf{m}, t)$ is the pressure time series at the coordinates of the maximum-pressure point ($\mathbf{m} = \mathbf{m}_{\max}$) or the focus ($\mathbf{m} = \mathbf{m}_{\text{ref}}$), $n \geq 1$ is a positive integer and T is the period of the driving frequency. Evaluating Eq. (3) at $\mathbf{m}_{\text{ref}}$ for a homogeneous simulation gives the SPTA intensity of an uninterrupted beam denoted as I_{hom}. Comparison of I_{focus} and $I_{\max}$ with I_{hom} provides an indication of how much energy is redistributed due to the presence of the marker. It is noted that henceforth intensity will always refer to SPTA intensity.

Finally, to measure how the size of the focal region changes when the marker is included compared to the homogeneous simulation, the -6 dB focal volume was calculated for each simulation. This was obtained using

$$V_f = N \, dx \, dy \, dz \tag{4}$$

where N is the number of voxels for which $I_{\text{spta}}(\mathbf{m})$ was greater than 50% of $I_{\max}$ (the maximum intensity for that simulation). As the reference intensity changes for each simulation, this metric does not give a direct indication of the ablation volume. However, taken together with $I_{\max}$, it provides a useful indication of the volume over which the acoustic energy is distributed.

The focusing metrics were evaluated using the final five cycles of the pressure time-series and excluded the pressure time-series recorded within the marker volume. In order to reduce the size of the output from each simulation, the pressure time-series was recorded within a sub-region of the grid (see Table I) centered at the focal-point of the transducer ($\mathbf{m}_{\text{ref}}$). Even with this restriction in place, the output file size was approximately 0.5 TB per simulation.

Table I. Computational cost in terms of memory and simulation time associated with each grid-size.

	Homogeneous simulations				Heterogeneous simulations			
Grid-size (pt^3)	RAM[a] (GB)	Input[b] (MB)	Output[b] (GB)	Time[c] (dd:hh:mm)	RAM[a] (GB)	Input[b] (GB)	Output[b] (GB)	Time[c] (dd:hh:mm)
$S_1 = 384 \times 256 \times 512$	10.5	2.9	0.7	00:00:10	11.9	4.1	2.0	00:00:39
$S_2 = 768 \times 512 \times 1024$	37.2	20.3	8.5	00:02:06	48.5	28.3	30.3	00:10:16
$S_3 = 1152 \times 768 \times 1536$	108.4	70.0	37.1	00:12:01	141.7	95.8	144.9	02:13:23
$S_4 = 1536 \times 1024 \times 2048$	246.4	165.2	108.4	01:16:07	331.9	225.9	445.9	08:11:38
$S_5 = 2304 \times 1536 \times 3072$	830.7	554.6	508.7	06:19:51	—	—	—	—

[a] Random access memory (RAM) requirements at each grid-size. MB = 2^{20} bytes, GB = 2^{30} bytes.

[b] Size of the input and output files of the simulation. At S_1 the pressure is recorded in the output within a 96 × 96 × 192pt^3 volume. These dimensions increase proportionally to the grid-size, except from S_5 where a smaller volume was used.

[c] Time required for completing a single simulation. May vary depending on the HPC's workload. (dd:hh:mm) = (days:hours:minutes).

2.E. Convergence test for grid-size selection

The grid-size required for the simulations was established using a convergence test. In particular, because of nonlinear wave propagation, some of the energy from the fundamental frequency of the driving signal is shifted to higher harmonics. For these harmonics to be captured by the model, the physical domain must be appropriately discretized. For the numerical method used, waves can be accurately propagated close to the Nyquist limit of two grid-points per minimum wavelength. However, the energy at higher frequency harmonics is not known a priori. As a result, the choice of the grid-size will determine the number of harmonics that can be represented on the grid and thus the accuracy with which the nonlinearity is captured. On the other hand, increasing the grid-size translates to higher computational requirements in terms of memory and simulation time. Therefore, the selected grid-size was a compromise between the number of supported harmonics and the associated computational cost.

To determine the appropriate grid-size, homogeneous and heterogeneous simulations were performed at increasing grid dimensions. The configuration of these simulations was as described in Section 2, with the heterogeneous simulation including a single spherical gold marker between the focal-point and the transducer at (0,0,8) mm. Here, the marker position is reported with respect to the coordinates of the focal point. The grid-sizes tested are shown in Table I, noting that a heterogeneous simulation at S_5 was not performed due to the extremely high memory requirements (> 1 TB of RAM). The physical dimensions of the simulated domain were kept constant as described in Section 2.B. For each simulation the five final cycles of the pressure were extracted at the focal-point. The pressure time-series was then used to evaluate the frequency spectrum and the intensity at the focal-point. The results from the homogeneous convergence test are presented in Fig. 3 (analogous behavior was observed for the heterogeneous set of simulations).

As shown in Fig. 3, by increasing the grid-size, a higher number of harmonics is supported and the effects of nonlinear propagation are more accurately captured. At the lowest grid-size S_1 only the fundamental frequency is supported, thus, the pressure waveform is a pure sinusoid but with a reduced amplitude. As the grid-size increases, the higher frequencies supported capture the nonlinear steepening of the wave and the amplitude of the pressure waveform increases. As the grid-size increases the pressure waveform also converges. Beyond the 6th harmonic (24 MHz), which is close to the maximum frequency supported by S_4, the amplitude of the higher harmonics becomes extremely small in comparison to the fundamental frequency. Also, the intensity at S_5 changes only by 1.79% from its value at S_4. On the other hand, the computational cost increases dramatically when switching to S_5 ($\sim$7 days vs. $\sim$2 days) making multiple simulations impractical even on the large computing cluster used for this study. Having in mind the trade-offs described here, the remaining simulations were performed at S_4. It is noted that, with more than 10 billion grid points, the simulation at S_5 is one of the largest ultrasound simulations of its kind performed to date.[52]

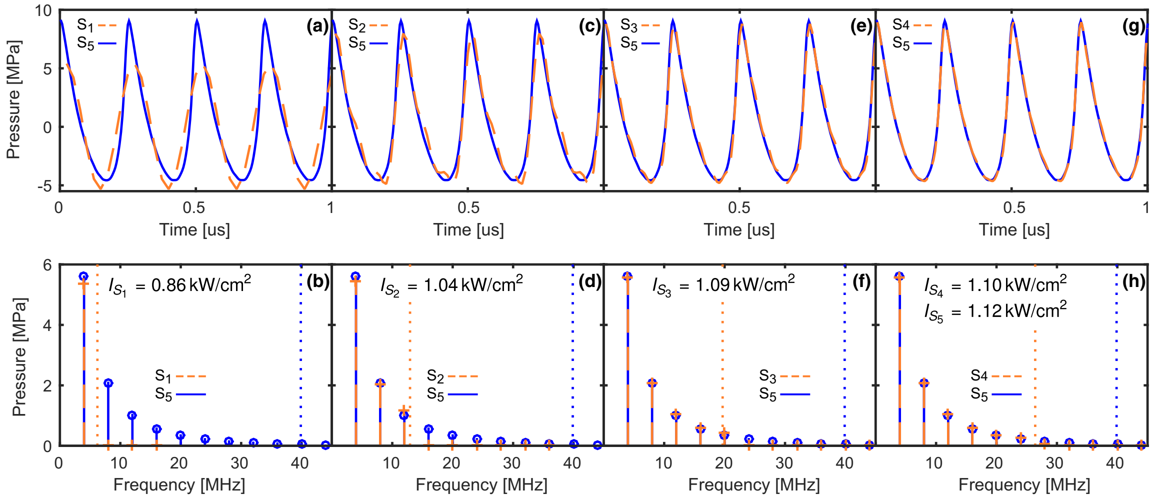


Fig. 3. Results of the convergence test performed in a homogeneous medium for selecting the appropriate grid-size of the simulations. The first row [(a), (c), (e) and (g)] shows the pressure time series at the focal-point and the second row [(b), (d), (f) and (h)] shows the respective frequency spectrum for each time series together with the SPTA intensity calculated using the time series. Each pair of plots compares the time and frequency response at a lower grid-size with the respective response at the highest grid-size S_5 simulated in this study. The grid dimensions are detailed in Table I. The vertical dotted lines indicate the maximum supported frequency at the corresponding grid-size. As the grid-size increases, a higher number of harmonics is supported and the effects of non-linear propagation are more accurately captured by the model. [Color figure can be viewed at wileyonlinelibrary.com]

2.F. Simulation deployment

The simulations were performed using the MPI version of k-Wave[52,53] on the IT4Innovations' Salomon HPC based at the National Supercomputing Center at VSB-Technical University of Ostrava in the Czech Republic. The actual hardware utilized for each simulation comprised of 144 cores and 768 GB of RAM (6 nodes with two Intel Xeon E5-2680v3 processors, each equipped with 24 cores and 128 GB RAM, interconnected by a 7D Enhanced hypercube Infiniband network). For the S_5 simulation, 9 nodes were utilized. Table I summarizes the memory and simulation-time requirements for a single homogeneous and heterogeneous simulation at each grid-size. At S_4, the output of a single heterogeneous simulation was 445.9 GB. With a total of 143 marker positions tested, the simulations generated ∼63 TB of output data and required ∼5 million core-hours to run. After the completion of the simulations, the output data was processed in Matlab to evaluate the various metrics quantifying the effect of the marker as described in Section 2.D.

3. SIMULATION

A total of 143 marker positions were simulated: 113 with a spherical marker and 30 with a cylindrical marker at three orientations. Figure 4 provides a visual description of the simulations performed to determine the marker effect on the HIFU beam. Figure 4(a) shows the maximum pressure field as recorded across the whole domain when the HIFU beam propagates in a homogeneous medium. The inset is a visualization of the −6 dB focal volume. The metrics extracted from this simulation serve as a reference for comparison to assess how placing a marker in the path of the beam deteriorates the focusing. Figure 4(b)–4(d) demonstrate how the maximum pressure field and the focal volume changes when a marker is introduced and gradually moved away from the focal-point. Figure 4(b) shows the dramatic effect of the marker when placed very close to the focal-point, while Figs. 4(c)–4(d) illustrate how the marker effect decreases as its distance from the focal-point increases. Finally, Fig. 4(d) shows that beyond a certain distance, focusing is re-established with the marker effect becoming less pronounced.

The metrics extracted for each marker position can be used to quantitatively study the marker's effect. Figures 5 and 6 show the evaluated metrics for the spherical and cylindrical gold markers respectively. For both sets of plots, the metrics are plotted along the axes connecting the focal-point to the transducer. The metrics are plotted with respect to the distance of the marker from the focal-point, with the positive direction indicating that the marker is positioned toward the transducer.

Figures 5(a) and 5(b) show how the intensity (which is proportional to the rate of heat deposition) changes with marker position at the maximum-pressure point ($\mathbf{m}_{\max}$) and at the focus ($\mathbf{m}_{\text{ref}}$), respectively. When the spherical marker is positioned very close to the focus, the intensity reduces dramatically. However, as the marker moves away from the focal-point and towards the transducer, the intensity increases. As an indication, at 11.5 mm and 8.5 mm the intensity at the focal-point is reduced by 10% and 20% respectively. Moving the marker further away from the focus, both intensities continue to gradually increase and eventually converge to approximately I_{hom}, indicating that focusing has been re-established fully.

A slightly different behavior is observed when the marker is positioned at exactly the focus ($\mathbf{m}_{\text{ref}}$) of the transducer. It is clear from Fig. 5(b) that practically no energy reaches the intended focal position. On the other hand, Fig. 5(a) shows that the maximum intensity more than doubles due to the reflections caused by the marker, which redirect the energy to the pre-focal region. This is due to the large impedance difference between the background medium (prostate) and the

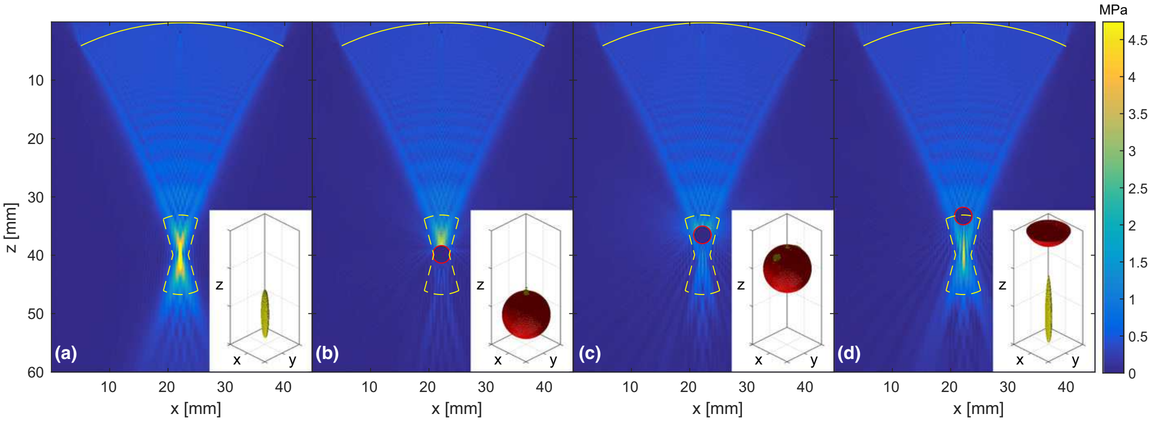


Fig. 4. Maximum pressure field of (a) a homogeneous simulation and (b)–(d) three heterogeneous simulations for three marker positions. The insets have dimensions 3×3×8 mm and are visualizations of the −6 dB focal-volume (dashed outline) evaluated using Eq. (4) and the spherical marker (solid-line circle). For each marker position, the metrics in Section 2.D. were evaluated and compared with the corresponding reference values of the homogeneous simulation. When the marker is inside the region indicated by the dashed line, the intensity at the focus drops by more than 30%. [Color figure can be viewed at wileyonlinelibrary.com]

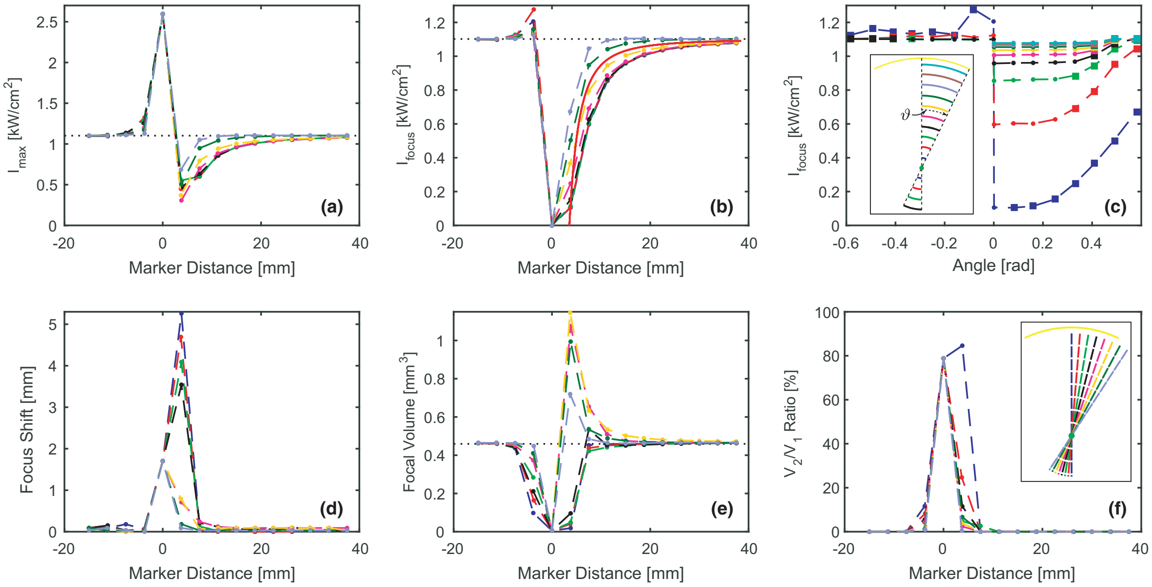


FIG. 5. Metrics quantifying the effect on the HIFU beam of a 3 mm gold spherical marker. All the metrics, except in (c), are plotted with respect to marker distance from the focus, with each curve corresponding to one of the radial axes shown in the inset of (f). The horizontal dotted line corresponds to the respective metric value obtained from a homogeneous simulation. (a) and (b) show respectively the intensity evaluated using Eq. (3) at the maximum-pressure point ($\mathbf{m}_{\max}$) and at the focus ($\mathbf{m}_{\text{ref}}$), with the solid-line in (b) showing the intensity as predicted by the geometric model in Eq. (5). (d) shows how the distance between $\mathbf{m}_{\max}$ and $\mathbf{m}_{\text{ref}}$ changes as calculated with Eq. (2). (e) is the focal-volume with marker position as given by Eq. (4) and (f) is the ratio between the second (V_2) and first (V_1) largest volumes in the focal region. Finally, (c) shows the intensity at the focus but plotted against the angle (ϑ) between the central radial axis and the axis on which the marker lies. Here, each curve corresponds to the group of markers at a constant distance from the focus as indicated in the inset. [Color figure can be viewed at wileyonlinelibrary.com]

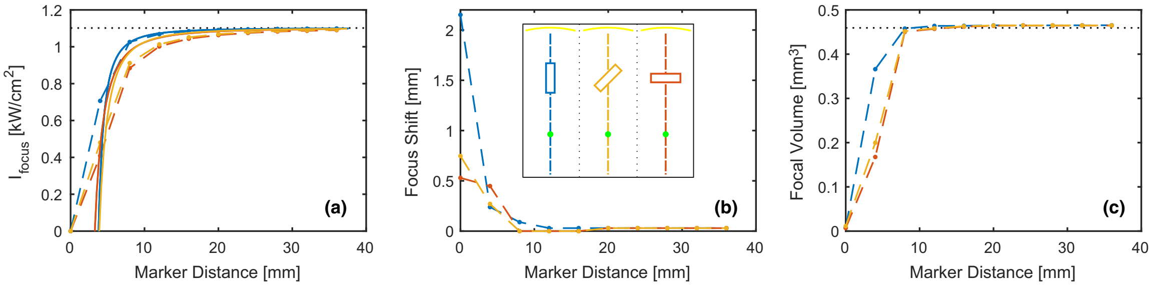


FIG. 6. Metrics quantifying the effect on the HIFU beam of a 3×1 mm gold cylindrical marker at different orientations. All the metrics, are plotted with respect to marker distance from the focus, with each curve corresponding to one of the orientations shown in the inset of (b). The horizontal dotted line corresponds to the respective metric value obtained from a homogeneous simulation. (a) is the intensity at the focus ($\mathbf{m}_{\text{ref}}$) given by Eq. (3), with the solid-lines showing the intensity as predicted by the geometric model in Eq. (5) for each orientation. (b) is the change in the distance between $\mathbf{m}_{\max}$ and $\mathbf{m}_{\text{ref}}$ evaluated using Eq. (2) and (c) is the focal-volume with marker position as given by Eq. (4). [Color figure can be viewed at wileyonlinelibrary.com]

marker (gold), which results in a high reflection coefficient (R = 0.91). When the marker is gradually shifted in the negative direction, the intensity at the maximum $I_{\max}$ and the focus I_{focus} quickly converge to the homogeneous value I_{hom} with no significant reflections observed beyond approximately −4 mm.

Figure 5(d) illustrates the effect of the marker on the position of the maximum pressure ($\mathbf{m}_{\max}$) relative to the focal-point ($\mathbf{m}_{\text{ref}}$). When the marker is positioned at a distance between −4 mm and 8 mm from the focal-point, a shift in the focus is observed of up to approximately 5.5 mm. The distortion caused by the marker can also be observed by looking at the variation in the size of the focal volume in Fig. 5(e) and the insets in Fig. 4. When the marker is placed at a distance from the focus between −4 mm and 8 mm, the focal volume decreases as its position moves closer to the focal-point, indicating its negative effect on the beam. The markers placed along the four outer radial axes seem to diverge from this behavior. As those markers move towards the focus from the positive direction, the decrease in focal volume is first preceded by a sharp increase. This is likely due to refocusing caused by reflections and diffraction around the marker. Moving the marker beyond -4 mm and 8 mm causes the focal volume to gradually converge back to its homogeneous value, indicating that focusing is re-established.

Placing the marker close to the focus also causes the focal-volume to split from a single region [Fig. 4(a)] into multiple smaller volumes of high pressure [Fig. 4(b)], which may induce heating at undesired locations. Figure 5(f) shows the ratio between the second (V_2) and first (V_1) largest connected components of the -6 dB volumes for each marker position. The ratio between the two volumes increases as the marker is positioned closer to the focus. When the marker is placed away from the focal-point the size of V_2 reduces to zero demonstrating that any secondary regions of high pressure are eliminated.

Figure 5(c) offers an alternative perspective on the effect of the marker on the intensity at the focus. In this case, the markers are grouped together with respect to their distance from the intended focus. Therefore, each curve corresponds to a fixed distance from the focus. The intensity is plotted with respect to the angle between the z-axis passing through the focus and the radius connecting the center of the marker with the focus. The markers outside the HIFU beam are denoted with squares whereas those inside the beam are denoted with dots. This plot shows that, as long as the marker is positioned inside the HIFU beam, its effect on the intensity remains approximately the same when its distance from the intended focus is kept constant. The figure also demonstrates the large reduction in intensity due to the markers closer to the focus (positive angles), however, as the angle increases, their effect on the intensity reduces since they move outside the HIFU beam. For the markers beyond the focus (negative angles), a small increase in intensity can be seen which reduces as their distance from the focus increases in the negative direction. The analogous behavior is observed for the focus shift and focal volume.

To investigate the effect of marker shape, another set of simulations was performed using a single gold cylindrical marker. A total of 30 simulations were executed: 10 marker positions were simulated along the z-axis passing through the focal-point with 3 orientations for each position as shown in Fig. 2(b)–2(d). The orientation in Fig. 2(b) is the most likely to be encountered in practice because of the procedure with which the markers are inserted. The simulations were restricted to a single radial axis in order to limit the number of simulations executed. This restriction was justified based on the observation that the effect of the spherical marker remains constant at a fixed distance from the focal-point as demonstrated in Fig. 5(c).

For each position of the cylindrical marker, the same set of metrics were calculated. Comparison of the plots in Fig. 6 with the corresponding plots in Fig. 5 suggests that the cylindrical marker distorts the HIFU beam in the same manner as the spherical marker. Namely, as the marker moves closer to the focal-point, the intensity and focal-volume decrease while the shift in the focus increases. It is also interesting to observe that marker orientation has an effect. For example, in terms of the intensity at the focal-point, the orientation parallel to the beam's axis has the smallest impact since the surface area encountered by the wave is the smallest, but it has the largest focus shift since the maximum pressure point occurs close to the base of the marker furthest from the focus. For the remaining two orientations, the metrics in Fig. 6 vary in a similar manner. This is likely due to their projected areas on the HIFU beam being similar.

4. GEOMETRIC MODEL

The results discussed above suggest that the distortion introduced by the marker is dominated by strong reflections. This is not surprising due to the large density difference between the background medium and gold, which results in a high reflection coefficient at the interface of the two materials. Additionally, the impact of the different marker orientations suggests a dependence on the surface area of the marker encountered by the wave. Based on these observations and with the aim of providing a faster and more efficient method for estimating the effect of different markers, a simple analytical model was derived which evaluates the focal intensity by considering the effect of a single marker.

Figure 7 defines the various parameters of the model assuming a spherical marker. More specifically, it shows the HIFU beam of the geometric model (solid-yellow line), whose size is determined by the focal length of the transducer R_t, its width W_t and its length L_t. Figure 7 also shows a cross-section of the beam with a spherical-strip shape (solid-green line), which is tangential to the point on the marker furthest from the focus. The cross-section has a length L_w and width W_w with a radius R_w. The circle indicates a cross-section of the spherical marker with diameter d and its center at a distance r from the focal-point. It is noted that, although a spherical marker is considered as an example here, the model can be adapted to any other shape.

With reference to Fig. 7, let I_{hom} denote the intensity at the focal-point of an uninterrupted beam (evaluated from a homogeneous simulation), A_w the total area of the beam's cross-section and A_m the projected area of the marker on the cross-section (red-solid line). Then, the intensity at the focus ($\mathbf{m}_{\text{ref}}$) when the beam is obstructed by a marker is approximately given by

$$I_{\text{focus}} \approx I_{\text{hom}}\left(1 - \frac{A_m}{A_w}\right). \tag{5}$$

The values of A_m and A_w vary according to the distance of the marker from the focal-point. Additionally, A_m changes depending on the marker's shape and its orientation. Thus, evaluating Eq. (5) requires a single homogeneous simulation to obtain I_{hom} and then calculation of the areas A_m and A_w. This is a significant improvement in terms of computation time since, after obtaining I_{hom} from a single homogeneous simulation, the time required for evaluating Eq. (5) is

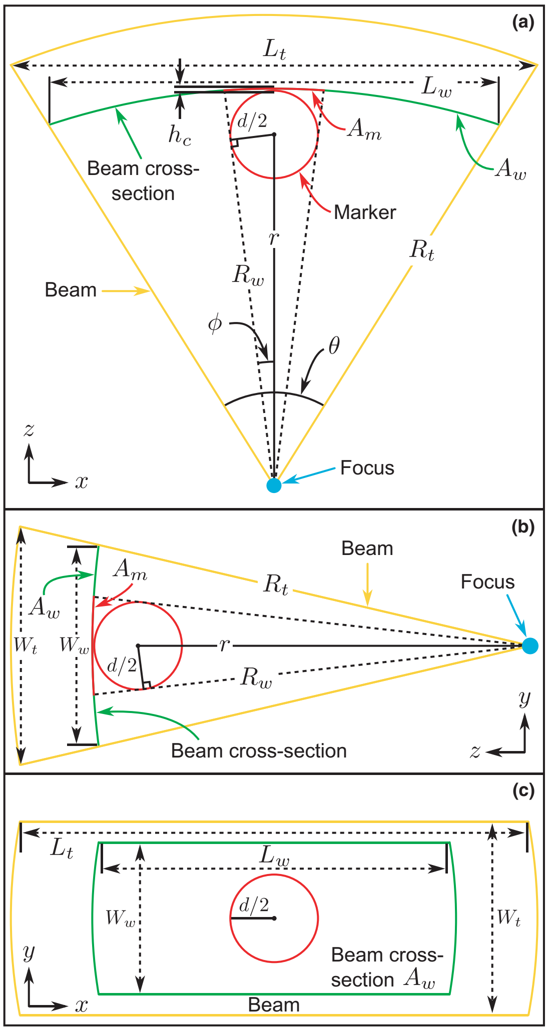


FIG. 7. The parameters of the geometric model defined in Eq. (5) assuming a spherical marker. The parameters are shown on (a) the *xz*, (b) the *yz* and (c) *xy* planes. [Color figure can be viewed at wileyonlinelibrary.com]

negligible compared to simulations. The model omits wave propagation due to diffraction or refraction and accounts only for perfect reflection. It also assumes that the energy lost only depends on the shape of the marker, its distance from the transducer's focus and the shape of the cross-section of the beam tangential to the marker.

The total area of the beam's cross-section A_w can be approximated as the area of a sector of a spherical strip

$$A_w = \theta R_w W_w \tag{6}$$

where θ is the angle defining the sector of the strip in radians, W_w is the width of the strip and R_w is the radius of the sphere on which the strip lies. As shown in Fig. 7, in this example $\theta = 2\arcsin(L_t/(2R_t))$ and $R_w = r + d/2$, since a spherical marker is assumed here, and $W_w = R_w W_t/R_t = (r + d/2)W_t/R_t$. Substituting in Eq. (6) yields

$$A_w = \frac{2W_t(r+d/2)^2 \arcsin(L_t/(2R_t))}{R_t}. \tag{7}$$

Calculating A_w for a different marker shape only requires obtaining a value for R_w with the rest of the steps remaining unchanged.

For a spherical marker, its projected area A_m on the tangential cross-section of the beam, as indicated in Fig. 7 (red-solid line), has a spherical-cap shape whose surface area is given by

$$A_m = 2\pi R_w h_c \tag{8}$$

where R_w is the radius of the sphere on which the spherical cap lies and h_c is the cap's height. In this example, $R_w = r + d/2$ as explained above and $h_c = R_w - R_w \cos\phi$, where $\phi = \arcsin(d/(2r))$. Substituting in Eq. (8) then gives

$$A_m = 2\pi\left(r + \frac{d}{2}\right)^2\left[1 - \cos\left(\arcsin\left(\frac{d}{2r}\right)\right)\right]. \tag{9}$$

Analogous arguments hold for evaluating the projected area A_m for the cylindrical marker, or any other marker shape.

The effect of both the spherical and cylindrical marker on the intensity at the focal-point as predicted by Eq. (5) is compared with the simulation results in Figs. 5(b) and 6(b) respectively (denoted by the solid-lines). To quantify the agreement of the model with the simulated intensity at the focus (I_{focus}), the root-mean-square relative error (RMSRE) was evaluated along the beam axis to which the other axial responses converge to and it is shown in Table II. The model slightly underestimates the marker's effect and becomes less accurate for marker positions closer to the focus, but overall it confirms the assumption that

TABLE II. Exclusion zone radius evaluated using the geometric model defined in Eq. (5) for different types of markers and orientations.

Marker type	Dimensions (mm)[a]	Orientation (degrees)[b]	Distance (mm)[c]	RMSRE (%)[d]
Spherical	1	—	2.3	—
Spherical	2	—	4.5	—
Spherical	3	—	6.8	15.7
Cylindrical	3×1	0°	5.0	8.1
Cylindrical	3×1	45°	5.7	18.2
Cylindrical	3×1	90°	5.3	22.8

[a] Dimensions are: diameter for spherical marker, height × diameter for cylindrical marker.

[b] Angle measured on *xz* plane between the central radial axis connecting the focus to the transducer and the symmetry axis of the cylinder.

[c] The distance from the focal-point towards the transducer at which the intensity drops by 30%, which is equivalent to approximately a 50% reduction in lesion volume, calculated using Eq. (5).

[d] Root-mean-square relative error (RMSRE) of the intensity at the focus (I_{focus}) between the simulated values along the beam axis and those evaluated using the geometric model.

reflections are the dominating effect causing the observed distortion. This suggests that Eq. (5) can be used to approximate the marker effect for arbitrary shapes without the necessity of performing time-consuming and computationally intensive simulations.

Table II provides an indicative list of distances from the focal-point at which the intensity drops by 30%, which is equivalent to approximately a 50% reduction in lesion volume, for different dimensions of cylindrical and spherical markers. The listed distances were evaluated using the analytical model in Eq. (5). Considering that the effect of a marker within the ultrasound beam remains the same at a fixed distance from the focal-point, Table II may be used to define a region around the focal-point within which the marker's impact on the beam is significant and thus may affect the delivery of the HIFU treatment. An example of such an *exclusion zone* is shown in Fig. 4(a) for the 3 mm spherical marker, where the radius of the exclusion zone from the focus was extracted from Table II and its lateral width was evaluated using four times the beam width ($4 \times 1.41 c_0/f_0 R_t/L_t \approx 2.6$ mm). The region defined by these boundaries may be used to evaluate whether a particular region in the prostate can be effectively treated using transrectal HIFU when a marker obstructs the beam.

5. CASE STUDIES

The results presented in the previous sections suggest that the marker distorts the HIFU beam with its effect increasing the closer it is positioned to the focus. To examine how these results might be applied in a clinical setting, four datasets have been retrospectively selected of patients with recurrent prostate cancer after failed EBRT, which were eligible for salvage-HIFU at UCLH. Three cases were selected in which the presence of the marker may affect the treatment and one case in which the marker is not expected to impose any risk. As shown in Fig. 8, for each patient three images from different modalities are presented co-registered. In each of these images, a contour identifies the region targeted during the treatment and a dot indicates the assumed position of a 3×1 mm cylindrical marker. Due to the difficulty in locating the exact marker position of the medical images, the marker positions were added in software retrospectively based on standard insertion protocols. The outline of the exclusion zone is also shown with its radius extracted from Table II for the cylindrical marker at 45° and its orientation determined by the likely direction of propagation of the HIFU beam indicated by the dashed line. Table III provides details of the four case-studies including the post-operative outcome with regards to any recurrence and its position for comparison with the modelled outcome.

For the first patient, the marker is close to the rectal wall and inside the region targeted during the treatment. Having in mind the strong reflections induced when positioned close to the focus, the marker may cause two side-effects. Firstly, the reflected wave may cause secondary regions of high pressure on the rectal wall, and secondly, the region in the top part of the exclusion zone may not receive enough energy to be adequately treated. Reviewing the patient's post-operative outcome confirmed (Table III), a recurrence in the lateral position of the lesion consistent with possible disruption from the position of the fiducial marker. In the second example, the marker is positioned near the upper edge of the region to be treated. In this case, the reflections due to the marker may cause excessive heating of regions in the bottom part of the exclusion zone, although this is unlikely to affect overall treatment efficacy. The clinical outcome was once again consistent with the modelling outcome. Although the patient developed a recurrence it was in the midline, some distance away from the marker, and thus the recurrence is likely due to either an inadequate surgical margin taken during the HIFU treatment or due to incomplete cell kill. The third patient, demonstrates another extreme case in which the marker is positioned near the lower bound of the treatment area. Here the reflected wave may induce heating in areas outside the desired treatment region within the bottom part of exclusion zone and leave the top part of the exclusion zone inadequately treated. In this case, recurrence of the tumor was again observed which, although not entirely in the predicted field of recurrence, it may have been influenced by the presence of the marker. In the final example, the marker is positioned away from the intended treatment region, thus, it is not expected to affect the treatment. This is confirmed by the post-HIFU MRI with no residual tumor within the treatment zone.

6. SUMMARY AND DISCUSSION

Gold fiducial markers are commonly used as part of the IGRT procedure during EBRT for men with localized or locally advanced prostate cancer. These markers remain permanently implanted in the prostate. Thus, they may affect the efficacy and safety of the subsequent use of HIFU treatment as a salvage therapy in case of local cancer recurrence. This work investigated the impact on the HIFU beam of a single spherical or cylindrical gold fiducial marker through a series of simulations performed using the open-source k-Wave Toolbox. For each marker configuration, four metrics were evaluated to quantify its impact on the beam. By comparing these metrics with their corresponding values from a homogeneous simulation, it is evident that the distortion introduced by the marker increases as its distance from the transducer's focus decreases and depends on the marker's shape.

Assuming perfect reflections, an analytical model was developed based on geometric arguments, which estimates the impact of the marker on the intensity at the focus. Using the model, which is in good agreement with the simulated results, it is possible to identify the boundaries of a region around the focus within which the presence of a marker will lead to an intensity drop below an acceptable threshold. For

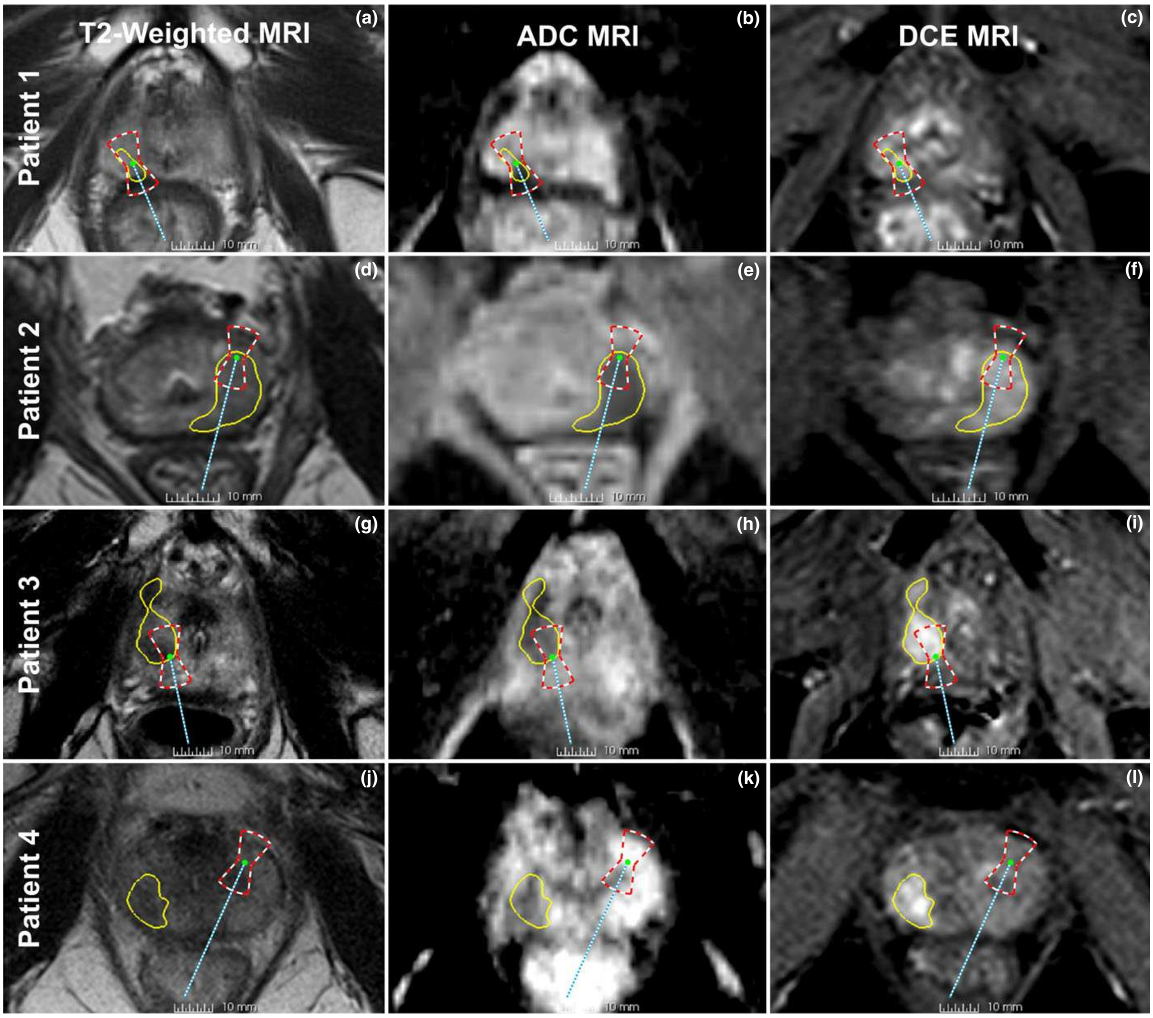


FIG. 8. Diagnostic images of four prostate-cancer patients. From left to right, the images are T2-weighted, apparent diffusion coefficient (ADC), and dynamic-contrast enhanced (DCE) magnetic resonance images (MRI). For each patient the three images are co-registered and show the targeted treatment region (solid-line contour), the position of a 3×1 mm cylindrical gold marker (dot marker), the exclusion zone (dashed outline) for a cylindrical marker at 45° and the direction of propagation of the HIFU-beam (dashed straight line). The exclusion zone is positioned such that the focus of the transducer coincides with the position of the marker and it is aligned with the direction of propagation of the HIFU beam. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE III. Clinical characteristics of the patients with recurrent prostate cancer after failed EBRT included in the case study.

Case	Cancer stage	Location[a]	Lesion volume	Salvage HIFU treatment plan	Outcome	Possible marker effect
Case 1	Intermediate risk Gleason 3+4 prostate cancer	Right PZ apex	1 cc	Quadrant ablation, 4 cm and 3 cm blocks	Infield failure - lateral recurrence	Yes
Case 2	Intermediate risk Gleason 3+4 prostate cancer	Left PZ mid to apex extending across midline	2.3 cc	Extended left hemi-ablation	HIFU infield failure - midline recurrence at edge of treatment zone	No
Case 3	Intermediate risk Gleason 3+4 prostate cancer	Right PZ	1.1 cc	Right subtotal hemi-ablation in 4 cm, 3 cm and 3 cm blocks	Infield failure at site of marker	Yes
Case 4	Intermediate risk Gleason 3+4 prostate cancer	Right anterior TZ	0.8 cc	Right quadrant ablation in 4 cm and 3 cm blocks	No recurrence	No

[a] PZ: peripheral zone, TZ: transitional zone.

example, as shown in Table II, placing a 3×1 mm marker within approximately 5 mm of the focus in the pre-focal region will induce an intensity drop of more than 30% of the homogeneous value, which will cause a reduction in the volume of the ablated region. As examined in Fig. 8 using scans of prostate cancer patients, this region around the marker can be divided into two parts: an undertreated region due to less energy arriving above the focus and an overtreated region due to reflections below the focus. Both effects may be undesirable depending on the location of the marker. Moreover, there is evidence that the region of recurrence after EBRT is the main tumor (index lesion).[54] Therefore, the results of this study may justify avoiding the index lesion during fiducial marker placement. Although a degree of accuracy was observed between the position of the marker and the site of recurrence, the cohort consisted only of four patients, thus extracting firm conclusions is difficult. Further work using larger retrospective and prospective cohorts is necessary to further develop and validate the model to allow its utilization in clinical practice. Such a study will aim to reveal the percentage of affected patients by the results of this study and whether the marker's impact can justify the exclusion of some patients from salvage-HIFU or the revision of the placement protocol of fiducial markers during EBRT. Experimental measurements on ex vivo tissue phantoms with implanted markers are also needed to confirm these results.

While investigating the distortion introduced by the marker, the study has omitted some additional factors which may affect the significance of the marker's impact on the treatment. Firstly, as discussed in Section 2.A., the Sonablate 500 probe, on which the transducer model was based, includes an imaging transducer which was not taken into account in the simulations. Although this is expected to affect the intensity at the focus (for the same source surface intensity), it is unlikely to change the distortion introduced by the marker. Similarly, since the operation of the other existing transrectal and transurethral HIFU systems is based on the same principles, using a different transducer model is not expected to affect the behavior of the marker observed here. Second, only the effect on the intensity at the focus (which correlates with heating rate) has been investigated. However, in practice, additional heating may occur due to absorption within the marker and viscous relative motion between the marker and surrounding tissue. These effects, combined with the multiple sonications used during a treatment, may help to counteract the reduced heating due to the lower intensity. It is also unclear from this work whether cavitation, which is triggered by large negative pressures, is reduced due to the presence of the marker. Finally, although other types of markers exist (see Section 1), only gold markers have been considered. However, given that all the materials used have greater impedance than the prostate, using other types of markers is unlikely to change the behavior observed here. A scenario in which the treatment may be severely affected is when a large number of marker-like elements are introduced in the prostate. Such a situation occurs during salvage-HIFU after failed (low-dose) brachytherapy, where a large number of seeds are permanently implanted in the prostate. Extending the insights of this work for the brachytherapy case and exploring other factors which may affect the distortion introduced by the marker, will be the subject of future work.

ACKNOWLEDGMENTS

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) in the United Kingdom, grant numbers EP/L020262/1 and EP/M011119/1. Computational resources were supported by the Partnership For Advanced Computing in Europe (PRACE, PROS-HIFU), IT4Innovations National Supercomputing Center, and the UCL Legion High Performance Computing Facility (Legion@UCL). We acknowledge that the results of this research have been achieved using the PRACE-3IP project (FP7 RI-312763) resource Supernova based in Poland at the Wroclaw Center for Networking and Supercomputing. This work was also supported by The Ministry of Education, Youth and Sports, Czech Republic from the Large Infrastructures for Research, Experimental Development and Innovations project "IT4Innovations National Supercomputing Center - LM2015070". Jiri Jaros is financed from the SoMoPro II Programme, co-financed by the European Union and the South-Moravian Region. This work reflects only the author's view and the European Union is not liable for any use that may be made of the information contained therein. Eli Gibson was supported by a Postdoctoral Fellowship awarded by the Canadian Institute for Health Research (CIHR). We thank Dr. Naren Sanghvi of SonaCare Medical LLC for providing technical details on the Sonacare 500 HIFU device.

CONFLICT OF INTEREST

The authors have no relevant conflicts of interest to disclose.

[a]Author to whom correspondence should be addressed. Electronic mail: p.s.georgiou@ucl.ac.uk.

REFERENCES

1. Ferlay J, Soerjomataram I, Ervik M, et al. GLOBOCAN 2012 v1.0, Cancer Incidence and Mortality World-wide: IARC CancerBase No. 11, 2013.
2. European Cancer Observatory (ECO). Cancer Fact Sheets, 2016.
3. American Cancer Society. Cancer Statistics Center, 2016.
4. Torre LA, Bray F, Siegel RL, Ferlay J, Lortet-Tieulent J, Jemal A. Global cancer statistics, 2012. *CA: Cancer J Clin*. 2015;65:87–108.
5. Warmuth M, Johansson T, Mad P. Systematic review of the efficacy and safety of high-intensity focussed ultrasound for the primary and salvage treatment of prostate cancer. *Eur Urol*. 2010;58:803–815.
6. Mottet N, Bellmunt J, Briers E, et al. EAU - ESTRO - SIOG Guidelines on Prostate Cancer, Tech. Rep. 2016.
7. Thompson I, Thrasher JB, Aus G, et al. Guideline for the management of clinically localized prostate cancer: 2007 update. *J Urol*. 2007;177: 2106–2131.

8. Borghede G, Aldenborg F, Wurzinger E, Johansson K-A, Hedelin H. Analysis of the local control in lymph-node staged localized prostate cancer treated by external beam radiotherapy, assessed by digital rectal examination, serum prostate-specific antigen and biopsy. *BJU Int*. 1997;80:247–255.
9. Boorjian SA, Karnes RJ, Viterbo R, et al. Long-term survival after radical prostatectomy versus external-beam radiotherapy for patients with high-risk prostate cancer. *Cancer*. 2011;117:2883–2891.
10. Grimm P, Billiet I, Bostwick D, et al. Comparative analysis of prostate-specific antigen free survival outcomes for patients with low, intermediate and high risk prostate cancer treatment by radical therapy. Results from the prostate cancer results study group. *BJU Int*. 2012; 109:22–29.
11. Punnen S, Cowan JE, Chan JM, Carroll PR, Cooperberg MR. Long-term Health-related Quality of Life After Primary Treatment for Localized Prostate Cancer: Results from the CaPSURE Registry. *Eur Urol*. 2015;68:600–608.
12. National Cancer Intelligence Network, Treatment routes in prostate cancer: urological cancers SSCRG. Tech. Rep. (NCIN) 2012.
13. Zelefsky MJ, Reuter VE, Fuks Z, Scardino P, Shippy A. Influence of local tumor control on distant metastases and cancer related mortality after external beam radiotherapy for prostate cancer. *Journal Urol*. 2008;179:1368–1373.
14. Murat F-J, Poissonnier L, Rabilloud M, et al. Mid-term results demonstrate salvage high-intensity focused ultrasound (HIFU) as an effective and acceptably morbid salvage treatment option for locally radiorecurrent prostate cancer. *Eur Urol*. 2009;55:640–649.
15. Kimura M, Mouraviev V, Tsivian M, Mayes JM, Satoh T, Polascik TJ. Current salvage methods for recurrent prostate cancer after failure of primary radiotherapy. *BJU Int*. 2010;105:191–201.
16. Shoji S, Nakano M, Omata T, et al. Salvage high-intensity focused ultrasound for the recurrent prostate cancer after radiotherapy. *AIP Conference Proceedings*. 2010;1215:234–238.
17. Agarwal PK, Sadetsky N, Konety BR, Resnick MI, Carroll PR. Treatment failure after primary and salvage therapy for prostate cancer. *Cancer*. 2008;112:307–314.
18. Touma NJ, Izawa JI, Chin JL. Current status of local salvage therapies following radiation failure for prostate cancer. *J Urol*. 2005;173:373–379.
19. Berge V, Baco E, Karlsen SJ. A prospective study of salvage high-intensity focused ultrasound for locally radiorecurrent prostate cancer: early results. *Scand J Urol Nephrol*. 2010;44:223–227.
20. Chalasani V, Martinez CH, Lim D, Chin J. Salvage HIFU for recurrent prostate cancer after radiotherapy. *Prostate Cancer and Prostatic Dis*. 2009;12:124–129.
21. Murota-Kawano A, Nakano M, Hongo S, Shoji S, Nagata Y, Uchida T. Salvage high-intensity focused ultrasound for biopsy-confirmed local recurrence of prostate cancer after radical prostatectomy. *BJU Int*. 2010;105:1642–1645.
22. Sanderson KM, Penson DF, Cai J, et al. Salvage radical prostatectomy: quality of life outcomes and long-term oncological control of radiorecurrent prostate cancer. *J Urol*. 2006;176:2025–2032.
23. Crouzet S, Murat F-J, Pommier P, et al. Locally recurrent prostate cancer after initial radiation therapy: early salvage high-intensity focused ultrasound improves oncologic outcomes. *Radiother Oncol*. 2012; 105:198–202.
24. Chapelon J-Y, Rouvière O, Crouzet S, Gelet A. Prostate focused ultrasound therapy. In: Escoffre J-M, Bouakaz A, eds. *Therapeutic Ultrasound*. Switzerland: Springer International Publishing; 2016:21–41.
25. Ahmed HU, Zacharakis E, Dudderidge T, et al. High-intensity-focused ultrasound in the treatment of primary prostate cancer: the first UK series. *Br J Cancer*. 2009;101:19–26.
26. Crouzet S, Chapelon JY, Rouvière O, et al. Whole-gland ablation of localized prostate cancer with high-intensity focused ultrasound: oncologic outcomes and morbidity in 1002 patients. *Eur Urol*. 2014;65:907–914.
27. Song W, Jung US, Suh YS, et al. High-intensity focused ultrasound as salvage therapy for patients with recurrent prostate cancer after radiotherapy. *Korean J Urol*. 2014;55:91.
28. Uchida T, Shoji S, Nakano M, et al. High-intensity focused ultrasound as salvage therapy for patients with recurrent prostate cancer after external beam radiation, brachytherapy or proton therapy. *BJU Int*. 2011; 107:378–382.
29. Zacharakis E, Ahmed HU, Ishaq A, et al. The feasibility and safety of high-intensity focused ultrasound as salvage therapy for recurrent prostate cancer following external beam radiotherapy. *BJU Int*. 2008;102: 786–792.
30. Fuller CD, Scarbrough TJ. Fiducial markers in image-guided radiotherapy of the prostate. *Oncol Hematol Rev (US)*. 2006;00:75.
31. Kupelian PA, Langen KM, Willoughby TR, Zeidan OA, Meeks SL. Image-guided radiotherapy for localized prostate cancer: treating a moving target. *Semin Radiat Oncol*. 2008;18:58–66.
32. Seip R, Shaeffer D, Lawrence P, et al. Feasibility study for the treatment of brachytherapy failure prostate cancer using high-intensity focused ultrasound. In *Third International Symposium on Therapeutic Ultrasound*; 2003.
33. Chapman AT, Rivens IH, Thompson AC, ter Haar GR. High intensity focused ultrasound (HIFU) as a salvage treatment for recurrent prostate cancer after brachytherapy - a feasibility study. In: *AIP Conference Proceedings*. Vol. 911. Melville, NY: AIP; 2007:405–410.
34. Chan MF, Cohen GN, Deasy JO. Qualitative evaluation of fiducial markers for radiotherapy imaging. *Technol Cancer Res Treat*. 2015;14:298–304.
35. Habermehl D, Henkner K, Ecker S, Jakel O, Debus J, Combs SE. Evaluation of different fiducial markers for image-guided radiotherapy and particle therapy. *J Radiat Res*. 2013;54:i61–i68.
36. Hellinger JC, Blacksberg S, Haas J, Melnick J. Interventional uroradiology in the management of prostate cancer. *Appl Radiol*. 2015;44:40–41.
37. Rudat V, Nour A, Hammoud M, Alaradi A, Mohammed A. Image-guided intensity-modulated radiotherapy of prostate cancer. *Strahlentherapie und Onkologie*. 2016;192:109–117.
38. Ye JC, Qureshi MM, Clancy P, Dise LN, Willins J, Hirsch AE. Daily patient setup error in prostate image guided radiation therapy with fiducial-based kilovoltage onboard imaging and conebeam computed tomography. *Quant Imaging Med Surg*. 2015;5:665–672.
39. van der Heide UA, Kotte AN, Dehnad H, Hofman P, Lagenijk JJ, van Vulpen M. Analysis of fiducial marker-based position verification in the external beam radiotherapy of patients with prostate cancer. *Radiother Oncol*. 2007;82:38–45.
40. Kotte AN, Hofman P, Lagendijk JJ, van Vulpen M, van der Heide UA. Intrafraction motion of the prostate during external-beam radiation therapy: analysis of 427 patients with implanted fiducial markers. *Int J Radiat Oncol Biol Phys*. 2007;69:419–425.
41. Ng M, Brown E, Williams A, Chao M, Lawrentschuk N, Chee R. Fiducial markers and spacers in prostate radiotherapy: current applications. *BJU Int*. 2014;113:13–20.
42. Shirato H, Harada T, Harabayashi T, et al. Feasibility of insertion/implantation of 2.0-mm-diameter gold internal fiducial markers for precise setup and real-time tumor tracking in radiotherapy. *Int J Radiat Oncol Biol Phys*. 2003;56:240–247.
43. Graf R, Wust P, Budach V, Boehmer D. Potentials of on-line repositioning based on implanted fiducial markers and electronic portal imaging in prostate cancer radiotherapy. *Radiat Oncol*. 2009;4:13.
44. Treeby BE, Cox BT. k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *J Biomed Opt*. 2010;15: 021314.
45. Treeby BE, Jaros J, Rendell AP, Cox BT. Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method. *J Acoust Soc Am*. 2012;131:4324.
46. Chin JL, Billia M, Relle J, et al. Magnetic resonance imaging guided transurethral ultrasound ablation of prostate tissue in patients with localized prostate cancer: a prospective phase 1 clinical trial. *Eur Urol*. 2016;70:447–455.
47. Yutkin V, Ahmed HU, Donaldson I, et al. Salvage high-intensity focused ultrasound for patients with recurrent prostate cancer after brachytherapy. *Urol*. 2014;84:1157–1162.
48. Illing R, Emberton M. Sonablate 500: transrectal high-intensity focused ultrasound for the treatment of prostate cancer. *Expert Rev Med Devices*. 2006;3:717–729.
49. Uchida T, Nakano M, Shoji S, Nagata Y, Usui Y, Terachi T. Twelve years experience with high-intensity focused ultrasound (HIFU) using sonablate devices for the treatment of localized prostate cancer. In: *11th International Symposium on Therapeutic Ultrasound*, Vol. 1481, Melville, NY: AIP Conference Proceedings; 2012: 401–406.

50. Uchida T, Ohkusa H, Nagata Y, Hyodo T, Satoh T, Irie A. Treatment of localized prostate cancer using high-intensity focused ultrasound. *BJU Int*. 2006;97:56–61.
51. Berenger J-P. Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. *J Comput Phys*. 1996;127: 363–379.
52. Jaros J, Rendell AP, Treeby BE. Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound. *Int J High Perform Comput Appl*. 2016;30:137–155.
53. Jaros J, Nikl V, Treeby BE. Large-scale ultrasound simulations using the hybrid openMP/MPI decomposition. Exascale Applications and Software Conference; 2015.
54. Jalloh M, Leapman MS, Cowan JE, et al. Patterns of local failure following radiation therapy for prostate cancer. *J Urol*. 2015;194:977–982.

C.3 Simulations in the Kidney

Suomi, V.; **Jaros, J.**; Treeby, B. E.; Cleveland, R.: Nonlinear 3-D simulation of high-intensity focused ultrasound therapy in the Kidney. In *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2016. ISSN 1557-170X. pp. 5648–5651. doi:10.1109/EMBC.2016.7592008.

Nonlinear 3-D Simulation of High-Intensity Focused Ultrasound Therapy in the Kidney

Visa Suomi[1], Jiri Jaros[2], Bradley Treeby[3] and Robin Cleveland[1]

Abstract— **Kidney cancer is a severe disease which can be treated non-invasively using high-intensity focused ultrasound (HIFU) therapy. However, tissue in front of the transducer and the deep location of kidney can cause significant losses to the efficiency of the treatment. The effect of attenuation, refraction and reflection due to different tissue types on HIFU therapy of the kidney was studied using a nonlinear ultrasound simulation model. The geometry of the tissue was derived from a computed tomography (CT) dataset of a patient which had been segmented for water, bone, soft tissue, fat and kidney. The combined effect of inhomogeneous attenuation and sound-speed was found to result in an 11.0 dB drop in spatial peak-temporal average (SPTA) intensity in the kidney compared to pure water. The simulation without refraction effects showed a 6.3 dB decrease indicating that both attenuation and refraction contribute to the loss in focal intensity. The losses due to reflections at soft tissue interfaces were less than 0.1 dB. Focal point shifting due to refraction effects resulted in −1.3, 2.6 and 1.3 mm displacements in x-, y- and z-directions respectively. Furthermore, focal point splitting into several smaller subvolumes was observed. The total volume of the secondary focal points was approximately 46% of the largest primary focal point. This could potentially lead to undesired heating outside the target location and longer therapy times.**

I. INTRODUCTION

Kidney cancer is the 13th most common cancer in the world with approximately 338,000 cases diagnosed in 2012 of which 214,000 were in men and 124,000 in women [1]. In the same year approximately 143,000 people died due to the disease. Early diagnosis as well as safe and effective therapy methods are therefore crucial for the survival of patients. Typically kidney cancer is treated surgically which is effective [2], but this can lead to complications in as many as 19% of cases [3]. Alternative, minimally invasive, therapies such as cryotherapy [4] and radiofrequency ablation [5] reduce the risk of complications and often result in shorter hospital stays. However, neither of these methods is completely non-invasive and therefore still contain a risk of infection, seeding metastases and other complications.

High-intensity focused ultrasound (HIFU) is a non-invasive therapy method which does not require puncturing the skin and typically has minimal or no side-effects. HIFU therapy can be used clinically to treat cancerous tissue in kidney, but the oncological outcomes have been variable [6]. This has been thought to be partly due to the attenuation properties of peri-nephric fat [7] which results in poor delivery of HIFU energy to the target focal point. The effect of attenuation might be significant especially in the nonlinear case where higher harmonic frequencies generated during HIFU therapy are strongly attenuated. In addition to attenuation, the defocusing of ultrasound due to refraction and the reflections at the tissue interfaces might result in significant loss of HIFU energy [8].

The aim of this research is to investigate how the attenuation, reflection and refraction effects of different tissue types affect the overall efficacy of HIFU therapy of the kidney. This was done by performing nonlinear HIFU therapy simulations in a segmented computed tomography (CT) dataset of a patient in 3-D.

II. SIMULATIONS

A. Parallelised nonlinear ultrasound simulation model

The HIFU simulations were calculated using the parallel k-Wave toolbox. The k-Wave toolbox models ultrasound wave propagation in soft tissue using a generalised version of the Westervelt equation which accounts for nonlinearity, material heterogeneities and power law absorption. The governing equations are solved using a k-space pseudospectral approach where the Fourier collocation spectral method is used to calculate spatial gradients, and a k-space corrected finite difference scheme is used to integrate forwards in time.

The toolbox is designed for deployment on large distributed computer clusters with thousands of compute cores [9]. The simulation domain is partitioned over one or two dimensions and distributed among the cores. Since the gradient calculation requires the fast Fourier transform (FFT) to be calculated over the whole domain, global data exchange is performed in each simulation time step. Although this has been proven to be a bottleneck, the code efficiency remains acceptable up to to 8192 compute cores [9]. The simulation data sampling and storing is performed via a parallel I/O module based on the HDF5 library and Luster file system.

B. Simulation geometry and execution

The simulation geometry was derived from a CT dataset of a patient (see Figure 1). Thresholds were used to segment the data set into bone, fat and other soft tissue. The kidney was then segmented manually. The medium outside the patient was assumed to be water. Typical values for sound speed, attenuation, density and B/A were used for each tissue type (see Table I) [10]. The HIFU transducer was modelled on a clinical system (Model JC-200 Tumor System, HAIFU) [7] with an annular transmitting surface of outer diameter 20 cm

[1]Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, UK

[2]Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic

[3]Department of Medical Physics and Biomedical Engineering, University College London, Wolfson House, 2-10 Stephenson Way, London, NW1 2HE, UK

978-1-4577-0220-4/16/$31.00 ©2016 IEEE

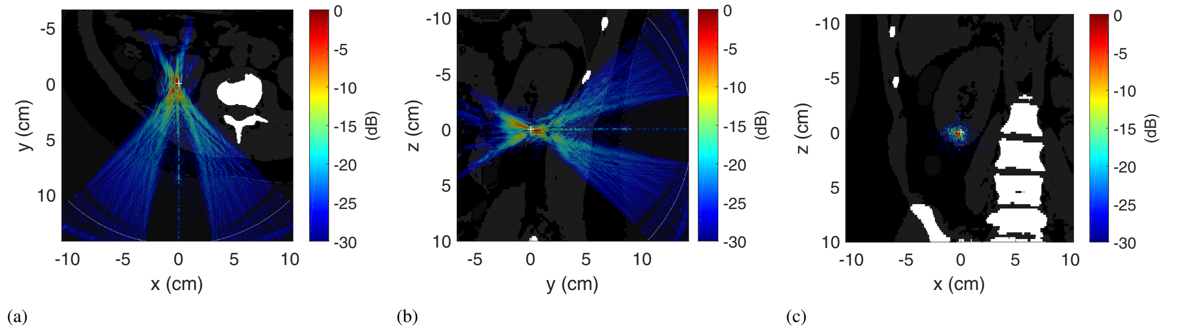


Fig. 1. (a) Axial, (b) sagittal and (c) coronal slices of the CT scan showing the ultrasound pressure field in kidney. The pressure field is displayed on a log-scale with a dynamic range of 30 dB. The different gray levels in the CT data correspond to the density of each tissue type: white - bone, gray - kidney/soft tissue, black - water/fat. The ultrasound focal point target location is marked with a white cross.

and inner hole diameter 6 cm. The operating frequency was 0.95 MHz and the focal length was 14.5 cm. The transducer was positioned so that the geometric focal point of the transducer (the white cross in Figure 1) was located in the bottom part of left kidney.

For data analysis three simulations were conducted: (i) reference simulation in pure water, (ii) simulation without the refraction effects (i.e., constant sound speed of water in all tissue types) but all other properties varying and (iii) simulation with all properties varying (i.e., with refraction effects). Before performing the actual simulations several convergence studies were conducted in order to find out the optimal grid size and temporal resolution. The computational grid consisted of 1200 × 1200 × 1200 grid points (i.e., 22.2 cm × 22.2 cm × 22.2 cm) giving a spatial resolution of 185 μm which supported nonlinear harmonic frequencies up to 4 MHz. Perfectly matched layers (PML) were used on the edges of the grid. The simulation length was set to 260 μs with a temporal resolution of 8.15 ns giving a total of 31876 time steps per simulation. The simulations were run using 400 computing cores for approximately 180 hours in total using the computing facilities provided by advanced research computing (ARC) at the University of Oxford [11]. For data analysis the time-domain waveforms and the peak pressures were saved in a three-dimensional grid around the focal point. In addition axial, sagittal and coronal slices of the ultrasound field over the whole spatial domain were saved.

TABLE I
TISSUE PARAMETERS USED IN THE SIMULATIONS [10]

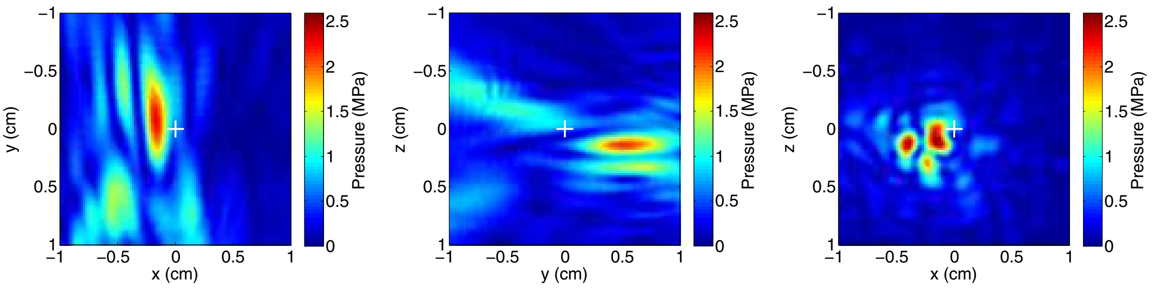
	Density (kg/m^3)	Sound speed (m/s)	Attenuation (dB/(MHz$^{1.1}$·cm))	B/A
Water	1000	1520	0.00217	5.2
Bone	1908	4080	20.00	7.4
Soft tissue	1055	1575	0.60	7.0
Fat	950	1478	0.48	10.0
Kidney	1050	1560	1.00	7.4

III. RESULTS

Figure 1 shows the axial, sagittal and coronal slices of the ultrasound pressure field generated by the HIFU transducer. The pressure field is displayed in log-scale with a dynamic range of 30 dB. The transducer was positioned in order to avoid the ribs which would otherwise cause significant pressure losses due to strong reflection. The annular nature of the source results in the appearance of two beams. In the focal region it can be seen that the region of high pressure does not form the archetypical ellipse shape, but is more diffuse. Further, the highest pressure is offset from the target location (white cross marker) in all slices.

Figure 2 shows close-ups of the axial, sagittal and coronal slices of the pressure field in the ultrasound focal area. Here the shift of the location of the highest pressure from the target location is clear and it was determined to be -1.3, 2.6 and 1.3 mm in x-, y- and z-directions respectively. By examining the focal area in more detail in the coronal slice (see Figure 2(c)), it can be seen that in addition to the focal shifting, a region of high pressure has split into a number of subvolumes. This is more clearly visualised in Figure 3(a) which shows the isosurfaces of the focal pressure regions thresholded at -6 dB. It can be seen that the focal region consists of five smaller focal points with the largest being approximately 12 mm in length and 3 mm in width. In comparison the size of the -6 dB focal point in water is approximately 6.5 mm in length and 1.1 mm in width.

The splitting of the focal region was quantified by identifying the largest subvolume as the parent focal region and the others as child regions. For a given pressure threshold, between 50% and 100% of the maximum pressure, the volume of the child focal regions was compared to that of the parent focal region. Figure 3(b) shows a histogram of the analysis. For pressure thresholds above 80% no voxels were present in the child focal regions. However, when the threshold was reduced to 70% it was found that approximately 5% of the voxels were in the child focal regions. As the threshold was decreased the amount of volume in the child regions increased. At the -6 dB pressure threshold the total volume in the child regions was 46% of the volume



(a) (b) (c)

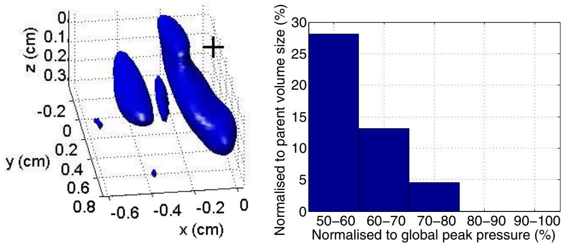
Fig. 2. (a) Axial, (b) sagittal and (c) coronal slices of the ultrasound field in the focal area in kidney. The ultrasound focal point target location is marked with a cross.

of the parent focal region. These data suggest that undesired heating effects will occur at secondary focal points due to focal point splitting.

Figure 4(a) shows the time waveforms at the location of maximum peak pressure in both water and kidney. The peak-positive pressure drops from 14.49 MPa in water to 3.51 MPa in kidney. Similarly, the spatial peak-temporal average (SPTA) intensity has dropped from 4116 W/cm^2 in water to 324 W/cm^2 in kidney, that is, an 11.0 dB decrease. The simulation without the refraction effects resulted in a single focal point (i.e., no focal splitting) with a peak-positive pressure of 6.46 MPa and SPTA intensity of 957 W/cm^2 corresponding to a 6.3 dB decrease. This suggests that refraction and attenuation contribute similarly to the loss in focal intensity. Figure 4(b) shows the windowed (Hann) frequency spectrum of the same signals. A peak at centre frequency 0.95 MHz is clearly visible as are the nonlinearly generated harmonics, however, in the case of tissue the nonlinear effects are much less pronounced.

IV. DISCUSSION

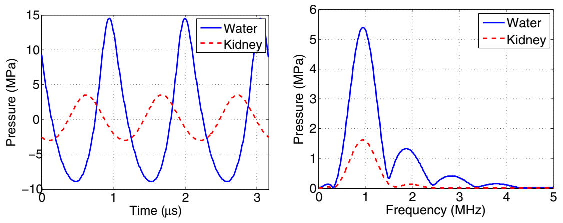
Ritchie et al. [7] studied the attenuation of focused ultrasound using subcutaneous and peri-nephric fat layers in front of the HIFU transducer. They found the attenuation of peri-nephric fat to be significantly higher (1.36 dB/cm) compared to typical fat tissue attenuation (0.48 dB/cm) [10]. In the simulations reported here all the fat layers were segmented as normal fat tissue using the latter attenuation value. This difference is not thought to affect conclusions as for the patient derived data set employed here the thickness of peri-nephric fat was 0.5 cm on average and adding in the higher attenuation would contribute an extra 0.44 dB of loss which is small in comparison to the total loss observed. The most significant attenuation losses were caused by subcutaneous fat and soft tissue in front of the kidney whose thickness were approximately 2.6 and 3.7 cm respectively.



(a) (b)

Fig. 3. (a) The focal point volume is shown with isosurfaces thresholded at -6 dB. The target focal point is marked with a black cross. The shifting and splitting of the focal point into one parent and four child focal volumes can be seen. (b) Histogram showing the size of the child volumes relative to the parent focal volume for various pressure contours varying from 50% to 80% of the global peak pressure.

In addition to attenuation, energy losses also occur due to reflections and scattering at interfaces, such as, the rib cage, tissue interfaces and air pockets. Here the transducer was positioned so that reflections due to rib bones were not present. The effect of tissue interfaces in the penetration of HIFU has been studied in rabbit kidney *in vivo* by Damianou [12]. They found the ultrasound penetration through muscle-kidney and fat-kidney interfaces to be excellent in a situation where no air bubbles were present. However, in some cases air spaces existed in between these interfaces which caused strong reflections and acted as possible sites for cavitation during the HIFU therapy. In the simulations here the interfaces between tissues contained no air spaces, and therefore, all the possible energy losses due to reflections were caused either by the rib cage or acoustic impedance mismatches between tissue interfaces. The intensity transmission coefficients for water-fat, fat-soft tissue, soft tissue-fat and fat-



(a) (b)

Fig. 4. (a) Time domain waveforms at the maximum peak pressure location in water and kidney. (b) Windowed (Hann) frequency spectrum of the same waveforms.

kidney interfaces were 99.84%, 99.29%, 99.29% and 99.41% respectively. For all the interfaces the total transmission is 97.85% which corresponds to a loss of less than 0.1 dB.

Focal shifting and splitting due to variations in the sound speed is another factor considerably affecting the efficacy of HIFU therapy. At interfaces changes in sound speed result in refraction, in addition, the phase accumulation will change in different tissues affecting the constructive and destructive interference of the waves. These effects will impact both the intensity and the location of the focus. Focal shifting due to subcutaneous and peri-nephric fat was studied by Ritchie et al. [7] who found the shift to be approximately 1 mm in both transverse directions. In the simulations here similar magnitude shifts were observed which are large in terms of -6 dB focal point width (1.1 mm), but not with respect to typical renal tumour sizes of several centimetres [13]. Splitting of the ultrasound focal point into smaller, less defined, volumes can significantly reduce its heating efficiency. The simulations showed an 11.0 dB drop in SPTA intensity when all effects were incorporated (specifically attenuation and refraction) and only a 6.3 dB drop without the refraction effects (i.e., no focal point splitting). This suggests that attenuation and refraction have a similar impact on the intensity loss at the focus, contributing about 5 to 6 dB each. When focal point splitting was present, the cumulative size of the two separate smaller focal volumes was found to be approximately 46% of main focal point. Although focal splitting provides larger total heating volume the efficiency is reduced, because the acoustic energy is distributed over a larger volume which leads to longer therapy times and also may result in undesired heating in regions away from the target region.

Other phenomena that have been shown to reduce the efficacy of renal HIFU therapy are respiratory movement [14] and perfusion [15]. These effects were not incorporated in the simulation model but could be considered in the future.

V. Conclusions

The effects of attenuation, reflection and refraction on the efficacy of HIFU therapy in kidney were investigated using a nonlinear simulation model. Attenuation and splitting due to refraction were found to be the most significant factors reducing the intensity of the ultrasound field. Reflections due to the rib cage could possibly cause significant losses, but this can be avoided by optimal positioning of the transducer. The reflections due to tissue interfaces were found to be negligible.

Acknowledgment

V. S. acknowledges the support of the RCUK Digital Economy Programme grant number EP/G036861/1 (Oxford Centre for Doctoral Training in Healthcare Innovation) as well as the support of Instrumentarium Science Foundation, Jenny and Antti Wihuri Foundation and Finnish Cultural Foundation. J. J. is financed from the SoMoPro II Programme, co-financed by the European Union and the South-Moravian Region. This work reflects only the author's view and the European Union is not liable for any use that may be made of the information contained therein. B. T. acknowledges the support of EPSRC grant numbers EP/L020262/1 and EP/M011119/1. R. C. acknowledges the support of EPSRC grant number EP/K02020X/1.

References

[1] J. Ferlay, I. Soerjomataram, M. Ervik, R. Dikshit, S. Eser, C. Mathers, M. Rebelo, D. Parkin, D. Forman, and F. Bray, "Globocan 2012 v1.0, cancer incidence and mortality worldwide: Iarc cancerbase no. 11 [internet]," Available from: http://globocan.iarc.fr, 2016, accessed: 2016-28-01.

[2] H. Van Poppel, L. Da Pozzo, W. Albrecht, V. Matveev, A. Bono, A. Borkowski, M. Colombel, L. Klotz, E. Skinner, T. Keane *et al.*, "A prospective, randomised eortc intergroup phase 3 study comparing the oncologic outcome of elective nephron-sparing surgery and radical nephrectomy for low-stage renal cell carcinoma," *European urology*, vol. 59, no. 4, pp. 543–552, 2011.

[3] I. S. Gill, S. F. Matin, M. M. Desai, J. H. Kaouk, A. Steinberg, E. Mascha, J. Thornton, M. H. Sherief, B. Strzempkowski, and A. C. Novick, "Comparative analysis of laparoscopic versus open partial nephrectomy for renal tumors in 200 patients," *The Journal of urology*, vol. 170, no. 1, pp. 64–68, 2003.

[4] I. S. Gill, E. M. Remer, W. A. Hasan, B. Strzempkowski, M. Spaliviero, A. P. Steinberg, J. H. Kaouk, M. M. Desai, and A. C. Novick, "Renal cryoablation: outcome at 3 years," *The Journal of urology*, vol. 173, no. 6, pp. 1903–1907, 2005.

[5] D. A. Gervais, F. J. McGovern, R. S. Arellano, W. S. McDougal, and P. R. Mueller, "Radiofrequency ablation of renal cell carcinoma: part 1, indications, results, and role in patient management over a 6-year period and ablation of 100 tumors," *American Journal of Roentgenology*, vol. 185, no. 1, pp. 64–71, 2005.

[6] R. W. Ritchie, T. Leslie, R. Phillips, F. Wu, R. Illing, G. Ter Haar, A. Protheroe, and D. Cranston, "Extracorporeal high intensity focused ultrasound for renal tumours: a 3-year follow-up," *BJU international*, vol. 106, no. 7, pp. 1004–1009, 2010.

[7] R. Ritchie, J. Collin, C. Coussios, and T. Leslie, "Attenuation and de-focusing during high-intensity focused ultrasound therapy through peri-nephric fat," *Ultrasound in medicine & biology*, vol. 39, no. 10, pp. 1785–1793, 2013.

[8] R. Illing, J. Kennedy, F. Wu, G. Ter Haar, A. Protheroe, P. Friend, F. Gleeson, D. Cranston, R. Phillips, and M. Middleton, "The safety and feasibility of extracorporeal high-intensity focused ultrasound (hifu) for the treatment of liver and kidney tumours in a western population," *British journal of cancer*, vol. 93, no. 8, pp. 890–895, 2005.

[9] V. Nikl and J. Jaros, "Parallelisation of the 3d fast fourier transform using the hybrid openmp/mpi decomposition," in *Mathematical and Engineering Methods in Computer Science*. Springer, 2014, pp. 100–112.

[10] T. D. Mast, "Empirical relationships between acoustic parameters in human soft tissues," *Acoustics Research Letters Online*, vol. 1, no. 2, pp. 37–42, 2000.

[11] A. Richards, "University of oxford advanced research computing," Zenodo.10.5281/zenodo.22558, 2016.

[12] C. Damianou, "Mri monitoring of the effect of tissue interfaces in the penetration of high intensity focused ultrasound in kidney in vivo," *Ultrasound in medicine & biology*, vol. 30, no. 9, pp. 1209–1215, 2004.

[13] M. Remzi, D. Katzenbeisser, M. Waldert, H.-C. Klingler, M. Susani, M. Memarsadeghi, G. Heinz-Peer, A. Haitel, R. Herwig, and M. Marberger, "Renal tumour size measured radiologically before surgery is an unreliable variable for predicting histopathological features: benign tumours are not necessarily small," *BJU international*, vol. 99, no. 5, pp. 1002–1006, 2007.

[14] M. Marberger, G. Schatzl, D. Cranston, and J. Kennedy, "Extracorporeal ablation of renal tumours with high-intensity focused ultrasound," *BJU international*, vol. 95, no. s2, pp. 52–55, 2005.

[15] I. Chang, I. Mikityansky, D. Wray-Cahen, W. F. Pritchard, J. W. Karanian, and B. J. Wood, "Effects of perfusion on radiofrequency ablation in swine kidneys 1," *Radiology*, vol. 231, no. 2, pp. 500–505, 2004.

C.4 Transcranial Ultrasonic Neurostimulation

Robertson, J. L. B.; Cox, B. T.; **Jaros, J.**; Treeby, B. E.: Accurate simulation of transcranial ultrasound propagation for ultrasonic neuromodulation and stimulation. *The Journal of the Acoustical Society of America*, vol. 141, no. 3, pp. 1726–1738. ISSN 0001-4966, doi:10.1121/1.4976339, **(IF 1.572)**.

Accurate simulation of transcranial ultrasound propagation for ultrasonic neuromodulation and stimulation

James L. B. Robertson,[1,a)] Ben T. Cox,[1] J. Jaros,[2] and Bradley E. Treeby[1]
[1]*Department of Medical Physics and Biomedical Engineering, University College London, London, United Kingdom*
[2]*Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic*

(Received 13 June 2016; revised 1 December 2016; accepted 31 January 2017; published online 13 March 2017)

Non-invasive, focal neurostimulation with ultrasound is a potentially powerful neuroscientific tool that requires effective transcranial focusing of ultrasound to develop. Time-reversal (TR) focusing using numerical simulations of transcranial ultrasound propagation can correct for the effect of the skull, but relies on accurate simulations. Here, focusing requirements for ultrasonic neurostimulation are established through a review of previously employed ultrasonic parameters, and consideration of deep brain targets. The specific limitations of finite-difference time domain (FDTD) and *k*-space corrected pseudospectral time domain (PSTD) schemes are tested numerically to establish the spatial points per wavelength and temporal points per period needed to achieve the desired accuracy while minimizing the computational burden. These criteria are confirmed through convergence testing of a fully simulated TR protocol using a virtual skull. The *k*-space PSTD scheme performed as well as, or better than, the widely used FDTD scheme across all individual error tests and in the convergence of large scale models, recommending it for use in simulated TR. Staircasing was shown to be the most serious source of error. Convergence testing indicated that higher sampling is required to achieve fine control of the pressure amplitude at the target than is needed for accurate spatial targeting. © *2017 Acoustical Society of America.*

[http://dx.doi.org/10.1121/1.4976339]

[JFL] Pages: 1726–1738

I. INTRODUCTION

The use of implanted electrodes for deep brain stimulation (DBS) is a well-established, invasive treatment for multiple neurological conditions and has directly resulted in a greater understanding of functional neuroanatomy and deep brain circuitry.[1] Unfortunately, its usefulness is limited by the inherent risks of the required neurosurgery combined with difficulties in targeting and repositioning the stimulatory focus.[2] Non-invasive alternatives such as transcranial magnetic and direct current stimulation have both met with success in research and clinical settings. However, they are limited in terms of their ability to achieve tight spatial focusing, and their penetration deep into tissue.[3] Table I demonstrates a selection of existing and planned DBS target structures alongside their approximate dimensions and deviation from the approximate center of the brain—the mid-commissural point (MCP).[4,5] These dimensions demonstrate the millimeter scale size of the target structures, and their position close to the center of the brain. Thus, the ability to non-invasively target these nuclei for modulation and stimulation would represent a revolutionary neuroscientific tool with both clinical and research applications.

Ultrasonic neuromodulation and stimulation (UNMS) offers a potential solution to these requirements, and has recently received a great deal of interest. Transcranial focusing of ultrasound offers the potential for reversible, non-invasive neural excitation and modulation, with focusing on the scale of the acoustic wavelength.[3] Table II shows a selection of UNMS papers published in the last decade, and demonstrates the variety of acoustic intensities and frequencies used, target structures sonicated, and neural responses observed. The physical mechanism underlying UNMS remains unclear, although a non-thermal mechanism is suspected, and lower acoustic frequencies have been shown to evoke a response more reliably.[3,6] Most recently ultrasound has been used to elicit electro-encephalogram (EEG) and sensory responses in human subjects, although this has been restricted to superficial cortical brain areas using unfocused single element transducers.[7–9] If UNMS is to develop as an effective non-invasive neurostimulation technique, its application to human subjects must be extended to deep brain targets. Based on the dimensions of DBS targets shown in Table I, and the range of effective ultrasonic intensities shown in Table II, the following focusing requirements may be defined:

- A spatial targeting error of less than 1.5 mm.
- Control of the intensity at the focus with $\leq$10% error will ensure that neurostimulation occurs. Greater accuracy may be desirable in studies of the mechanisms and thresholds of UNMS.
- An ultrasonic stimulation focus of no greater than 3 mm diameter will ensure stimulatory specificity.
- Steering of the ultrasonic focus up to ~30 mm from the MCP to allow stimulation of arbitrary deep brain targets.

[a)]Electronic mail: james.robertson.10@ucl.ac.uk

TABLE I. Approximate dimensions of DBS targets (Ref. 6). AP/DV/ML—Anteroposterior/dorsoventral/mediolateral, MCP—Mid-commisural point.

Target	AP × DV × ML [mm]	MCP deviation [mm]
Ventral intermediate nucleus	10 × 15.8 × 11	17
Ventral anterior nucleus	7 × 12.6 × 10	15
Centro-median nucleus	8 × 4.5 × 4	14
Nucleus Accumbens	9.5 × 10 × 12	21
Globus pallidus externus	21.5 × 10 × 3	23
Globus pallidus internus	12.5 × 8 × 6	20
Sub-thalamic nucleus	8 × 4 × 6.3	13

The primary obstacle to achieving these ultrasonic focusing criteria within the brain is the presence of the skull, which aberrates and attenuates incoming wavefronts. Time-reversal (TR) focusing, first adapted for transcranial focusing by Aubry *et al.*, is able to correct for the aberrating effect of the skull by taking advantage of the time-symmetry of the lossless acoustic wave equation.[9] In model-driven TR, numerical models simulate the propagation of ultrasound from a target area to a virtual transducer using acoustic property maps of the head derived from CT or MRI images.[9,10] The pressure time series at the simulated transducer surface is then time-reversed, and used to generate drive signals for a multi-element acoustic transducer array. For high-intensity thermal applications, model-driven TR may be combined with MRI thermometry for treatment verification. Chauvet *et al.*[11] confirmed the potential for model-driven TR-based focusing inside the human head to millimeter precision, verified by MRI thermometry. Marquet *et al.*[12] showed that model-driven TR is capable of restoring 90% of the peak pressure that can be obtained with gold-standard hydrophone based methods when focusing through an *ex vivo* skull bone. However, model-driven TR remains subject to systematic errors and uncertainties with a resulting loss in focusing quality or efficiency. Four categories of uncertainty are:

(i) The underlying physical model and how the governing equations model the physics of propagation including phenomena such as absorption, nonlinearity, and shear wave effects.

(ii) Numerical approximations due to the discretization of the simulation domain, including numerical dispersion, the representation of medium heterogeneities, and the effectiveness of any absorbing boundary conditions.

(iii) The inputs to the model, such as the map of acoustic medium properties and the representation of acoustic transducers.

(iv) How the numerical simulations are used within a broader TR protocol, including how the simulated source is related to the desired pressure at the target, and how phenomena that are not time-invariant, such as absorption, are accounted for.

TR simulations for transcranial focusing have typically made use of finite-difference time domain (FDTD) numerical models.[11,12] Recently a *k*-space corrected, pseudospectral time domain (PSTD) numerical scheme was used in model-driven TR and shown to give comparable accuracy.[13] Both FDTD and PSTD schemes use consistent approximations to

TABLE II. Review of selected recent ultrasonic neuromodulation and neurostimulation literature. SPPA—Spatial peak pulse average, SPTA—spatial peak temporal average, SPTP—spatial peak temporal peak, VEP—Visual evoked potential, LGN—lateral geniculate nucleus, FEF—frontal eye field, PET—positron emission tomography, fMRI—functional magnetic resonance imaging, GABA—gamma-aminobutyric acid, S1—primary somatosensory cortex, MC—motor cortex. *—0.5 MHz achieved with 2 MHz carrier.

Year	Author	Freq. [MHz]	Intensity at focus [W/cm^2]	Target: *In-vivo*(IV) vs *Ex-vivo*(EV)	Neural Response & Observations
2008	Tyler *et al.*	0.44 and 0.67	2.9 SPPA	EV mouse hippocampus	Imaging of ion channel opening and synaptic activation
2008	Khraiche *et al.*	7.75	50–150 SPTP	EV rat hippocampus	Increased neuronal spike rate
2010	Tufail *et al.*	0.25 and 0.50	0.228 SPPA	IV mouse brain	Motor response, cortical spiking, ion channel opening
2011	Yoo *et al.*	0.69	3.3–12.6 SPPA	IV rabbit cortex	Motor response VEP suppression and fMRI activity
2011	Min *et al.*	0.69	2.6 SPPA	IV rat epileptic focus	Suppression of induced epileptic behavior
2011	Yang *et al.*	0.65	3.5 SPPA	IV rat thalamus	Decrease in extracellular GABA levels
2012	King *et al.*	0.50	1–17 SPTP	IV rat brain	Motor response above an intensity threshold
2012	Kim *et al.*	0.35	8.6–20 SPPA	IV rat abducens nerve	Motor response in abducens muscle
2013	Menz *et al.*	43	20–60 SPPA	EV salamander retina	Retinal interneuron stimulation
2013	Deffieux *et al.*	0.32	4 SPPA	IV primate FEF	Altered visual antisaccade latency
2013	Younan *et al.*	0.32	17.5 SPPA	IV rat cortex	Motor response
2014	Legon *et al.*	0.50	5.9 SPPA	IV human S1	Altered sensory evoked EEG oscillations
2014	Kim *et al.*	0.35	3.5–4.5 SPTA	IV rat thalamus	Glucose uptake change, motor response
2014	King *et al.*	0.5	3 SPTA	IV mouse MC	Motor response varying with targeting
2014	Juan *et al.*	1.1	13.6–93.4 SPTA	IV rat vagus nerve	Reduced vagus compound action potential
2014	Mehic *et al.*	0.5*	2–8 SPTA	IV rat brain	Motor response scaling with intensity
2014	Mueller *et al.*	0.5	5.9 SPPA	IV human S1	Altered EEG beta phase dynamics
2015	Lee *et al.*	0.25	0.5–2.5 SPPA	IV human S1	Evoked sensations and EEG changes
2015	Lee *et al.*	0.25	6.6–14.3 SPPA	IV sheep cortex	Motor and EEG responses
2016	Ye *et al.*	0.3–2.9	0.1–127 SPPA	IV mouse MC	Motor response, more effective at low frequencies
2016	Ai *et al.*	0.5 and 0.86	6 SPPA	IV human brain	fMRI activity at stimulation site and deep brain
2016	Darvas *et al.*	1.05	1.4 SPTA	IV rat brain	EEG response, focal effects on gamma band activity

the wave equation, and can be made stable by choosing the discretization parameters appropriately. As the rate of spatial and temporal sampling increases, they will converge on the true solution at a rate dependent on the particular approximations of the numerical model [(ii) above]. However, due to the large scale of these simulations, it is desirable to minimize the grid size and resulting computational burden without compromising accuracy, so knowledge of the minimum sampling criteria necessary to achieve the required accuracy is valuable. In the present paper, these numerical schemes are briefly described, and the various factors affecting the rate of numerical convergence are examined. This is quantified in terms of the spatial and temporal sampling required to obtain acceptable accuracy in the simulation of ultrasound propagation from the scalp to a deep brain target. While the criteria used are established for the application of transcranial UNMS, these results are also applicable to other therapies that require accurate transcranial ultrasound simulation, such as high intensity focused ultrasound (HIFU) ablation and opening the blood brain barrier with ultrasound.

II. NUMERICAL METHODS FOR ULTRASOUND PROPAGATION

A. FDTD

FDTD methods have seen extensive use in the simulation of ultrasound propagation, and have accordingly been used for the purpose of model-driven TR with success.[9–12] In finite difference methods, partial derivatives are calculated using a linear combination of function values at neighboring grid points. The finite difference approximations are derived by combining local Taylor series expansions truncated to a fixed number of terms.[14] When simulating ultrasound propagation, this approximation causes an unphysical dependence of the simulated sound speed on the number of grid points per wavelength ($\mathrm{PPW} = \lambda/\Delta x$) and the number of temporal points per period [$\mathrm{PPP} = 1/(f\Delta t)$] where f and λ are acoustic frequency and wavelength, respectively, and Δx and Δt are the spatial and temporal discretization, respectively.[14] This manifests as a cumulative error in the phase of propagating waves, termed numerical dispersion. In addition, stability conditions must also be met to ensure the numerical scheme is stable. These conditions are contingent on the exact scheme used and the number of simulated dimensions.[14] A useful metric when considering stability is the Courant-Friedreichs-Lewy (CFL) number, defined as

$$\mathrm{CFL} = \frac{c\Delta t}{\Delta x} = \frac{\mathrm{PPW}}{\mathrm{PPP}}, \qquad (1)$$

where c is the sound speed. Stability criteria are often expressed as limits placed on the CFL number.[14–16]

B. PSTD

In PSTD methods, spatial derivatives are calculated by decomposing the spatially varying acoustic variables into a finite sum of weighted global basis functions.[17] This decomposition allows efficient computation of spatial derivatives using the derivatives of the basis functions. For wave problems, a Fourier basis is typically used, with the basis function weights calculated via the fast Fourier transform.[17] The subsequent gradient calculation is exact, eliminating numerical dispersion due to spatial discretization. However, for an explicit time-stepping scheme, temporal gradients must still be approximated via a finite difference method, with resulting dispersive error.[15] Fortunately, for a second-order accurate approximation, this error can be calculated analytically, and used to introduce a correction factor, $\kappa = \mathrm{sinc}(c_{\mathrm{ref}}k\Delta t/2)$, in the spatial frequency domain.[15] Here, $k = \sqrt{k_x^2 + k_y^2 + k_z^2}$ is the magnitude of the wavevector (k_x, k_y, k_z) at each grid point in the spatial frequency domain, and c_{ref} is a user defined reference sound speed. This method is often called the *k*-space PSTD method, and in homogeneous media it is unconditionally stable and free from numerical dispersion for arbitrarily large Δt.[15,16] In media with a heterogeneous sound speed, the application of κ in the spatial frequency domain means that a single sound speed must be chosen for the correction factor. As a result, numerical dispersion will arise, the extent of which will depend on the temporal sampling and the difference between the local sound speed $c(x)$, and the reference sound speed c_{ref}.[16] As with FDTD schemes, simulation-dependent limits on the CFL number must be observed to ensure numerical stability.

C. The BLI

FDTD and PSTD methods both use functions to interpolate between the values of the acoustic variables at the grid points. The interpolating functions are used to approximate the field gradients at these points, and the values of the field variables are updated at the grid points at each time step. FDTD methods use polynomials to interpolate between neighboring points, while PSTD methods use a Fourier series to interpolate between all points simultaneously.[17] This Fourier series is bandlimited (truncated) to ensure a unique Fourier representation and is therefore known as the bandlimited interpolant (BLI).[17] This can be considered the representation of the discretely sampled pressure field within PSTD schemes. When a time-varying source is used, the resulting pressure signal is formed from a sum of one or more weighted BLIs. As a result of this, a discrepancy can arise between the BLI and the intuitive expectation of what the sampled function represents. This is shown in Fig. 1(a) for a Kronecker delta represented on a discrete grid. In this case, because the Fourier coefficients of the sampled function do not decay to zero before the Nyquist limit of the grid, the intended field is replaced with a BLI representation with Gibbs type oscillations. It is important to understand that this representation is not erroneous *per se*, but that there is a disparity between the desired input to the PSTD scheme (in this case a Kronecker delta), and what the scheme is capable of representing via a bandlimited Fourier series. To reduce the size of the disparity, smoothing of the intended field can be used to force the Fourier coefficients to decay.[18] This is shown in Fig. 1(b) for the same Kronecker delta function when frequency is filtered with a Blackman window. Although this remains an inexact representation of the

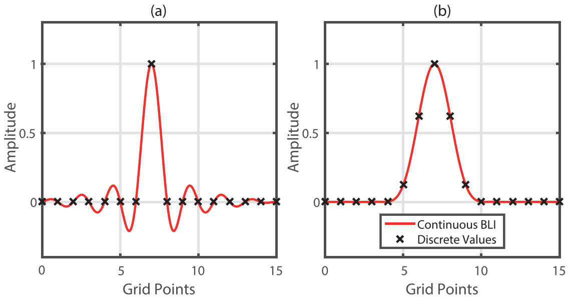


FIG. 1. (Color online) Discrete pressure maps and their BLIs. (a) Unsmoothed delta function and (b) delta function frequency filtered with a Blackman window.

original Kronecker delta function, the non-oscillating BLI more closely matches the intended underlying pressure distribution as defined by the values at the discrete grid points.

III. NUMERICAL TESTING OF INDIVIDUAL ERRORS

A. Overview

In this section, the impact of various factors which affect the convergence of FDTD and PSTD models for the case of transcranial ultrasound simulation is presented. These comprise: the influence of the BLI, changes in the effectiveness of the absorbing perfectly matched layer (PML), the impact of numerical dispersion, the representation of discontinuities in medium properties, and staircasing of acoustic sources and material geometry. The first two represent fundamental considerations in numerical simulations, and are dealt with independently. For the subsequent phenomena, the specific inaccuracies occurring when simulating the propagation of ultrasound from a source in the deep brain to an external transducer are established. This is modelled as consisting of 10 cm propagation through cerebral soft tissue, 1 cm propagation through bone, and 1 cm additional propagation through superficial soft tissue, shown in Fig. 2. Accuracy is quantified in terms of the resulting error in the amplitude and position (calculated using time of arrival) of the temporal maximum intensity at the target position and the sampling criteria constraining these errors below 10% and 1.5 mm, respectively, are established. Beam steering capabilities are determined primarily by hardware, and are not considered here. Modeling of the skull as a single homogeneous layer in this way was recently validated for low frequency model-driven TR by Miller *et al.*[10] The influence of reverberations within the head is considered when necessary, with each reverberation consisting of 2 cm propagation through bone, and 20 cm propagation through brain tissue. The combined effects of these numerical inaccuracies and the validity of the established sampling criteria are then examined through convergence testing of fully simulated two-dimensional (2D) and three-dimensional (3D) TR protocols.

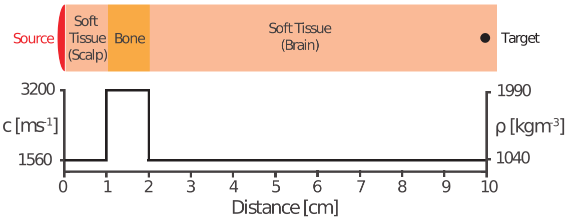


FIG. 2. (Color online) A scaled schematic of the simulation model used to evaluate the impact of numerical errors.

Numerical simulation of ultrasound was carried out with the open source *k*-Wave toolbox using PSTD and *k*-space corrected PSTD (with a user defined c_{ref}) numerical schemes.[16,19] These are henceforth referred to as "PSTD" and "*k*-space" schemes, respectively. The toolbox also includes a second-order accurate in time, fourth-order accurate in space (2–4) FDTD scheme, which was also tested. This scheme is described in detail by Strikwerda,[14] and is widely used to simulate acoustic wave propagation, including in simulated TR.[11,12] Unless stated otherwise, the CFL number was set to 0.3, one-dimensional (1D) tests were carried out on a spatial grid of 4096 grid points, and 2D tests on a 1024 × 1024 grid. Frequency filtered Kronecker delta functions, like that shown in Fig. 1(b), were used to create broadband pressure sources. Homogeneous simulation grids were given the acoustic properties of brain tissue, a density of 1040 kg/m^3, and a sound speed of 1560 m/s (also used to represent superficial soft tissues).[20] For heterogeneous simulations, bone tissue was assigned a density of 1990 kg/m^3 and a sound speed of 3200 m/s.[20] When it was necessary to define a specific ultrasonic frequency of interest to calculate the required sampling criteria, 500 kHz was used. This frequency has seen extensive use in studies of UNMS (see Table II), sits within the range of ultrasound frequencies demonstrating optimal transcranial transmission,[21,22] and has a theoretical minimum focus size of ∼3 mm diameter in soft tissue.

B. The BLI

The BLI represents a fundamental component of both *k*-space and PSTD schemes. As such, it is necessary to examine its impact on simulation accuracy before moving on to more complex factors that affect the rate of convergence. Bandlimited interpolation, as described above, can result in a discrepancy between the intended pressure field and the representation of that field within PSTD schemes when the Fourier coefficients of the intended field have not decayed sufficiently. Practically, this manifests globally as undesired, oscillating pressure values across the simulation grid [see Fig. 1(a)]. Therefore, to examine the impact of BLI effects, it is necessary to determine the amplitude of these undesired pressures relative to that of an intended input.

In practice, the error in the representation of a particular pressure distribution will depend on how well it can be represented by a discrete Fourier transform at a specific spatial discretization.[17] Tonebursts have a well-defined power spectrum determined by their length and central frequency. Therefore, to approximate the BLI error likely to be generally observed, a series of time-varying 10, 30, and 50 cycle acoustic toneburst sources with central wavenumbers approaching the spatial Nyquist limit were used as input signals. These sources have 22.7%, 7.4%, and 4.3% full

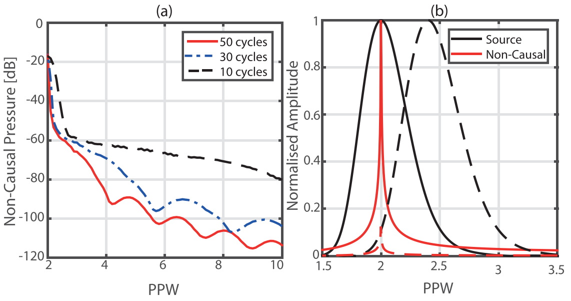


FIG. 3. (Color online) (a) Non-causal pressure amplitude as a function of toneburst central PPW for different toneburst lengths and (b) normalized amplitude spectra of non-causal pressure signals and the corresponding spectra of their source tonebursts, demonstrating how non-casual pressures relate to the 2 PPW component of the source signal.

width at half maximum (FWHM) bandwidth as a percentage of central frequency, respectively. The source was positioned a quarter of the way along a homogeneous 1D computational grid with no PML. The simulations were run for the time taken for waves to travel from the source to the center of the grid. The pressure was recorded at every grid point of the other half of the grid which, according to causality, should have remained quiescent if the BLI of the pressure field matched the intended input of compactly supported tonebursts. Error was quantified as the maximum pressure recorded across the second half of the grid relative to the peak pressure of the source toneburst. The results are shown in Fig. 3(a) as a function of the PPW of the central wavenumber of the toneburst. The amplitude of the non-causal pressure drops rapidly as the number of PPW increases from the Nyquist limit. Reducing the error requires a higher number of PPW for shorter tonebursts due to their wider power spectra, but for all three toneburst lengths the error drops to below -60 dB by 3 PPW. An additional observation was that wavenumbers corresponding to less than 2 PPW are not aliased or otherwise propagated on the grid.

To determine what frequencies comprised the observed non-causal pressure, the results obtained from 10 cycle tonebursts were examined further. Time-varying pressure signals were recovered from the grid points closest to the wave front, which experienced the peak non-causal pressures. The normalized amplitude spectra of these signals resulting from source tonebursts with central wavenumbers sampled at 2 and 2.4 PPW are displayed in Fig. 3(b), alongside the corresponding amplitude spectra of the source tonebursts. The recorded spectra demonstrate a sharp peak at 2 PPW regardless of the central frequency of the source toneburst, and the amplitude of the peak scales with the amplitude of the 2 PPW component of the source toneburst. Practically, these results demonstrate that this error reduces very rapidly as the spatial sampling of the pressure distribution increases, and at 3 PPW BLI errors are reduced to below -60 dB. In higher dimensions, BLI errors are less severe than the 1D case.

C. The PML

The interaction of outgoing pressure waves with the edge of the simulation grid presents a problem for numerical schemes. In the FDTD scheme used here, outgoing pressure waves are perfectly reflected from the edge of the grid, while for *k*-space and PSTD schemes outgoing pressure waves are "wrapped" to the opposite edge of the simulation grid (i.e., the grid is toroidal). To replicate free field conditions, *k*-Wave employs Berenger's split field PML, where the pressure field is artificially divided into Cartesian components to allow selective absorption of the normally incident component.[19,23] The response of the PML to different frequencies was established by propagating broadband pressure sources toward a 20 point PML on a homogeneous 1D grid using the PML profile given in Tabei *et al.*[15,19] Rather than the 2-4 FDTD scheme used elsewhere, a second order accurate in space and time (2-2) FDTD scheme with a CFL of 1 was used to prevent numerical dispersion. It should be noted that this FDTD formulation is not practical outside the homogeneous 1D case due to stability constraints. The time-varying pressure traces of the incident wave, the wave reflected from the surface of the PML, and the wave transmitted to the edge of the computational grid were recorded. The power spectra of these signals were used to calculate reflection and transmission amplitudes relative to the incident wave as a function of spatial sampling. No notable difference was observed between *k*-space and PSTD schemes. Results are shown in Fig. 4 for *k*-space and 2-2 FDTD schemes. In both schemes, the pressure reflection coefficient demonstrates a dependence on spatial sampling, rising steadily from below -120 dB for frequencies sampled at above 4 PPW to total reflection at 2 PPW. Transmission to the edge of the grid remains constant at below -70 dB for both schemes until spatial sampling drops beneath 3 PPW, below which the *k*-space scheme shows an increase in transmission and the FDTD scheme shows a reduction in transmission. These results indicate that the effectiveness of the PML is greatly reduced for wavenumbers sampled at below 3 PPW, and it cannot be relied on at these PPW values. However, erroneous reflection and transmission reduce rapidly as sampling increases. It should be noted that pressure reaching the edge of the grid for both schemes is subject to further attenuation within the PML when reflected or wrapped back into the grid. Furthermore, the BLI will have influenced the behavior of these tests for frequencies sampled at close to the Nyquist limit, which may explain why the *k*-space scheme shows an increase in both reflection and transmission close to 2 PPW.

The PML was also tested in 2D to determine its dependence on the angle of incidence of incoming waves. A broadband point source was placed close to the edge of the PML on the 2D grid and propagated into, and across the surface of, the PML. The pressure was recorded at the edge of the simulated domain to examine transmission, with each recording position corresponding to a particular angle of incidence. The peak pressure transmission at each angle was calculated through comparison with a reference recording obtained with PML absorption set to zero. The results are shown in Fig. 4(c). Transmission to the edge of the grid is lowest for normally incident waves, rising with increasing angle of incidence crossing to above -60 dB at ~40°. No clear relationship between angle of incidence and reflection from the PML was observed. These results should be

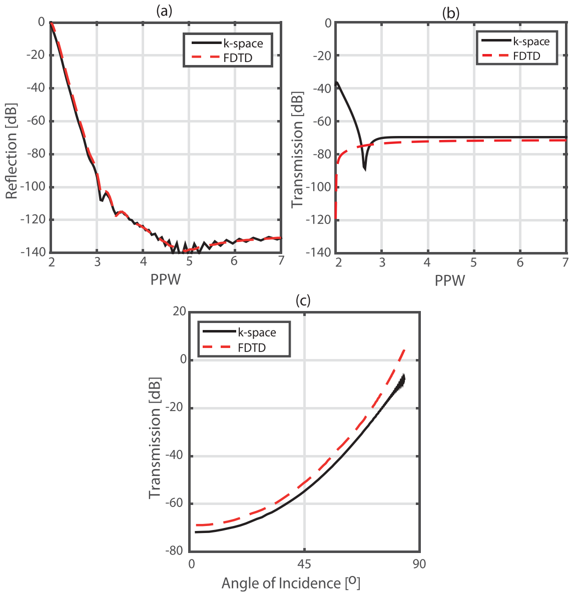


FIG. 4. (Color online) (a) Reflection from and (b) transmission through the PML as a function of spatial sampling, and (c) transmission to the edge of the grid as a function of angle of incidence. Here 0 dB means that the reflected or transmitted wave has the same amplitude as the incident wave, i.e., total reflection or transmission.

considered when designing acoustic sources and considering the angles at which pressure waves will impinge on the PML.

D. Numerical dispersion

To examine the impact of numerical dispersion, a broadband pressure source was defined on a homogeneous 1D grid. Grids with the medium properties of both bone and brain tissue grids were tested, and for the *k*-space scheme c_{ref} was set to the speed of sound in brain tissue. For FDTD schemes, the temporal and spatial dispersive errors oppose each other, with reduced dispersive error at higher CFL numbers.[14] Therefore the CFL was set to 0.5 for these simulations, the highest value at which both schemes are stable in 3D.[14,15] The time-varying pressure was recorded at a distance of 1 cm, and the phase spectra of the recorded pressure signals were compared to a dispersion-free reference simulation obtained with perfect *k*-space correction. This allowed calculation of phase error per cm propagated in either tissue type as a function of acoustic frequency. Using the model for transcranial propagation of ultrasound to a deep brain target described in Sec. III A, this was used to calculate the sampling criteria required to obtain <1.5 mm positional error for the direct path, and for each reverberation, shown in Table III. To compare, when sampled at 18 PPP, the *k*-space scheme is exact for soft tissue and gives a 19 μm error in the focal position per cm propagated in bone, PSTD gives a 25 μm error per cm in soft-tissue and a 50 μm error per cm in bone, and FDTD gives 25 μm error per cm in soft-tissue and a 130 μm error per cm in bone.

TABLE III. Temporal sampling required to obtain <1.5 mm targeting error.

Target	Direct path [PPP]	Reverb. I [PPP]	Reverb. II [PPP]	Reverb. III [PPP]
FDTD	17.9	21.9	23.9	25.1
PSTD	11.2	18.9	24.2	28.6
k-space	4.0	4.0	4.6	5.4

Results are given in terms of temporal PPP to allow a comparison between dispersive error for the FDTD scheme, which is dependent on both spatial and temporal sampling, and PSTD and *k*-space schemes, which are dependent on temporal sampling only. Equation (1) shows how the CFL defines a fixed ratio between spatial and temporal sampling. Different CFL numbers will result in a different combination of spatial and temporal requirements for the FDTD scheme. Values for both PSTD schemes are dependent only on temporal sampling, and do not require a particular spatial sampling to reduce dispersive error. However, with that in mind, the results shown in Table III do demonstrate a clear reduction in the temporal sampling required to minimize dispersive positional error below acceptable levels for the *k*-space scheme compared to FDTD and PSTD schemes.

E. 1D medium discontinuities

To examine the delay in convergence due to inaccuracies in reflection and transmission from medium discontinuities, broadband pressure sources were propagated across a bone-soft tissue interface (propagation direction makes no difference). The incident, reflected, and transmitted waves were recorded and the power spectra used to calculate intensity reflection and transmission coefficients for each wavenumber. Percentage error in these coefficients was calculated through comparison with the analytical values. To examine the dependence of this error on the size of the impedance change, these tests were repeated with the impedance of the bone varied up to ten times that of the soft tissue, with sound speed and density varied independently. No difference was observed between *k*-space and PSTD schemes.

Figure 5 shows the resulting error in intensity transmission and reflection coefficients as a function of PPW, and as a function of impedance change for a PPW of 6. FDTD and *k*-space schemes demonstrate a similar error even at high

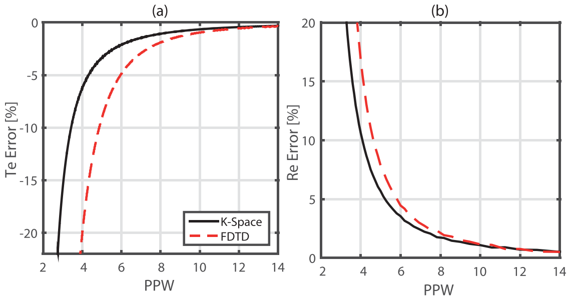


FIG. 5. (Color online) Error in simulated intensity (a) transmission (Te) and (b) reflection (Re) coefficients.

impedance changes. Changes in density result in increased deviation from the correct coefficient due to the interpolation of the density values on a staggered grid.

To calculate how the error in 1D reflection and transmission will affect transcranial focusing, the error in the intensity following transmission across a bone layer (i.e., across two bone/soft-tissue interfaces) was computed as

$$\text{Error} = \frac{\widetilde{\text{Te}}^2 - \text{Te}^2}{\text{Te}^2}, \qquad (2)$$

where Te is the analytical energy transmission coefficient between bone and soft tissue, and $\widetilde{\text{Te}}$ is the simulated energy transmission coefficient as a function of spatial PPW. Reflections inside the skull and skull cavity were not considered. To obtain <10% intensity error, the FDTD scheme requires 5.9 PPW while the *k*-space and PSTD schemes require 4.3 PPW. This result is notable, as the representation of discontinuities in medium properties has previously been identified as a limitation of PSTD methods.[15] During the update steps of these schemes, the pressure field is multiplied by the maps of medium density and sound speed, before being evaluated by a truncated Fourier series. Step changes in medium properties will therefore introduce Gibbs phenomenon into the pressure field, as described in Sec. II C. However, these results indicate that, for the step change in medium properties between bone and soft tissue, the error resulting from the representation of this change within the FDTD scheme tested is greater.

F. Staircasing

Staircasing refers to the spatial approximation that is necessary when attempting to define continuous geometries on a discrete, regular Cartesian grid in 2D and 3D. Curved surfaces and lines at an angle to the Cartesian directions will be approximated in a stair-stepped manner, and certain vertex and edge positions do not correspond to points on the grid.[24] The impact of staircasing was examined separately for acoustic sources and medium distributions. Tests involved recording the time-varying pressure at a number of positions across the field resulting from a staircased representation of a source or medium, and comparison of these signals with references obtained from a staircase free simulation. Error was then quantified as the percentage error in the amplitude of the temporal peak intensity (calculated using a plane wave assumption) and its positional error (derived from the change in the time of arrival of the intensity peak) as a percentage of wavelength. A positional error of 50% of wavelength corresponds to 1.5 mm for a source frequency of 500 kHz in brain tissue. These errors were calculated for each recording position, and then averaged across the field to give mean errors in peak intensity amplitude and position. No notable difference in error was observed between FDTD, PSTD, and *k*-space schemes across all tests.

The impact of staircasing on acoustic sources was examined using line-sources with a length of 65 × *dx*, where *dx* is the spatial discretization step, at a series of angles to the Cartesian grid. These included four angles that form Pythagorean triangles on the grid, specifically: ∼14.3° (with Pythagorean triple 16, 63, 65), ∼22.6° (25, 60, 65), ∼30.5° (33, 56, 65), and ∼36.9° (39, 52, 65). For these angles, the line-source endpoints are coincident with specific grid point positions, and any error is only due to the staircased representation of the line, rather than endpoint misregistration (both are aspects of staircasing error). A source defined parallel to a Cartesian axis was used as a non-staircased reference. The sources were excited with 10 cycle acoustic tonebursts with a range of central wavenumbers sampled at 3–100 PPW. The amplitudes of the source signals were normalized based on any change in the number of distinct source points used when defining an angled line source when compared to the aligned case. The time-varying field was recorded at 100 points positioned in front of the line source, and the sensor map was rotated with the line source to maintain source-sensor geometry. The simulation layout is shown in Figs. 6(a) and 6(b).

Mean errors in the amplitude and position of the peak intensity across the sensor field were calculated independently for each angle tested relative to the aligned, non-staircased reference case. The maximum mean errors across the range of angles tested for each PPW value are shown in Fig. 7. These results demonstrate that staircasing errors worsen with lower spatial sampling and are less serious for Pythagorean angles, when endpoints are correctly registered. The error in the position of the intensity peak never rises above 50% of wavelength for any source. Seventeen PPW are required to obtain <10% error in the amplitude of the intensity peak for all angles tested, while Pythagorean angles require only 7 PPW. Although the error examined here does not relate directly to the model for transcranial ultrasound propagation described above, these results do indicate that staircasing and spatial sampling must be considered when defining acoustic source distributions, and that error can be reduced by ensuring endpoint registration. The testing of multiple angles also allowed examination of how the exact mapping of the staircased line relates to the error observed in the resulting field. No clear relationship between the angle of the line source and the level of staircasing error was observed. However, a staircasing metric was defined as the average distance between the staircased source points and their equivalent equispaced points on an ideal angled line. It was observed that the convergence rate of the error of the staircased source maps, quantified as the average *L*2 error in the recorded pressure signals at the maximum spatial sampling tested, showed a strong dependence on this staircasing metric. Although only a simple metric, this indicates that the severity of staircasing error can be predicted through comparison of an ideal or parametric map of the intended geometry with its staircased representation.

The impact of staircasing of heterogeneous medium properties was examined in two separate tests. The first was conceptually similar to the examination of source staircasing. An acoustic point source excited by ten cycle acoustic tonebursts with central frequencies ranging from 3 to 80 PPW was propagated across a planar medium boundary (soft tissue-to-bone), defined at varying angles to the Cartesian axes. The time varying pressure field was recorded at 100

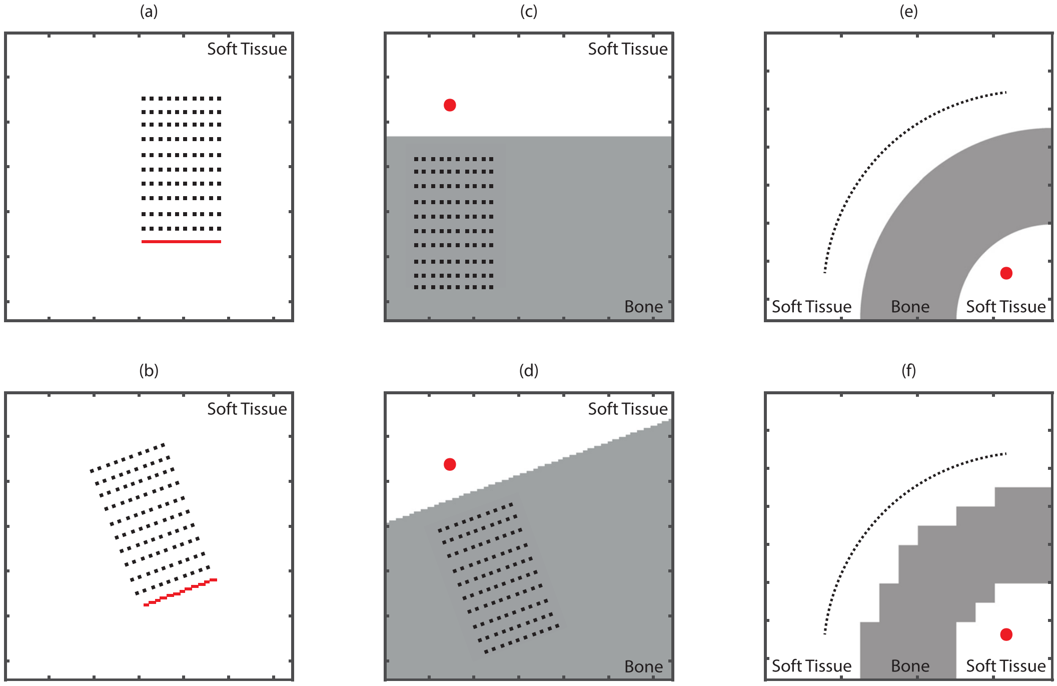


FIG. 6. (Color online) Simulation layouts used to test the impact of staircasing (not to scale). Sources are shown in red, and pressure recording positions in black. (a) Non-staircased line source used as reference, and (b) staircased line source used to examine error. (c) Non-staircased medium map used as a reference, and (d) staircased medium used to examine error. (e) High resolution map of a bone-tissue layer used as reference, and (f) downsampled medium.

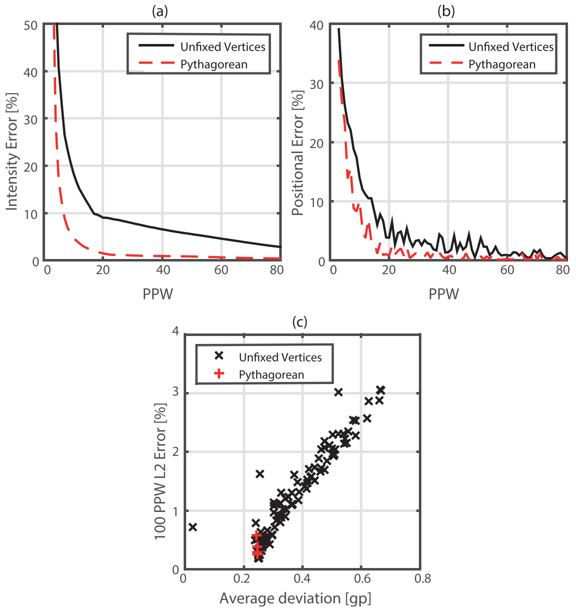


FIG. 7. (Color online) Error in staircased line sources as a function of spatial PPW. (a) Percentage error in peak intensity. (b) Positional error as a percentage of wavelength. (c) $L2$ error at 100 PPW sampling against average deviation from ideal line source. Pythagorean sources have both vertices of the parametric line source exactly defined on the grid.

sensor points following the interaction with the medium boundary. The sensor points were rotated with the medium boundary to maintain the simulation geometry. A non-staircased boundary defined along a Cartesian axis was used as a reference map. This simulation layout is shown in Figs. 6(c) and 6(d). The second test was designed as a more accurate model of staircasing in transcranial transmission. A 10 cycle, 2520 PPW toneburst was propagated through a medium map comprising a quarter circle bone layer. The medium was then artificially staircased through spatial downsampling, before being remapped to the original grid. A 3780 × 3780 simulation grid was used due to the large number of integer factors of 3780, which allowed the medium to be successively downsampled while maintaining positioning. The time-varying pressure was recorded across a quarter-circle, and error metrics computed through comparison with the least staircased medium distribution. This simulation layout is shown in Figs. 6(e) and 6(f). An effective PPW value for each level of downsampling was calculated through comparison of the source PPW with the new effective spatial discretization. Mean error measurements across the recorded fields as a function of PPW are shown in Fig. 8 for both medium staircasing tests. For the single interface model, the maximum mean errors across the range of angles tested are shown.

In terms of the impact on the model for transcranial propagation, the results for the bone layer model shown in Fig. 8 suggest that 20 PPW or above are required to obtain less than 10% mean error in intensity transmitted to an external transducer surface. As might be expected, the errors for

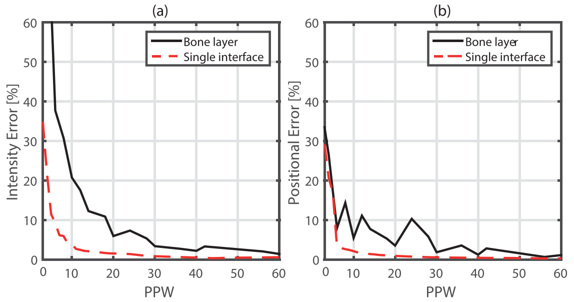


FIG. 8. (Color online) Error resulting from propagation through bone-layer and single interface staircased medium boundaries. (a) Percentage error in peak intensity magnitude. (b) Positional error of peak intensity as a percentage of wavelength.

the single interface are lower, with 6 PPW required to obtain the same error in peak intensity amplitude. The error in peak intensity position is less serious, with mean positional error never rising above 50% of wavelength (1.5 mm for 500 kHz ultrasound in brain tissue) for both tests, as with source staircasing. This may be due to staircasing introducing a random error in acoustic pathlength, leading to a defocusing and change in amplitude rather than a shifting in the peak position. To place this in context, the voxel size of clinical CT images is on the order of 0.5 mm at best. This corresponds to 6 PPW at 500 kHz, and fewer at higher frequencies. This suggests that staircasing may have a significant impact on simulations using image derived medium property maps. When considered alongside the results for source staircasing, these results indicate that staircasing error is likely the most serious of the numerical errors tested. The single interface medium model was also briefly tested using an elastic PSTD model, which indicated that medium staircasing may also have a pronounced impact on simulated mode conversion, although more rigorous testing is necessary.[25]

IV. CONVERGENCE TESTING

A. Overview

To examine the combined effects of numerical errors on the effectiveness of transcranial TR focusing, convergence testing of fully simulated TR was carried out in 2D and 3D. The general method is outlined in Fig. 9. It consists of forward propagation of a 10 cycle toneburst from a source point inside a virtual skull model to a circular (2D) or hemispherical (3D) virtual transducer-sensor array. As these tests were to examine the specific impact of numerical convergence, the simulated transducer array surface was modelled as a continuous surface made up of point transducers at the resolution of the spatial grid, and no attempt was made to replicate real transducer characteristics. This ensures that convergence is dependent only on the numerical accuracy of the forward simulations and will apply to alternate source conditions. The spatial discretization of these simulations was varied to correspond to a range of spatial PPW values for the central frequency of the source toneburst. The time-varying pressure signals recorded at the virtual transducer position were then time reversed and propagated into the head to refocus onto the target position. The reversal simulations were carried out at the finest discretization feasible, and the CFL was 0.3 for all simulations. Due to the change in spatial and temporal discretization, it was necessary to interpolate the pressure signals recorded in the forward simulations onto the spatial and temporal grids used in the reversal simulations [see Fig. 9(b)] using Cartesian triangulation and Fourier interpolation, respectively. In addition, the position of the source, defined at a grid point on the high resolution reversal grid, was not conserved due to the varying discretization of the forward simulations, and instead the nearest neighboring point was used. The peak pressure occurring in a time window of 20 acoustic cycles centered on the expected time of refocusing was recorded across the brain volume. Convergence was established by examining focusing quality as a function of the discretization used in the forward simulations. The focusing metrics examined were the spatial and temporal peak pressure across the brain volume, the distance of the peak from the target position, and the FWHM of the focal spot size. Peak pressure and focus FWHM were normalized relative to the results obtained for the most highly resolved forward simulation.

Simulations in 3D were carried out with toneburst sources with central frequencies of 500 kHz, while testing

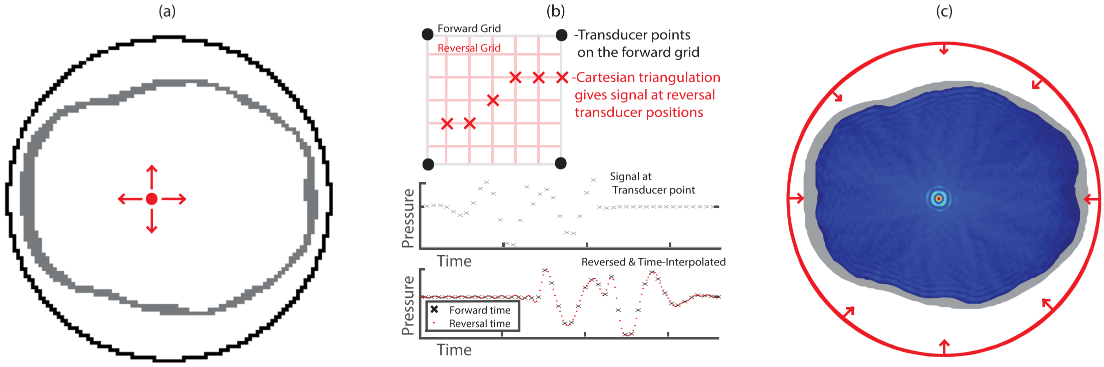

(a)
(b)
(c)
Forward Grid
Reversal Grid
-Transducer points on the forward grid
-Cartesian triangulation gives signal at reversal transducer positions
Signal at Transducer point
Pressure
Time
Reversed & Time-Interpolated
Pressure
Forward time
Reversal time
Time

FIG. 9. (Color online) Method for convergence testing in both 2D and 3D. (a) Forward simulation of ultrasound propagation from target point to simulated transducer surface with low-PPW forward discretization. (b) Triangulation is used to extract pressure signals at the transducer positions for the reversal simulation. This signal is then Fourier interpolated onto the finer temporal grid, and time-reversed. (c) The time reversed pressure signals are propagated back into the high-PPW head model in a reversal simulation. An example of the pressure field recorded across the brain volume and used to evaluate focusing is shown.

TABLE IV. Simulation criteria used in convergence testing.

	2D		3D
Simulation parameter	250 kHz	500 kHz	500 kHz
Simulation grid size	108^2–3096^2	162^2–3096^2	144^3–1024^3
Spatial discretization [mm]	3.1–0.0618	1.6–0.0624	1.6–0.18683
Forward and reverse simulation times [ms]	0.40499 and 0.42499	0.38864 and 0.39864	0.3879 and 0.3979
Temporal discretization [ns]	292.5–5.79	146.25–5.85	146.25–17.515
Transducer radius [mm]	90.9	91.8	95.0
Target Cartesian deviation from transducer focus [mm]	[2,3]	[2,3]	[5,5,5]
Simulation runtime [mins]	0.2–83	0.3–78	10–6736

in 2D used both 250 and 500 kHz tonebursts. The simulation parameters employed across these tests are shown in Table IV. Grid sizes include the absorbing PML layer. The forward simulations were run for the time taken for an acoustic wave to propagate across the grid three times, plus the duration of the source toneburst. The reversal simulation was run for the additional time of five acoustic cycles in order to fully capture the reconstructed toneburst. The medium property map used in the convergence tests was derived from a T1-weighted MR image obtained from the Imperial College brain development dataset.[26] Brain and skull volumes were extracted using the FSL MRI processing toolbox[27] and converted into a surface mesh using the iso2mesh toolbox.[28] The reference surface mesh was then sampled onto a 2D or 3D grid with the required spatial discretization steps. Examples of the 2D maps used in forward and reversal simulations, and their varying discretizations, are shown in Figs. 9(a) and 9(c). In each case, the skull was modelled as a single homogeneous bone layer, and the rest of the simulation domain was assigned brain tissue medium properties. Although fully heterogeneous models of the skull have demonstrated tighter model-driven TR focusing in some cases, homogeneous models remain effective.[29] Furthermore, they allow accurate resampling of the bone map to multiple spatial discretizations without interpolation, and ensure that convergence is dependent on the accuracy of the numerical simulation, rather than the mapping and homogenization of acoustic medium properties. For the k-space scheme c_{ref} was set to the speed of sound in brain tissue.

B. Convergence testing in 2D

2D convergence testing was carried out using 10 cycle acoustic toneburst sources corresponding to 250 and 500 kHz frequencies. Reversal simulations were carried out using a 3072 × 3072 grid, with a spatial discretization corresponding to 50 PPW for 500 kHz and 101 PPW for 250 kHz. Forward simulations were carried out using the k-space and FDTD schemes. Reversal simulations were carried out using the k-space scheme only.

The results of the 2D convergence testing are shown in Fig. 10, with error in peak pressure position given relative to the source point in the forward simulations. Refocusing quality in the reversal simulations increases with the spatial PPW of the simulated frequency in the forward simulations. Several key points can be derived from these results. First, for all three metrics, the k-space scheme demonstrates convergence at approximately ∼2 PPW below the FDTD scheme. Second, although there is some difference in the position and size of the focus at very low sampling, both 250 and 500 kHz demonstrate similar behavior as a function of spatial sampling. This indicates that these results can, to some extent, be generalized, and suggests that the reversal simulations have converged for both frequencies. Finally, of the three refocusing metrics examined, normalized peak pressure across the brain volume [Fig. 10(a)] requires higher spatial sampling to converge than either focal volume or the deviation of the focus from the target. This suggests that when fine pressure control is not required, coarser sampling criteria may suffice.

C. Convergence testing in 3D

3D convergence testing employed a similar protocol to the 2D convergence testing described above. Ten cycle

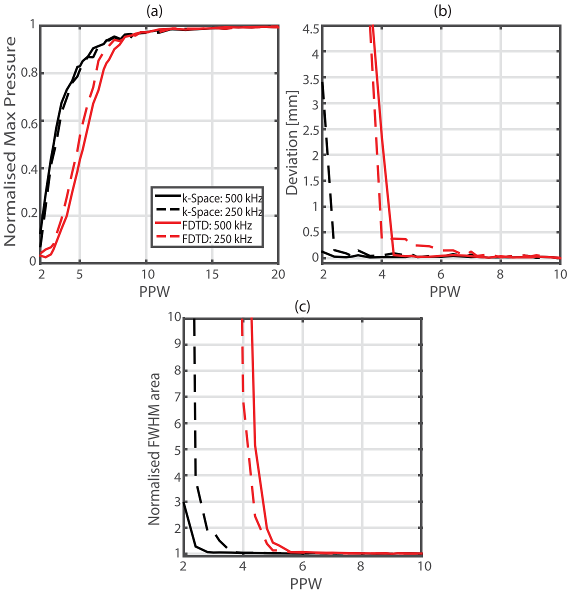


FIG. 10. (Color online) Results of 2D convergence testing. (a) Peak pressure recorded across the brain volume. (b) The deviation of the peak pressure from the location of the forward simulation source. (c) Normalized FWHM area of the focal spot size. Normalization is relative to results obtained with the most highly resolved forward simulation.

acoustic tonebursts with a central frequency of 500 kHz were propagated from a source point inside a full 3D model of the human skull to a simulated hemispheric transducer. Reversal simulations were carried out using a 1024 × 1024 × 1024 simulation grid with a spatial discretization corresponding to 16.7 PPW. Forward and reversal simulations were carried out using the *k*-space scheme only. Testing was also carried out using homogeneous media for forward and reverse simulations, to test the accuracy of the spatiotemporal interpolation. The results for heterogeneous 3D convergence testing are shown in Fig. 11. The results for normalized peak pressure amplitude in Fig. 11(a) show similar trends to the 2D results. Six PPW are required to obtain 95% reconstruction of pressure at the target (corresponding to <10% drop in intensity) and ∼10 PPW to attain convergence. The convergence of the volume of the focal spot [Fig. 11(c)] shows similar behavior, although it only requires ∼6 PPW to fully converge. Given the known impact of staircasing in 2D, the faster convergence here is likely due to a reduced staircasing error for 3D geometry. The results in Fig. 11(b) show the error in the position of the pressure peak relative to both the reversal target and the shifted forward source point. This error is reduced compared to the 2D case, never rising above 1.1 mm and, when computed relative to the position of the forward source, is stable using sampling as low as 2 PPW. However it can be seen from the other results that at this sampling the peak pressure amplitude is much lower, with a larger focal spot. The apparent periodicity in the positional error when calculated relative to the parametrically defined target point is likely due to the oscillating distance of a definable source point from this position.

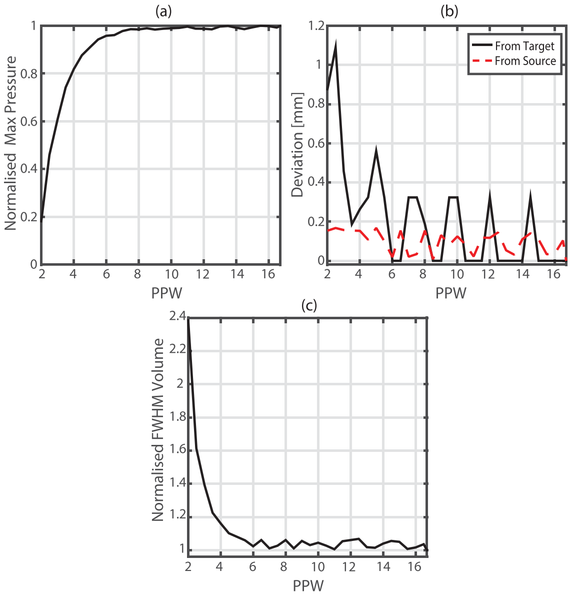


FIG. 11. (Color online) 3D convergence testing results. (a) Normalized peak pressure amplitude across the brain. (b) Deviation of pressure peak from both the parametrical defined target and the source used in the forward simulation. (c) Normalized half-maximum focal volume. Normalization is relative to results obtained with the most highly resolved simulation.

Accordingly, the reduction in error and lack of dependence on spatial sampling when calculating positional error relative to the actual source position from the forward simulation suggests that any error in the position of the peak is in fact due solely to the misregistration of the source points. Generally, these results confirm that when targeting accuracy is the prime concern, spatial sampling requirements are laxer than when a tight focal volume with known peak pressure amplitude is required. This is in agreement with previous studies which have demonstrated that good spatial targeting of HIFU can be obtained via simulated TR using relatively coarse spatial sampling.[11,30] Homogeneous testing demonstrated total convergence across all metrics by 3.5 PPW, which is to be expected given the behavior of the PML and BLI, discussed in Sec. III.

V. SUMMARY AND DISCUSSION

In this paper, a comprehensive assessment of the impact of different factors that affect the convergence of numerical models for the simulation of transcranial ultrasound propagation was carried out. The spatial and/or temporal sampling required to reduce inaccuracies below the levels required for targeting of deep brain nuclei for neurostimulation were determined.

Initial simulations examined reduction in the effectiveness of the PML, and the impact of the BLI when using *k*-space and PSTD methods. Both the PML and BLI lead to erroneous pressures appearing on the grid when simulating frequencies sampled at close to the spatial Nyquist limit. Although both of these effects have the potential to seriously reduce the accuracy of the simulations, they decrease in severity rapidly as the rate of spatial sampling increases. Above ∼3 PPW, erroneous pressures resulting from both BLI and PML effects were at least −60 dB below the amplitudes of the ultrasound sources being simulated.

Numerical dispersion has a serious effect on the accuracy of FDTD and PSTD schemes, resulting in high temporal sampling requirements to reduce positional error. However, this was not the case for the *k*-space scheme, where ∼3 PPW will serve to limit dispersion sufficiently for transcranial transmission for any stable CFL value. Errors in reflection and transmission from discontinuous medium properties manifest in the magnitude of reflected and transmitted simulated intensities. Despite the representation of step changes in media previously being identified as a key limitation of PSTD schemes, the error was shown to be more severe for the 2-4 FDTD scheme tested. To reduce error in the intensity below 10% following transcranial transmission, *k*-space and PSTD schemes require 4.3 PPW, while FDTD requires 5.9 PPW.

Staircasing of source and medium geometries was shown to require the most stringent sampling criteria to obtain required accuracy, affecting FDTD, PSTD, and *k*-space schemes equally. Both source and medium staircasing were shown to have a greater impact on the intensity amplitude of the toneburst signal being examined than the position of the intensity peak. The results shown in Figs. 7 and 8 indicate that ≥20 PPW are required to reduce the error in peak intensity following transcranial transmission below 10%. The preliminary examination of a potential staircasing metric also suggests that the error resulting from a particular staircased geometry is directly related to its deviation from the ideal geometry.

Convergence testing of a fully simulated TR protocol using 2D and 3D head models was used to examine the impact of all numerical errors in concert. Testing in 2D for 250 and 500 kHz ultrasound showed a faster rate of convergence for all focusing metrics for the *k*-space scheme when compared to FDTD. In addition, the error in the peak pressure amplitude at the focus showed slower convergence than both the positional error, and the volume of the focus. This is likely due to the most serious source of error, medium staircasing, which was shown to have a greater impact on the peak intensity amplitude of transcranially transmitted ultrasound, than the position of the peak. Results in 3D showed similar trends to the 2D results for the convergence of the peak pressure amplitude. The focal spot size showed slightly slower convergence in the 3D case, while the positional error demonstrated almost no dependence on the sampling rate of the forward simulation. This indicates that less stringent sampling may suffice for applications concerned only with the position of the focus, rather than the size of the focal spot and the exact amplitude at the target. When fine control over the pressure amplitude is required, stricter sampling may be necessary. Despite the relatively severe error resulting from staircasing at higher spatial sampling, all three metrics of focusing quality were well converged at below 20 PPW. This discrepancy may be due to the differences between the convergence testing protocol and the specific test used to examine staircasing across a bone layer, and suggests that the influence of staircasing is case specific.

The work described above is subject to some limitations, primarily the degree to which the examination of individual numerical errors can be generalized to different setups, although trends and qualitative observations remain valid. Many of the tests only examine toneburst sources, and the error is evaluated over a small field, with pressure recorded at a limited number of sensor positions (see Fig. 6). A separate 2-2 FDTD scheme was used to examine the effectiveness of the PML, which may not be exactly relatable to the commonly used 2-4 FDTD scheme. Furthermore, the impact of shear wave propagation was not examined. This will not have affected 1D or homogeneous simulations, but a more thorough examination of medium staircasing should include testing of elastic wave propagation. Similarly, no effort was made to examine the manifestation of numerical errors when modeling nonlinear propagation or acoustic absorption, which will become relevant for applications requiring the simulation of high-amplitude ultrasound, such as HIFU. It should be noted that, in simulated TR, accounting for acoustic absorption occurs in the post-processing stage, when converting recorded pressure signals into driving amplitudes,[12] and work examining absorption should focus on this stage of the simulated TR process.

The results presented here are primarily relevant to the simulation of transcranial ultrasound propagation for TR targeting of deep brain structures with finely controlled ultrasound for the purposes of neurostimulation. However, the criteria and simulations presented are also relevant to alternative low-intensity, transcranial ultrasonic therapies such as opening the blood-brain barrier with ultrasound,[22] as well as existing transcranial HIFU ablation therapies. Use of appropriately discretized simulations will ensure accurate targeting and effective therapy as the field of ultrasonic neurostimulation develops.

ACKNOWLEDGMENTS

The authors would like to thank Charlotte Stagg and Adam Thomas for helpful discussions, and Nishant Ravikumar and Zeike Taylor for the provision of the skull and brain meshes. This work was supported by the Engineering and Physical Sciences Research Council (ESPRC), UK. J.J. is financed from the SoMoPro II programme. This research has acquired a financial grant from the People Programme (Marie Curie Action) of the Seventh Framework Programme of EU according to the REA Grant Agreement No. 291782. The research is further co-financed by the South-Moravian Region. This work reflects only the author's view and the European Union is not liable for any use that may be made of the information contained therein. Computational resources were provided by the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), funded by the European Regional Development Fund and the national budget of the Czech Republic via the Research and Development for Innovations Operational Programme, as well as Czech Ministry of Education, Youth and Sports via the project Large Research, Development and Innovations Infrastructures (LM2011033).

APPENDIX

Simulations were carried out using the open source *k*-Wave toolbox for MATLAB, C++. The toolbox includes *k*-space, PSTD, and 2-4 FDTD codes for the time-domain simulation of acoustic fields. 1D simulations were carried out in the MATLAB environment on a Dell Precision T1700 with an Intel Xeon E3-1240 3.40 GHz CPU and 16 GB of RAM running Windows 10 64 bit. 2D simulations were carried out in the MATLAB environment with CUDA hardware acceleration on a Dell PowerEdge R730 compute server with 2 × 6-core Xeon E5-2620 2.4 GHz CPUs, 64 GB of 1866 MHz memory, on an Nvidia Titan X GPU with 3072 CUDA cores and 12 GB of memory. The largest 2D simulations had a domain size of 3780^2 including the PML and comprised 258 462 time steps, with a total runtime of 10.6 h. 3D simulations were carried out on the IT4I Salomon supercomputing cluster. Each simulation was carried out on Intel Xeon E5-4627v2, 3.3 GHz, 8cores and 256 GB of RAM per simulation. The largest 3D simulations had a domain size of 1024^3 including the PML and comprised 22 718 time steps, with a total runtime of 112.3 h. The skull mesh used in convergence testing is Copyright Imperial College of Science, Technology and Medicine 2007. All rights reserved. www.brain-development.org.

[1]P. J. Karas, C. B. Mikell, E. Christian, M. A. Liker, and S. A. Sheth, "Deep brain stimulation: A mechanistic and clinical update," Neurosurg. Focus 35(5), E1–E16 (2013).

[2] C. Hamani, E. Richter, J. M. Schwalb, and A. M. Lozano, "Bilateral subthalamic nucleus stimulation for Parkinson's disease: A systematic review of the clinical literature," Neurosurgery **56**(6), 1313–1324 (2005).

[3] Y. Tufail, A. Matyushov, N. Baldwin, M. L. Tauchmann, J. Georges, A. Yoshihiro, S. I. Helms Tillery, and W. J. Tyler, "Transcranial pulsed ultrasound stimulates intact brain circuits," Neuron **66**(5), 681–694 (2010).

[4] A. Morel, *Stereotactic Atlas of the Brain and Basal Ganglia* (Informa Healthcare, New York, 2007), pp. 1–160.

[5] A. M. Lozano and N. Lipsman, "Probing and regulating dysfunctional circuits using deep brain stimulation," Neuron **77**(3), 406–424 (2013).

[6] P. P. Ye, J. R. Brown, and K. B. Pauly, "Frequency dependence of ultrasound neurostimulation in the mouse brain," Ultrasound Med. Biol. **42**(7), 1512–1530 (2016).

[7] W. Legon, T. Sato, A. Opitz, J. Mueller, A. Barbour, A. Williams, and W. J. Tyler, "Transcranial focused ultrasound modulates the activity of primary somatosensory cortex in humans," Nat. Neurosci. **17**(January), 322–329 (2014).

[8] W. Lee, H. Kim, Y. Jung, I.-U. Song, Y. A. Chung, and S.-S. Yoo, "Image-guided transcranial focused ultrasound stimulates human primary somatosensory cortex," Sci. Rep. **5**, 8743–8753 (2015).

[9] J. F. Aubry, M. Tanter, M. Pernot, J. L. Thomas, and M. Fink, "Experimental demonstration of noninvasive transskull adaptive focusing based on prior computed tomography scans," J. Acoust. Soc. Am. **113**(1), 84–93 (2003).

[10] G. W. Miller, M. Eames, J. Snell, and J. F. Aubry, "Ultrashort echo-time MRI versus CT for skull aberration correction in MR-guided transcranial focused ultrasound: In vitro comparison on human calvaria," Med. Phys. **42**(5), 2223–2233 (2015).

[11] D. Chauvet, L. Marsac, M. Pernot, A. L. Boch, R. Guillevin, N. Salameh, M. Tanter, and J. F. Aubry, "Targeting accuracy of transcranial magnetic resonance–guided high-intensity focused ultrasound brain therapy: A fresh cadaver model," J. Neurosurg. **118**(5), 1046–1052 (2013).

[12] F. Marquet, M. Pernot, J. F. Aubry, G. Montaldo, M. Tanter, and M. Fink, "Non-invasive transcranial ultrasound therapy based on a 3D CT scan: Protocol validation and in vitro results," Phys. Med. Biol. **54**(9), 2597–2613 (2009).

[13] Y. Jing, C. Meral, and G. Clement, "Time-reversal transcranial ultrasound beam focusing using a k-space method," Phys. Med. Biol. **57**(4), 901–917 (2012).

[14] J. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, 2nd ed. (SIAM: Society for Industrial and Applied Mathematics, Pacific Grove, CA, 2004), pp. 1–427.

[15] M. Tabei, T. D. Mast, and R. C. Waag, "A k-space method for coupled first-order acoustic propagation equations," J. Acoust. Soc. Am. **111**(1), 53–63 (2002).

[16] B. E. Treeby, J. Jaros, A. P. Rendell, and B. T. Cox, "Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method," J. Acoust. Soc. Am. **131**(6), 4324–4336 (2012).

[17] L. N. Trefethen, *Spectral Methods in MATLAB*, 1st ed. (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000), pp. 1–39.

[18] B. E. Treeby and B. T. Cox, "A k-space Green's function solution for acoustic initial value problems in homogeneous media with power law absorption," J. Acoust. Soc. Am. **129**(6), 3652–3660 (2011).

[19] B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave-fields," J. Biomed. Opt. **15**(2), 021314 (2010).

[20] F. Duck, *Physical Properties of Tissue. A Comprehensive Reference Book* (Academic Press, London, 1991), pp. 73–135.

[21] J. Sun and K. Hynynen, "Focusing of therapeutic ultrasound through a human skull: A numerical study," J. Acoust. Soc. Am. **104**(3 Pt 1), 1705–1715 (1998).

[22] T. Deffieux and E. E. Konofagou, "Numerical study of a simple transcranial focused ultrasound system applied to blood-brain barrier opening," IEEE Trans. Ultrason. Ferroelectr. Freq. Control **57**(12), 2637–2653 (2010).

[23] J. P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," J. Comput. Phys. **114**, 185–200 (1994).

[24] J. Van Aken and M. Novak, "Curve-drawing displays algorithms for raster," ACM Trans. Graph. **4**(2), 147–169 (1985).

[25] T. Bohlen and E. H. Saenger, "Accuracy of heterogeneous staggered-grid finite-difference modeling of Rayleigh waves," Geophysics **71**(4), 109–115 (2006).

[26] R. A. Heckemann, S. Keihaninejad, P. Aljabar, D. Rueckert, J. V. Hajnal, and A. Hammers, "Improving intersubject image registration using tissue-class information benefits robustness and accuracy of multi-atlas based anatomical segmentation," Neuroimage **51**(1), 221–227 (2010).

[27] M. Jenkinson, C. F. Beckmann, T. E. J. Behrens, M. W. Woolrich, and S. M. Smith, "FSL," Neuroimage **62**(2), 782–790 (2012).

[28] F. Qianqian and D. A. Boas, "Tetrahedral mesh generation from volumetric binary and grayscale images," in *IEEE International Symposium on Biomedical Imaging From Nano to Macro* (IEEE, Boston, 2009), pp. 1142–1145.

[29] R. M. Jones and K. Hynynen, "Comparison of analytical and numerical approaches for CT-based aberration correction in transcranial passive acoustic imaging," Phys. Med. Biol. **61**(1), 23–36 (2016).

[30] R. Narumi, K. Matsuki, S. Mitarai, T. Azuma, K. Okita, A. Sasaki, K. Yoshinaka, S. Takagi, and Y. Matsumoto, "Focus control aided by numerical simulation in heterogeneous media for high-intensity focused ultrasound treatment," Jpn. J. Appl. Phys. **52**(7S), 07HF01 (2013).

C.5 Photoacoustic Imaging

Treeby, B. E.; **Jaros, J.**; Cox, B. T.: Advanced photoacoustic image reconstruction using the k-Wave toolbox. In *SPIE Photons Plus Ultrasound: Imaging and Sensing*, vol. 9708, 2016. ISBN 978-1-62841-942-9. ISSN 1605-7422. page 97082P. doi:10.1117/12.2209254.

Advanced photoacoustic image reconstruction using the k-Wave toolbox

B. E. Treeby*, J. Jaros†, and B. T. Cox*

*Department of Medical Physics and Biomedical Engineering, University College London, UK
†Faculty of Information Technology, Brno University of Technology, Czech Republic

ABSTRACT

Reconstructing images from measured time domain signals is an essential step in tomography-mode photoacoustic imaging. However, in practice, there are many complicating factors that make it difficult to obtain high-resolution images. These include incomplete or undersampled data, filtering effects, acoustic and optical attenuation, and uncertainties in the material parameters. Here, the processing and image reconstruction steps routinely used by the Photoacoustic Imaging Group at University College London are discussed. These include correction for acoustic and optical attenuation, spatial resampling, material parameter selection, image reconstruction, and log compression. The effect of each of these steps is demonstrated using a representative *in vivo* dataset. All of the algorithms discussed form part of the open-source k-Wave toolbox (available from http://www.k-wave.org).

1. INTRODUCTION

Forming an image from measured time domain signals is an essential step in photoacoustic tomography (PAT). Superficially, this would appear to be a solved problem, particularly given the large number of published photoacoustic images.[1] Indeed, commercial photoacoustic scanners are now available that can generate images in real time,[2] and exact reconstruction formulae for canonical geometries have existed in the mathematical literature for some time.[3] However, in the practical case, there are many complicating factors that make it difficult to obtain high-resolution images with good signal-to-noise. These include the data being incomplete (e.g., because the detection aperture is limited or spatially undersampled),[4] filtering effects (e.g., because the transducer elements have limited sensitivity and bandwidth, and have a finite size),[5, 6] limited penetration depth (due to optical and acoustic attenuation),[1, 7] and uncertainties in the material properties needed for the reconstruction.[8, 9] While advances have been made to address many of these challenges, the rapid growth in the development and application of photoacoustic technology means that there is often a disconnect between researchers developing new algorithms and those performing *in vivo* imaging studies. In this paper, the preprocessing, image reconstruction, and post-processing steps routinely used by the Photoacoustic Imaging Group at University College London (UCL) to generate high-resolution photoacoustic images are discussed. The purpose is to provide insight into the impact of applying different techniques on reconstructed photoacoustic images. All of the algorithms discussed form part of the open-source k-Wave image reconstruction toolbox developed at UCL (available from http://www.k-wave.org).[10] This makes it easy for other researchers to apply them to their own datasets.

2. DATASET AND ACQUISITION PARAMETERS

The dataset used to demonstrate the different image reconstruction steps is taken from Ref. 11 (see Fig. 3(f) and front cover). This is an *in vivo* dataset of the blood vasculature and a xenograft (or tumour) composed of K562 cells labelled with a tyrosinase-based genetic reporter taken in the flank of a nude mouse. The dataset was acquired using a photoacoustic scanning system based on a planar Fabry-Perot interferometer.[12] This was used as a 2D detection array with 142 × 141 detection elements (giving a total of 20,022 time domain waveforms), an element separation of 100 μm (giving a scan area of 14.2 × 14.1 mm), and an optically defined element size of 22 μm.[11] The -3 dB bandwidth of the detection system was $0.35 - 22$ MHz, and the three-sigma noise equivalent

Send correspondence to b.treeby@ucl.ac.uk.

Photons Plus Ultrasound: Imaging and Sensing 2016, edited by Alexander A. Oraevsky, Lihong V. Wang
Proc. of SPIE Vol. 9708, 97082P · © 2016 SPIE · CCC code: 1605-7422/16/$18 · doi: 10.1117/12.2209254

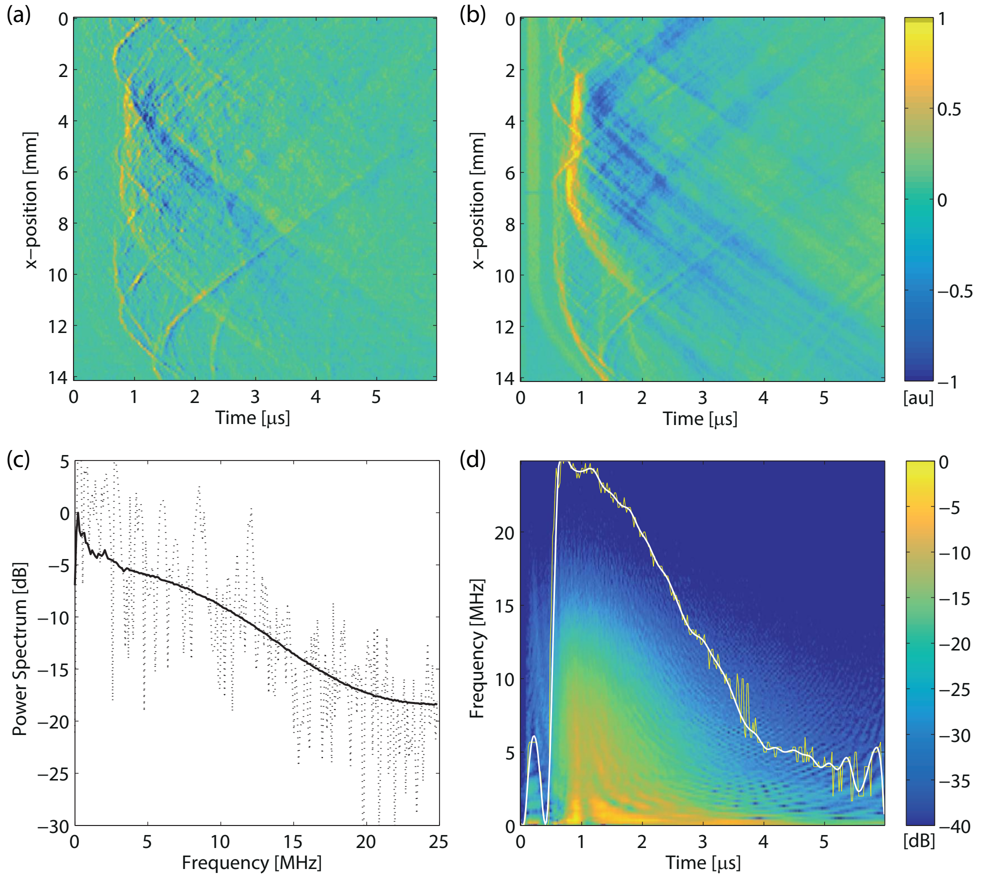


Figure 1. (a) Recorded time series data as a function of x-position and time at a y-position of 9 mm. (b) Summation of the recorded time series data across all y-positions. (c) Power spectrum of the recorded time domain photoacoustic signals. The solid line shows the spectrum averaged across all signals, and the dashed line shows the spectrum of the signal recorded in the centre of the array. (d) Average time frequency distribution of the recorded signals. The yellow (thin) and white (thick) lines show the filter cutoff frequency before and after spline fitting, respectively.

pressure over a 20 MHz bandwidth was 200 Pa. Time domain signals at each detector position were acquired sequentially at a sampling frequency of 50 MHz with no averaging, and the entire dataset took 6 minutes and 40 seconds to obtain using a 50 Hz excitation laser. Signal acquisition was triggered using a photodiode, such that the beginning of each time trace was synchronised with the excitation laser pulse reaching the tissue surface.

One line scan (as a function of x-position and time) from the recorded time series data is shown in Fig. 1(a). Bipolar signals from different optically absorbing tissue structures are clearly visible, which illustrates the low-noise and wide-bandwidth of the measurement system.[12] A summation of the recorded time series data across the y scanning dimension is shown in Fig. 1(b). The strongest signals appear around 1 μs ($\sim$1.5 mm), however, features are still visible after at least 3 μs ($\sim$4.5 mm). The power spectrum of the recorded data is shown in Fig. 1(c). The black line shows the spectrum averaged across all the signals, and the dotted line shows the spectrum of the signal recorded in the centre of the array. This demonstrates the broadband nature of the acquired photoacoustic signals, with energy from 350 kHz (the low frequency cutoff of the acquisition system) up to approximately 20 MHz.

3. IMAGE RECONSTRUCTION AND PROCESSING STEPS

3.1. Workflow

The pre-processing, reconstruction, and post-processing steps used to reconstruct the dataset shown in Fig. 1 are outlined below. The same procedure is routinely followed for most recent *in vivo* imaging studies using the Fabry-Perot scanner published by the Photoacoustic Imaging Group at UCL, e.g., Refs. 13–15.

1. Correct for acoustic attenuation in the time series data
2. Select sound speed that maximises the sharpness of the reconstructed image
3. Spatially upsample the acquired data to improve image resolution
4. Reconstruct the photoacoustic image
5. Correction for optical attenuation in the image data
6. Apply image processing and segmentation techniques as appropriate
7. Display as maximum intensity projection (MIP)

These steps are discussed in the following sections, with further details given in the references. The MATLAB and k-Wave functions used to perform these steps are also described. Note, to illustrate the effect of the individual steps on the final reconstructed image, the images displayed in each section are reconstructed cognisant of details discussed in other sections. In particular, the optimum sound speed is always used, except where otherwise noted.

3.2. Acoustic attenuation compensation

It is well known that soft biological tissue is acoustically absorbing, with the experimentally observed attenuation following a frequency power law of the form $\alpha = \alpha_0 f^y$. Due to the broadband nature of the ultrasound waves generated in photoacoustics, this causes a depth-dependent magnitude error and blurring of features in the reconstructed image. Applying compensation for acoustic attenuation can correct for these errors and improve the visibility and resolution of deeper vessels.[16, 17] Here, attenuation compensation is performed using time-variant filtering, which applies the correction directly to the time series data before reconstruction.[18] This approach is very flexible, and can be applied regardless of the acquisition system, geometry, or the reconstruction method used. The algorithm works by applying a non-stationary convolution matrix to each recorded time series p_{att}, where

$$\begin{bmatrix} p_{\text{corr}}(t_1) \\ \vdots \\ p_{\text{corr}}(t_N) \end{bmatrix} = \begin{bmatrix} F(t_1,\tau_1) & \dots & F(t_1,\tau_N) \\ \vdots & & \vdots \\ F(t_N,\tau_1) & \dots & F(t_N,\tau_N) \end{bmatrix} \begin{bmatrix} p_{\text{att}}(\tau_1) \\ \vdots \\ p_{\text{att}}(\tau_N) \end{bmatrix} . \tag{1}$$

The matrix F is constructed to allow attenuation compensation as a function of both frequency *and* travel distance (or time).[18] In k-Wave, this is applied using the function `attenComp`

```
sensor_data = attenComp(sensor_data, dt, c0, a0, y);
```

where `sensor_data` is a 2D matrix containing the recorded time series p_{att} in each row, `dt` is the size of the time step in units of s, `c0` is the sound speed in units of m/s, `a0` is the power law absorption prefactor in units of dB/(MHz^y cm), and `y` is the power law absorption exponent. This function also automatically selects a cutoff frequency for the attenuation compensation (to stop high frequency noise being amplified) based on the average time-frequency distribution of the signals.[18]

To compensate for acoustic attenuation in the dataset shown in Fig. 1, the power law absorption parameters were set to those of breast tissue, with `a0 = 0.75` and `y = 1.5`.[19] As the acoustic absorption parameters in

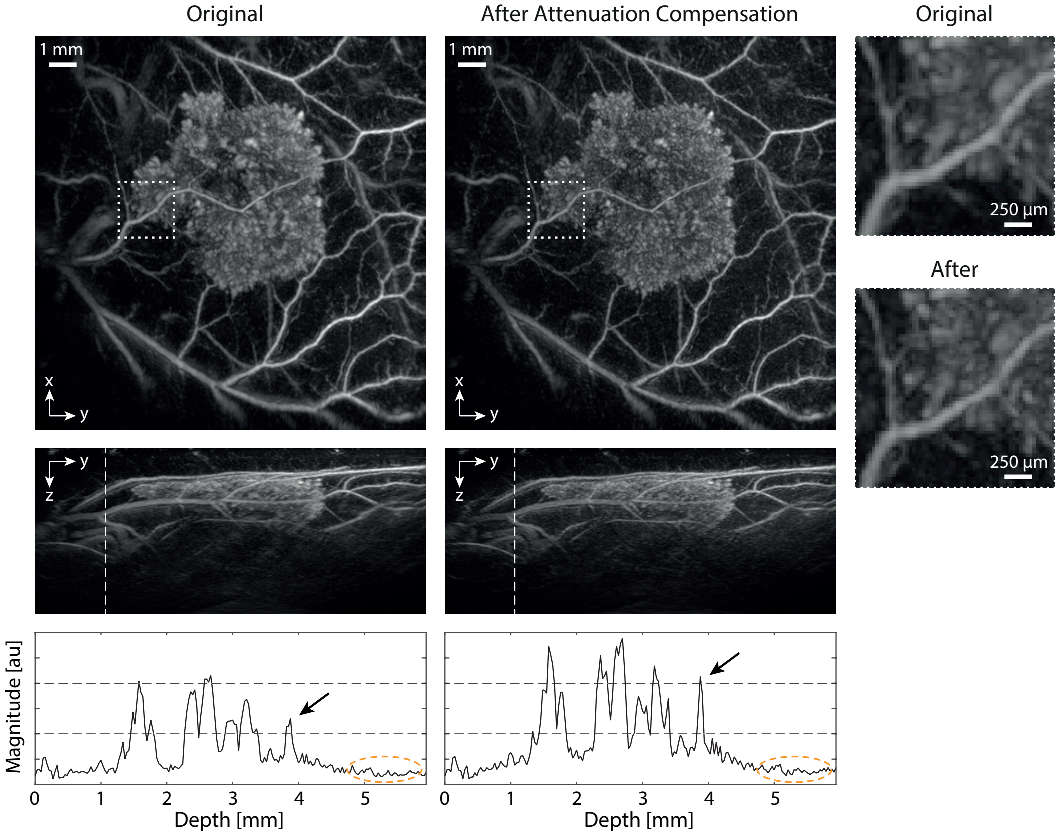


Figure 2. Three-dimensional reconstructions of the dataset shown in Fig. 1 with and without attenuation compensation. The three vertical panels show maximum intensity projections (MIPs) through the depth and lateral directions, and a one-dimensional profile through the lateral MIP at the location shown with a dashed white line. Magnification of the depth MIPs within the square dotted boxes is shown in the two right panels. Including attenuation compensation increases the sharpness of the vasculature and tumour, and improves the visibility and resolution of deeper vessels (arrows) without increasing the noise floor (circled region).

murine tissue are not well characterised, the average value in breast tissue is an apposite choice for soft-tissue containing a range of tissue types. The noise threshold and energy threshold used to select the filter cutoff frequency were set to 5% and 95%, respectively. The average time frequency distribution and the automatically selected filter cutoff frequency are shown in Fig. 1(d). Reconstructed images with and without acoustic attenuation compensation are shown in Fig. 2 for comparison. The tumour can be clearly seen as the sponge like structure in the centre of the image, along with the surrounding blood vasculature. When attenuation compensation is included, the sharpness of the vasculature and tumour is increased. This is particularly noticeable in the magnified images, which show an improvement down to the voxel level. The visibility and resolution of deeper vessels is also improved, as shown in the one-dimensional profiles. For example, the visibility and sharpness of the vessel denoted with the black arrows has significantly increased, without a corresponding increase in the noise floor (circled region). It is useful to point out that these improvements are not an image processing trick; they arise directly from rectifying the acoustic losses that physically occur as the photoacoustic waves propagate through tissue.

Regarding computational time, acoustic attenuation compensation using time variant filtering is very fast to apply. Using a desktop PC with an 8-core Intel Xeon E5-1660 v3 @ 3 GHz processor running MATLAB 2015a, the `attenComp` function took 3.1 s to calculate the average time frequency distribution, 0.45 s to select the cutoff frequency for the filter, 0.042 s to create the filter, and 0.19 s to apply the correction to all 20,022 time domain waveforms. If the filter cutoff frequency is chosen manually (based on the noise floor in the power spectrum for example[20]), there is no need to calculate the time frequency distribution, and the correction is even faster to apply.

3.3. Sound speed selection

The reconstruction of photoacoustic images requires knowledge of the sound speed within the medium so time-of-flight measurements can be correctly mapped back to the initial pressure distribution. Most reconstruction algorithms routinely used assume a constant value of sound speed. However, for *in vivo* imaging, the true value of sound speed is usually unknown. It is possible to estimate an appropriate value by systematically modifying the sound speed until the sharpness of the reconstructed image is maximised.[8, 21] This is based on the premise that features in the imaging volume are inherently sharp, and thus the correct sound speed is the one that produces the sharpest looking image. In k-Wave, sharpness is evaluated using the function `sharpness`.[21] By default, this uses a sharpness metric or focus function based on a simple finite difference gradient calculation known as the Brenner gradient. In 2D this is given by

$$\mathrm{F}_{\mathrm{brenner}} = \sum_{x,y} \left(f_{x+2,y} - f_{x,y}\right)^2 + \left(f_{x,y+2} - f_{x,y}\right)^2 \quad . \tag{2}$$

The sound speed value that maximises the sharpness metric can then be found by looping through a range of values as shown below, or using simple optimisation routines (e.g., `fminbnd` in MATLAB).

```
% set range of sound speeds to test
c_array = c_min:c_step:c_max;

% loop through sound speeds
for c_index = 1:length(c_array)

    % compute reconstruction using current value of sound speed
    recon = ...

    % take maximum intensity projection along the depth direction
    mip = max(recon, [], 3)

    % compute sharpness metric
    sharpness_metric(c_index) = sharpness(mip);

end

% find the index of the maximum sharpness
[~, max_index] = max(sharpness_metric);

% assign optimum sound speed
c_opt = c_array(max_index);
```

Figure 3(a) shows depth direction (enface) maximum intensity projections (MIPs) of the reconstructed tumour image using six different values of sound speed from 1400 m/s to 1600 m/s. As the sound speed is increased, the image is gradually focused and then defocused again. The corresponding focus function calculated using Eq. (2) is shown in Fig. 3(b). In this case, the focus function is unimodal, with a peak at 1515 m/s (shown with the dashed line). This is within the range of physiological values between fat (1430

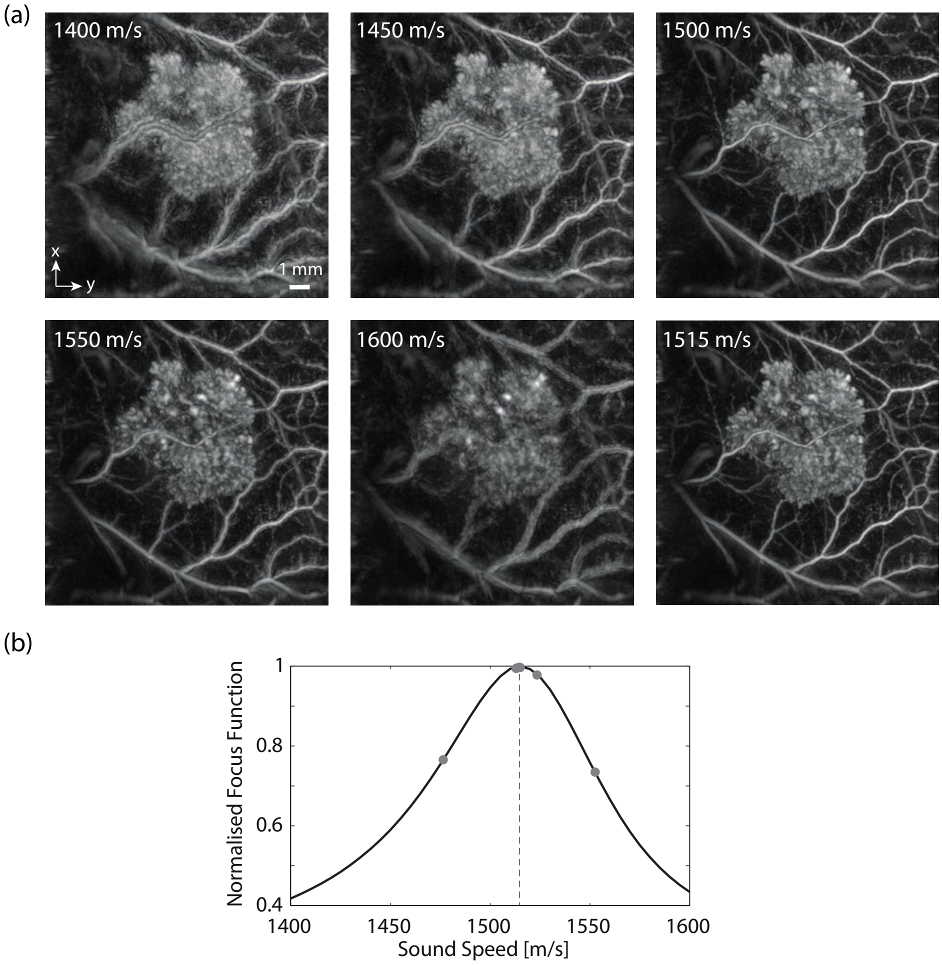


Figure 3. (a) Depth direction maximum intensity projections of the reconstructed image using different values for sound speed. A focusing and then defocusing can be noticed as the sound speed is increased. The reconstructed image using the sound speed that maximises the focus function (sharpness metric) is shown in the bottom right panel. (b) Variation of the focus function for sound speed values between 1400 and 1600 m/s. The focus function is unimodal, and the value of sound speed that maximises the focus function is shown with a dashed line. The steps used by `fminbnd` in MATLAB to find the maximum are shown with the grey dots.

m/s) and muscle (1580 m/s).[19] The reconstructed image using the optimised value for sound speed is shown in the bottom right panel of Fig. 3(a). The computational cost of computing the MIP and sharpness metric are negligible, thus the main cost of this approach is the repeated image reconstruction that must be performed. Using `fminbnd` in MATLAB, the optimum value for sound speed is found in 6 steps, shown as the grey dots in Fig. 3(b). Combined with an optimised C++ version of the FFT-based algorithm described in Sec. 3.5, the complete autofocusing procedure takes less than 5 seconds.

3.4. Upsampling

Due to practical constraints of using the Fabry-Perot scanning system (including animal scanning times and acquisition memory depth), the temporal sampling used for *in vivo* imaging studies is typically higher than the spatial sampling (the same is true of most photoacoustic and ultrasound scanners). For the sampling parameters used for the tumour dataset, the maximum supported frequencies due to the temporal and spatial sampling are

$$f_{\max,t} = \frac{1}{2\Delta t} = 25\ \text{MHz} \qquad > \qquad f_{\max,x} = \frac{c}{2\Delta x} = 7.575\ \text{MHz}\ . \tag{3}$$

As shown in Fig. 1(c), the acquired photoacoustic signals are very broadband, containing energy up to ∼20 MHz. This means the acquisition is spatially undersampled. Thus, if the image is reconstructed onto a grid defined by the *spatial* acquisition parameters, higher frequency information contained in the temporal signals will not be used, reducing resolution. To overcome this, the grid parameters used for the reconstruction can be spatially upsampled. This is demonstrated in Fig. 4, where the tumour image has been reconstructed using time reversal with upsampling factors of 1 (no upsampling), 2 and 3. This corresponds to a grid spacing (and maximum supported frequency) of 100 μm (7.575 MHz), 50 μm (15.15 MHz), and 33 μm (22.725 MHz), respectively. To allow a fair comparison, all three reconstructed images have been resampled to the same resolution for display using Fourier interpolation (`interpftn` in k-Wave). There is a very clear improvement with an upsampling factor of 2 compared to no upsampling. The small vessels are more visible, and there is much greater detail in the tumour mass. In comparison, there is little perceptible different between the reconstructed images with 2 and 3 times upsampling, despite the latter allowing almost the full range of frequencies contained in the temporal signals to be used in the reconstruction.

To examine this in more detail, the tumour dataset was reconstructed with an upsampling factor of 3 after first low-pass filtering the time signals. The reconstructed images for filter cutoff frequencies between 12 MHz and 4 MHz are shown in Fig. 5. For filter cutoff frequencies above 12 MHz, there was no discernible change in the image. At 12 MHz, the magnitude of the reconstructed image starts to decrease, and there is a slight reduction in high frequency variations at the voxel level. At 10 MHz, the main features are all still discernible, but begin to become noticeably softer. This trend continues down to 4 MHz, where the tumour and main vessels are still visible, but significantly blurred. Thus for this dataset, qualitatively it would appear that the frequency content up to ∼12 MHz has a perceptible impact on the reconstructed image. This explains why there is no noticeable difference seen between upsampling factors of 2 and 3 shown in Fig. 4.

The computational cost of using upsampling (particularly with time reversal image reconstruction) is that the image reconstruction must be performed using a larger computational grid. The grid size, compute time, and memory usage for the three reconstructions shown in Fig. 4 (using the 20,022 recorded time series) are given in Table 1. The reconstructions were performed using an optimised C++/CUDA version of k-Wave running on an NVIDIA GeForce GTX TITAN X graphics processing unit (GPU).[22] Even at the largest scale, the reconstruction takes less than 30 seconds and uses less than 4 GB of memory.

Table 1. Summary of grid size and compute time to reconstruct the tumour image using time reversal with different upsampling factors.

Upsampling Factor	Grid Size	Compute Time	Memory Usage
1	162 × 162 × 96	1.5 s	419 MB
2	324 × 324 × 144	8.1 s	1320 MB
3	450 × 450 × 216	25.3 s	3368 MB

3.5. Reconstruction methods

Two algorithms are routinely used for reconstructing the datasets acquired using the planar Fabry-Perot scanning system. The first is a fast one-step method based on an interpolation between spatial and temporal frequency performed in the Fourier domain as shown below (`kspacePlaneRecon` in k-Wave).[23, 24]

$$p(x,y,t) \xrightarrow{\text{FFT}} P(k_x,k_y,\omega) \xrightarrow{\frac{\omega}{c^2}=k_x^2+k_y^2+k_z^2} H(k_x,k_y,k_z) \xrightarrow{\text{IFFT}} h(x,y,z) \tag{4}$$

The second is time reversal, where the detected signals are propagated back into the domain in time reversed order using a numerical model of the acoustic forward problem (`kspaceFirstOrder3D` in k-Wave).[17, 25, 26]

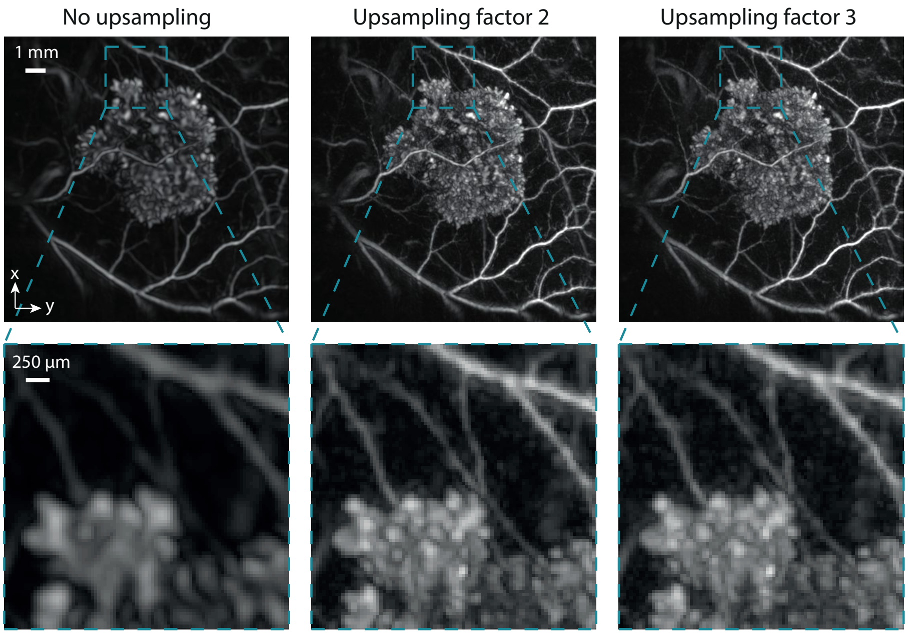


Figure 4. Effect of spatial upsampling on the resolution of reconstructed photoacoustic images. The maximum supported frequency for the three reconstructions is 7.575 MHz, 15.15 MHz, and 22.725 MHz, respectively.

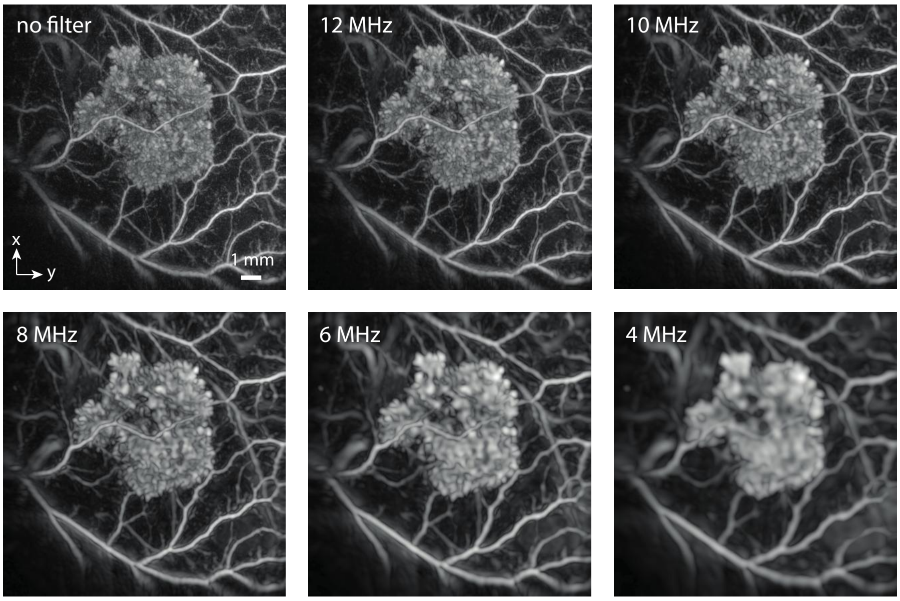


Figure 5. Reconstructed images using low-pass filtered data with filter cutoff frequencies from 12 MHz to 4 MHz.

A comparison of the images produced by these two methods is shown in Fig. 6. Because of the different assumptions inherent in the two algorithms, they produce visibly different images. First, the time reversal image has noticeably more detail, particularly in the tumour mass and smaller vasculature (e.g., the vessel denoted with the dashed arrows). This may be due to the way high-frequency information is mapped into the image in time reversal, and the inclusion of evanescent waves in the reconstruction.[10] Second, the FFT-based image contains wrapping artefacts due to the assumed spatial periodicity of the data. This is clearly visible when the reconstruction is repeated after spatially zero-padding the time domain signals (right panels). For example, the vessel denoted with the solid arrow belongs at the top of the image outside the field of view (bottom right panel), but is mapped to the bottom of the FFT-based image without zero-padding (bottom left panel). The main advantage of the FFT-based algorithm is its speed. Consequently, it is used for the sound speed optimisation step discussed in Sec. 3.3, and real-time display. Time reversal is generally used for all other purposes.

3.6. Image processing

After reconstruction, very basic image processing is performed. First, a positivity condition is usually applied where negative acoustic pressures in the reconstructed image are thresholded to zero.[27] Next, to improve the visibility of deeper lying vessels, a simple first-order correction for the variable light fluence in tissue is applied using a solution to the 1D diffusion equation

$$\Phi(z) = \Phi_0 \exp(-\mu_{\text{eff}} z) \ . \tag{5}$$

Here $\Phi(z)$ is the light fluence at depth z, and μ_{eff} is the effective attenuation coefficient, which can be in the range $50 - 250$ m^{-1} depending on the type of tissue. Note, this assumes the optical illumination is a planar collimated beam at the top surface of the tissue, the fluence is diffuse everywhere, and the optical properties are constant throughout the tissue.[7] These assumptions do not hold in general, thus this step does not quantitively correct for the spatial distribution of the fluence, but rather, qualitatively improves the visibility of deeper structures which in general will have received less light.

The effect of changing μ_{eff} on the lateral MIP of the tumour image is shown in Fig. 7. The right panels show 1D depth profiles summed across both lateral dimensions of the image, and give the total image intensity as a function of depth. If the image features were distributed evenly throughout the imaging volume, the peaks in the 1D profile would have approximately the same amplitude. However, because of optical attenuation, the image intensity rapidly decays with depth. As μ_{eff} is increased, the visibility of deeper structures is improved and the image intensity becomes more uniform. However, this comes at the expense of increasing the noise level, particularly at greater depths in the image.

In addition to correction for optical attenuation, the image data is often log compressed to reduce the dynamic range of the image before display (analogous to the log compression performed in ultrasound imaging). The log compression is performed according to

$$\bar{h}_{\text{compressed}} = \frac{\log_{10}\left(1 + 2^l \times \bar{h}\right)}{\log_{10}\left(1 + 2^l\right)} \ , \tag{6}$$

where $\bar{h}$ is the image data normalised between 0 and 1, and l is the compression level, which is typically set between 0 (low compression) and 4 (high compression). In k-Wave, this is applied using the function `logCompression`. The nonlinear mapping given by Eq. (6) is plotted in Fig. 8(a), and the log compressed images using l set to 1 and 4 are shown in Fig. 8(b). The compression makes it significantly easier to visualise the different structures in the image, particularly the small vasculature.

No other image processing (e.g., denoising) is routinely applied to the reconstructed images. In some cases, a manual segmentation and false colour might be used to highlight different regions of the image as shown in the left panel of Fig. 8(c). k-Wave also includes a vessel filtering function (`vesselFilter`), the output of which is shown in the middle panel of Fig. 8(c).[28] However, this is less useful in the case of the tumour image, which appears almost cartoon like. Finally, in some cases a colour map is used for depth direction (en face) maximum intensity projections to illustrate the depth at which the maximum value is extracted. An example is shown in

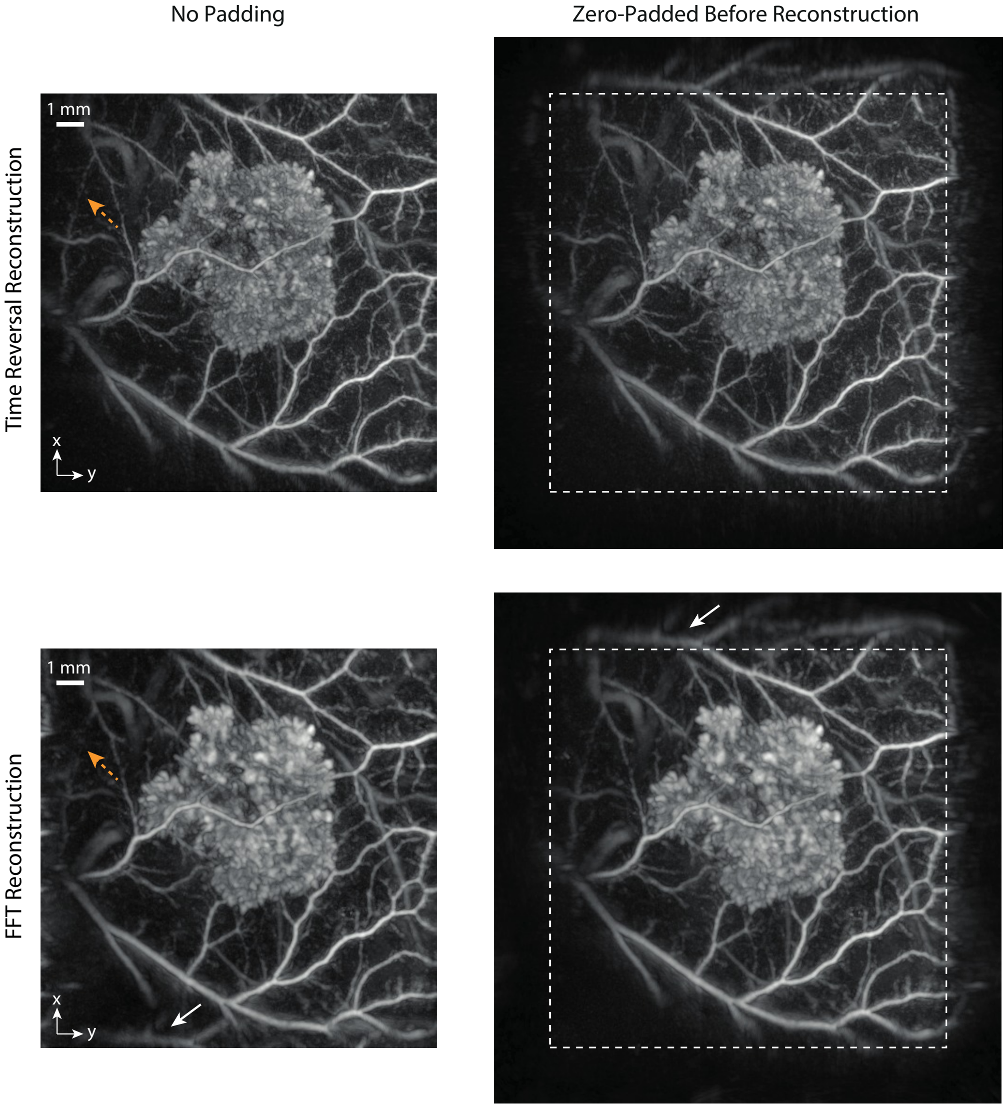

No Padding
Zero-Padded Before Reconstruction
Time Reversal Reconstruction
FFT Reconstruction
1 mm
1 mm
x
y
x
y

Figure 6. Comparison of images reconstructed using time reversal (top panels) and the FFT-based method (bottom panels) with and without zero padding the data. Time reversal gives a more detailed image (dashed arrows), and doesn't suffer from the wrapping artefacts present in the FFT-based reconstruction (solid arrows).

the right panel of Fig. 8(c). After processing, the 3D images are typically displayed as 2D maximum intensity projections. For depth-direction (en face) MIPs in particular, this can help reduce the visual perception of limited view artefacts.[4] Note, if image resampling is needed for high-resolution display, this is performed using Fourier interpolation (`interpftn` in k-Wave)

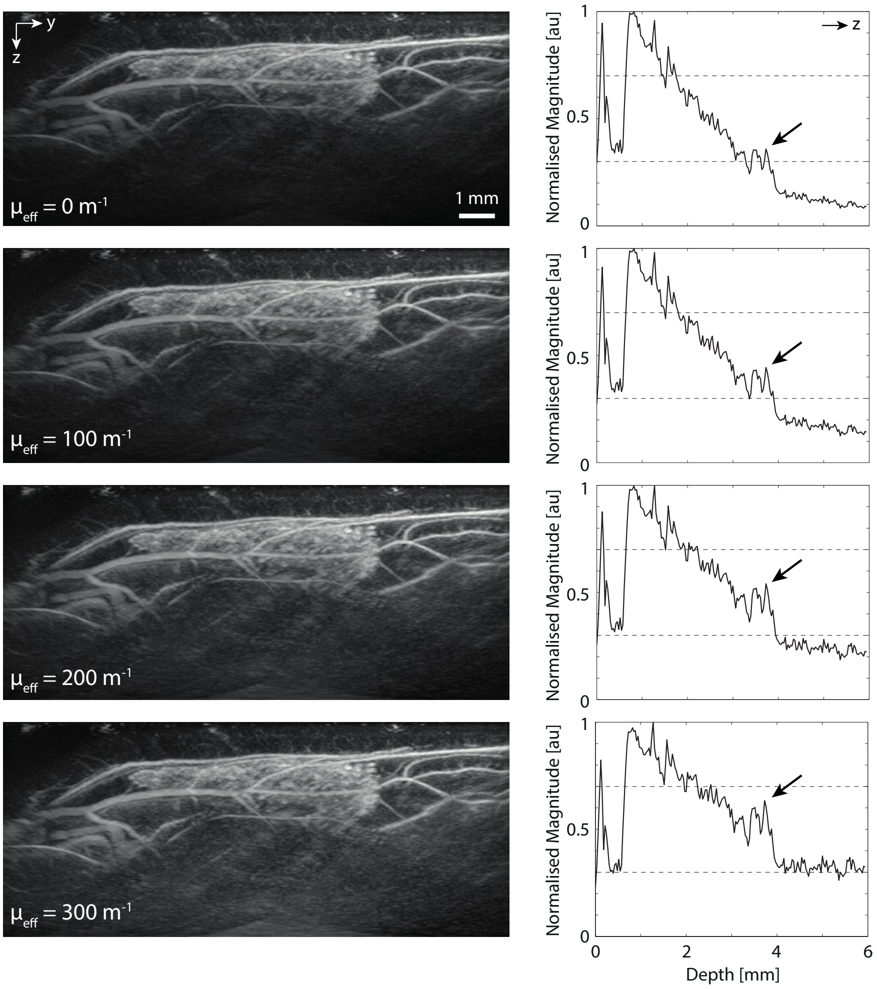


Figure 7. First order correction for optical attenuation using a solution to the 1D diffusion equation. The left panels show lateral maximum intensity projections through the reconstructed image using different values for the effective optical attenuation coefficient μ_{eff}. The right panels show 1D depth profiles summed across both lateral dimensions of the image. As μ_{eff} is increased, the visibility of deeper vessels is improved (arrows), at the expense of increasing the noise level.

4. SUMMARY

The image reconstruction and processing methods used in photoacoustic tomography can have a significant impact on the quality, resolution, and clinical value of photoacoustic images. Consequently, in addition to optimising the light delivery and ultrasound detection systems, careful thought should also be given to the image reconstruction process. Techniques such as automatic sound speed selection and acoustic attenuation compensation are fast and easy to apply, and noticeably improve the reconstructed images. Using the latest hardware and software advances, three-dimensional time reversal image reconstruction can also be performed on relatively large datasets in under 30 seconds. All of the algorithms discussed are available in the open source k-Wave toolbox, which makes it easy for others to apply them to their own datasets.

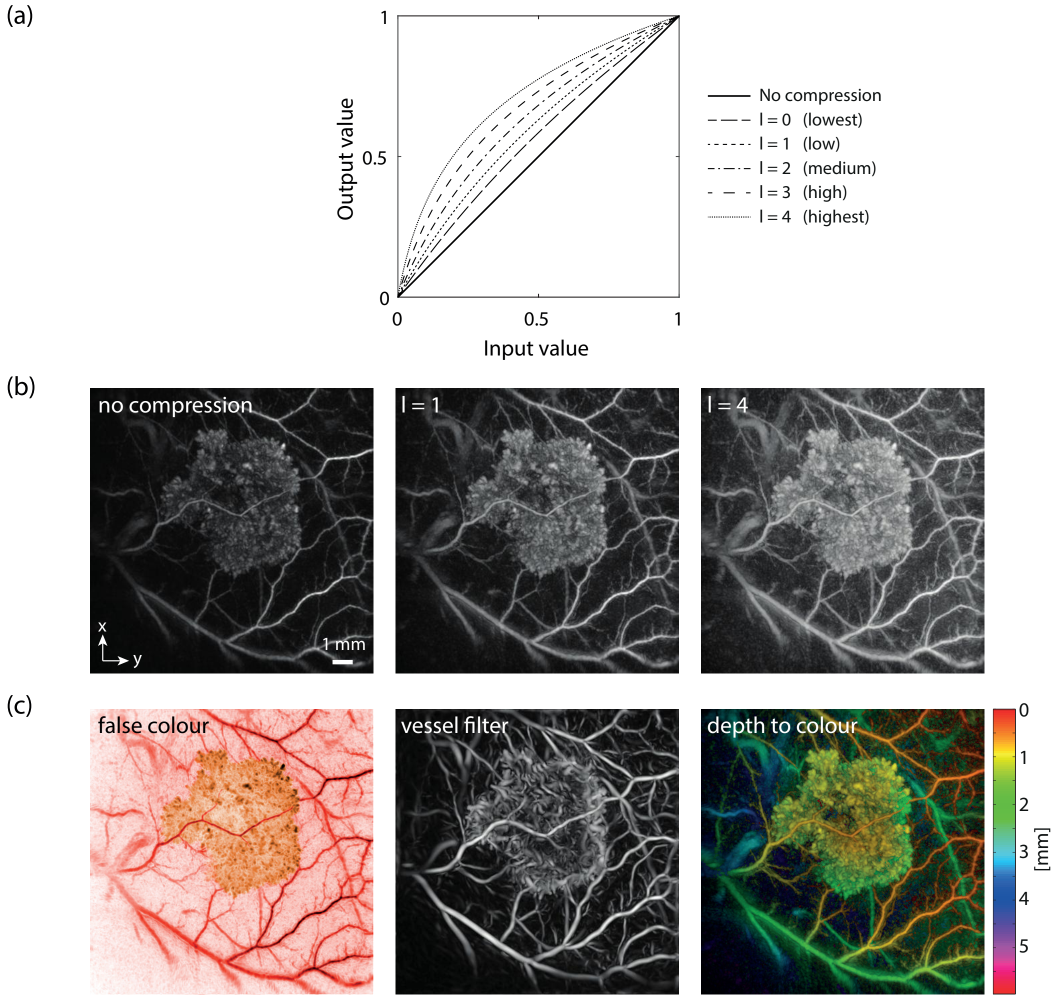


Figure 8. (a) Log compression curves for compression values from 0 to 4 calculated using Eq. (6). (b) Reconstructed photoacoustic images with no compression (left panel), log compression with $l = 1$ (middle panel), and log compression with $l = 4$ (right panel). (c) Other image processing techniques include false colour (left panel), vessel filtering (middle panel), and depth colour coded maximum intensity projections (right panel).

ACKNOWLEDGMENTS

The authors would like to thank Prof. Paul Beard, head of the Photoacoustic Imaging Group, for his contribution to the algorithms described in this paper, and valuable comments on the manuscript. This work was supported by the Engineering and Physical Sciences Research Council, UK (EP/L020262/1 and EP/M011119/1). The experimental data used in this study was provided by Amit Jathoul, Jan Laufer, Olumide Ogunlade, Bradley Treeby, Ben Cox, Edward Zhang, Peter Johnson, Arnold Pizzey, Brian Philip, Teresa Marafioti, Mark Lythgoe, R Barbara Pedley, Martin Pule, and Paul Beard. See Ref. 11 for details. Jiri Jaros is financed from the SoMoPro II Programme, co-financed by the European Union and the South-Moravian Region. This work reflects only the author's view and the European Union is not liable for any use that may be made of the information contained therein.

REFERENCES

1. P. Beard, "Biomedical photoacoustic imaging," *Interface Focus* **1**(4), pp. 602–631, 2011.
2. M. Lakshman and A. Needles, "Screening and quantification of the tumor microenvironment with micro-ultrasound and photoacoustic imaging," *Nat. Meth.* **12**(4), pp. iii–v, 2015.
3. P. Kuchment and L. Kunyansky, "Mathematics of Photoacoustic and Thermoacoustic Tomography," in *Handbook of Mathematical Methods in Imaging*, ch. 19, pp. 817–865, Springer, 2011.
4. Y. Xu, L. V. Wang, G. Ambartsoumian, and P. Kuchment, "Reconstructions in limited-view thermoacoustic tomography," *Med. Phys.* **31**(4), pp. 724–733, 2004.
5. B. T. Cox and B. E. Treeby, "Effect of sensor directionality on photoacoustic imaging: A study using the k-Wave toolbox," in *Proc. of SPIE*, **7564**, pp. 6–11, 2010.
6. N. A. Rejesh, H. Pullagurla, and M. Pramanik, "Deconvolution-based deblurring of reconstructed images in photoacoustic/thermoacoustic tomography," *J. Opt. Soc. Am. A* **30**(10), pp. 1994–2001, 2013.
7. B. Cox, J. G. Laufer, S. R. Arridge, and P. C. Beard, "Quantitative spectroscopic photoacoustic imaging: A review," *J. Biomed. Opt.* **17**(6), p. 061202, 2012.
8. C. Yoon, J. Kang, S. Han, Y. Yoo, T.-K. Song, and J. H. Chang, "Enhancement of photoacoustic image quality by sound speed correction: Ex vivo evaluation," *Opt. Express* **20**(3), pp. 3082–3090, 2012.
9. D. Van de Sompel, L. S. Sasportas, A. Dragulescu-Andrasi, S. Bohndiek, and S. S. Gambhir, "Improving image quality by accounting for changes in water temperature during a photoacoustic tomography scan," *PloS one* **7**(10), p. e45337, 2012.
10. B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *J. Biomed. Opt.* **15**(2), p. 021314, 2010.
11. A. P. Jathoul, J. Laufer, O. Ogunlade, B. Treeby, B. Cox, E. Zhang, P. Johnson, A. R. Pizzey, B. Philip, T. Marafioti, M. F. Lythgoe, R. B. Pedley, M. A. Pule, and P. Beard, "Deep in vivo photoacoustic imaging of mammalian tissues using a tyrosinase-based genetic reporter," *Nat. Photon.* **9**, pp. 239–246, 2015.
12. E. Zhang, J. Laufer, and P. Beard, "Backward-mode multiwavelength photoacoustic scanner using a planar Fabry-Perot polymer film ultrasound sensor for high-resolution three-dimensional imaging of biological tissues," *Appl. Optics* **47**(4), pp. 561–577, 2008.
13. J. Laufer, P. Johnson, E. Zhang, B. Treeby, B. Cox, B. Pedley, and P. Beard, "In vivo preclinical photoacoustic imaging of tumor vasculature development and therapy," *J. Biomed. Opt.* **17**(5), p. 056016, 2012.
14. J. Laufer, F. Norris, J. Cleary, E. Zhang, B. Treeby, B. Cox, P. Johnson, P. Scambler, M. Lythgoe, and P. Beard, "In vivo photoacoustic imaging of mouse embryos," *J. Biomed. Opt.* **17**(6), p. 061220, 2012.
15. S. P. Johnson, O. Ogunlade, E. Zhang, J. Laufer, V. Rajkumar, R. B. Pedley, and P. Beard, "Photoacoustic tomography of vascular therapy in a preclinical mouse model of colorectal carcinoma," in *Proc. of SPIE*, **8943**, p. 89431R, 2014.
16. P. Burgholzer, H. Grün, M. Haltmeier, R. Nuster, and G. Paltauf, "Compensation of acoustic attenuation for high resolution photoacoustic imaging with line detectors," in *Proc. of SPIE*, **6437**, p. 643724, 2007.
17. B. E. Treeby, E. Z. Zhang, and B. T. Cox, "Photoacoustic tomography in absorbing acoustic media using time reversal," *Inverse Probl.* **26**(11), p. 115003, 2010.
18. B. E. Treeby, "Acoustic attenuation compensation in photoacoustic tomography using time-variant filtering," *J. Biomed. Opt.* **18**(3), p. 036008, 2013.
19. T. L. Szabo, *Diagnostic Ultrasound Imaging*, Elsevier Academic Press, London, 2004.
20. B. E. Treeby, J. G. Laufer, E. Z. Zhang, F. C. Norris, M. F. Lythgoe, P. C. Beard, and B. T. Cox, "Acoustic attenuation compensation in photoacoustic tomography: Application to high-resolution 3D imaging of vascular networks in mice," in *Proc. of SPIE*, **7899**, p. 78992Y, 2011.
21. B. E. Treeby, T. K. Varslot, E. Z. Zhang, J. G. Laufer, and P. C. Beard, "Automatic sound speed selection in photoacoustic image reconstruction using an autofocus approach," *J. Biomed. Opt.* **16**(9), p. 090501, 2011.
22. B. E. Treeby, J. Jaros, A. P. Rendell, and B. T. Cox, "Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method," *J. Acoust. Soc. Am.* **131**(6), pp. 4324–4336, 2012.

23. S. J. Norton and M. Linzer, "Ultrasonic reflectivity imaging in three dimensions: Exact inverse scattering solutions for plane, cylindrical, and spherical apertures," *IEEE T. Biomed. Eng.* **28**(2), pp. 202–220, 1981.
24. K. P. Koestli, M. Frenz, H. Bebie, H. P. Weber, K. P. Köstli, M. Frenz, H. Bebie, and H. P. Weber, "Temporal backward projection of optoacoustic pressure transients using Fourier transform methods," *Phys. Med. Biol.* **46**(7), pp. 1863–1872, 2001.
25. D. Finch, S. K. Patch, and Rakesh, "Determining a function from its mean values over a family of spheres," *SIAM J. Math. Anal.* **35**(5), pp. 1213–1240, 2004.
26. Y. Xu and L. Wang, "Time reversal and its application to tomography with diffracting sources," *Phys. Rev. Lett.* **92**(3), pp. 3–6, 2004.
27. G. Paltauf, J. A. Viator, S. A. Prahl, and S. L. Jacques, "Iterative reconstruction algorithm for optoacoustic imaging," *J. Acoust. Soc. Am.* **112**(4), p. 1536, 2002.
28. T. Oruganti, J. Laufer, and B. E. Treeby, "Vessel filtering of photoacoustic images," in *Proc. of SPIE*, **8581**, p. 85811W, 2013.