Computers and Geosciences 116 (2018) 23–31

Contents lists available at ScienceDirect

Computers and Geosciences

journal homepage: www.elsevier.com/locate/cageo

An algorithm for fast elastic wave simulation using a vectorized finite difference operator☆,☆☆

Ajay Malkoti[a,*], Nimisha Vedanti[a,b], Ram Krishna Tiwari[a,b]

[a] AcSIR-NGRI, Hyderabad, 500007, India
[b] CSIR-National Geophysical Research Institute, Hyderabad, 500007, India

ARTICLE INFO

Keywords:
Seismic
Elastic wave
Vectorization
Staggered grid

ABSTRACT

Modern geophysical imaging techniques exploit the full wavefield information which can be simulated numerically. These numerical simulations are computationally expensive due to several factors, such as a large number of time steps and nodes, big size of the derivative stencil and huge model size. Besides these constraints, it is also important to reformulate the numerical derivative operator for improved efficiency. In this paper, we have introduced a vectorized derivative operator over the staggered grid with shifted coordinate systems. The operator increases the efficiency of simulation by exploiting the fact that each variable can be represented in the form of a matrix. This operator allows updating all nodes of a variable defined on the staggered grid, in a manner similar to the collocated grid scheme and thereby reducing the computational run-time considerably. Here we demonstrate an application of this operator to simulate the seismic wave propagation in elastic media (Marmousi model), by discretizing the equations on a staggered grid. We have compared the performance of this operator on three programming languages, which reveals that it can increase the execution speed by a factor of at least 2–3 times for FORTRAN and MATLAB; and nearly 100 times for Python. We have further carried out various tests in MATLAB to analyze the effect of model size and the number of time steps on total simulation run-time. We find that there is an additional, though small, computational overhead for each step and it depends on total number of time steps used in the simulation. A MATLAB code package, 'FDwave', for the proposed simulation scheme is available upon request.

1. Introduction

The earth is a heterogeneous, anisotropic and attenuating medium which designates it as a complex model. The interaction of seismic waves with various structures is a complicated process due to inherent heterogeneity in material distribution. Wave phenomena associated with the media properties, shape or location of the layer, thus requires better understanding. Few such examples include-no reflections in a salt diaper, occurrence of low velocity layers, fluid-solid interfaces, free surface, pinching out bed, etc. One way to study the interaction of a wave with the earth is to simulate full seismic wavefield. It would enable us to analyze the wave behavior at all time instances and its response at the surface, as recorded by the receivers. The simulation also has several other significant applications, for example, finding the subsurface structure (Olsen et al., 1995; Yomogida and Etgen, 1993) and optimizing survey parameters and geometries (Moldoveanu and Egan, 2007; Regone, 2007). In fact, seismic simulation is considered as an integral part of seismic interpretation (Ibrahim, 2005) and quantification of results from seismic data (Sayers and Chopra, 2009).

Simulation of the seismic wavefield in earth is a large scale problem and thus requires a considerable memory along with high computational power. Memory demand for a seismic simulation can be mitigated using techniques described in earlier studies, viz. Graves (1996) and Etgen and O'Brien (2007). The number of computations (number of arithmetic operations) can be reduced by using highly accurate stencils with, smaller length for derivative calculation (Holberg, 1987; Lele, 1992; Tam and Webb, 1993; Liu and Sen, 2009). Smaller stencil uses lesser number of points and thereby reduces the computational cost.

☆ The code is available on request from author.
☆☆ AM conceived the idea of vectorization, developed its theoretical framework, implemented it as code and drafted the manuscript. NV and RKT have provided critical feedbacks, supervised the project and helped in manuscript writing. All authors read and approved the final manuscript.

* Corresponding author.
E-mail addresses: ajmalkoti@gmail.com (A. Malkoti), nimisha@ngri.res.in (N. Vedanti), rktiwari54@gmail.com (R.K. Tiwari).

https://doi.org/10.1016/j.cageo.2018.04.002
Received 30 May 2017; Received in revised form 12 March 2018; Accepted 2 April 2018
Available online 5 April 2018
0098-3004/

Table 1
List of codes developed by various authors for addressing the wave propagation problems in various media using different techniques.

Language	Rheology	Forward/ Inverse	Author (Program Name)
Parallelization method: MPI			
C	Viscoelastic	FWD	Bohlen (2002) (fdmpi/sofi2D)
C	Elastic	FWI	Köhn et al. (2014) (Denise)
Fortran 2003	Viscoelastic	FWD	Maeda et al. (2017) (OPENSWPC)
C	Poroelastic	FWD	Sheen et al. (2006)
Parallelization method: GPU			
CUDA	Elastic	FWD	Micha and Komatitsch (2010)
CUDA	Elastic	FWD	Okamoto et al. (2010)
CUDA	Acoustic	FWI	Yang et al. (2015)
OpenCL	Viscoelastic	FWI	Fabien-Ouellet et al. (2017) (SeisCL)
Parallelization method: Multiple GPU			
OpenMP + CUDA	Anisotropic elastic	FWD	Weiss and Shragge (2013)
MPI + CUDA	Elastic	FWD	Zhou et al. (2013), Roten et al. (2016)
OpenGL	Elastic	FWD	Rubio et al. (2014)

The computational cost can further be reduced by exploiting the parallelism in computing which can be achieved using mainly two approaches. The first approach is to make many processors work simultaneously so that each processor shares a piece of work. To achieve this, one can use OpenMP or MPI depending upon the memory architecture (shared/distributed). It is also possible to take advantage of thousands of cores available in GPU using CUDA, OpenCL, OpenGL, etc. Few references, where seismic simulation was carried out using such techniques, are listed in Table 1. The second approach is to make each processor deliver more, for which data streaming to processor must be tuned finely. It can be achieved in following ways- (1) Instruction Level Parallelism (ILP), (2) Cache friendly code, (3) Fast algorithms or optimized numerical libraries, and (4) Single-instruction-multiple-data (SIMD) style instruction. ILP technique makes use of efficient sequencing before executing instructions and thus reduces the latency. The cache memory in CPU also plays a major role in determining the execution speed since high cache miss rate can cause high latency. It can be reduced by exploiting the spatial and temporal locality of data in the memory. For example, if matrices are accessed along the major axis, the memory access is faster due to low cache miss rate. Numerical libraries (e.g., BLAS, OPENBLAS, LINPACK, ATLAS, Intel Math Kernel Library, AMD Core Math Library, etc.) reduce the computation run-time by using efficient computational algorithms. For example, Coppersmith-Winograd algorithm is a fast algorithm for matrix multiplication. In SIMD approach (a.k.a. vectorization) a single machine instruction is automatically applied to all the arguments of the same type (a vector) by taking advantage of vector processors present inside each CPU.

Several researchers have studied the vectorization of the wave equation. For example, Vafidis et al. (1992) presented an algorithm based on the 'matrix times vector by diagonals' technique. There are other techniques which modify the pattern to access the data according to cache size for lesser cache miss rate. It involves loop-rearrangement, loop-blocking, and block-striding (Borges and Thierry, 2011; Zhou and Symes, 2014; Titarenko and Hildyard, 2017).

The seismic wave simulation involves BLAS level 1 type operations (e.g., scalar-matrix multiplication, αA; matrix-matrix addition, $A+B$). For such operations no special algorithms are required but SIMD plays a very important role in determining the efficiency of the computations. Programming languages such as FORTRAN, Python and MATLAB can perform better for whole-array operations than serial execution. Executing commands utilizing whole array operations are faster because such operations can be easily vectorized by the compilers. Thus, to achieve computational efficiency and speed, the numerical scheme should be written in such a way that it can take advantage of whole array operations or in other words, the code could be vectorized.

In this paper, we present a methodology to speed up the simulation for the elastic wave propagation in a given medium, by vectorizing the derivatives over the staggered grid. A stress-velocity formulation (Levander, 1988; Virieux, 1986) is solved over staggered grid using finite difference scheme. Problem with such a grid is that the variables cannot be updated using whole matrix operation of derivatives since field variables (i.e., velocities and stresses) are not defined at the same location in the grid. For example, to update the values of σ_{xz}, defined at center in a staggered grid, we need v_x and v_z which are defined at the horizontal edge and the vertical edge of staggered grid, respectively. To overcome this problem, we derived a new finite difference operator by decomposing the original staggered grid into a simpler shifted-grids, which can be easily extended to solve an n-dimensional problem. This method can significantly reduce the simulation run-time.

To show the reduction in computational cost, we have compared simulation run-time of the vectorized derivative operator with the conventional approach (using simple loops) at different platforms viz. FORTRAN, Python, and MATLAB, for different sizes of models and time steps. Further, we carried out a complete seismic simulation using MATLAB for Marmousi model and estimated the simulation run-time for various sizes of the model and time. Our findings indicate that the higher number of time steps can cause a little more computational overhead than the expected simulation run-time. The vectorization strategy helps in avoiding loops and renders the code in a well organized and succinct form which leads to an easy debugging and thus helps in quicker verification of new finite difference schemes.

2. Theory

2.1. Elastic wave formulation

The propagation of an elastic wave in the earth can be described using the momentum conservation equation and the Hook's law (Aki and Richards, 2002). The governing equation (momentum conservation equation) provides a relation between velocities (v_i) and the stresses (σ_{ij}), which can be written as

$$\rho\frac{\partial v_i}{\partial t} = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho f_i \tag{1}$$

where ρ is density of the medium and f_i is the external body force in the i direction and $i,j \in \{x,y\}$. It can be used for estimating the particle velocities in the medium at the given instance $v_i(x,z,t)$. For Eq. (1), the constitutive law is Hook's law. It gives a relationship between the stress (σ_{ij}) and strain ($\varepsilon_{i,j}$), which can be expressed as

$$\frac{\partial \sigma_{ij}}{\partial t} = \lambda\dot{\varepsilon}_{k,k}\delta_{ij} + \mu\left(\dot{\varepsilon}_{i,j} + \dot{\varepsilon}_{j,i}\right) \tag{2}$$

where, λ and μ are the Lame's parameter and the dot over the quantity (i.e. $\dot{\varepsilon}$) implies the time derivative of the quantity. Hence $\dot{\varepsilon}$ represent the strain rate, which can be defined as

$$\dot{\varepsilon}_{i,j} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \tag{3}$$

Using strain rate definition (Eq. (3)), the constitutive relation (Eq. (2)) can be transformed into a stress-velocity relationship. Hence, Eqs. (1) and (2) together form a coupled partial differential equation, known as the velocity-stress formulation. In our study, this equation will be used for simulation of seismic wave propagation in the earth assuming it as an elastic medium.