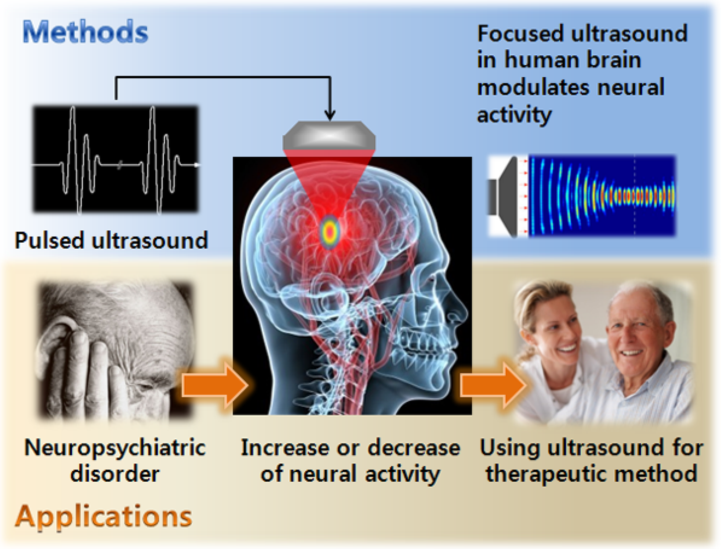
GPU-accelerated Simulation of Elastic Wave Propagation

Kristian Kadlubiak[1], Bradley E. Treeby[2], Jiri Jaros[1]

[1]Faculty of Information Technology, Brno University of Technology, CZ
[2]Department of Medical Physics and Bioengineering, University College London, UK

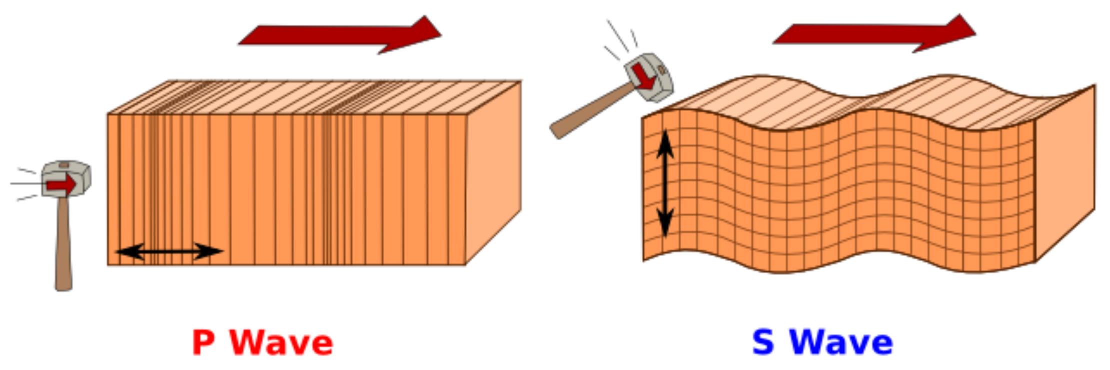
1 Overview

The simulation of elastic wave propagation has many applications in ultrasonics. Non-invasive, focal neurostimulation with ultrasound is a potentially powerful neuroscientific tool that requires effective transcranial focusing of ultrasound to develop. Time-reversal focusing using numerical simulations of transcranial ultrasound propagation can correct for the effect of the skull but relies on accurate simulations.



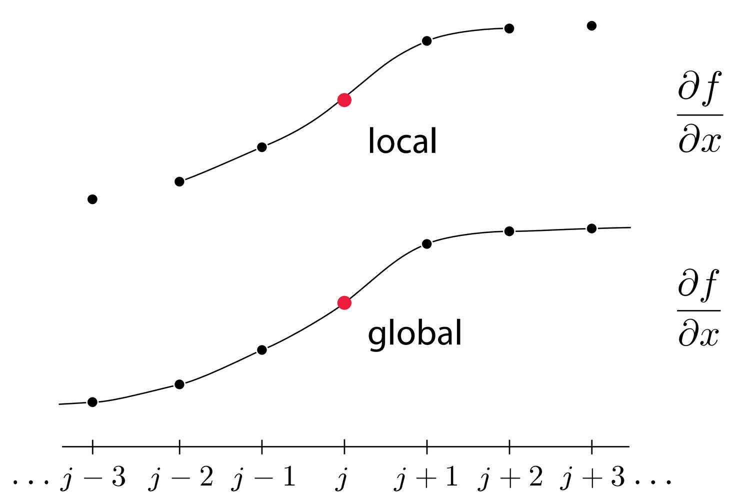
2 Elastic Wave Propagation in Bones and Skull

The governing equations must account for both compressional (P) and shear (S) waves in heterogeneous absorbing media.



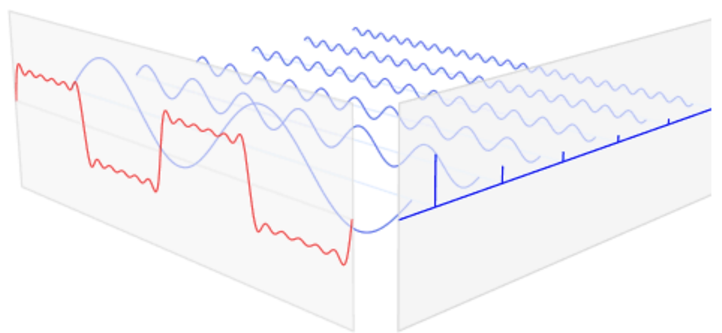
The numerical model is based on the explicit solution of coupled PDEs using the Fourier pseudospectral method. This uses the Fourier collocation spectral method to compute spatial derivatives, and a leapfrog finite-difference scheme to integrate forwards in time.

$$\partial_x \sigma_{xyz} = \mathbb{F}_x^{-1}\left\{ik_x e^{+ik_x \Delta x/2}\mathbb{F}_x\left\{\sigma_{xyz}\right\}\right\}$$



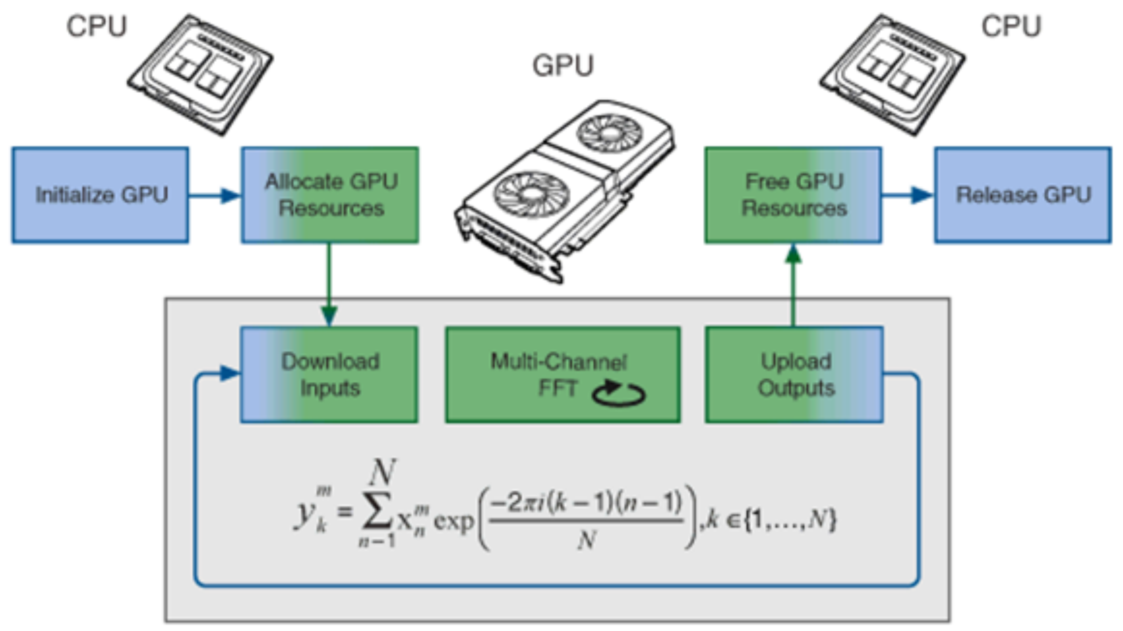
$$\frac{\partial f}{\partial x} \approx \frac{\frac{1}{12}f^{j-2} - \frac{2}{3}f^{j-1} + \frac{2}{3}f^{j+1} - \frac{1}{12}f^{j+2}}{\Delta x}$$

$$\frac{\partial f}{\partial x} \approx \mathbb{F}^{-1}\left\{ik_x \mathbb{F}\left\{f\right\}\right\}$$



3 Native CUDA application

The native CUDA application performs all calculation on the GPU side. The processor only loads data from the input HDF5 file, uploads it into GPU memory, samples output quantities and controls the simulation.

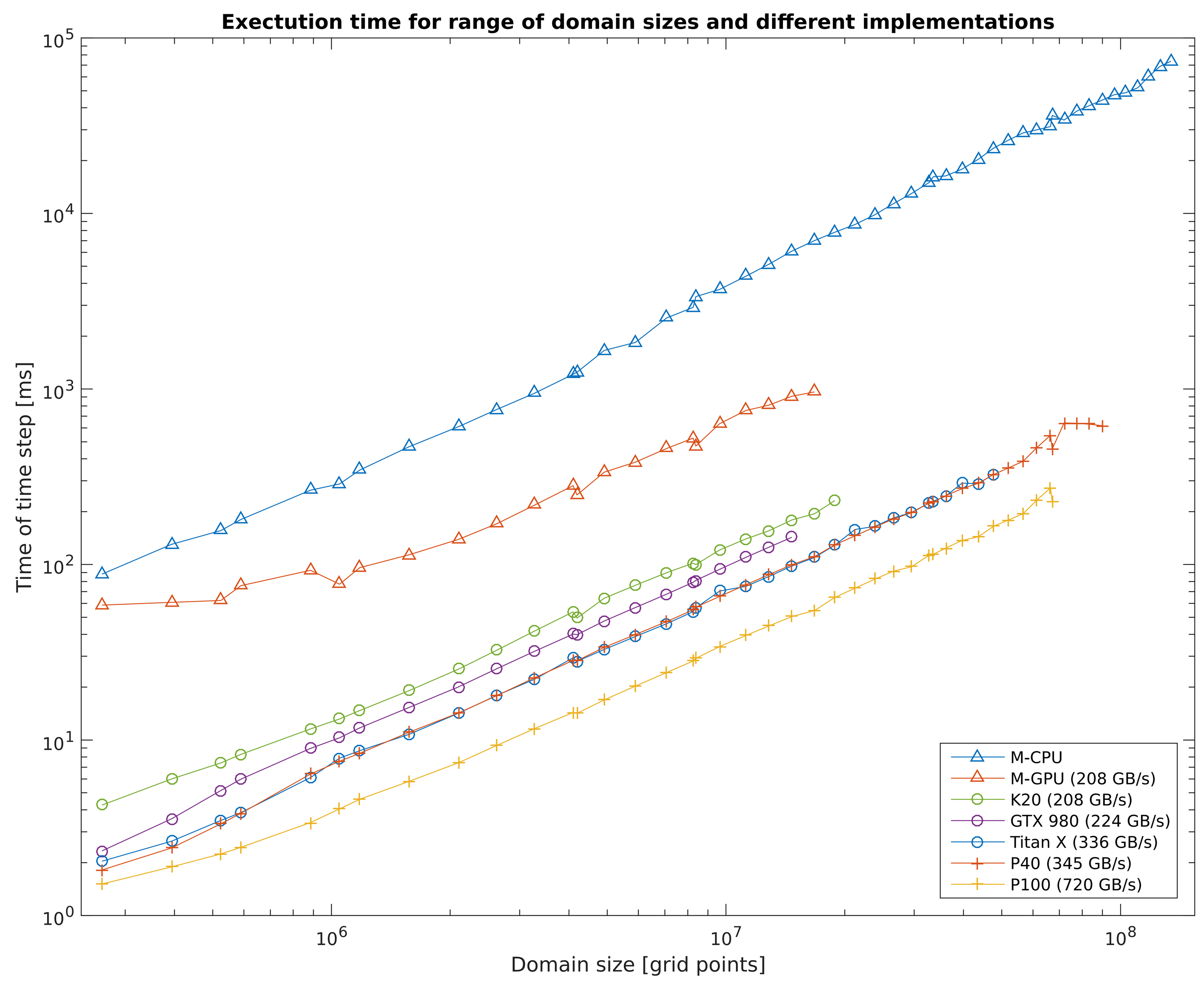


$$y_k^m = \sum_{n=1}^{N} x_n^m \exp\left(\frac{-2\pi i(k-1)(n-1)}{N}\right), k \in \{1,\dots,N\}$$

The code uses the cuFFT library to calculate FFTs and a number of fine tuned CUDA kernels to calculate element-wise matrix operations.

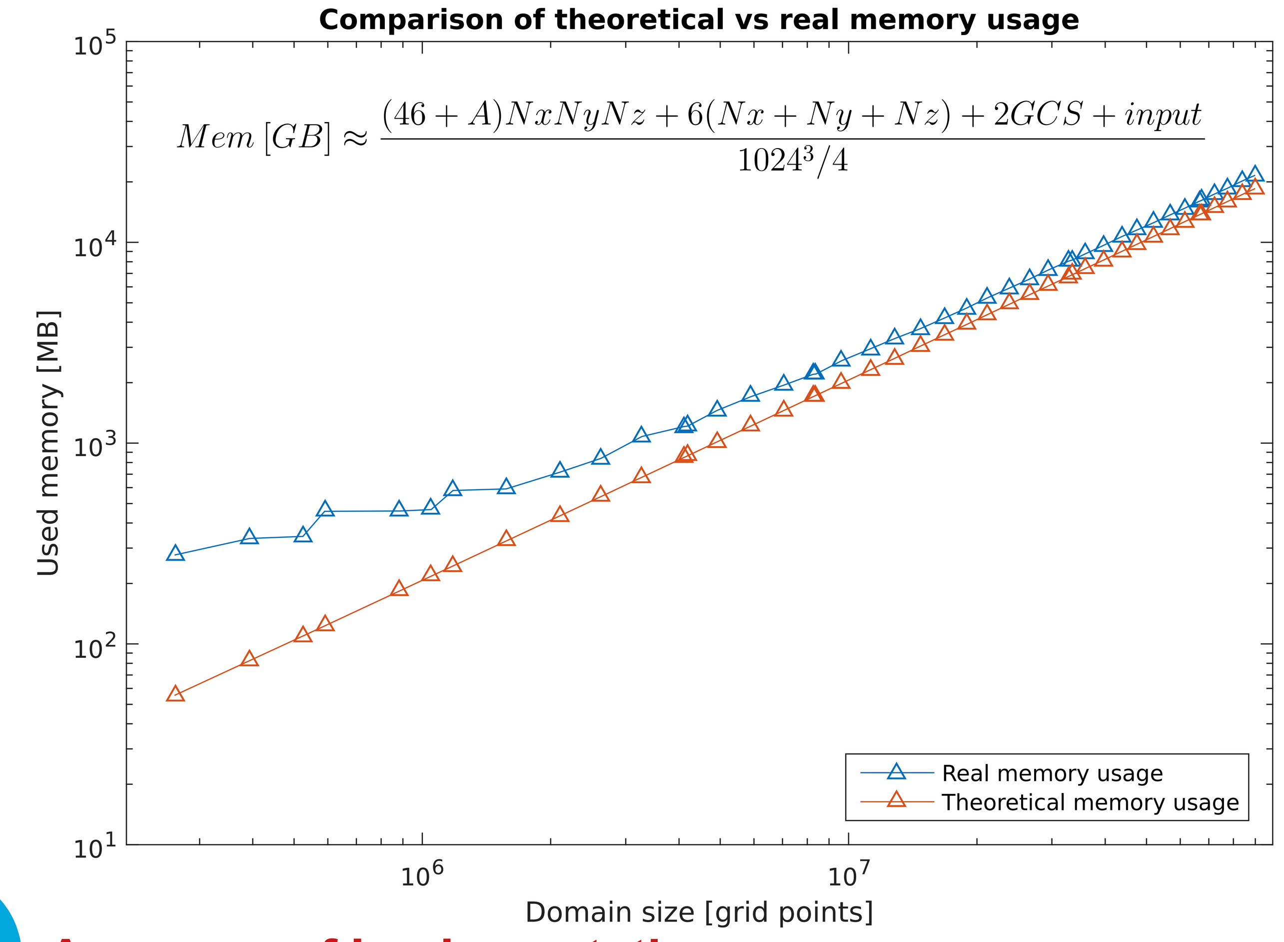
4 Performance Comparison

Execution times for one simulation time step. The domain sizes vary between 64^3 and 512^3 grid points. The first two curves M-CPU and M-GPU stand for the Matlab code running on CPU and GPU using the Matlab Parallel Computing Toolbox, while the others represent our native CUDA application. The numbers at the curves represent the maximal and average speed-up with respect to the Matlab CPU code.



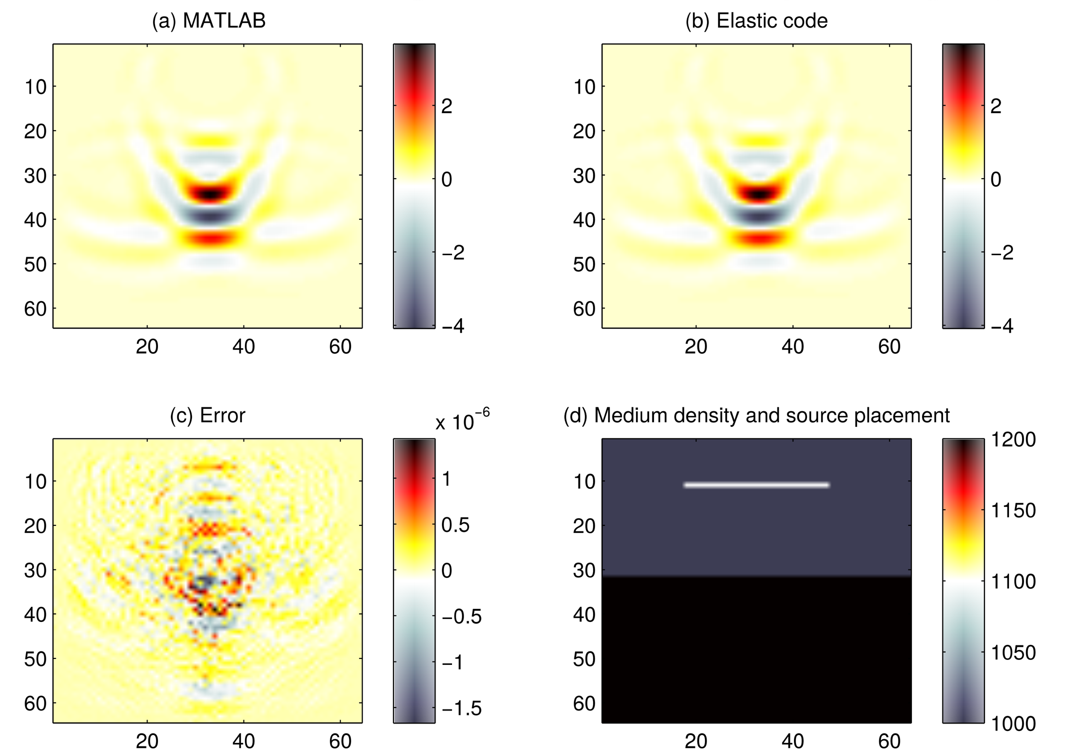
6 Memory Consumption

Comparison of the predicted and measured GPU memory consumption for the Nvidia Pascal P40 with 24GB of memory for domain sizes between 64^3 and 512^3.



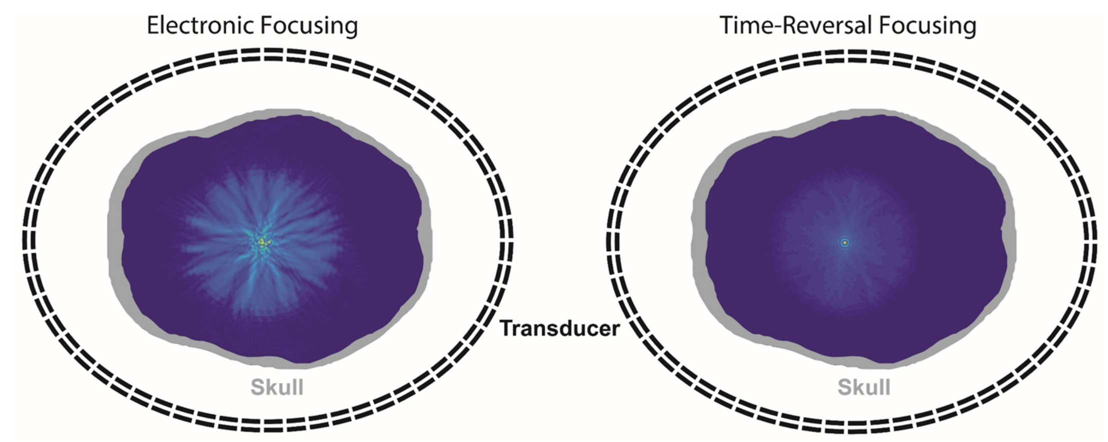
7 Accuracy of Implementation

Accuracy of implementation was tested against Matlab code and as presented in error figure, the maximum relative error is around 1.5×10^{-6} which is bordering with the accuracy of single precision.



8 Conclusions

The code has both significantly decreased the simulation time and extended the range of manageable simulation scenarios. For example, the elastic wave propagation simulation with 448^3 grid points and 4,655 time steps can now be completed in 48 minutes, instead of several hours.



CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science - LQ1602" and by the IT4Innovations infrastructure which is supported from the Large Infrastructures for Research, Experimental Development and Innovations project IT4Innovations National Supercomputing Center - LM2015070". This work was supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID pr14. This work was also supported by the Engineering and Physical Sciences Research Council, UK, grant numbers EP/L020262/1 and EP/M011119/1.

IT4Innovations national supercomputing center