Fast, implicit finite difference solver for the wave equation

Michael P. Lamoureux[1] and RJ Vestrum[2]
University of Calgary[1] and 3vGeomatics[2]

Summary

We demonstrate here a fast, stable **implicit** time-stepping method for numerically solving the wave equation in two and three dimensions that makes use of novel operator factorization in a grid algebra. The technique is demonstrated with a numerical examples in three dimensions, on computation grids of size up to several million grid points.

Numerical solutions to the wave equations that arise in seismic imaging have been richly developed over the last 70+ years, and typically depend on explicit time-stepping or Fourier techniques for fast, accurate solutions. The counterpart to explicit methods are **implicit** methods which enjoy features such as unconditional stability even with large time steps. Unfortunately such methods are typically computationally prohibitive in two and three spatial dimensions. The grid algebra technique presented here reduces the complexity to level of the familiar explicit methods while retaining high accuracy and stability.

introduction

Finding efficient numerical methods for solving systems of differential equations that describe physical processes is a central challenge in scientific computation [Press 2009, Selvadurai 2000, Myint-U 2006]. In seismic imaging, one key step is computing numerical solutions for the wave equation that models seismic wave propagation through the earth's subsurface. Several numerical methods for computing wave propagation involve representing the wavefield on a grid of nodes or cells in space and modelling its propagation using various finite difference approximations to the wave equation.

There is considerable interest in developing **implicit** solutions for finite-difference methods of modelling wave propagation, in order to make use of the inherent advantages of implicit methods, such as unconditional stability and accuracy. Some early work on implicit methods for multidimensional PDEs goes back to [Stone 1968]; more recent developments focus on such things as the alternating direction implicit (ADI) schemes as seen in [Qin 2009]. A central challenge in the implicit method is that, somewhere along the line, a system of linear equations has to be solved numerically. Here, the high dimensions are a particular problem and even general sparse matrix methods only partially alleviate the challenge.

In one dimension Claerbout demonstrated that implicit method are cheaper than other finite difference methods since the increased accuracy allows the use of larger time steps. However for space dimensions higher than one, the implicit method in standard form becomes prohibitively costly [Claerbout 1985]. On the plus side, often implicit methods are unconditionally stable. [Smith and Smither 1985]. For a finite-difference operator expressed as a sparse matrix with the non-zero values falling on a few diagonals, standard LU decomposition 'fills in' a very large number of zeros between the diagonals [Smith and Smither 1985]. The same issue arises with Cholesky factorization. There has long been hope for an approach that can avoid this complication: “It seems fortunate that the table contains many zeroes, and we are led to hope for a rapid solution method for the simultaneous equations." [Claerbout 1985]

We show here that we have obtained this “rapid solution method.” The focus of our research is to find factorizations similar to LU decomposition of a linear system, that preserve the local structure of the grid. We note the underlying structure in the linear systems based on an organized grid of points in 2D or 3D gives additional geometric relationships between coefficients in the linear operator. The general sparse matrix techniques do not readily take advantage of these geometric relationships, so there is an opportunity to develop algorithmic efficiencies from this extra information. As an alternative to the sparse matrix methods, we aim to use the underlying geometric structure to perform “matrix operations” directly on the structured linear system to produce more efficient linear algebraic methods.

We sometimes call such an approach the **grid algebra method**, as discussed in our earlier work [Lamoureux, Hardeman-Vooys 2016]. We have used this algebraic method to find a nearest neighbour factorization of the Laplacian operator in 2D and 3D (and higher) that arises in the acoustic wave equation, in a form suitable for the rapid computation of an implicit finite difference scheme for time stepping the wave equation. The result is an implicit algorithm that has computational complexity of order O(N), where N is the number of points in the computational grid. This gives a method with all the advantages of the implicit schemes, while maintaining the computational efficiently of explicit schemes.

The derivation of the algorithm using grid algebra is contained in the MSc thesis of the second author [Vestrum 2021].

Method

Implicit methods are often computationally slow because at some point, a large linear system of equations needs to be solved, or a large matrix inverted. For the finite difference method used to solve the wave equation in only 1D, the Laplacian matrix is tridiagonal, and can be factored in the Cholesky form FF*, where each factor is bi-diagonal and easily inverted via back substitution. This accounts for the rapid implicit method in 1D.

In 2 dimensions, we observe that a nearest neighbour operator $v = Fu$ of the form

$$v_{ij} = au_{ij} + bu_{i-1,j} + cu_{i,j-1} + du_{i-1,j-1}$$

and its adjoint operator can serve as factors for the shifted 2D Laplacian on a 9-point stencil, for a judicious choice of parameters a, b, c, d. The computational speed-up occurs because this operator can be inverted quickly by back-substitution on the grid, as we can re-arrange the previous equation to solve for u as:

$$u_{ij} = (v_{ij} - bu_{i-1,j} - cu_{i,j-1} - du_{i-1,j-1})/a$$

Taking into account a rectangular grid (different spacings in the x and y directions), the right choice for parameters is

$$a = \frac{\left(1 + \sqrt{1 + 4\epsilon\lambda_x}\right)\left(1 + \sqrt{1 + 4\epsilon\lambda_y}\right)}{4}$$
$$b = \frac{\left(1 - \sqrt{1 + 4\epsilon\lambda_x}\right)\left(1 + \sqrt{1 + 4\epsilon\lambda_y}\right)}{4}$$
$$c = \frac{\left(1 + \sqrt{1 + 4\epsilon\lambda_x}\right)\left(1 - \sqrt{1 + 4\epsilon\lambda_y}\right)}{4}$$
$$d = \frac{\left(1 - \sqrt{1 + 4\epsilon\lambda_x}\right)\left(1 - \sqrt{1 + 4\epsilon\lambda_y}\right)}{4}$$

Here, the constants λ_x, λ_y relate the relative grid spacing in the x,y directions respectively, and $\epsilon = c^2 \frac{\Delta t^2}{\Delta h^2}$ is the CFL parameter for finite differences that appears in the shifted Laplacian $1 - \epsilon L$.

A similar factorization occurs in 3D, also for rectangular grids (i.e. different grid spacing in different directions) with 8 parameters required and a 27 point stencil for the shifted Laplacian obtained. In both the 2D and 3D case, the order of operations is O(N) per time step, where N is the number of points on the computational grid. This is the same order as for the usual explicit methods.

Numerical examples

Figures 1 and 2 below show various time slices for a numerical simulation with this method, illustrating one or two initial Gaussian pulses propagating through a homogeneous medium.

Figure 1 shows four time slices, on three different computational grids of size 151x151x151, 151x85x151 and 151x76x151. These have the number of grid points N ranging from 1.7 to 3.5 million, and verify that the order of numerical operations per time step is indeed O(N), just as in the explicit FD methods. Note that even though the grid cells are non-square in two of the grids, the wave continues to propagate with a uniform circular wavefront, as in the square grid case. In the right-most column, we do see some numerical dispersion in the last time step, which could be a result in a loss of accuracy as the grid spacing in the y dimension becomes large.

Figure 2 shows six time slices on a non-rectangular grid of size 101x101x51 and illustrates the interaction between two wavefronts propagating in a homogenous medium. The polarity of the wavefronts and their interaction (destructive interference) at the front-intersections behave as expected, demonstrating some accuracy in the physics of the modelling, as expected. There is little evidence of dispersion, even in this non-rectangular grid.

The numerical examples were computed in Python 3 in a Jupyter notebook, in a single thread, running on a reasonable fast Intel laptop. The code is available on the Github repository of the second author.

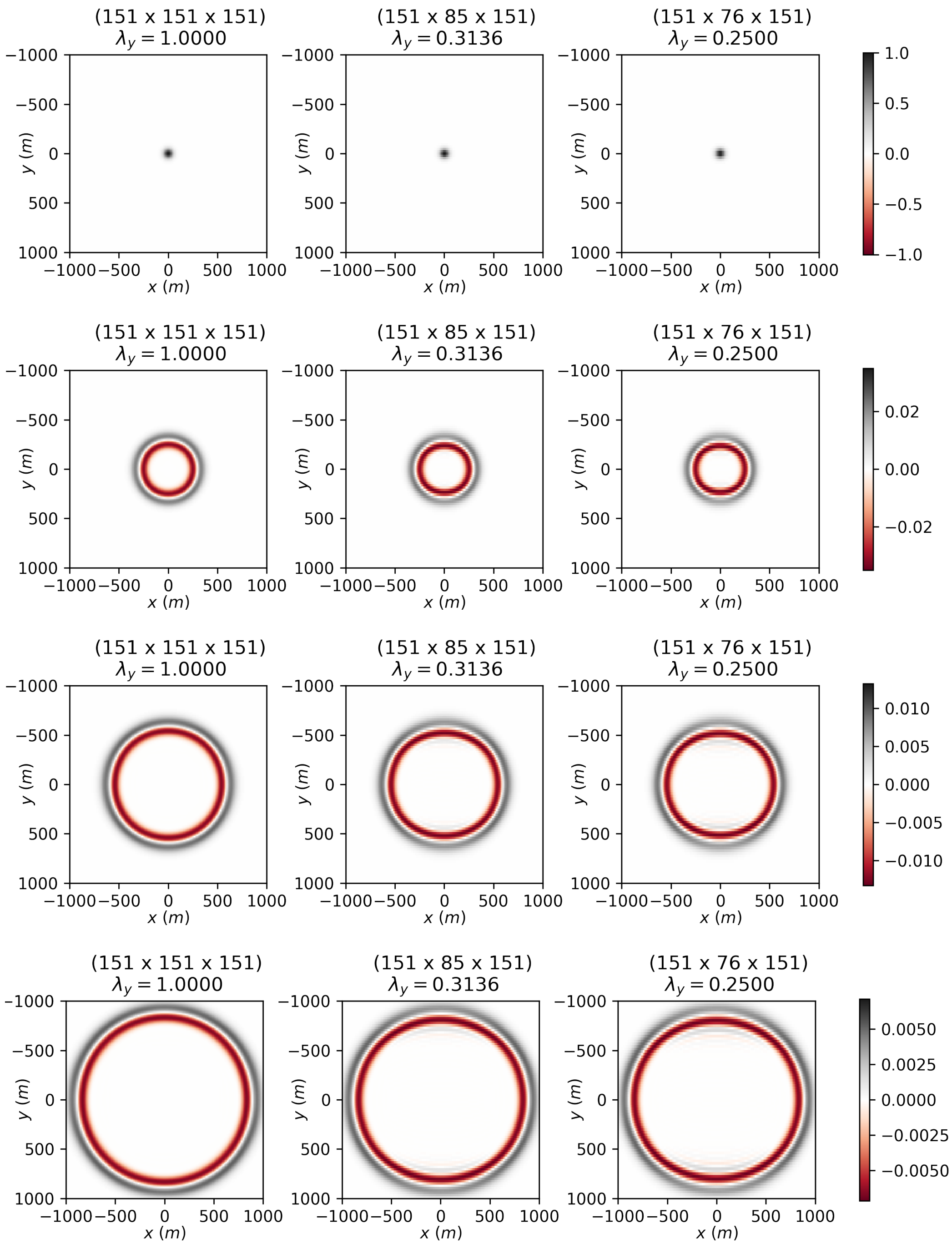

Figure 1. 3D Wave propagation on non-square grids, with up to 3.5 million grid points.

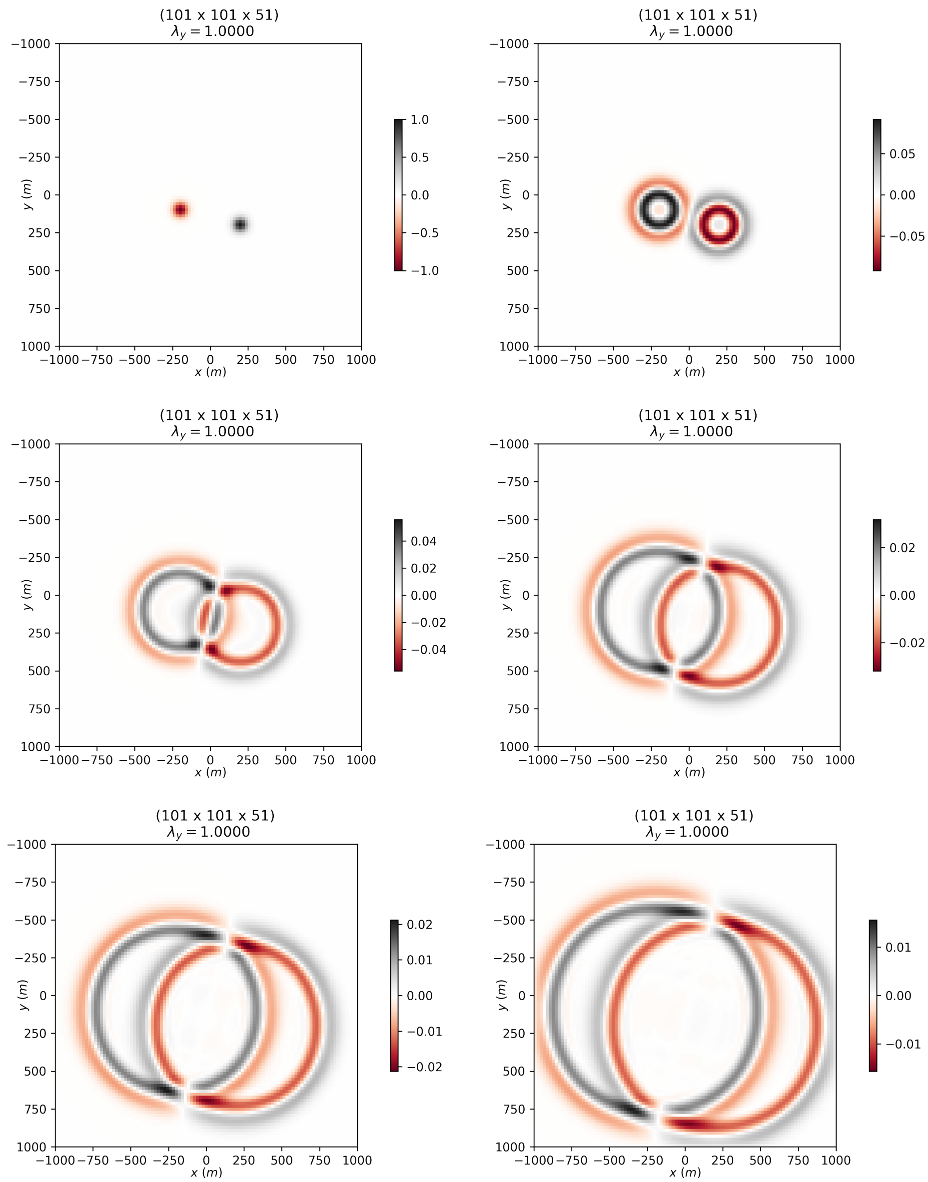

Figure 2. Two 3D Interacting wavefronts on non-square grids, with 500,000 grid points.

Additive Information

There are further developments in this method. There are formulas for factorizations on triangular girds, and we have developed parallel implementations that are suitable for GPU processing. We continue work on implementing the grid factorization for variable velocity cases, higher order Laplacians, and handle general boundary conditions.

Acknowledgements

This abstract is a summary of the main results described in the MSc thesis of the second author [Vestrum 2021]. It builds on previous work on grid algebras by the first author and his student [Lamoureux, Hardeman-Vooys 2016].

The research was funded by CREWES industrial sponsors and NSERC (Natural Science and Engineering Research Council of Canada) through the grant CRDPJ 543578-19 and the Discovery Grant RGPIN- 2020-04561 of the first author.

References

Claerbout, J. F., 1985, Imaging the earth's interior, vol. 1: Blackwell scientific publications Oxford.

Folland,G.B., 2020, Introduction to Partial Differential Equations: Second Edition: Princeton University Press.

Lamoureux, M.P. and Hardeman-Vooys, H.K., 2016, Grid algebra in finite difference schemes: CREWES Research Report, 28.

Lamoureux, M.P. and Vestrum, R.J., 2016, Factoring the wave equation for fast, implicit numerical solutions: CREWES Research Report, 33.

Myint-U., T., and Debnath, L., 2006, Linear Partial Differential Equations for Scientists and Engineers: Birkhäuser Boston.

Press, W., 2009, Numerical Recipes in C++: The Art of Scientific Computing: Cambridge University Press.

Qin, J., 2009, The new alternating direction implicit difference methods for the wave equations: Journal of Computational and Applied Mathematics, 230, No. 1, 213–223.

Selvadurai, A., 2000, Partial Differential Equations in Mechanics 1: Fundamentals, Laplace's Equation, Diffusion Equation, Wave Equation, Engineering online library: Springer.

Smith, G. and Smither, M., 1985, Numerical Solution of Partial Differential Equations: Finite-difference Methods, Oxford applied mathematics and computing science series: Clarendon Press.

Stone, H. L., 1968, Iterative solution of implicit approximations of multidimensional partial differential equations: SIAM Journal on Numerical Analysis, 5, No. 3, 530–558.

Vestrum, R., 2021, Local Factorization of Multidimensional Differential Operators to Optimize Implicit Solution Methods: M.Sc. thesis, University of Calgary, Calgary, AB, Canada.