Development of Open Source, GPU-accelerated, Finite-Difference Time-Domain Software for Shear Wave Elastography Simulation

Final Report

August 24, 2015

Stephen McAleavey and Jonathan Langdon
Department of Biomedical Engineering
University of Rochester

I. Summary

Two finite difference time domain codes have been created for the purpose of simulating shear wave propagation: a 2D "C-plane" simulator allowing for arbitrary viscoelastic models in a plane normal to the push beam, and a 3D "B-scan" simulator using a Kelvin-Voigt viscoelastic model. The 3D simulator includes a tool for simulation of realistic push beams by full wave simulation of acoustic propagation from a definable aperture, and allows modeling of sound speed variations in the target, as well as variations in shear wave speed. A description of the simulators and representative output is presented, along with a comparison of simulated and measured output in the QIBA Phase II set 1 phantoms. Differences in the measured and simulated output are observed, and will be examined in future work.

II. Description of the simulators

II.A. C-plane simulator:

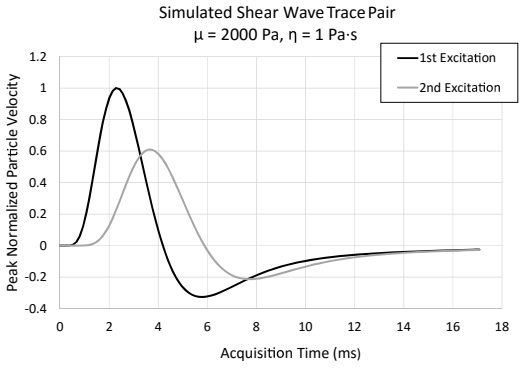
The C-plane simulator is based on a Kelvin-Voigt model based simulation similar to that described in [1] was implemented. This 2D simulation utilizes split field equations and perfectly matched layers (PMLs) implemented on a Yee cell grid as originally proposed in [2]. This simulator is capable of modeling non-axisymmetric beams, as for instance those produced at the focus of a rectangular aperture. The simulator is limited to modeling a single plane normal to the push beam axis, and assumes that the material and push beam do not vary along the beam axis.

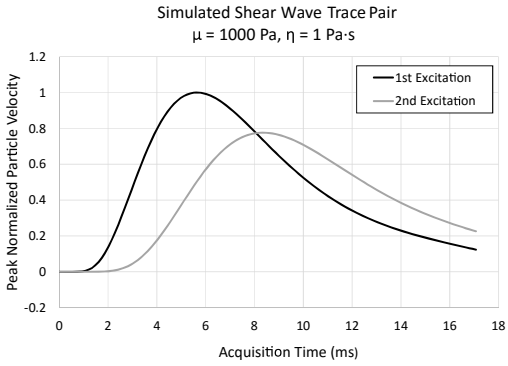
For the simulations shown here the modeled domain is 10 cm x 10 cm in the elevational-lateral plane with 1 cm PML on all domain edges. The simulation step size is 0.4 mm spatially and 1 μs temporally. Finally, this implementation includes an iterative guess and revise scheme for calculating time derivatives. This involves first calculating the time update using a linear model and then iteratively solving for the viscoelastic update.

To produce the simulated shear wave data, an initial axial velocity is applied to the simulation domain that approximates the focal geometry of the STL-SWEI pushing pulse. Specifically, a Gaussian cylindrical excitation with a lateral full width at half max

(FWHM) of 1 mm was applied to the center of the simulation grid for 300 μs. This excitation time was used to approximate the upper limit of ARFI excitation time used in our experiments. The data from the simulation was sampled at a rate of 7.44 kHz. Subsequently, this data is interpolated and stored in a data structure is identical to that used for actual RF processing in our in-house software. Therefore, the simulation data can be used to directly test the actual code used for processing ultrasound images.

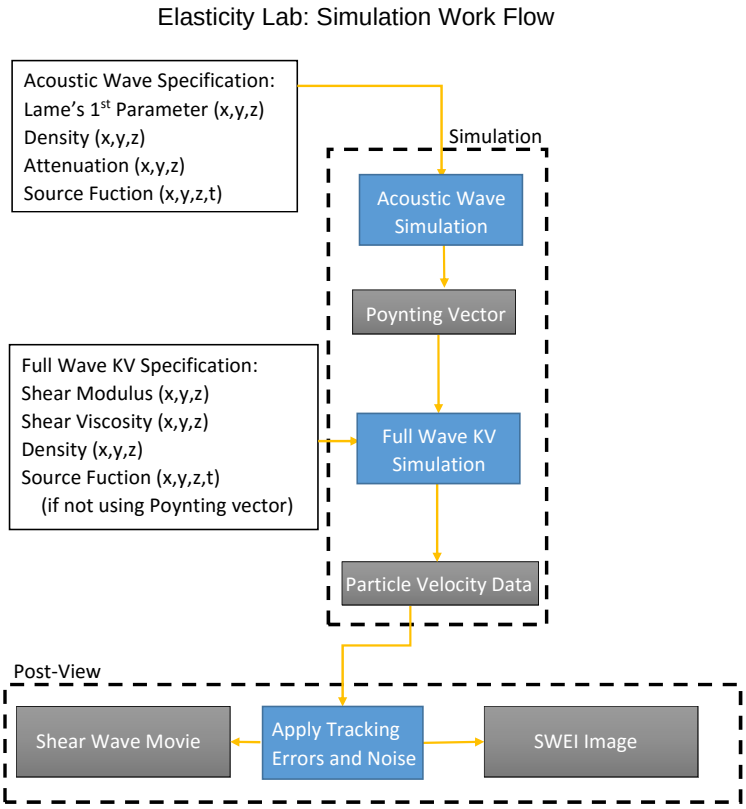
Representative simulator outputs are shown below.





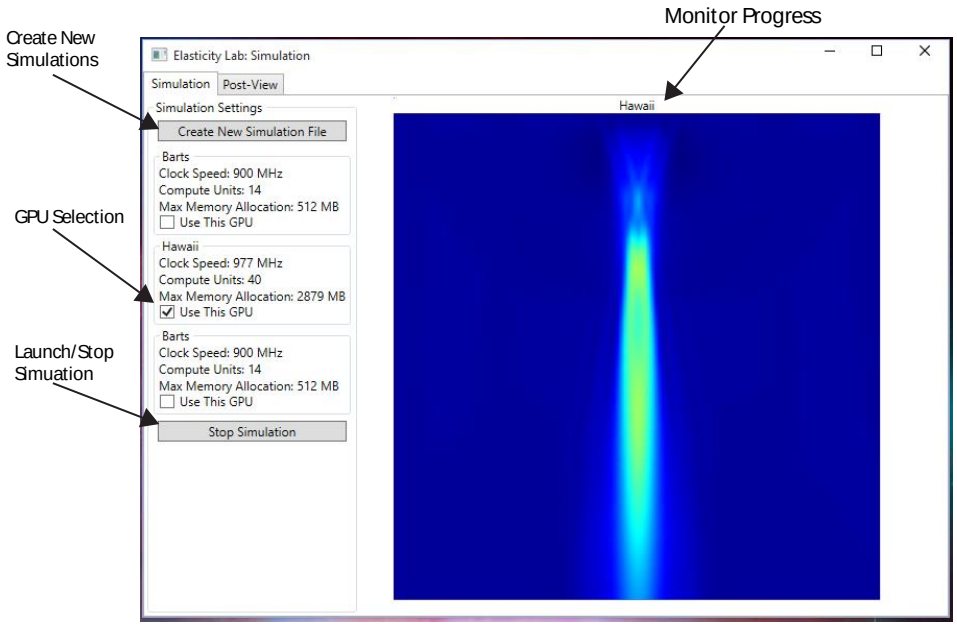
II.B. B-mode simulator – "Elasticity Lab Simulator"

The "B-plane" simulator permits simulation of both the ultrasonic "push beam" as well as the resulting shear wave. The simulation flowchart is depicted in the figure below. Either the result of an acoustic wave simulation, or an arbitrarily specified forcing function, is applied as input to the full-wave KV simulator to determine motion in response to an excitation.



Flowchart depicting the stages and code involved in the overall shear wave elastography simulation.

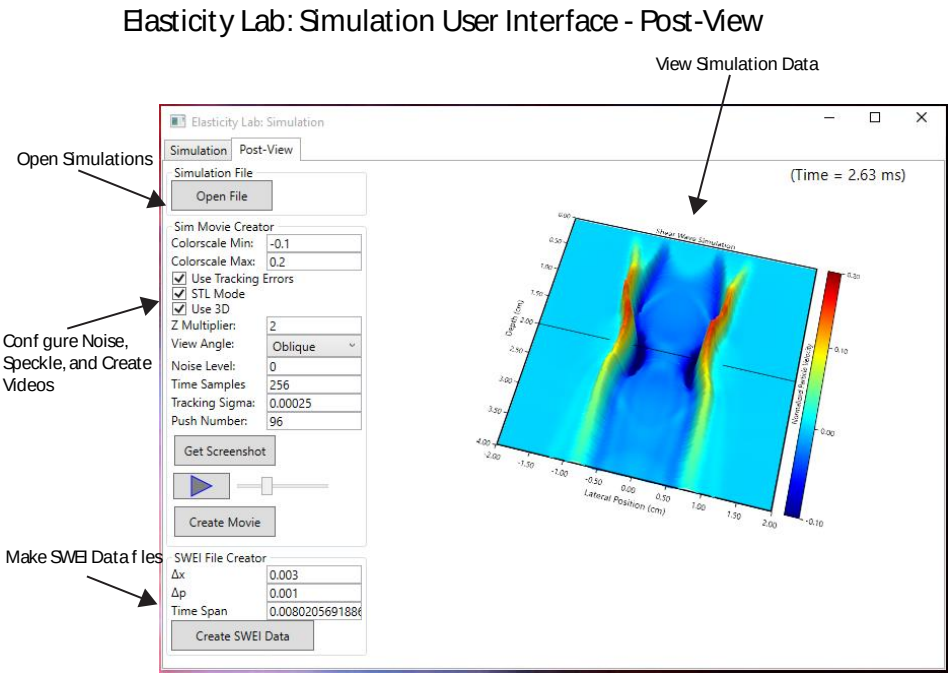
The acoustic wave simulator allows calculation of push beam geometry and forces. Beginning with acoustic parameters of the medium (Lame's 1st (λ) parameter, attenuation, and density as functions of position) and the acoustic source (aperture) as a function of position and time, an acoustic wave simulation is run to determine the Poynting vector associated with the push beam, from which both the magnitude and direction of the acoustic radiation force are calculated. This forcing function may then be applied to the full wave KV simulator.



User interface for the acoustic wave simulator, used to model the acoustic radiation force push beam.

The forcing function from the acoustic wave simulator, or an arbitrary user-specified forcing function, along with material parameters (modulus, viscosity, and density as functions of position) are the inputs to the full wave KV simulator. This simulator models the mechanical response of the described target, providing particle velocity as a function of position and time. Performing the entire simulation with realistic longitudinal wave speeds would take a prohibitive amount of time. Therefore, in the full wave simulation, the longitudinal wave speed is substantially decreased (10 - 100 m/s) to allow larger time steps to be used and increasing the speed of the simulation.

The files generated by the simulator can be opened in a post-view code allowing visualization of the motion. The user interface for this part of the simulation is depicted in Figure 4.3. In this part of the software, speckle bias can be approximated and noise added to the data. Shear wave propagation videos may be produced inside the interface. The data may then be processed into SWEI images with user defined tracking offsets and push beam spacings (beam ensemble settings). Because this SWEI specific functionality is in the last step of the simulation work flow, images using various beam ensemble settings may be produced. The files may also be opened in Matlab.



Post-view interface, which allows visualization of the shear wave simulation.

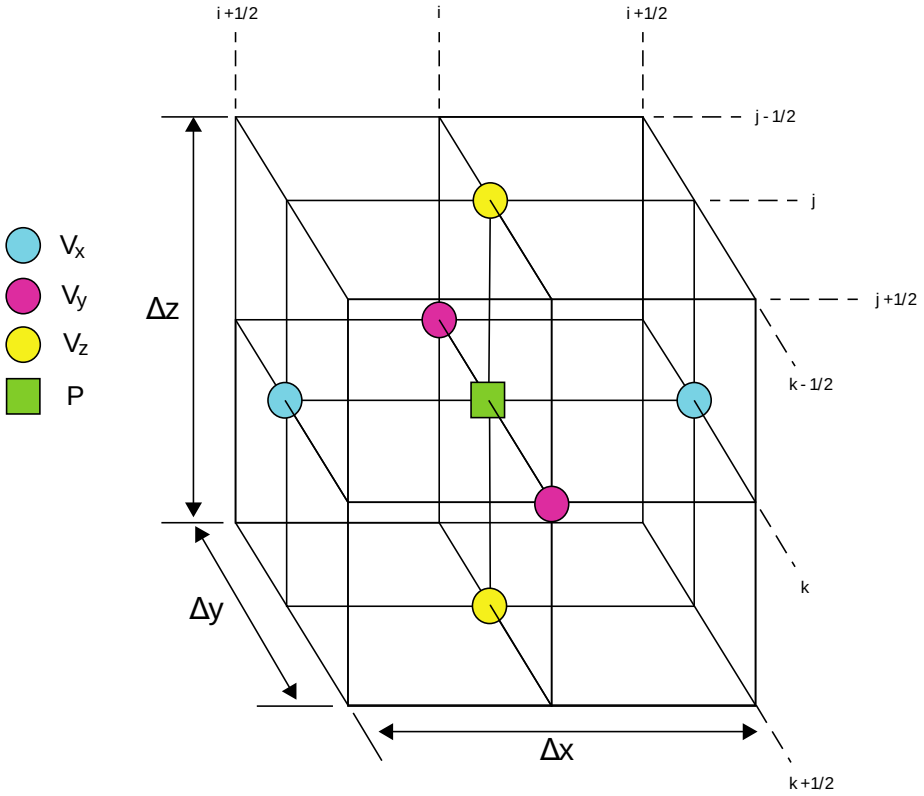
II.B.1. Acoustic Wave Simulation

The acoustic wave simulation is based on the acoustic wave equations in the velocity stress formulation. In this formulation, the particle velocity and the stress (or pressure) are related by a pair of first order differential equations. They may be expressed as

$$\rho \frac{\partial v}{\partial t} = \nabla P$$
$$\frac{\partial P}{\partial t} = \Lambda \nabla \cdot v.$$

Here v is the velocity, P the pressure, ρ is the density, and Λ is Lame's first parameter. We use the capital Λ is used here to distinguish it from the wavelength, λ. Although the equations could be combined to arrive at a second order differential equation, the first order equation are more convenient for discretization using finite difference method. Using the so-called staggered grid depicted in figure below, the differential equations may be split into component parts and discretized as

Computational Grid for Acoustic Wave Simulation



$$v_x^{i+\frac{1}{2},j,k,n+1} = v_x^{i+\frac{1}{2},j,k,n} + \frac{1}{\rho}\Delta t \frac{P^{i+1,j,k,n+\frac{1}{2}} - P^{i,j,k,n+\frac{1}{2}}}{\Delta x}$$

$$v_y^{i,j+\frac{1}{2},k,n+1} = v_y^{i,j+\frac{1}{2},k,n} + \frac{1}{\rho}\Delta t \frac{P^{i,j+1,k,n+\frac{1}{2}} - P^{i,j,k,n+\frac{1}{2}}}{\Delta y}$$

$$v_z^{i,j,k+\frac{1}{2},n+1} = v_z^{i,j,k+\frac{1}{2},n} + \frac{1}{\rho}\Delta t \frac{P^{i,j,k+1,n+\frac{1}{2}} - P^{i,j,k,n+\frac{1}{2}}}{\Delta z}$$

$$P^{i,j,k,n+\frac{1}{2}} = P^{i,j,k,n-\frac{1}{2}} + \kappa\Delta t \left(\frac{v_x^{i+\frac{1}{2},j,k,n} - v_x^{i-\frac{1}{2},j,k,n}}{\Delta x} + \frac{v_y^{i,j+\frac{1}{2},k,n} - v_y^{i,j-\frac{1}{2},k,n}}{\Delta y} + \frac{v_z^{i,j,k+\frac{1}{2},n} - v_z^{i,j,k-\frac{1}{2},n}}{\Delta z} \right)$$

Here i, j, and k are discrete integer coordinates in the x, y, and z dimensions, respectively. The discretization of the time dimension is given by the integer n. Consequently, the discrete time-space coordinates (i, j, k, n) map to the real valued coordinates (iΔx, jΔy, kΔz, nΔt). Each dimension is staggered into half steps, with the velocity calculated at integer multiples of time and the pressure calculated at half-integer multiples.

Grid dispersion is minimized by choosing a grid step size that is less than one-tenth of the shortest shear wave wavelength. Once the step size is chosen the time step is set using the Courant-Friedrichs-Lewy (CFL) condition. This is given as

$$\Delta t \leq \frac{\sqrt{3}}{c\sqrt{(\frac{1}{\Delta x})^2 + (\frac{1}{\Delta y})^2 + (\frac{1}{\Delta z})^2}}$$

Since this FDTD solver is used primarily for simulated mono-frequency excitation, attenuation may be accounted for by simply introducing a frequency independent attenuation. This would not be adequate for simulating pulse-echo imaging where there is pulse has a broad spectrum. However, for the purpose of simulating an ARF beam, the long duration of the pulse diminishes the effect of any transients yielding a narrow spectrum. To achieve what amounts to an exponential decay law, the velocity update equation are rewritten as

$$v_x^{i+\frac{1}{2},j,k,n+1} = \left(1 - \frac{\Delta t}{\tau}\right) v_x^{i+\frac{1}{2},j,k,n} + \frac{1}{\tau}\Delta t \frac{P^{i+1,j,k,n+\frac{1}{2}} - P^{i,j,k,n+\frac{1}{2}}}{\Delta x}$$

$$v_y^{i,j+\frac{1}{2},k,n+1} = \left(1 - \frac{\Delta t}{\tau}\right) v_y^{i,j+\frac{1}{2},k,n} + \frac{1}{\tau}\Delta t \frac{P^{i,j+1,k,n+\frac{1}{2}} - P^{i,j,k,n+\frac{1}{2}}}{\Delta y}$$

$$v_z^{i,j,k+\frac{1}{2},n+1} = \left(1 - \frac{\Delta t}{\tau}\right) v_z^{i,j,k+\frac{1}{2},n} + \frac{1}{\tau}\Delta t \frac{P^{i,j,k+1,n+\frac{1}{2}} - P^{i,j,k,n+\frac{1}{2}}}{\Delta z}$$

At the boundaries of the computational domain, the velocity outside the grid is taken to equal that immediately inside. Let I, J, and K be the lengths of the computational grid in the i, j, and k directions, respectively. The boundary conditions may be written as:

$$v_x^{\frac{1}{2},j,k,n} = v_x^{-\frac{1}{2},j,k,n}$$

$$v_x^{(N-1)-\frac{1}{2},j,k,n} = v_x^{(N-1)+\frac{1}{2},j,k,n}$$

$$v_y^{i,\frac{1}{2},k,n} = v_y^{i,-\frac{1}{2},k,n}$$

$$v_y^{i,(J-1)-\frac{1}{2},k,n} = v_y^{i,(J-1)+\frac{1}{2},k,n}$$

$$v_z^{i,j,\frac{1}{2},n} = v_z^{i,j,-\frac{1}{2},n}$$

$$v_z^{i,j,(K-1)-\frac{1}{2},n} = v_z^{i,j,(K-1)+\frac{1}{2},n}$$

Note that these conditions are the same as stating that the derivative of the velocity across the boundary is zero. We simply skip calculating the derivative across the boundary to implement this. However, these are reflective boundary conditions. In order to prevent reflections, the outer several cells of the computational domain are replaced by perfectly matched layers (PMLs). The PML is implemented by splitting the update equations for the perimeter of the grid into separate equations for each individual spatial derivative and

the weighting each derivate by a scaling factor. This is referred to as operator stretching. For the velocity case, there is only one spatial derivative to begin with and so the number of equations is unchanged. The velocity PMLs are given by

$$v_x^{i+\frac{1}{2},j,k,n+1} = C_x v_x^{i+\frac{1}{2},j,k,n} + D_x \frac{1}{\Delta x}\left(P^{i+1,j,k,n+\frac{1}{2}} - P^{i,j,k,n+\frac{1}{2}}\right)$$

$$v_y^{i,j+\frac{1}{2},k,n+1} = C_y v_y^{i,j+\frac{1}{2},k,n} + D_y \frac{1}{\Delta y}\left(P^{i,j+1,k,n+\frac{1}{2}} - P^{i,j,k,n+\frac{1}{2}}\right)$$

$$v_z^{i,j,k+\frac{1}{2},n+1} = C_z v_z^{i,j,k+\frac{1}{2},n} + D_z \frac{1}{\Delta z}\left(P^{i,j,k+1,n+\frac{1}{2}} - P^{i,j,k,n+\frac{1}{2}}\right)$$

Here Cx, Cy, Cz and Dx, Dy, Dz are the PML scaling factors for the x, y, and z derivatives, respectively. Notice that Dx, Dy, and Dz replace the time step in the equations. These coefficients for the x dimension are given by

$$C_x = e^{-s_x \Delta t}$$

$$D_x = \frac{1 - C_z}{s_x}$$

and the equations for the other dimensions are the same with substitution of the subscript. The term s_x is an attenuation that is set as desired to achieve good absorption of the incident wave. Typically this value is ramped up exponentially from zero at the inner edge of the PML to some maximum value at the domain's outer edge.

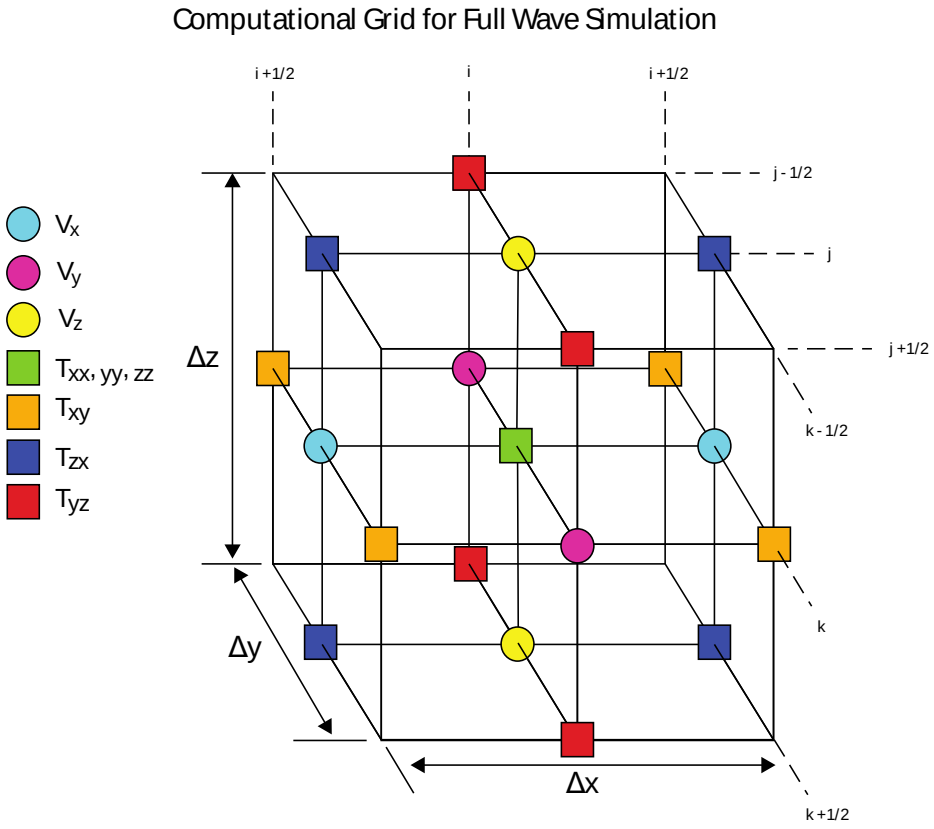
Implementing of the FDTD model on the GPU requires consideration of memory access. In our implementation, the data is laid out with the depth, z, dimension leading. On AMD GPUs, GPU threads come in batches of 256 called work-groups. Since GPU performance relies critically on adjacent threads accessing adjacent data, in our implementation each work group is assigned 256 adjacent point on the FDTD grid vertically. However, this requires that the depth dimension be a multiple of 256. Therefore, once the step size is determined the depth of the grid must be $256 * \Delta z * m$ for some integer m. When the grid is not a multiple of 256 then the work-group size is reduced to some power of 2 that will fit the grid. For odd numbered grid lengths only one thread per work-group may be assigned. Thus, choosing the correct grid size can result in an up to 256x boost in performance.

In addition to memory access optimizations, the memory usage in our implementation is minimized by using functions to represent the material parameters rather than arrays of values. This reduces the total memory needed and the number of accesses to the GPUs global memory that is required. These functions are defined in the specification files. They may then be compiled into the core FDTD program at run-time.

II.B.2. Full Wave Kelvin-Voigt Simulation

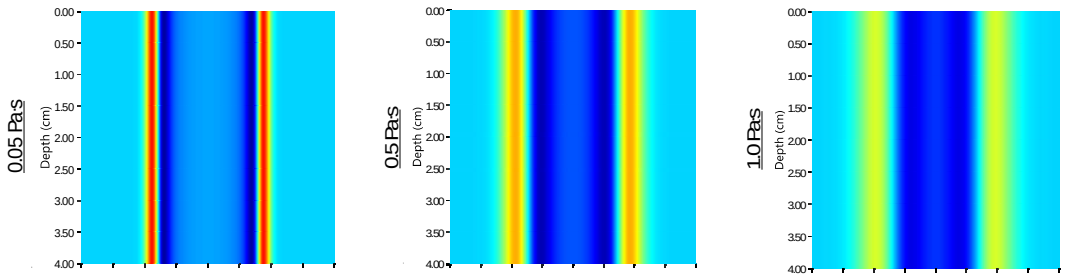
The full wave simulation is similar to that describe for the acoustic wave case. However, the full stress tensor, T , is calculated rather than just the pressure, P . The computational

grid used in the implementation is shown in the figure below. The three diagonal components of the stress tensor (T_{xx}, T_{yy}, T_{zz}) now replace the pressure at the center of the unit cube. For the isotropic media being simulated, the stress tensor is taken to be symmetric with $T_{xy} = T_{yx}$, $T_{zx} = T_{xz}$, and $T_{zy} = T_{yz}$. These off-diagonal components of the stress tensor are computed along the edges of the unit cube. The choice of edge assures that the particular stress component occupies the same plane as the velocity components with the same subscripts (e.g. T_{xy} share a plane with v_x and v_y).



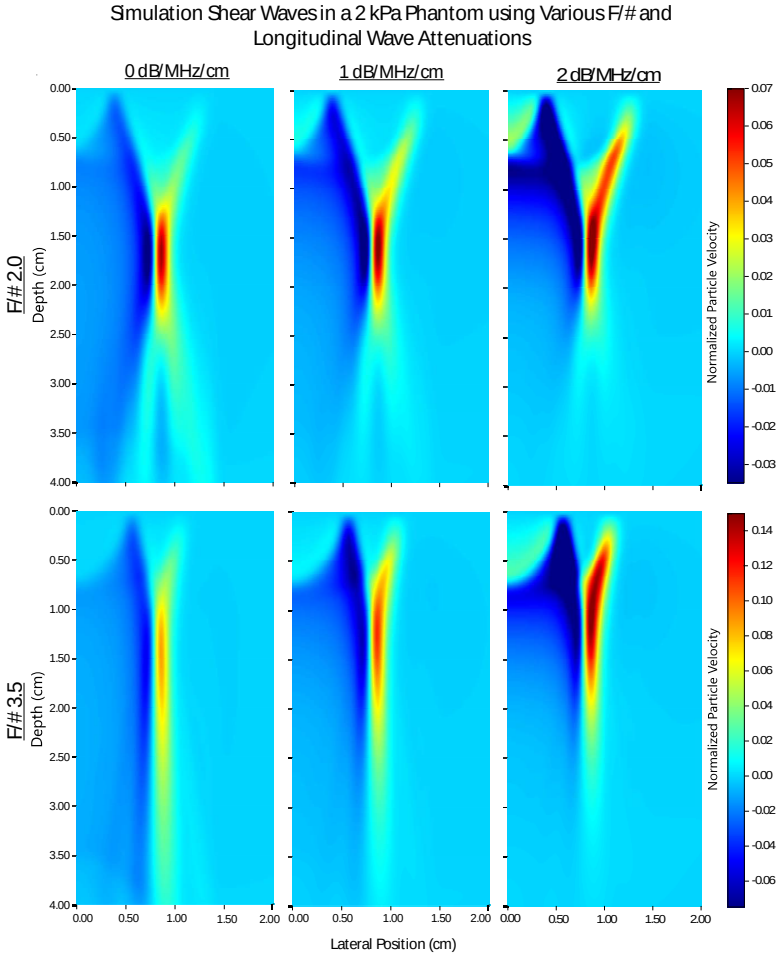
II.B.3 Cylindrical Wave Simulations

The cylindrical wave excitation of a homogenous media provides a test case for the accuracy of the 3D simulation. The solution is easily modeled and is built into our MLE solver. Therefore, the simulation's accuracy can be readily verified in this test case. Here a set of five viscoelastic simulations were performed with cylindrical wave excitations in a 2 kPa shear modulus 3D simulated material. The shear viscosity was set to either 0.05 Pa•s, 0.25 Pa•s, 0.50 Pa•s, 0.75 Pa•s, and 1.00 Pa•s. The simulation viscosity was then validated using the STL-VE algorithm. A 4 cm by 4 cm region of interest was modeled.

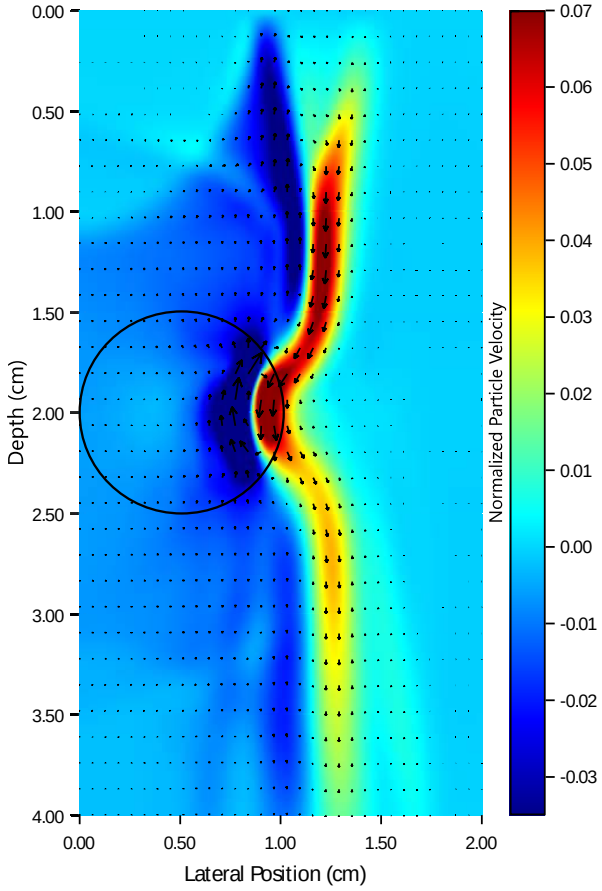


Representative frames from simulated cylindrical waves. A KV medium with μ=2kPa an viscosity of 0.05-1 Pa s was modeled for verification of the FDTD full wave KV model.

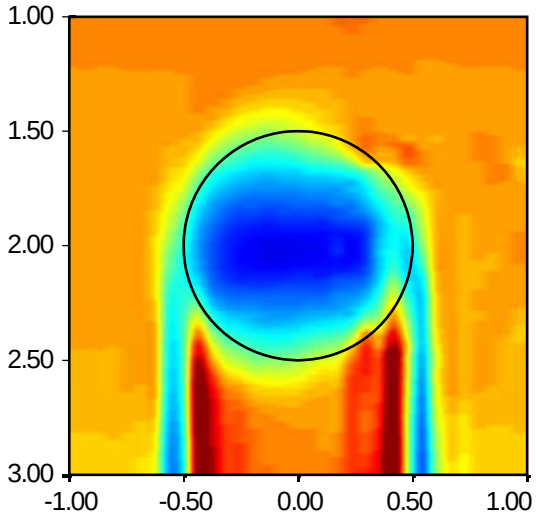
The acoustic wave simulator allows for creation of realistic push beams and radiation force geometries, allowing examination of potential biases in a realistic imaging scenario.The panels here depict simulated typical ARF beam geometries applied to the Full Wave simulation. The F/# and the acoustic attenuation both play a role in determining the shape of the excitation beam. The simulated beams all utilize a focal depth of 2 cm and a center frequency of 4.21 MHz, matching the phantom experiments.



Distortion of the shear wavefront by inclusions of differing shear modulus from the background is modeled. The figure below provides an example of diffraction of the shear wave by an inclusion.

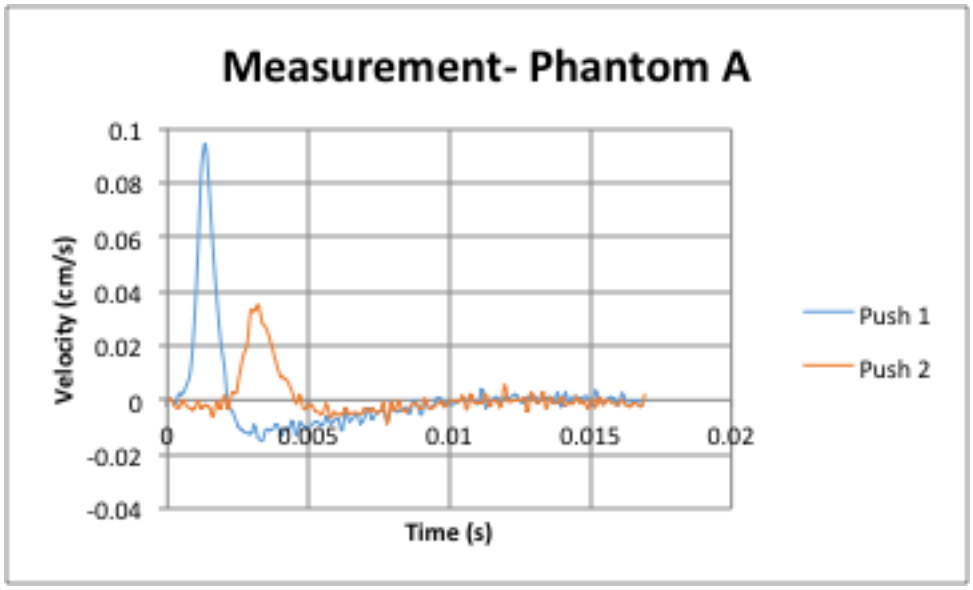


The use of the full wave acoustic simulator allows diffraction of the push beam to be modeled. Below is a simulated STL-SWEI image showing effect of push beam diffraction due to differing acoustic wave speed in lesion and background, in addition to shear wave speed difference between lesion and background. Note that tracking beam diffraction is not simulated. The boundary of the sphere is depicted by the black circle. Push beam diffraction effects are observed beneath the lesion. Additionally, the sphere appears smaller in height than in width with a region of increased modulus within the bounding circle.

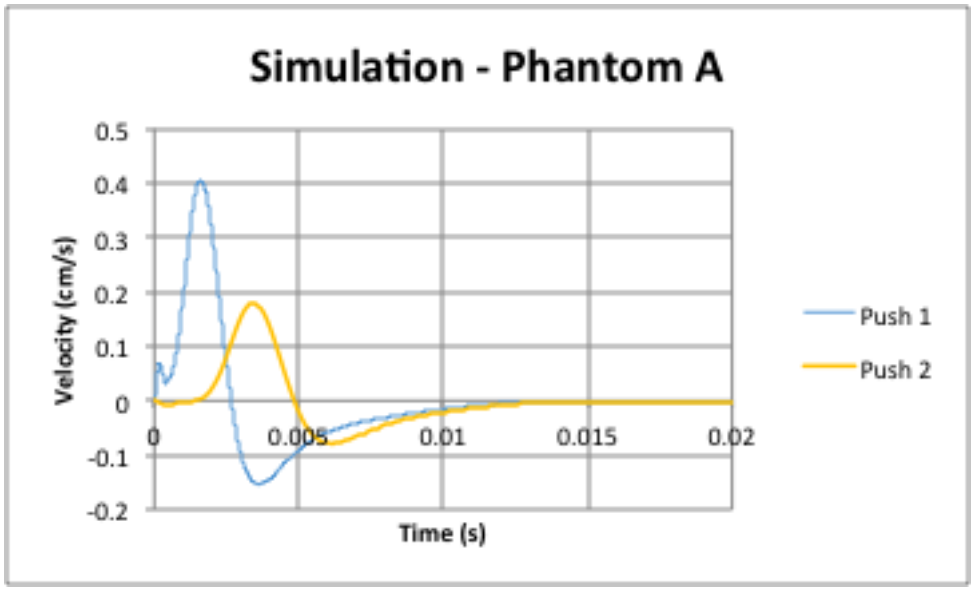


III. Comparision of phantom and simulated shear wave data

QIBA SWS Phase II Set 1 phantoms were scanned at the University of Rochester with a Siemens Antares scanner and custom research beam sequences. A representative set of measured shear wave traces is shown in the figure below. A push aperture of F/3.5 focused at a depth of 20 mm was used. The offset from the tracking location to the "Push 1" location was 4 mm; the offset to the second push location was 8 mm. The estimated wave speed from these traces was 2.21 m/s.



An approximate matching model was simulated, based on estimates of the shear wave speed and viscosity. The resulting estimated shear wave group speed from the simulation was 2.09 m/s



References:

[1] M. Orescanin, Y. Wang, and M. Insana, "3-D FDTD simulation of shear waves for evaluation of complex modulus imaging," *IEEE Trans Ultrason Ferroelectr Freq Control*, vol. 58, pp. 389-98, Feb 2011.

[2] J. P. Berenger, "A Perfectly Matched Layer for the Absorption of Electromagnetic-Waves," *Journal of Computational Physics*, vol. 114, pp. 185-200, Oct 1994.