

# k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields

Bradley E. Treeby

B. T. Cox

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

**Abstract.** A new, freely available third party MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields is described. The toolbox, named k-Wave, is designed to make realistic photoacoustic modeling simple and fast. The forward simulations are based on a k-space pseudo-spectral time domain solution to coupled first-order acoustic equations for homogeneous or heterogeneous media in one, two, and three dimensions. The simulation functions can additionally be used as a flexible time reversal image reconstruction algorithm for an arbitrarily shaped measurement surface. A one-step image reconstruction algorithm for a planar detector geometry based on the fast Fourier transform (FFT) is also included. The architecture and use of the toolbox are described, and several novel modeling examples are given. First, the use of data interpolation is shown to considerably improve time reversal reconstructions when the measurement surface has only a sparse array of detector points. Second, by comparison with one-step, FFT-based reconstruction, time reversal is shown to be sufficiently general that it can also be used for finite-sized planar measurement surfaces. Last, the optimization of computational speed is demonstrated through parallel execution using a graphics processing unit. © 2010 Society of Photo-Optical Instrumentation Engineers. [DOI: 10.1117/1.3360308]

Keywords: photoacoustics; ultrasound modeling; k-space; pseudo-spectral; image reconstruction; parallel processing; graphical processing unit (GPU).

Paper 09335SSR received Aug. 5, 2009; revised manuscript received Dec. 2, 2009; accepted for publication Dec. 28, 2009; published online Mar. 16, 2010.

## 1 Introduction

Photoacoustic tomography (PAT) is a noninvasive biomedical imaging modality that allows the *in vivo* visualization of embedded light absorbing structures.<sup>1</sup> The technique works by externally illuminating a tissue sample with short pulses of visible or near-infrared (NIR) laser light. The localized absorption of this light (particularly by the hemoglobin chromophores present in blood) produces broadband ultrasonic waves via thermoelastic expansion. By measuring the ultrasonic waves that propagate back to the tissue surface, images of the initial photoacoustic pressure (which is related to the absorbed optical energy distribution) can then be reconstructed. These images may be used to quantify tissue properties,<sup>2,3</sup> or to identify pathological structures.<sup>4</sup> The technique has been demonstrated via high-resolution *in vivo* imaging of vasculature in both small animals<sup>5,6</sup> and humans.<sup>7</sup> Similar images may also be formed using microwave frequencies (an analogous technique often called thermoacoustic tomography), where water is the primary absorber.<sup>8</sup>

The continued development of PAT (for quantitative imaging, for example) is in part contingent on a detailed understanding of the parameters that affect the reconstructed photoacoustic image, including the optical, thermal, and acoustic

properties of the tissue; the arrangement and characteristics of the excitation laser source; the arrangement and characteristics of the ultrasound sensors; and the assumptions and limitations of the numerical reconstruction algorithm. To this end, the simulation of PAT can provide both qualitative and quantitative insight into the contribution of these parameters, including the effect of their perturbation and the optimization of their values. Conceptually, the PAT modeling problem can be divided into two components: optical and acoustic. Considering the acoustic element, simulation models are also integral for the generation of numerical phantom data,<sup>9</sup> for the development of image reconstruction algorithms,<sup>10,11</sup> and for use within iterative reconstruction routines.<sup>12</sup> It is the development of fast, accurate, easy-to-use, and tissue-realistic methods for modeling photoacoustic wave fields (i.e., the acoustic component of the PAT modeling problem) that is the subject of interest here.

The principle requirement for the development of acoustic models for PAT is simply that the underlying assumptions of the governing equations are also satisfied in the acoustic environments relevant to the modality. Within soft tissue, the sound speed and density are related to the relative proportions of water, proteins (such as collagen and hemoglobin), and lipids, and are thus inherently heterogeneous.<sup>13</sup> A high protein content causes the sound speed and density to be higher than

Address all correspondence to: Bradley E. Treeby, University College London, Department of Medical Physics and Bioengineering, Gower Street, London, WC1E 6BT, United Kingdom; E-mail: btreeby@mpb.ucl.ac.uk

those in water (e.g., blood), while a high lipid content causes them to be lower (e.g., fat).<sup>14</sup> For layered media such as human skin (where the collagen-rich dermis overlies a layer of subcutaneous fat), this necessitates heterogeneous distributions of sound speed, density, and acoustic absorption. However, there is a distinct disparity between these requirements and current simulation and reconstruction techniques used for PAT. Most approaches assume a homogeneous distribution of acoustic properties and a nonabsorbing medium. Moreover, simulation models based on finite-difference (FD) time domain solutions to the wave equation that do facilitate these inclusions become computationally formidable for modestly large three-dimensional (3-D) domains.

Here, the architecture and use of a new third party toolbox for MATLAB (MathWorks, Inc., Natick, Massachusetts) developed for the simulation and reconstruction of photoacoustic wave-fields is described. The toolbox, named k-Wave, is freely available at <http://www.k-wave.org>, and includes:

- An easy-to-use time domain forward model of acoustic wave propagation for acoustically heterogeneous media with power law absorption.
- The ability to model arbitrary detection surfaces with directional elements.
- The option to use the forward model as a flexible time reversal image reconstruction algorithm for an arbitrary measurement surface.
- A fast one-step image reconstruction algorithm for data recorded on a linear (2-D) or planar (3-D) measurement surface.
- Optional input parameters to adjust visualization and performance, including options to make a wave propagation movie for use in presentations and to run the simulations on the graphics processing unit (GPU).
- Many simple-to-follow tutorial examples to illustrate the capabilities of the toolbox.

The governing photoacoustic equations and their application to photoacoustic simulation and reconstruction are discussed in Sec. 2. The k-space pseudo-spectral (PS) solution method and the use of a perfectly matched layer are also described. In Sec. 3, the k-Wave toolbox is introduced and many of the included functions are illustrated and discussed. In Sec. 4, several novel simulation and reconstruction examples are presented, including the use of data interpolation for sparse detector arrays; a comparison of time reversal and one-step, fast Fourier transform (FFT)-based image reconstruction algorithms; and performance enhancements via data casting and parallelization using the GPU. A summary and a discussion of future work are given in Sec. 5.

## 2 Modeling Photoacoustic Wave Propagation

### 2.1 Photoacoustic Wave Equation

In PAT, a spatially dependent ultrasound signal is generated by illuminating a turbid medium with short pulses of visible or infrared laser light. Within soft biological tissue at these optical wavelengths, embedded chromophores such as melanin and hemoglobin preferentially absorb the light and undergo thermoelastic expansion, producing both thermal and acoustic waves. The resulting heat diffusion occurs on a time scale much longer than the acoustic propagation time (hundreds of milliseconds compared to microseconds), which in

turn is much longer than the time scale for the heating laser pulse (typically, on the order of nanoseconds). For the acoustic propagation, heat conduction can thus be neglected and the governing acoustic equations reformulated as an initial value problem.<sup>15</sup>

For a given light fluence, the initial photoacoustic pressure distribution is related to the spatially dependent optical, thermal, and acoustic properties of the medium.<sup>16</sup> For PAT *in vivo*, this initial pressure is typically on the order of 10 kPa. Accordingly, the time evolution of photoacoustic wave fields can be modeled using the equations of linear acoustics. For soft biological tissue, it can also generally be assumed that the propagation medium is isotropic and quiescent, that the pressure flow is irrotational, and that shear waves can be neglected. In a lossless medium, the appropriate equation of motion, equation of continuity, and equation of state can then be written as<sup>17</sup>

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

$$\frac{\partial p}{\partial t} = -\rho_0 \nabla \cdot \mathbf{u},$$

$$p = c^2 \rho, \quad (1)$$

where the initial conditions are given by

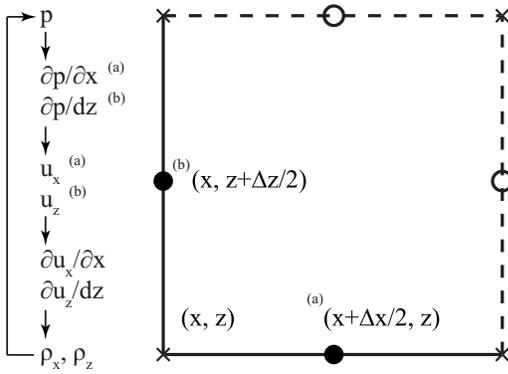
$$p_0 = \Gamma \mu_a \Phi,$$

$$\frac{\partial p_0}{\partial t} = 0. \quad (2)$$

Here,  $\mathbf{u}$  is the acoustic particle velocity,  $\rho_0$  is the ambient density,  $\rho$  is the acoustic density,  $c$  is the thermodynamic sound speed,  $p$  is the acoustic pressure,  $p_0 = p(t=0)$  is the initial photoacoustic pressure distribution,  $\Gamma$  is the Grüneisen parameter (the proportionality constant between the absorbed light and the initial pressure),  $\mu_a$  is the optical absorption coefficient, and  $\Phi$  is the light fluence, all of which may be spatially varying. These equations are often combined to give a single second-order photoacoustic wave equation.<sup>15</sup> Using this framework, it is straightforward to modify the adiabatic equation of state to account for acoustic absorption<sup>18</sup> or nonlinear effects.<sup>19</sup>

### 2.2 Pseudo-Spectral and k-Space Methods

The most commonly used numerical methods for solving partial differential equations in acoustics are the finite-difference, finite-element, and boundary-element methods. Although excellent for many applications, for time domain modeling of broadband or high-frequency waves, they can become cumbersome and slow. This is due to the requirements for many grid points per wavelength and small time-steps to minimize unwanted numerical dispersion. The PS method (which represents an extension of the FD method) can help reduce the first of these problems, and the k-space approach can help to overcome the second.



**Fig. 1** Schematic showing the computational steps in the solution of the coupled first-order acoustic equations for heterogeneous media using a staggered spatial grid. The superscripts (a) and (b) denote the parameters calculated on the x and z staggered grids, respectively.

In a simple FD scheme, the gradient of the field is estimated using linear interpolation between its values at the grid points (i.e., the mesh nodes). A better estimate of the gradient can be obtained by fitting a higher-order polynomial to a greater number of nodes and calculating the derivative of the polynomial. The more points used, the higher the degree of polynomial required, and the more accurate the estimate of the derivative. The PS method takes this idea further and fits a Fourier series to all of the data; it is therefore sometimes referred to as a global, rather than local, method. There are two significant advantages to using Fourier series: first, the amplitudes of the Fourier components can be calculated efficiently using the FFT, and second, the basis functions are sinusoidal, so only two nodes per wavelength are required, rather than the six to ten required in other methods.<sup>20,21</sup>

While the PS method improves efficiency in the spatial domain, conventional FD schemes are still necessary to calculate the gradients (rates of change) in the time domain. The FD approximation introduces instability into the numerical simulation that can only be controlled by limiting the size of the time-step. The techniques broadly classed as k-space methods attempt to relax this limitation in order to allow larger time-steps to be used without compromising accuracy. By comparing a simple PS time domain model for acoustically homogeneous media to an exact solution to the corresponding homogeneous wave equation, it is possible to find replacement expressions for either the temporal or the spatial derivative such that the numerical solutions are exact for arbitrarily large time-steps.<sup>22-25</sup> In effect, this substitution incorporates *a priori* information about the form of the derivative specific to the governing wave equation. These k-space adjustments also lead to improved numerical stability in the case of acoustically heterogeneous media. For the range of heterogeneity evident in soft biological tissue, this allows the use of much larger time steps for the same degree of accuracy. A detailed error analysis of the implemented k-space technique is provided by Tabei et al.,<sup>23</sup> and an experimental validation of k-space methods for photoacoustics is given by Cox et al.<sup>26</sup>

### 2.3 Perfectly Matched Layer

The simulation of propagating wave fields using a finite-sized computational grid requires an efficient numerical scheme to

compute the derivatives near the grid boundaries. For PS or k-space methods, the computation of the spatial derivatives via the FFT causes waves leaving one side of the domain to reappear at the opposite side. This wave wrapping can be avoided by implementing an absorbing boundary condition known as a perfectly matched layer (PML).<sup>27,28</sup> This is a thin absorbing layer that encloses the computational domain and is governed by a nonphysical set of equations, causing anisotropic attenuation. The use of a PML requires the propagating density or pressure to be artificially divided into Cartesian components, i.e.,  $\rho = \rho_x + \rho_y + \rho_z$ . The absorption is then defined such that only components of the wave field traveling within the PML and normal to the boundary are absorbed. Including a PML, the first-order acoustic equations given in Eq. (1) become<sup>23,24</sup>

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho_0} \nabla p - \boldsymbol{\alpha} \cdot \mathbf{u},$$

$$\frac{\partial \rho_x}{\partial t} = -\rho_0 \frac{\partial u_x}{\partial x} - \alpha_x \rho_x,$$

$$p = c_0^2 \sum \rho_{x,y,z}, \tag{3}$$

where the second equation is repeated for each Cartesian direction. Here  $\boldsymbol{\alpha} = \{\alpha_x, \alpha_y, \alpha_z\}$  is the anisotropic absorption in Nepers per meter, which is only nonzero within the PML. The performance of the PML is dependent on both the size and attenuation of the layer, as well as the time-step used in the simulation.<sup>29</sup> When the PML is implemented effectively, it is possible to simulate infinite domain propagation in k-space using only a small computational grid.<sup>23,24</sup>

### 2.4 Staggered Grids

The numerical solution of Eq. (3) is computed in several steps (see Fig. 1). First, the pressure distribution within the computational domain is used to calculate the spatial derivatives  $\partial p / \partial x$ ,  $\partial p / \partial y$ , and  $\partial p / \partial z$ . These are used to update the corresponding velocity terms using a first-order FD. (Computationally, the k-space adjustments for the FD calculation of the temporal derivative are actually made using a modified Laplacian operator, i.e., within the computation of the *spatial* derivatives.<sup>24</sup>) Next, the spatial derivatives of the velocity for each Cartesian direction are computed. These are used to update the values of the acoustic density within the domain, again using a first-order FD. Last, the pressure is computed using the appropriate equation of state.

For both FD and k-space (or PS) methods, additional accuracy, and therefore stability, can be obtained when computing odd-order derivatives by using staggered spatial and temporal grids.<sup>23,30</sup> In this case, the positions where the governing equations are discretized need not coincide with the positions where the function values are available.<sup>31</sup> Figure 1 illustrates the use of staggered grids in 2-D for a computation using the coupled acoustic equations given in Eq. (3). The grid staggering means that a local change in  $p$  will immediately affect the adjacent particle velocities. (This is not the case for nonstaggered grids.<sup>23</sup>) The calculations for the particle velocity and its derivatives are also staggered temporally, which minimizes

errors when using low-order FD methods to compute the temporal derivative. A more detailed description of the computational methodology implemented in k-Wave can be found in Refs. 23 and 24.

### 2.5 Time Reversal Image Reconstruction

The computational challenge in PAT is to reconstruct an estimate of the initial photoacoustic pressure distribution  $p_0$  given a set of time varying measurements of the acoustic pressure  $p_S$  recorded over an arbitrary surface  $S$  for some time  $t=0$  to  $T$ . In time reversal image reconstruction, this estimate is obtained by using the recorded measurements of  $p_S$  in time reversed order as a time varying Dirichlet boundary condition imposed at the position of the detectors on the measurement surface.<sup>11,32–34</sup> The time evolution of the wave field propagating into the domain from the imposed boundary condition is calculated using a forward propagation model with zero initial conditions. The reconstruction is then given as the acoustic pressure within the domain after time  $T$ . As the time reversal reconstruction is dependent on a forward propagation model, it is straightforward to include heterogeneities into the reconstruction simply by using the appropriate model. This choice will similarly dictate the speed and accuracy of the time reversal reconstruction.

### 2.6 One-Step Image Reconstruction for a Planar Measurement Surface

For an acoustically homogeneous medium, if the measurement surface is planar, a much faster reconstruction algorithm is available that calculates the initial pressure distribution in a single step (i.e., without the need for time iterations).<sup>35,36</sup> The algorithm works by mapping the time domain information of the measured data (recorded as a function of time and 2-D position on the plane) into a third spatial dimension. This mapping is performed by relating the temporal and spatial frequency information in the depth direction via a dispersion relation.<sup>37</sup> If the pressure  $p_S(x, y, t)$  recorded over a planar measurement surface  $S$  is forced to be symmetrical about  $t=0$ , the reconstruction can be computed efficiently simply using interpolation and the Fourier transform:

$$p_0(k_x, k_y, \omega) = \frac{c^2 k_z}{2\omega} \mathcal{F}_{x,y,t} \{p_S(x, y, t)\},$$

$$p_0(k_x, k_y, \omega) \rightarrow p_0(k_x, k_y, k_z),$$

$$p_0(x, y, z) = \mathcal{F}_{x,y,z}^{-1} \{p_0(k_x, k_y, k_z)\}, \quad (4)$$

Here,  $k_x$ ,  $k_y$ , and  $k_z$  are the spatial wave number components in each Cartesian direction,  $\omega$  is the temporal frequency,  $\rightarrow$  represents the interpolation step, and  $\mathcal{F}$  and  $\mathcal{F}^{-1}$  represent the forward and inverse Fourier transforms, respectively. It is assumed that the evanescent region of wave numbers is excluded in the Fourier transform over the  $t$ -dimension.<sup>15</sup> In practice, this is achieved by setting the values of  $p_0(k_x, k_y, \omega)$  to zero wherever  $\omega^2/c^2 < k_x^2 + k_y^2$ . The interpolation between the temporal and spatial frequencies  $\omega$  and  $k_z$  is then computed using the dispersion relation  $k_z^2 = (\omega/c)^2 - k_x^2 - k_y^2$ . The

method used for this interpolation can affect both the accuracy and the speed of the reconstruction.<sup>38</sup>

Note that, numerically, an additional scaling factor of  $4/c$  must also be applied to the final reconstructed amplitude. This accounts for the difference in spacing between the forward and inverse FFT for the interpolated coordinate ( $dt$  versus  $dz$ ), the inherent assumption that  $p_0$  is symmetrical about  $z$ , and the fact that the planar measurement surface necessarily lies on only one side of the measurement domain and therefore does not detect waves propagating in the opposite direction. The use of the FFT to compute the Fourier transformation steps of Eq. (4) means that the reconstruction will be fastest when both the number of time samples and the number of detector points are powers of 2.

## 3 The k-Wave Toolbox

### 3.1 Overview of Functions

The k-Wave toolbox is designed to make photoacoustic modeling easy and fast. The functions included within the toolbox can be divided into four broad categories:

- The simulation of photoacoustic (or ultrasonic) wave fields.
- The reconstruction of photoacoustic images.
- The creation of geometric shapes.
- Utility and system functions.

The simulation functions compute the time evolution of an acoustic wave field within homogeneous or heterogeneous media in either 1-, 2-, or 3-D. The computations are based on a k-space solution to coupled acoustic equations as discussed in Sec. 2. These functions can also be used for time reversal image reconstruction. The additional image reconstruction functions allow the initial photoacoustic pressure to be estimated from data recorded over a linear (2-D) or planar (3-D) measurement surface. The geometry creation functions allow both Cartesian- and grid-based geometries to be defined, including circles, arcs, disks, spheres, shells, and balls. The Cartesian-based functions return the geometric coordinates of the particular shape, while the grid-based functions return a binary matrix (i.e., matrix of 1s and 0s) where the 1s correspond to the location of shape. The utility functions are used to perform additional tasks such as grid creation, matrix smoothing, matrix interpolation, file loading, etc. Examples of using many of the functions within the toolbox are given in the following sections.

### 3.2 Time-Domain Simulation of Photoacoustic Wave Fields

Figure 2 illustrates the computational architecture of the simulation functions based on the coupled first-order acoustic equations given in Eq. (3) (`kspaceFirstOrder1D`, `kspaceFirstOrder2D`, and `kspaceFirstOrder3D`). The functions are given information about the discretization of the propagation medium, its acoustic properties, the initial (or time varying) pressure distribution, and the location and characteristics of the measurement surface that detects the ultrasonic wave field. These properties are assigned as fields within four input structures; `grid`, `medium`, `source`, and `sensor` (see Table 1). The propagation of the wave field through the medium is then computed step by step, with the pressure values at the sensor elements stored after each itera-

tion. These values are returned when the time loop has completed. A simple example of a 2-D simulation in a heterogeneous layered medium is given here (1-D and 3-D simulations are performed in an analogous fashion):

```
% create the computational grid
Nx = 256;
Nz = 128;
dx = 50e-6;
dz = 50e-6;
kgrid = makeGrid(Nx, dx, Nz, dz);

% define the medium properties
medium.sound_speed = 1500*ones(Nz, Nx);
medium.sound_speed(1:50,:) = 1600;
medium.density = 1000*ones(Nz, Nx);
medium.density(1:50,:) = 1040;

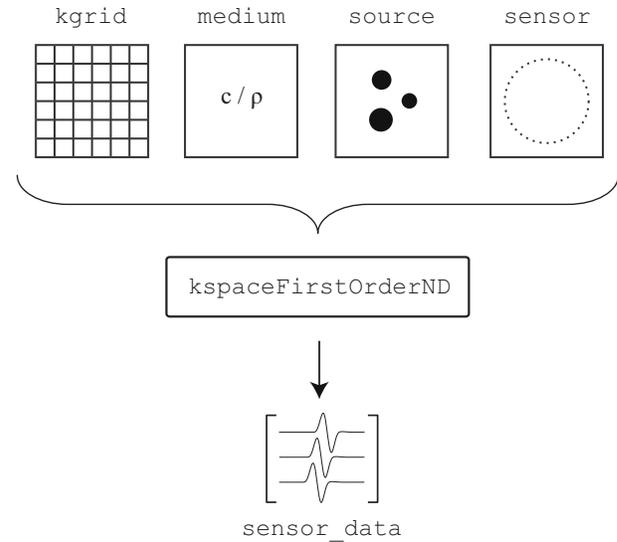
% define the initial pressure
disc_x_pos = 120;
disc_z_pos = 75;
disc_radius = 8;
disc_mag = 3;
source.p0 = disc_mag*makeDisc(Nx, Nz,
    disc_x_pos, disc_z_pos, disc_radius);

% define a centered circular sensor
sensor_radius = 2.5e-3;
num_sensor_points = 50;
sensor.mask = makeCartCircle(sensor_ra
    dius, num_sensor_points);

% run the simulation
sensor_data = kspaceFirstOrder2D(kgrid,
    medium, source, sensor);
```

The medium discretization is performed using the utility function `makeGrid`. The size ( $dx$ ,  $dz$ ) and number ( $N_x$ ,  $N_z$ ) of pixels in each Cartesian direction are used to calculate the Cartesian and k-space discretizations, and a k-Wave grid structure `kgrid` encapsulating this information is returned. The discretizations are calculated to satisfy the requirements of the FFT-based spatial derivatives. This structure is used extensively by both the simulation and utility functions within k-Wave. The time-steps used in the simulation are defined by `kgrid.t_array`, which is set to 'auto' by `makeGrid`. In this case, the time array is automatically calculated within the simulation functions using the utility function `makeTime` based on the size and properties of the k-space grid and sensible stability criterion [a Courant-Friedrichs-Lewy stability value of 0.3 (Ref. 23)].

For a homogeneous medium, `medium.sound_speed` and `medium.density` are given simply as scalar values. For a heterogeneous medium, these are given as  $N_z \times N_x$  matrices with arbitrary numeric values. The initial photoacoustic pressure distribution `source.p0` is similarly defined as an  $N_z \times N_x$  matrix. The measurement surface `sensor.mask` is given either as a binary grid (i.e., an  $N_z \times N_x$  matrix of 1s and 0s) representing the pixels within the computational grid that will collect the data, or as a  $2 \times N$  matrix of Cartesian coordinates where the pressure values are calculated at each time-



**Fig. 2** Schematic of the architecture of the simulation functions within the k-Wave toolbox that are based on coupled first-order acoustic equations for heterogeneous media.

step using interpolation. Note, in 3-D, the input matrices are instead  $N_z \times N_x \times N_y$  in size, and the Cartesian sensor points are given by a  $3 \times N$  matrix. For the example given here, the geometric function `makeDisc` is used to define an initial pressure distribution of a small filled disk, while `makeCartCircle` is used to define a Cartesian sensor mask with a set of evenly spaced points on a circle.

The simulation is invoked by calling `kspaceFirstOrder2D` with the inputs described earlier. By default, a visualization of the propagating wave field and a status bar are displayed, with frame updates every 10 time-steps. The default k-Wave color map displays positive pressures as yellows through reds to black, zero pressures as white, and negative pressures as light to dark blue-greys. A screen shot of the k-Wave simulation example coded earlier is shown in Fig. 3. The circular sensor mask, the absorption within the lower PML, and a small reflection from the layered sound speed and density interface are all clearly visible. As the function runs, status updates and computational parameters are printed to the command line. When the time loop has completed, the function returns the time series recorded at the detector locations defined by the sensor mask (see Fig. 4). If the sensor mask is given as a set of Cartesian coordinates, the computed `sensor_data` is returned in the same order. If the sensor mask is given as a binary grid, `sensor_data` is returned using MATLAB's standard column-wise linear indexing. In both cases, the recorded data is indexed as `sensor_data(sensor position, time)`. For a binary sensor mask, the pressure values at a particular time can be restored to the sensor positions within the computational grid using the utility function `unmaskSensorData`.

Additional properties of the medium, source, and sensor can be assigned using the remaining structure fields (see Table 1). For example, arbitrary power law absorption can be assigned using `medium.alpha_power` and `medium.alpha_coeff`, sensor directivity can be specified using `sensor.directivity_angle` and

**Table 1** Summary of the input structure fields for the first-order k-Wave simulation functions. The fields that are required for a photoacoustic forward simulation are marked with an asterisk (\*). By default, `makeGrid` sets `kgrid.t_array` to 'auto'.

Field	Description
<code>kgrid.k</code> , <code>kgrid.Nx</code> , <code>kgrid.dx</code> , etc.*	Cartesian and k-space grid fields returned by <code>makeGrid</code>
<code>kgrid.t_array</code> *	Simulation time array
<code>medium.sound_speed</code> *	Sound speed distribution
<code>medium.density</code> *	Ambient density distribution
<code>medium.alpha_power</code>	Power law absorption exponent
<code>medium.alpha_coeff</code>	Power law absorption coefficient
<code>source.p0</code> *	Initial pressure distribution
<code>source.p</code>	Time varying pressure source distribution
<code>source.p_mask</code>	Binary grid specifying the positions of the time varying pressure source distribution
<code>sensor.mask</code> *	Binary grid or set of Cartesian points where the pressure is recorded
<code>sensor.time_reversal_boundary_data</code>	Time varying pressure enforced as a Dirichlet boundary condition over <code>sensor.mask</code>
<code>sensor.time_reversal_adapt_thresh</code>	Adaptive boundary condition threshold
<code>sensor.directivity_angle</code>	Direction of maximum response
<code>sensor.directivity_size</code>	Equivalent element size

`sensor.directivity_size`, and a time varying source can be used by defining `source.p` and `source.p_mask`.

### 3.3 Optional Input Parameters

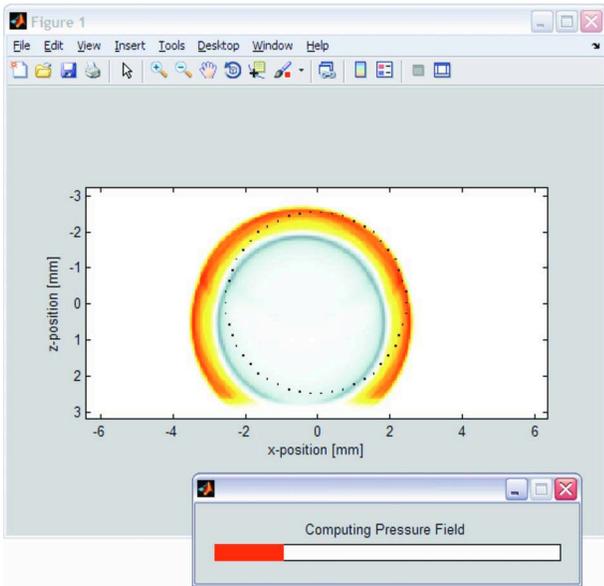
The behavior of the simulation functions within the k-Wave toolbox can be controlled through the use of optional input parameters. These are given as `param, value` pairs following the primary function inputs. For example, the visualization can be automatically saved as a movie by setting 'RecordMovie' to true. Similarly, a plot of the initial pressure distribution, sensor mask, and medium properties can be automatically generated by setting 'PlotLayout' to true; the properties of the PML can be controlled using 'PMLSize', 'PMLAlpha', and 'PMLInside'; the interpolation method used to calculate the pressure values on a Cartesian sensor mask can be set using 'CartInterp'; and the smoothing of input matrices can be controlled via 'Smooth'. Detailed descriptions of the functions and their usage are given in the html help files and examples included within the toolbox.

### 3.4 Time Reversal Image Reconstruction

The first-order k-Wave functions already described for the simulation of photoacoustic wave propagation can also be used for photoacoustic image reconstruction by assigning the time varying pressure recorded over the detector array to

`sensor.time_reversal_boundary_data`. This pressure is then enforced, in time reversed order, as a Dirichlet boundary condition over the given sensor mask. If the sensor mask is given as a set of Cartesian coordinates, then the sensor data, indexed as `sensor_data(sensor position, time)`, must be given in the same order. An equivalent grid-based sensor mask computed using nearest-neighbor interpolation is then used to enforce the boundary condition within the computational grid at each time-step. If the sensor mask is instead given as a binary grid, the sensor data must be ordered using MATLAB's standard column-wise linear matrix indexing.

An example of using k-Wave to compute a 2-D time reversal image reconstruction is given below. By passing the sensor data returned from a k-space forward simulation directly to `sensor.time_reversal_boundary_data` and then calling `kspaceFirstOrder2D`, it is straightforward to simulate the measurement and reconstruction process. (Note, in this simple example, the "inverse crime" is committed in which the same numerical parameters are used for both simulation and reconstruction.) When using the simulation functions in time reversal mode, the array of time points `kgrid.t_array` must be explicitly defined. This array is created here using the utility function `makeTime` (the same function that is called internally by the first-order simulation codes when `kgrid.t_array` is set to 'auto'):



**Fig. 3** Screen shot of a 2-D forward simulation in a heterogeneous layered medium using the k-Wave toolbox. The circular sensor mask utilized is shown as a series of small black pixels, and the progress of the simulation is illustrated by the status bar. The anisotropic absorption within the perfectly matched layer (PML) on the lower side of domain is clearly visible. By default, the visualization is updated every 10 time-steps.

```
% create the time array
kgrid.t_array = makeTime(kgrid,
    medium.sound_speed);

% run the forward simulation
sensor_data = kspaceFirstOrder2D(kgrid,
    medium, source, sensor);

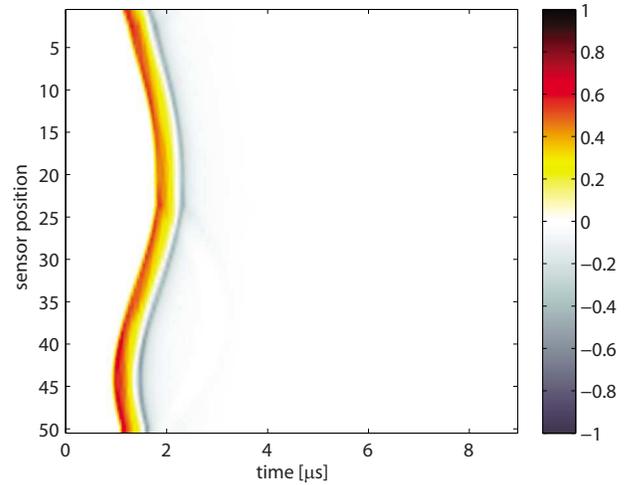
% reset the initial pressure
source.p0 = 0;

% assign the time reversal data
sensor.time_reversal_boundary_data =
    sensor_data;

% run the time reversal reconstruction
p0_recon = kspaceFirstOrder2D(kgrid,
    medium, source, sensor);
```

### 3.5 One-Step Image Reconstruction

If the measured data is recorded using a linear (2-D) or planar (3-D) detector array, a fast one-step, FFT-based image reconstruction can be performed using the functions `kspaceLineRecon` and `kspacePlaneRecon`. Here, the time series data input must be indexed as `p_txy(time, sensor x position, sensor y position)` in 3-D or `p_tx(time, sensor position)` in 2-D, where the sensor spacing is given by `dx` and `dy`, the temporal spacing is given by `dt`, and the sound speed in the propagation medium (which is assumed to be acoustically homogeneous) is given by `c`. The reconstruction is then invoked by calling the function with the parameters described earlier. For example, in 3-D, the reconstruction is performed by calling:



**Fig. 4** The time series recorded over a circular Cartesian sensor mask for a 2-D forward simulation in a heterogeneous layered medium using the k-Wave toolbox. If the sensor mask is given as a set of Cartesian coordinates, the computed time series is returned in the same order.

```
p_zxy = kspacePlaneRecon(p_txy, dx, dy, dt, c);
```

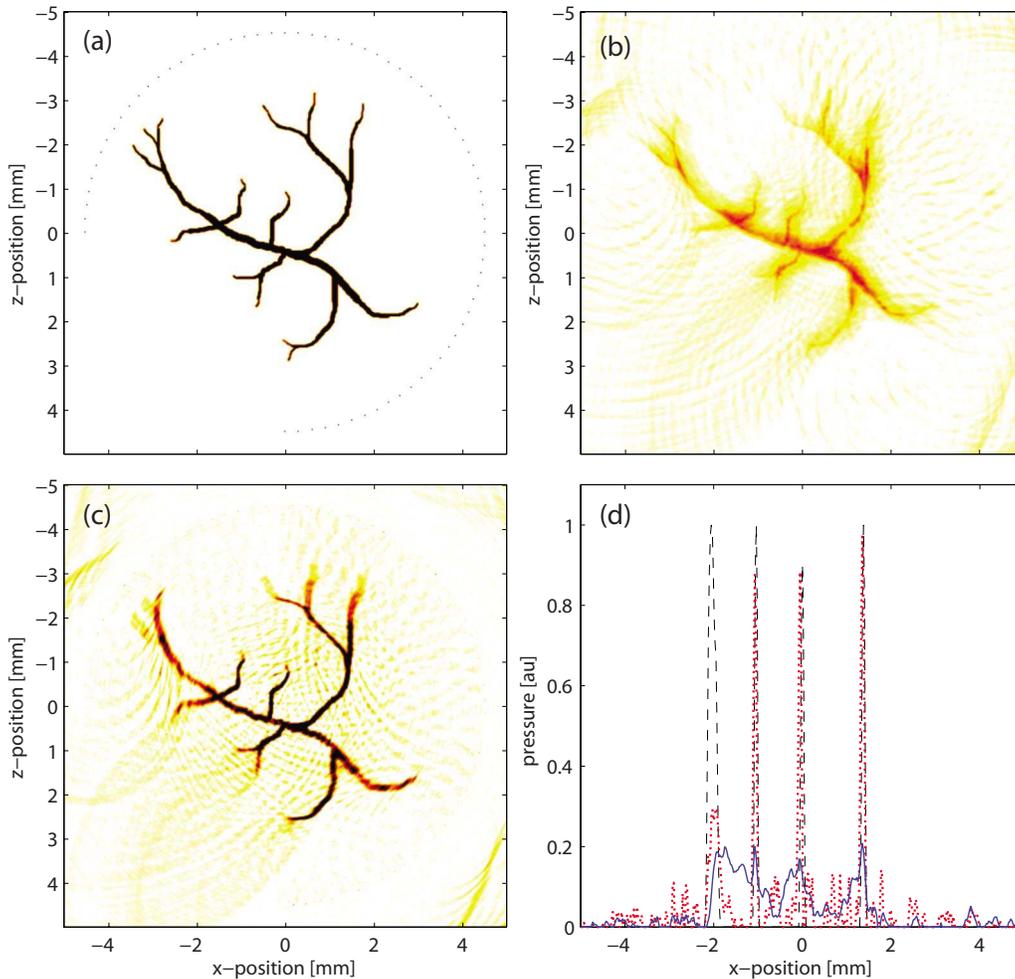
where the output is indexed as `p_zxy(z, x, y)`.

Regardless of the physical alignment of the sensor within the acoustic medium, the reconstruction is always returned as if the sensor was located across  $z=0$  (i.e., the first matrix row). The resolution of the reconstruction in the  $x$  and  $y$  directions is defined by the physical location and spacing of the sensor elements, while the resolution in the  $z$  direction is defined by sample rate at which the pressure field is recorded (i.e.,  $dt$ ). The reconstructed initial pressure distribution will thus typically have a much finer discretization in the  $z$  (time) direction.

As the reconstruction relies on the interpolation between a temporal and a spatial domain coordinate with different inherent spacings, both the speed and accuracy of the reconstruction are dependent on the interpolation method used. This can be controlled via the optional input parameter 'Interp', which is passed directly to the MATLAB function `interp3` (or `interp2` in 2-D). By default, this is set to '\*nearest', which optimizes the interpolation for speed. Setting 'Interp' to '\*linear' or '\*cubic' will reduce background interpolation artifacts in the image at the expense of computational speed. A visualization of the reconstruction can also be produced by setting the optional input parameter 'PlotRecon' to true. A positivity condition (which sets the negative parts of the reconstruction to zero) can similarly be enforced by setting 'PosCond' to true.

## 4 k-Wave Simulation Examples

The application of the functions within the k-Wave toolbox is illustrated in the following sections through several novel simulation and reconstruction examples.



**Fig. 5** Two-dimensional time reversal reconstruction example for a sparse array of detector positions on an arc. (a) Initial pressure distribution and sensor mask; (b) conventional time reversal reconstruction; (c) time reversal reconstruction after interpolating the measurement data onto a continuous sensor mask; and (d) a profile through  $z = -0.5$  showing the initial pressure (dashed line), conventional reconstruction (solid line), and interpolated reconstruction (dotted line).

#### 4.1 Improving Time Reversal Image Reconstruction Using Interpolated Sensor Data

In conventional time reversal image reconstruction, the recorded pressure time series are enforced in time reversed order as a Dirichlet boundary condition at the position of the detectors on the measurement surface. If a sparse array of detector points is used to collect the measurements (rather than a continuous surface), the enforced time reversal boundary condition will necessarily be discontinuous. This can cause significant blurring in the reconstructed image, as illustrated in Fig. 5. Here the initial pressure distribution is given by a  $512 \times 512$  pixel ( $10 \times 10$  mm) image representative of vasculature (loaded using the utility function `loadImage`). The detector array (`sensor.mask`) is defined as a 270-deg arc of radius 4.5 mm with 70 evenly spaced detector points. The corresponding time array (created using `makeTime`) has 2226 time-steps of 4.24 ns. A plot of the initial pressure distribution (smoothed using the utility function `smooth`) and the sensor mask (using the utility function `cart2grid` to place the Cartesian sensor points into the grid based image) is shown in Fig. 5(a).

The time reversal reconstruction using a different sized  $400 \times 400$  pixel grid with 2.5% random uniform noise added to the recorded sensor data before reconstruction (to avoid the inverse crime) is shown in Fig. 5(b). (The displayed reconstructions are shown with a positivity condition enforced.) The edges of the original image have been significantly blurred due to outgoing waves from each detector position on the measurement surface interacting with other positions at which a pressure value is also being enforced. This interaction can be avoided by interpolating the recorded data onto a continuous (rather than discrete) measurement surface within the k-space grid used for the reconstruction. This can be achieved in k-Wave by using the utility function `interpCartData` along with a binary sensor mask of a continuous surface that is spatially equivalent to the original Cartesian measurement surface (in this case, an arc). This function calculates the time series at the detector positions on the continuous binary sensor mask from those on the Cartesian sensor mask via interpolation. (Nearest neighbor is used by default.)

The reconstructed image using the interpolated sensor data and sensor mask is shown in Fig. 5(c). The edges of the image

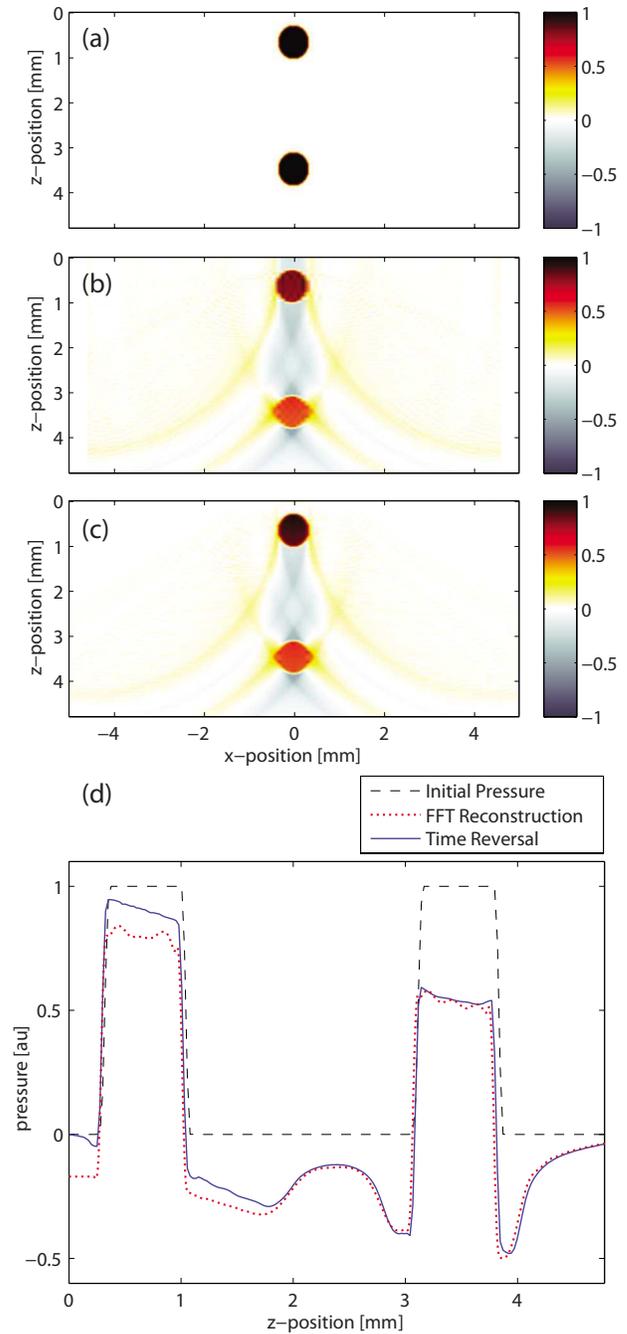
are now considerably sharper. This is also evident in Fig. 5(d), which shows a profile through  $z=-0.5$  mm. The junction between the main vessel and the right branch [the first two peaks visible in Fig. 5(d)] is noticeably sharper after the interpolation (dotted line) compared to before (solid line). The overall magnitude of the reconstruction and the signal-to-noise ratio have also been improved through partial correction for the discontinuous aperture.

#### 4.2 Comparison of Time Reversal and One-Step Image Reconstruction for a Planar Measurement Surface

Although time reversal image reconstruction is exact only for a closed measurement surface in odd dimensions and homogeneous media (in which Huygens' principle can be fulfilled), in practice, the technique has been successfully applied to heterogeneous media, reconstructions in even dimensions, and partially closed measurement surfaces.<sup>34</sup> Here, the use of time reversal for finite-sized planar measurement surfaces is demonstrated via comparison with the one-step, FFT-based reconstruction algorithm. The utilized initial pressure distribution, created using `makeDisc` within a  $472 \times 216$  pixel grid with a 20-pixel external PML, is shown in Fig. 6(a). The measurement surface is defined as a linear array of 100 evenly spaced Cartesian points along the line  $z=0$  from  $x=-4.5$  to 4.5 mm. The corresponding time array (created using `makeTime`) has 1610 time-steps of 7.45 ns.

The one-step, FFT-based reconstruction using 'Interp' set to '\*cubic' is shown in Fig. 6(b). The corresponding time reversal image reconstruction using a different sized  $400 \times 200$  pixel grid with the recorded sensor data interpolated onto a continuous sensor mask (as discussed in the previous section) is shown in Fig. 6(c). The magnitude of the latter has been multiplied by 2 to account for the fact that the planar measurement surface necessarily lies on only one side of the measurement domain and therefore does not detect waves propagating in the opposite direction. (This scaling is applied automatically within `kSpaceLineRecon` and `kSpacePlaneRecon`.)

Aside from the inherent resolution difference in the  $x$  direction (the time reversal reconstruction utilizes a larger data set due to the interpolation step), the two reconstructed images appear visually very similar. A  $z$ -profile through  $x=0$  is shown in Fig. 6(d), which facilitates a more quantitative comparison. The time reversal reconstruction illustrates an improvement when the objects are close to the sensor surface and is almost identical to the one-step FFT-based reconstruction when the objects are farther away. This raises three points of particular interest. First, the similarity of the two reconstructions suggests that the semicircular banding artifacts frequently seen in reconstructions using the one-step, FFT-based algorithm are largely due to limited aperture effects (although the method used for the interpolation step can also introduce additional artefacts, particularly if nearest-neighbor interpolation is used). Second, time reversal image reconstruction is sufficiently general that it can also be used for open planar measurement surfaces. From the results shown here, it seems probable that time reversal image reconstruction is exact in the case of an infinite plane. Third, the inherent inclusion of the evanescent wave component in time reversal (this is ex-

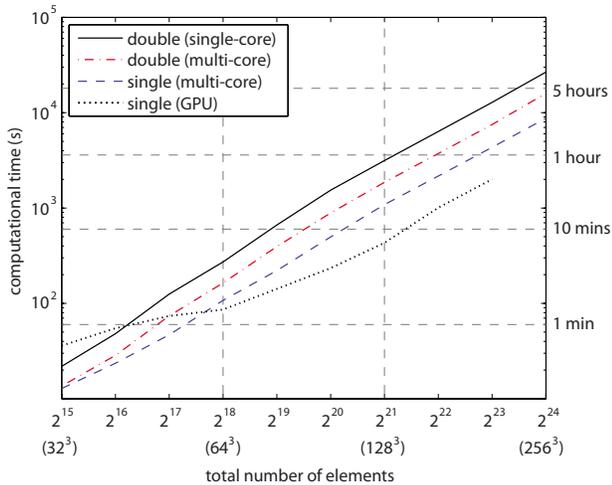


**Fig. 6** Comparison of time reversal and one-step, FFT-based image reconstruction for a linear measurement surface. (a) Initial photoacoustic pressure distribution—the measurement surface is located at  $z=0$ ; (b) FFT-based reconstruction; (c) time reversal reconstruction; and (d)  $z$  profile through  $x=0$ .

cluded in the one-step, FFT-based reconstruction) appears to improve the amplitude of the reconstruction and reduce artifacts close to the sensor surface. Consequently, it may be possible to improve one-step, FFT-based reconstructions by also including the contribution of evanescent waves.

#### 4.3 Optimizing k-Wave Performance

Although  $k$ -space methods have inherent computational advantages over analogous PS or FD methods, the overall com-



**Fig. 7** Comparison of the total computational time for a 3-D simulation with 1000 time steps against grid size using different data types and parallel processing.

computational efficiency is still dependent on the manner in which the algorithms are encoded and executed. For a standard k-Wave simulation, the majority of the computational time is spent running forward and inverse FFT routines, along with the point-wise multiplication of matrices. Additional time is also spent preparing and displaying the animated visualizations, and, if a Cartesian sensor mask is used with linear interpolation, precomputing the interpolation weights through Delaunay triangulation. The time spent doing the latter can be minimized by switching the visualization off by setting the optional input parameter 'PlotSim' to false, and by using either nearest-neighbour interpolation (by setting 'CartInterp' to 'nearest') or a binary sensor mask. After these modifications, the majority of the computational time is spent completing the time-stepped calculation of spatial derivatives via the FFT. It is possible to decrease this burden by capitalizing on MATLAB's use of overloaded functions for different data types. For example, computing an FFT of a matrix of `single` type takes less time than for `double` (the standard data format used within MATLAB). For most computations of interest here, the loss in precision as a result of doing the computations in `single` type is negligible.

Within the k-Wave simulation functions, the data type used for the variables within the time loop can be controlled via the optional input parameter 'DataCast'. A comparison of the total time (including precomputations) to run a 3-D forward simulation with 1000 time-steps for a varying number of total grid elements is shown in Fig. 7. The number of grid points in each Cartesian direction was always of the form  $2^N$ , the optional inputs 'PlotSim' and 'CartInterp' were set to false and 'nearest', respectively, and the computational times computed from three averages. The calculations were performed using MATLAB R2009a on a 64-bit PC with four 3.00-GHz CPUs and 16 GB of RAM.

The computational speed when 'DataCast' is set to 'single' (dashed line) is increased by approximately 1.7 times compared to performing the computations using 'double' (dashed-dotted line). As earlier versions of MATLAB do not support multithreading, a comparison of perform-

ing the computations using 'double' in MATLAB 2008a (on the same PC) is also shown (solid line). The use of multithreading similarly increases the computational speed by approximately 1.7 times. This speed-up is a result of both the FFT and the point-wise matrix multiplication used within the implemented k-space simulation model being inherently parallelizable.

The computational speed can be further improved through additional parallelization, in particular, by using data types that force program execution on the GPU. There are now several third party MATLAB toolboxes available that contain overloaded functions (such as the FFT) that run on any NVIDIA (Nvidia Corporation, Santa Clara, California) CUDA-capable GPU. Within MATLAB, the execution is as simple as casting the variables to the required data type. These toolboxes can be used with the k-Wave simulation functions by choosing the appropriate setting for the optional input parameter 'DataCast'.

A comparison of performing the same computations (in `single` type) using an NVIDIA Quadro FX 3700 with 512 MB of memory is also shown in Fig. 7 (dotted line). In 3-D, the additional computational overhead of utilising the GPU only becomes worthwhile when the grid size reaches  $2^{18}$  ( $64^3$ ) elements. No data point is available for the GPU computation using a grid size of  $256^3$ , as this exceeded the available memory of the particular card used for the comparison. (Note, GPU cards with 4 GB of memory are already available.) For larger grid sizes, the use of the GPU increases the computational speed between 2.5 (compared to multithreaded CPU computation using `single`) and 7 times (compared to single-threaded CPU computation using `double`). Using the GPU, the simulation of 1000 time-steps for a grid size of  $128^3$  can be completed in approximately 7 min.

## 5 Conclusions

The modeling and simulation of the phenomena underlying PAT has a number of important applications. These include investigating the effects of the various optical, thermal, acoustic, and system parameters on image reconstruction (fundamental to the development of quantitative PAT<sup>3</sup>); the simulation of phantom data; the development of new signal processing and tomographic reconstruction techniques; and indeed simply the reconstruction of photoacoustic images. Here, a new and freely available MATLAB toolbox for photoacoustic simulation and reconstruction is presented. The simulation functions are based on a k-space pseudo-spectral time domain solution to coupled first-order acoustic equations for homogeneous or heterogeneous media. The application of these functions to both forward simulations and time reversal image reconstruction is described. Additional one-step, FFT-based image reconstruction algorithms for linear (2-D) or planar (3-D) measurement surfaces are also discussed.

The application of the k-Wave toolbox to research questions within PAT is demonstrated through three examples. First, the use of interpolation is shown to considerably improve time reversal reconstruction when the measurement surface has only a sparse array of detector points. Here, the recorded data at the discrete detector positions is interpolated onto a continuous measurement surface within the time reversal grid. This prevents the outgoing waves from the discrete

detector positions being scattered by other positions at which a pressure value is also being enforced. This result is of particular interest, as most PAT systems based on conventional ultrasound detectors have sparse detector arrays.

Second, time reversal and one-step, FFT-based image reconstruction are shown to produce visually similar reconstructions when the measurement surface is planar. This result suggests that the banding artifacts seen in reconstructions using one-step, FFT-based algorithms are largely due to limited aperture effects. Similarly, it can be concluded that time reversal image reconstruction is sufficiently general that it can also be used for open planar measurement surfaces. Moreover, the inherent inclusion of the evanescent wave component in time reversal appears to improve the reconstruction close to the sensor surface. Consequently, it may be possible to improve one-step, FFT-based reconstructions by also including the contribution of evanescent waves. Last, an increase in computational speed of up to 7 times is illustrated through the use of parallelization using the GPU.

The framework of the simulation functions included within k-Wave allow the application of the toolbox in many fields of acoustics and ultrasonics. In addition to the initial pressure distribution used in the examples given here, flexible time varying sources may also be defined. This facilitates simulations of conventional diagnostic ultrasound, seismology, or environmental noise propagation. Similarly, the inclusion of arbitrary power law absorption means that realistic absorption parameters can be included in each of these cases. The directivity of sensor elements can also be modeled, increasing the realism of the simulations. Future releases of k-Wave will extend support for computations using the GPU, include additional functions for conventional ultrasound imaging, and allow integration with light models so that the complete photoacoustic forward problem can be simulated.

### Acknowledgments

This work was supported by the Engineering and Physical Sciences Research Council, UK.

### References

1. L. V. Wang, Ed., *Photoacoustic Imaging and Spectroscopy*, CRC, London (2009).
2. J. G. Laufer, D. Delpy, C. Elwell, and P. C. Beard, "Quantitative spatially resolved measurement of tissue chromophore concentrations using photoacoustic spectroscopy: application to the measurement of blood oxygenation and haemoglobin concentration," *Phys. Med. Biol.* **52**(1), 141–168 (2007).
3. B. T. Cox, S. R. Arridge, and P. C. Beard, "Estimating chromophore distributions from multiwavelength photoacoustic images," *J. Opt. Soc. Am. A* **26**(2), 443–455 (2009).
4. A. A. Oraevsky, "Optoacoustic tomography of the breast," in *Photoacoustic Imaging and Spectroscopy*, Lihong V. Wang, Ed., pp. 411–429, CRC Press, London (2009).
5. X. Wang, Y. Pang, G. Ku, X. Xie, G. Stocia, and L. V. Wang, "Non-invasive laser-induced photoacoustic tomography for structural and functional *in vivo* imaging of the brain," *Nat. Biotechnol.* **21**(7), 803–806 (2003).
6. J. Laufer, E. Zhang, G. Raivich, and P. Beard, "Three-dimensional noninvasive imaging of the vasculature in the mouse brain using a high-resolution photoacoustic scanner," *Appl. Opt.* **48**(10), D299–D306 (2009).
7. E. Z. Zhang, J. G. Laufer, R. B. Pedley, and P. C. Beard, "In vivo high-resolution 3D photoacoustic imaging of superficial vascular anatomy," *Phys. Med. Biol.* **54**(4), 1035–1046 (2009).
8. L. V. Wang, "Microwave-induced acoustic (thermoacoustic) tomography," in *Photoacoustic Imaging and Spectroscopy*, Lihong V. Wang, Ed., pp. 339–347, CRC Press, London (2009).
9. M. Xu, Y. Xu, and L. V. Wang, "Time-domain reconstruction algorithms and numerical simulations for thermoacoustic tomography in various geometries," *IEEE Trans. Biomed. Eng.* **50**(9), 1086–1099 (2003).
10. B. T. Cox, S. R. Arridge, and P. C. Beard, "Photoacoustic tomography with a limited-aperture planar sensor and a reverberant cavity," *Inverse Probl.* **23**, S95–S112 (2007).
11. B. T. Cox and B. E. Treeby, "Artifact trapping during time reversal photoacoustic imaging for acoustically heterogeneous media," *IEEE Trans. Med. Imaging* **29**(2), 387–396 (2010).
12. G. Paltauf, J. A. Viator, S. A. Prahl, and S. L. Jacques, "Iterative reconstruction algorithm for photoacoustic imaging," *J. Acoust. Soc. Am.* **112**(4), 1536–1544 (2002).
13. 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* **56**(8), 1666–1676 (2009).
14. T. D. Mast, "Empirical relationships between acoustic parameters in human soft tissues," *ARLO* **1**(2), 37–42 (2000).
15. B. T. Cox and P. C. Beard, "Fast calculation of pulsed photoacoustic fields in fluids using k-space methods," *J. Acoust. Soc. Am.* **117**(6), 3616–3627 (2005).
16. B. T. Cox and P. C. Beard, "Modeling photoacoustic propagation in tissue using k-space techniques," in *Photoacoustic Imaging and Spectroscopy*, Lihong V. Wang, Ed., pp. 25–34, CRC Press, London (2009).
17. P. M. Morse and K. U. Ingard, *Theoretical Acoustics*, Princeton University Press, Princeton, NJ (1968).
18. J. J. Markham, R. T. Beyer, and R. B. Lindsay, "Absorption of sound in fluids," *Rev. Mod. Phys.* **23**(4), 353–411 (1951).
19. S. Ginter, M. Liebler, E. Steiger, T. Dreyer, and R. E. Riedlinger, "Full-wave modeling of therapeutic ultrasound: nonlinear ultrasound propagation in ideal fluids," *J. Acoust. Soc. Am.* **111**(5), 2049–2059 (2002).
20. Y.-L. Sheu and P.-C. Li, "Simulations of photoacoustic wave propagation using a finite-difference time-domain method with Berenger's perfectly matched layers," *J. Acoust. Soc. Am.* **124**(6), 3471–3480 (2008).
21. B. E. Treeby and J. Pan, "A practical examination of the errors arising in the direct collocation boundary element method for acoustic scattering," *Eng. Anal. Boundary Elem.* **33**(11), 1302–1315 (2009).
22. 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).
23. 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).
24. 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**(6), 3453–3464 (2007).
25. B. E. Treeby and B. T. Cox, "Fast, tissue-realistic models of photoacoustic wave propagation for homogeneous attenuating media," in *Photons Plus Ultrasound: Imaging and Sensing 2009*, A. A. Oraevsky and L. V. Wang, Eds., *Proc. SPIE* **7177**, 717716 (2009).
26. B. T. Cox, J. G. Laufer, K. P. Köstli, and P. C. Beard, "Experimental validation of photoacoustic k-space propagation models," in *Photons Plus Ultrasound: Imaging and Sensing 2004*, A. A. Oraevsky and L. V. Wang, Eds., *Proc. SPIE* **5320**, 238–248 (2004).
27. J.-P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves," *J. Comput. Phys.* **114**(2), 185–200 (1994).
28. X. Yuan, D. Borup, J. W. Wiskin, M. Berggren, R. Eidsen, and S. A. Johnson, "Formulation and validation of Berenger's PML absorbing boundary for the FDTD simulation of acoustic scattering," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **44**(4), 816–822 (1997).
29. T. K. Katsibas and C. S. Antonopoulos, "A general form of perfectly matched layers for three-dimensional problems of acoustic scattering in lossless and lossy fluid media," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **51**(8), 964–972 (2004).
30. B. Fornberg, "High-order finite differences and the pseudospectral method on staggered grids," *SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal.* **27**(4), 904–918 (1990).

31. B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, Cambridge, UK (1996).
32. Y. Xu and L. V. Wang, "Time reversal and its application to tomography with diffracting sources," *Phys. Rev. Lett.* **92**(3), 033902 (2004).
33. P. Burgholzer, G. J. Matt, M. Haltmeier, and G. Paltauf, "Exact and approximative imaging methods for photoacoustic tomography using an arbitrary detection surface," *Phys. Rev. E* **75**(4), 046706 (2007).
34. Y. Hristova, P. Kuchment, and L. V. Nguyen, "Reconstruction and time reversal in thermoacoustic tomography in acoustically homogeneous and inhomogeneous media," *Inverse Probl.* **24**(5), 055006 (2008).
35. K. P. Köstli, M. Frenz, H. Bebie, and H. P. Weber, "Temporal backward projection of photoacoustic pressure transients using Fourier transform methods," *Phys. Med. Biol.* **46**(7), 1863–1872 (2001).
36. Y. Xu, D. Feng, and L. V. Wang, "Exact frequency-domain reconstruction for thermoacoustic tomography. I. Planar geometry," *IEEE Trans. Med. Imaging* **21**(7), 823–828 (2002).
37. S. J. Norton and M. Linzer, "Ultrasonic reflectivity imaging in three dimensions: exact inverse scattering solutions for plane, cylindrical, and spherical apertures," *IEEE Trans. Biomed. Eng.* **BME-28**(2), 202–220 (1981).
38. M. Jaeger, S. Schupbach, A. Gertsch, M. Kitz, and M. Frenz, "Fourier reconstruction in photoacoustic imaging using truncated regularized inverse k-space interpolation," *Inverse Probl.* **23**(6), S51–S63 (2007).