AMCS 312. Course project

Oleg Ovcharenko

January 6, 2017

Abstract

In this project we explore scalability of an implicit finite difference code in time domain that solves acoustic wave equation in homogeneous isotropic 3D media using capabilities of Portable, Extensible Toolkit for Scientific Computation (PETSc). Also we provide performance comparison in terms of execution time with SOFI3D_acoustic, developed in Karlsruhe Institute of Technology (KIT).

Introduction

There is a variety of computational methods for solving approximately physical problems such as Finite-Element Method (FEM) [Strang and Fix, 1973], spectral-element method(SEM) [Komatitsch and Tromp, 1999] and others, but the most popular one is Finite-difference method(FDM) that is widely used due to it's simplicity, speed, relative easiness in implementation and parallelization.

We won't get into details of finite-difference theory, but one could find sufficient introduction in Moczo et al. [2007]. The general idea is simple – solve equation in FD approximation where all spatial and temporal derivatives are replaced with their descritized on grid analogous. There are two principal points to choose: how to discretize derivatives in space and how to discretize them in time. As an example, this choice leads to such methods as pseud-spectral methods(PSM) [Kosloff and Baysal, 1982], where time derivatives could be approximated in variety of ways, but spatial derivatives are computed using Fourier transforms. Also there is a choice of grid types with their pros and cons, for example: conventional collocated grid, staggered grid Virieux [1984, 1986], rotated staggered grid [Saenger et al., 2000] and others. The way of approximation of time derivatives defines two basic time stepping methods: explicit and implicit. Explicit methods calculate the state of a system at a later time from the state of the system at the current time, while implicit methods find a solution by solving an equation involving both the current state of the system and the later one. Implicit methods require an extra computation and it can be much harder to implement them. Despite these difficulties, implicit methods are used because there are many problems arising in practice for which the use of an explicit method requires impractically small time steps Δt to keep the error in the result bounded (to satisfy Courant-Friedrichs-Lewy condition [Courant et al., 1928]). For such problems, to achieve given accuracy, it takes much less computational time to use an implicit method with larger time steps, even taking into account that one needs to solve the whole system of equations at each time step. So it depends upon the problem to be solved, whether one should use an explicit or implicit method.

Motivation

We have chosen seismic wave propagation as the problem of interest as this topic has always been an area of interest among Earth science researchers because wave propagation is widely used both in exploration and global geophysics. On exploration scale seismic waves are propagated at each step of full-waveform inversion and on global scale this is done for modeling of surface waves, to model earthquake responses and etc. Implicit time stepping methods are very promising, but their behavior on large scales is not explored well yet.

Approach

Implicit FD operators have been relatively recently introduced to solve seismic wave propagation problems [Liu and Sen, 2009, Kosloff et al., 2010, Chu et al., 2010], but one could already note their potential driven also by steady growth of HPC facilities. That is why we have chosen implicit time stepping in order to check scalability and overall performance of this approach as currently the majority of wave propagators are implemented with explicit schemes. In general, use of implicit method was motivated by several factors. First, we were impressed by performance of iterative Krylov-based methods implemented in PETSc (Portable, Extensible Toolkit for Scientific Computation) for solving systems of linear and non-linear equations. Second, it is said, that using implicit method one could choose larger time step and fewer grid points per wavelength, whereas in explicit methods one has to satisfy CFL condition and thus use very small time step for fine grids. Third, [Chu and Stoffa, 2012, Chu et al., 2010] showed that using implicit numerical operators higher accuracy might be achieved with negligible additional costs.

Framework

We consider acoustic wave propagation due to it's simplicity and as it is sufficient for the purpuses of the study. Mathematically acoustic wave propagation in homogeneous media can be expressed as a second-order hyperbolic PDE given as eq. 1. Chosing the grid type and spatial derivative discretization for this equation, we follow reasoning of Geller et al. [2005] who showed that “Displacement only” schemes on collocated grids can be equivalent or in some cases more efficient in terms of cost then widely used ”Velocity-stress” schemes on staggered grid. Spatial derivative’s approximations are chosen to be based on Taylor series expansion as this formulation fits well our needs. To sum up:

Equation: Strong form of equation of motion in displacement formulation

Discretization: Finite differences in time domain (FDTD)

Time stepping: Implicit

Accuracy: $O\,(2, 2)$, $O\,(2, 4)$

Domain: 3D homogeneous isotropic media with constant velocity

Programming: C/C++ with PETSc 3.6.1

3D acoustic wave equation

Strong form of acoustic wave equation for homogeneous media in displacement formulation could be expressed as follows

$$\frac{\partial^2 u}{\partial t^2} - c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) = f \tag{1}$$

We discretize it with 2nd order of accuracy both in space and time, $O\,(2, 2)$. Implicit time stepping requires backward time derivative approximation, whereas spacial derivatives are approximated with centered scheme.

$$\begin{aligned} &\frac{2U^{n+1}_{i,j,k} - 5U^{n}_{i,j,k} + 4U^{n-1}_{i,j,k} - U^{n-2}_{i,j,k}}{\Delta t^2} = \\ &c^2_{i,j,k} \frac{U^{n+1}_{i+1,j,k} - 2U^{n+1}_{i,j,k} + U^{n+1}_{i-1,j,k}}{\Delta x^2} + \\ &c^2_{i,j,k} \frac{U^{n+1}_{i,j+1,k} - 2U^{n+1}_{i,j,k} + U^{n+1}_{i,j-1,k}}{\Delta y^2} + \\ &c^2_{i,j,k} \frac{U^{n+1}_{i,j,k+1} - 2U^{n+1}_{i,j,k} + U^{n+1}_{i,j,k-1}}{\Delta z^2} + f_{i,j,k} \end{aligned} \tag{2}$$

Hence,

$$\begin{gathered}U_{i,j,k}^{n+1}\left(2\Delta x\Delta y\Delta z+2c_{i,j,k}^{2}\Delta t^{2}\left(\frac{\Delta y\Delta z}{\Delta x}+\frac{\Delta x\Delta z}{\Delta y}+\frac{\Delta x\Delta y}{\Delta z}\right)\right)+\\ U_{i+1,j,k}^{n+1}\left(-c_{i+1,j,k}^{2}\Delta t^{2}\ \frac{\Delta y\Delta z}{\Delta x}\right)+\\ U_{i-1,j,k}^{n+1}\left(-c_{i-1,j,k}^{2}\Delta t^{2}\ \frac{\Delta y\Delta z}{\Delta x}\right)+\\ U_{i,j+1,k}^{n+1}\left(-c_{i,j+1,k}^{2}\Delta t^{2}\ \frac{\Delta x\Delta z}{\Delta y}\right)+\\ U_{i,j-1,k}^{n+1}\left(-c_{i,j-1,k}^{2}\Delta t^{2}\ \frac{\Delta x\Delta z}{\Delta y}\right)+\\ U_{i,j,k+1}^{n+1}\left(-c_{i,j,k+1}^{2}\Delta t^{2}\ \frac{\Delta x\Delta y}{\Delta z}\right)+\\ U_{i,j,k-1}^{n+1}\left(-c_{i,j,k-1}^{2}\Delta t^{2}\ \frac{\Delta x\Delta y}{\Delta z}\right)=\\ \Delta x\Delta y\Delta z\left(5U_{i,j,k}^{n}-4U_{i,j,k}^{n-1}+U_{i,j,k}^{n-2}+\Delta t^{2}f_{i,j,k}\right)\end{gathered}\tag{3}$$

At each time step we will solve system

$$Ax=b \tag{4}$$

where lhs. will consist of values from lhs. of eq. 3 rhs. will be rhs. of the same equation. As one could notice, matrix A is a 7 diagonal matrix (Fig. 1) that has to be inverted a teach time step in implicit methods. It worth noticing again that we are not going to invert this matrix directly, but do it using iterative Krylov methods that find $A^{-1}b$ approximation cheaper in terms of memory and which are well parallelizable. We won't provide any examples with $O(2,2)$ accuracy, as it has been used on early stages of the project.

Now let us do the same but for $O\left(2,4\right)$, and get as a result 13 diagonal matrix (Fig. 2).

$$\begin{gathered}\frac{2U_{i,j,k}^{n+1}-5U_{i,j,k}^{n}+4U_{i,j,k}^{n-1}-U_{i,j,k}^{n-2}}{\Delta t^{2}}=\\ c_{i,j,k}^{2}\frac{-U_{i+2,j,k}^{n+1}+16U_{i+1,j,k}^{n+1}-30U_{i,j,k}^{n+1}+16U_{i-1,j,k}^{n+1}-U_{i-2,j,k}^{n+1}}{12\Delta x^{2}}+\\ c_{i,j,k}^{2}\frac{-U_{i,j+2,k}^{n+1}+16U_{i,j+2,k}^{n+1}-30U_{i,j,k}^{n+1}+16U_{i,j-1,k}^{n+1}-U_{i,j-2,k}^{n+1}}{12\Delta y^{2}}+\\ c_{i,j,k}^{2}\frac{-U_{i,j,k+2}^{n+1}+16U_{i,j,k+1}^{n+1}-30U_{i,j,k}^{n+1}+16U_{i,j,k-1}^{n+1}-U_{i,j,k-2}^{n+1}}{12\Delta z^{2}}+\end{gathered}\tag{5}$$

$$
\begin{aligned}
&U^{n+1}_{i,j,k}\left(2\Delta x\Delta y\Delta z+30\ c^2_{i,j,k}\Delta t^2\left(\frac{\Delta y\Delta z}{12\Delta x}+\frac{\Delta x\Delta z}{12\Delta y}+\frac{\Delta x\Delta y}{12\Delta z}\right)\right)+\\
&U^{n+1}_{i+2,j,k}\left(c^2_{i+2,j,k}\Delta t^2\ \frac{\Delta y\Delta z}{12\Delta x}\right)+\\
&U^{n+1}_{i+1,j,k}\left(-16\ c^2_{i+1,j,k}\Delta t^2\ \frac{\Delta y\Delta z}{12\Delta x}\right)+\\
&U^{n+1}_{i-1,j,k}\left(-16\ c^2_{i-1,j,k}\Delta t^2\ \frac{\Delta y\Delta z}{12\Delta x}\right)+\\
&U^{n+1}_{i-2,j,k}\left(c^2_{i-2,j,k}\Delta t^2\ \frac{\Delta y\Delta z}{12\Delta x}\right)+\\
&U^{n+1}_{i,j+2,k}\left(c^2_{i,j+2,k}\Delta t^2\ \frac{\Delta x\Delta z}{12\Delta y}\right)+\\
&U^{n+1}_{i,j+1,k}\left(-16\ c^2_{i,j+1,k}\Delta t^2\ \frac{\Delta x\Delta z}{12\Delta y}\right)+\\
&U^{n+1}_{i,j-1,k}\left(-16\ c^2_{i,j-1,k}\Delta t^2\ \frac{\Delta x\Delta z}{12\Delta y}\right)+\\
&U^{n+1}_{i,j-2,k}\left(c^2_{i,j-2,k}\Delta t^2\ \frac{\Delta x\Delta z}{12\Delta y}\right)+\\
&U^{n+1}_{i,j,k+2}\left(c^2_{i,j,k+2}\Delta t^2\ \frac{\Delta x\Delta y}{12\Delta z}\right)+\\
&U^{n+1}_{i,j,k+1}\left(-16\ c^2_{i,j,k+1}\Delta t^2\ \frac{\Delta x\Delta y}{12\Delta z}\right)+\\
&U^{n+1}_{i,j,k-1}\left(-16\ c^2_{i,j,k-1}\Delta t^2\ \frac{\Delta x\Delta y}{12\Delta z}\right)+\\
&U^{n+1}_{i,j,k-2}\left(c^2_{i,j,k-2}\Delta t^2\ \frac{\Delta x\Delta y}{12\Delta z}\right)=\\
&\Delta x\Delta y\Delta z\left(5U^n_{i,j,k}-4U^{n-1}_{i,j,k}+U^{n-2}_{i,j,k}+\Delta t^2 f_{i,j,k}\right)
\end{aligned}
\tag{6}
$$

In case of homogeneous isotropic media, matrix A will be symmetric and positive definite, hence the most convenient choice of solver is Conjugate Gradient with Schwarz preconditionner and ilu as sub-preconditionner. After simple test Schwarz levels of overlap we found that the best performance was reached when it was 2. Also, multigrid could be used as preconditioner, but for some reasons in crashed in our simulations. Besides, one has to understand, that in case of non-homogeneous media matrix A will become non-symmetric and other solver such as GMREs will have to be used and it will lead to slower convergence.

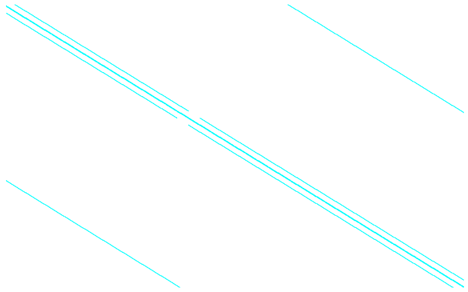

Figure 1: Zoomed 7 diagonal matrix for $O(2,2)$

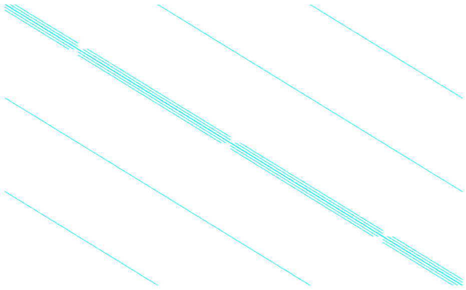

Figure 2: Zoomed 13 diagonal matrix for $O(2,4)$

Results and Significance

We use SOFI3D_acoustic as an explicit solver. It solves acoustic wave equation on staggered grid in velocity-stress formulation with FD approximation in time domain. The code is written with C and provides accuracy of $O(2,4)$. Simulation model: cube side 8km, propagation velocity $c = 3.5\ km/s$. Source: Ricker wavelet at center of the cube with dominant frequency $f_0 = 20\ Hz$. Time step is chosen such that for implicit method CFL number = 1, for explicit CFL number = 0.5. Resulting wavefield example is given on Figures 3-4.

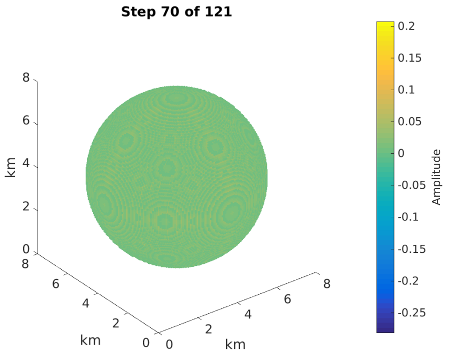

Figure 3: Snapsot of pressure wavefield at $NT = 70$

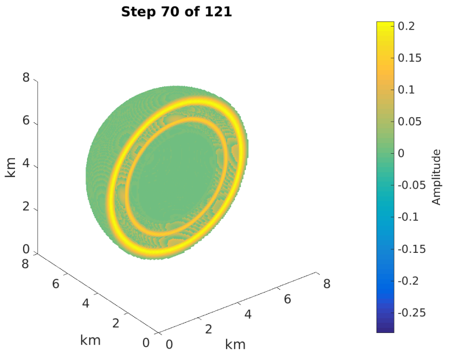

Figure 4: Snapsot of slice of pressure wavefield at $NT = 70$

Points per wavelength

The major benefit of implicit schemes is that one could take larger time steps and less number of grid points per wavelength. This advantages allow to model high frequencies more efficiently. Explicit methods bounded by CFL condition suffer from dispersion that occurs at high frequencies and at CFL numbers close to (Fig. 5), so refinement of mesh and smaller time steps are needed in order to overcome these dispersion artifacts (Fig. 6). Sufficient number of points per wavelength should be higher than 10. Whereas implicit method works well with CFL number equal to 1 and larger, so in general 5 points per wavelength is sufficient for a stable simulation (Fig. 7). Even with 2.5 point per wavelength in dispersion artifacts still do not occur (Fig. 8), but the major drawback in this case is that probably because of too small number of time steps, high-frequency initial wavelet is discretized just by few points that leads to loss of accuracy in total.

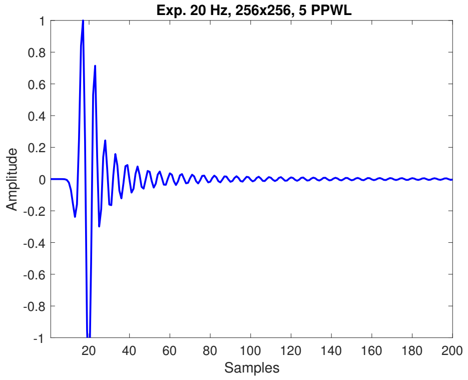

Figure 5: Normalized seismogram. Explicit method, 20Hz, grid 256^3, 5 points per wavelength

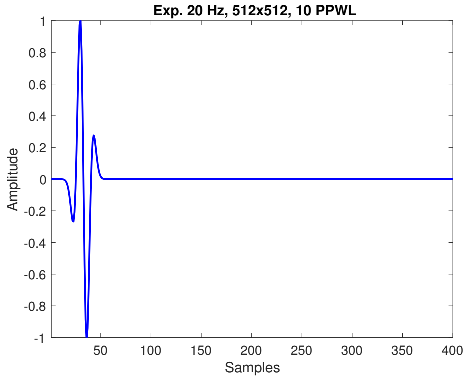

Figure 6: Normalized seismogram. Explicit method, 20Hz, grid 512^3, 10 points per wavelength

So, in general, implicit methods could find application in cases when high frequencies have to be modeled, as explicit methods perform very well, but computational difficulty grows quadratically with higher frequencies.

Strong scaling

Here and after we will consider two times coarser mesh and two times larger time step for implicit methods as an equivalent to explicit methods. This approximation came from observation that explicit method becomes stable at CFL numbers of about 0.5, whereas implicit methods work well at 1. All strong scaling tests were performed from 128 to 16384 cores.

Figure 9 shows strong scaling study of implicit method solving second order wave equation in displacement formulation in cube whose side contains 512 grid points with its 1024 grid points equivalent for explicit version. As

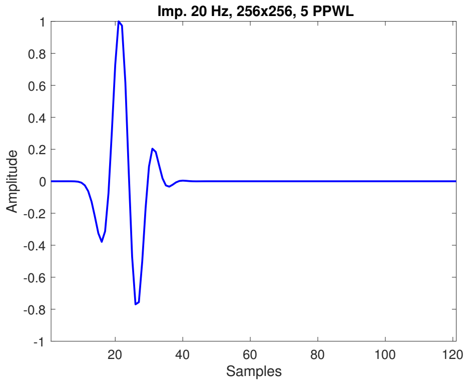

Figure 7: Normalized seismogram. Implicit method, 20Hz, grid 256^3, 5 points per wavelength

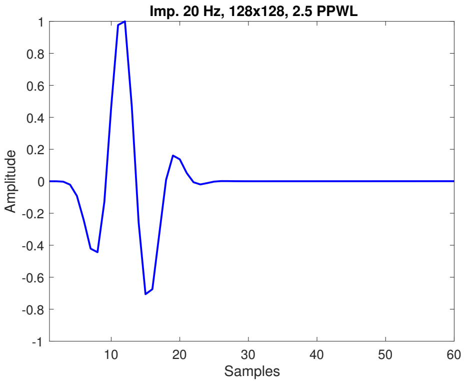

Figure 8: Normalized seismogram. Implicit method, 20Hz, grid 128^3, 2.5 points per wavelength

one could notice, explicit code scales better having suspicious jump at 8192 cores, scaling comparison in the same axis is given on Figure 10. The same comparison but on refined grids is shown on Figure 11, where grid for implicit method has 1024^3 points and for an explicit one has 2048^3 nodes. From there we see that explicit code scales almost linearly whereas the implicit one slows down at 8192 cores and higher. Most probably this slow down of implicit method caused by rise of MPI communication messages and drop of Flop/sec (Fig. 13-14), but for both approaches the larger problem we take the better it scales (Fig. 11-12).

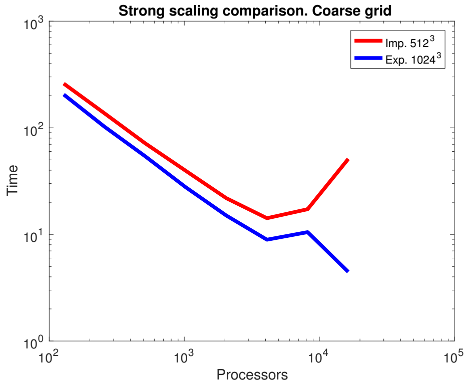

Figure 9: Strong scalability study on coarse grid - imp. 512^3, exp. 1024^3. Implicit(red) *vs* Explicit(blue) approaches.

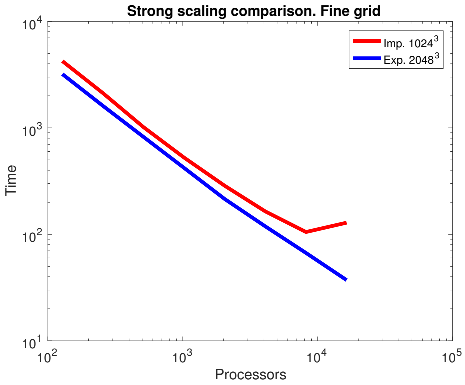

Figure 10: Strong scalability study on fine grid - imp. 1024^3, exp. 2048^3. Implicit(red) *vs* Explicit(blue) approaches.

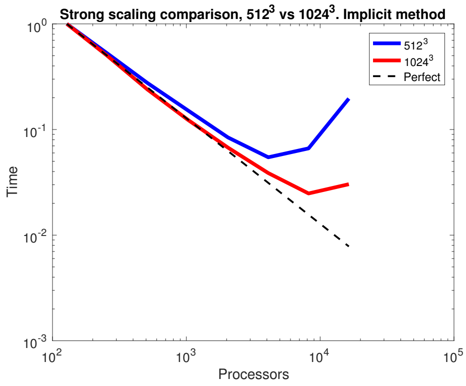

Figure 11: Implicit method. Problem size-dependent strong scaling. Coarse grid, 512^3 (blue) *vs* Fine grid, 1024^3 (red)

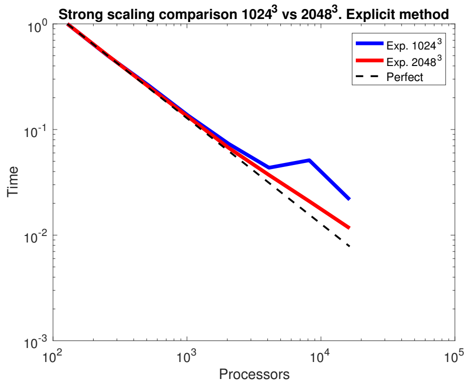

Figure 12: Explicit method. Problem size-dependent strong scaling. Coarse grid, 1024^3 (blue) *vs* Fine grid, 2048^3 (red)

Also it is worth mentioning, that despite conventional explicit operators were used in implicit method, Figures 9-10 show that implicit and explicit performances are pretty close one to another what gives us hope that with proper implementation implicit methods will be able to use its full potential.

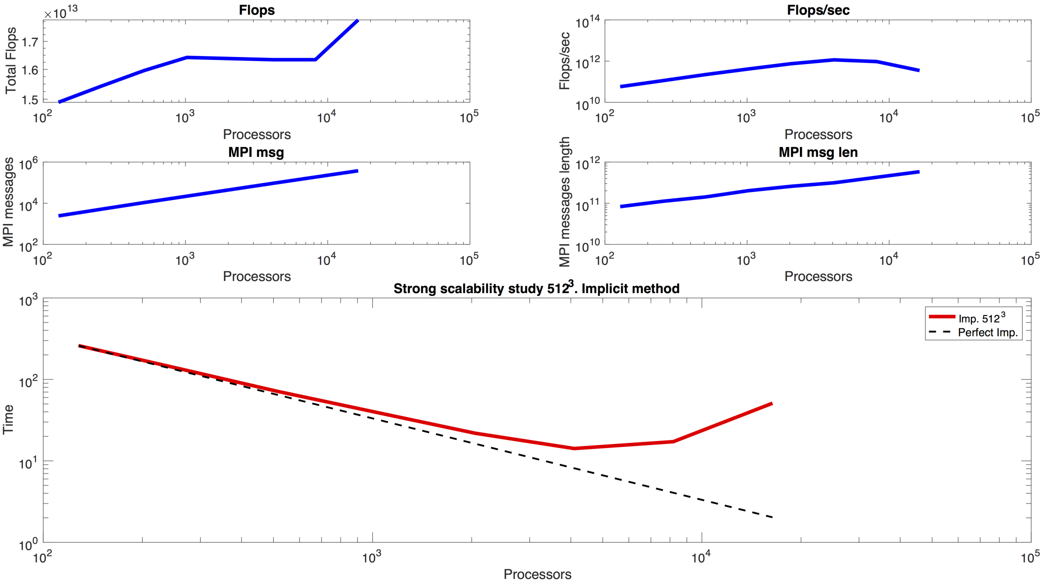

Figure 13: Implicit method. Strong scalability study on coarse grid 512^3, Total Flops, Flops/sec, count of MPI messages, Length of MPI messages

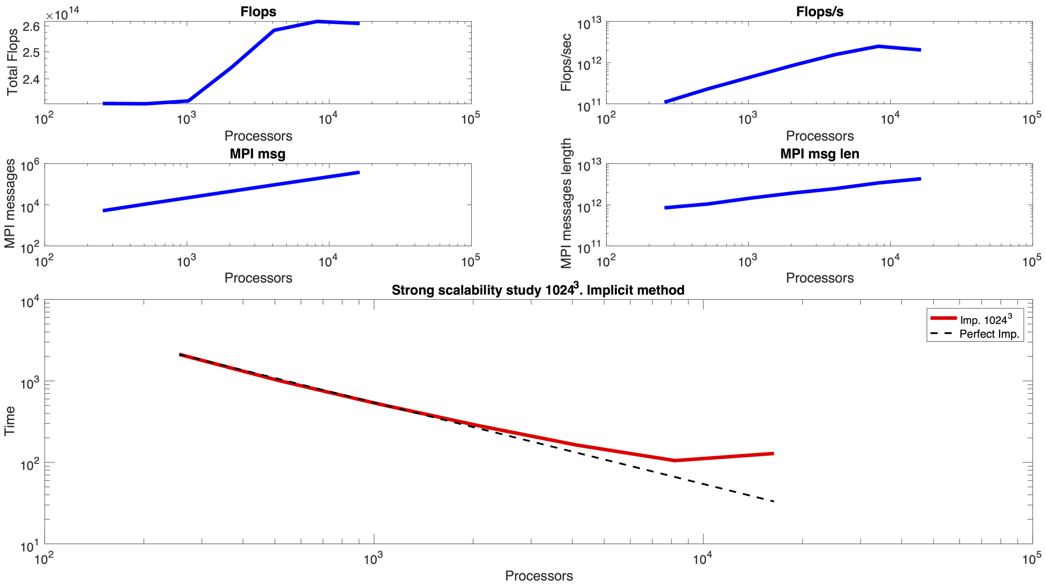

Figure 14: Implicit method. Strong scalability study on fine grid 1024^3, Total Flops, Flops/sec, count of MPI messages, Length of MPI messages

Conclusion

In this project we explored scalability of an implicit finite difference code in time domain that solves acoustic wave equation in homogeneous isotropic 3D media. The motivation of the study was to have have a look at performance of methods given in PETSc solving a hyperbolic problem and compare results with it's explicit analog.

Results obtained in the study showed, that in order to compete explicit methods, better numerical schemes have to be used in implicit methods, as the conventional explicit schemes derived from Taylor series do not provide desirable benefits for implicit approach – special implicit finite-difference operators have to be used instead [Chu et al., 2010]. Moreover, Chu and Stoffa [2012] showed in their study that high-order implicit methods may be replaced by implicit implementations of the same order resulting in much improved performance, what they meant is that accuracy of 6th order implicit schemes is equivalent to that of 10th order of explicit scheme, implicit stencils are shorter and time step could be larger. So, implicit methods could perform better than the explicit approaches when problem has to be solved with high accuracy or for high frequencies (in wave propagation), when fine grid and small time step usually are required. It is worth noticing again, that for implicit method special implicit operators such as from Chu and Stoffa [2012] has to be used instead of the explicit ones. Besides, Conjugate Gradient solver was used in this study as the most applicable one for the given framework. Performance results obtained with other methods could be worse.

In its turn, SOFI3D_acoustic works and scales really well, almost linearly on log-log scale. The drawback is that it gets unstable at CFL number > 0.6.

Lessons Learned and Future Work

This project allowed to implement most of skills and knowledge learned during AMCS312 course:

1. We knew basics of principal algorithms, preconditionners, computational workflows, history and etc.

2. We knew how to use Shaheen II and run programs in supercomputer environment.

3. We got familiar with the PETSc, that is the core of the project's code, as it contains realizations of Krylov methods and all parallel-related routines

4. Other minor improves: bash scripting, Visual Studio code use, C syntax

Further steps should lead to implementation of proper implicit spatial derivative schemes, as it is crucial to make the approach competitive. Another step could be to try to use SNES for time stepping, as at current implementation stepping over time is implemented with for loop.

Bibliography

Chunlei Chu and Paul L Stoffa. Implicit finite-difference simulations of seismic wave propagation. *Geophysics*, 77(2):T57–T67, 2012.

Chunlei Chu, Paul L Stoffa, et al. Frequency domain modeling using implicit spatial finite difference operators. In *2010 SEG Annual Meeting*. Society of Exploration Geophysicists, 2010.

Richard Courant, Kurt Friedrichs, and Hans Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische annalen*, 100(1):32–74, 1928.

Robert J Geller, Nobuyasu Hirabayashi, and Hiromitsu Mizutani. A comparison of staggered-grid, conventional one-step, and optimally accurate finite-difference schemes for computing synthetic seismograms. *Geophysics*, 2005.

Dimitri Komatitsch and Jeroen Tromp. Introduction to the spectral element method for three-dimensional seismic wave propagation. *Geophysical journal international*, 139(3):806–822, 1999.

Dan Kosloff, Reynam C Pestana, and Hillel Tal-Ezer. Acoustic and elastic numerical wave simulations by recursive spatial derivative operators. *Geophysics*, 75(6):T167–T174, 2010.

Dan D Kosloff and Edip Baysal. Forward modeling by a fourier method. *Geophysics*, 47(10):1402–1412, 1982.

Yang Liu and Mrinal K Sen. A practical implicit finite-difference method: Examples from seismic modelling. *Journal of Geophysics and Engineering*, 6(3):231, 2009.

Peter Moczo, Johan OA Robertsson, and Leo Eisner. The finite-difference time-domain method for modeling of seismic wave propagation. *Advances in Geophysics*, 48:421–516, 2007.

Erik H Saenger, Norbert Gold, and Serge A Shapiro. Modeling the propagation of elastic waves using a modified finite-difference grid. *Wave motion*, 31(1):77–92, 2000.

Gilbert Strang and George J Fix. *An analysis of the finite element method*, volume 212. Prentice-hall Englewood Cliffs, NJ, 1973.

Jean Virieux. Sh-wave propagation in heterogeneous media; velocity-stress finite-difference method. *Geophysics*, 49(11):1933 – 1942, 1984. ISSN 00168033.

Jean Virieux. P-sv wave propagation in heterogeneous media: Velocity-stress finite-difference method. *Geophysics*, 51(4):889–901, 1986.