We G102 06

A Review of Recent Forward Problem Developments Used for Frequency-domain FWI

B. Pajot* (ISTerre, Université Grenoble Alpes), Y. Li (Harbin Inst. Tech. & ISTerre, Univ. Grenoble Alpes), V. Berthoumieux (Intel Corp. SAS), C. Weisbecker (Livermore Software Technology Corp.), R. Brossier (ISTerre, Université Grenoble Alpes), L. Métivier (LJK, CNRS, Université Grenoble Alpes), P. Thierry (Intel Corp. SAS), S. Operto (Géoazur, CNRS, Université Nice Sophia Antipolis) & J. Virieux (ISTerre, Université Grenoble Alpes)

SUMMARY

For efficient frequency-domain full wave inversion (FWI), frequency-domain solutions of the wave-equation are required. The wave equation can be solved by direct or iterative solvers in the frequency domain or, alternatively, one can also use time domain integration until the stationary state has been reached. These three strategies have the same time complexitities. Increasing the order of the finite difference stencil may reduce the memory requirements and should optimize the use of available floating point operations per second (flops) on modern computer platforms. Known drawbacks of direct solvers, as poor scalability and significant in-core memory requirement, could be overcome by recent advances such as block low-rank (BLR) approximation. Finally some new algorithm designs for iterative solvers have shown interesting scalability and good convergence properties in heterogeneous media. We illustrate that a parallel conjugate gradient (CARP-CG) is robust and scalable for 2D elastic equations, extendable to 3D geometries. These recent breakthroughs could be integrated in the frequency-domain FWI workflows for real applications.

Introduction

Full Waveform Inversion (FWI) is a seismic imaging method based on local optimization techniques used to fit the seismic traces recorded for each shot (see the review Virieux and Operto (2009) for instance). The misfit function can be expressed either in the time or frequency domains. Time-domain has the advantage to allow selection of progressive time windows on data. If acquisition is sufficiently dense, frequency-domain approach allows to reduce the data complexity by selecting only few frequencies needed to sample the required model wavenumber spectrum. Both approaches are worth the attention as they may provide complementary strategies. In this study, we will focus on the frequency-domain FWI strategy.

When considering frequency-domain FWI, one requires the frequency-domain solutions of the wave-equation. One can consider directly the frequency-domain wave-equation, leading for each frequency and each seismic source to solve a linear system. This resolution can be tackled either by direct solver (Operto et al., 2007) or by iterative solver (Plessix, 2009). One can also use time-domain wave-equation reaching the stationary state and Fourier transformed (Sirgue et al., 2007).

In this study, the three approaches will be reviewed and discussed with regard on recent developments, and combined with high performance computing implications.

Time and frequency-domains wave-equation

It is known that a large part of FWI computation is related to the forward problem step. Considering the general 3D elastic partial differential equations that represent wave propagation, we need to consider the particle velocity vector v_i and symmetrical stress tensor σ_{ij} at any point of the medium embedded into a general vector $\mathbf{u}$ of nine components. In an heterogeneous anisotropic elastic media, the governing equations are the equation of motion and the constitutive Hooke's law. Density ρ and symmetrical stiffness tensor c_{ijkl} (having only 21 independent components) describe properties of the medium embedded into $\mathbf{m}$. The generic time-domain wave-equation can therefore be written as

$$\partial_t \mathbf{u}(\mathbf{x},t) - N(\mathbf{m}(\mathbf{x}))H(\nabla)\mathbf{u}(\mathbf{x},t) = \mathbf{s}(\mathbf{x},t), \tag{1}$$

where $N(\mathbf{m})$ depends on physical properties only, $H(\nabla)$ is related to spatial derivatives, $\mathbf{s}$ is the source term, $\mathbf{x}$ is location and t is time. Solving equation (1) with a numerical method and waiting for steady state allows to extract $\mathbf{u}(\mathbf{x},\omega)$ for frequency ω by Fourier transform.

One can also consider the frequency-domain wave equation which turns out to be a linear system

$$\mathbf{B}(\mathbf{x},\omega)\mathbf{u}(\mathbf{x},\omega) \quad = \quad \mathbf{s}(\mathbf{x},\omega), \tag{2}$$

where $\mathbf{B}$ is the so-called impedance matrix (Marfurt, 1984). Linear system (2) is large, sparse and ill-conditioned and can be solved either with a direct or an iterative solver.

Theoretical complexities

Table 1 illustrates the theoretical complexity for computational time and memory for each of the three strategies. This evaluation does not consider parallelism. Considering a whole seismic acquisition, the three methods have the same time complexity, while direct solvers are more expensive for memory.

Time-domain modeling with explicit schemes

Solving equation (1) is generally achieved by an explicit time-marching algorithm. We choose here to discretize the time-derivative by a 2^{nd} order leap-frog scheme while the spatial derivatives are discretized by a staggered-grid with 4^{th} or 8^{th} order Finite Difference (FD) schemes (Fornberg, 1988). CPML (Komatitsch and Martin, 2007) absorbing layers are implemented on edges of the simulation box while a flat free surface condition is considered on top.

Method	Time+FT	Direct	Iterative
Time complexity of seismic sources independant tasks	-	$\mathcal{O}(N^6)$	-
Time complexity of seismic sources dependant tasks	$\mathcal{O}(N^4)$	$\mathcal{O}(N^4)$	$\mathcal{O}(N^4)$
Time complexity for N^2 seismic sources	$\mathcal{O}(N^6)$	$\mathcal{O}(N^6)$	$\mathcal{O}(N^6)$
Memory complexity (single seismic source)	$\mathcal{O}(N^3)$	$\mathcal{O}(N^4)$	$\mathcal{O}(N^3)$

Table 1 *Time & memory theoretical complexities of the three modeling approaches, describing how the time & memory costs scale to a 3D problem of size N^3 (without considering parallelism).*

It is obvious that the main bottleneck of time-marching algorithm is the available computing ressources more than memory, and in particular how the implementation takes benefit of the floating point operations per second (flops) attainable on the execution platform. Figure 1 shows for example a roofline model (Williams et al., 2009) of the efficiency of a 4^{th} order scheme for a pressure kernel on an Intel bi-socket Sandy Bridge Xeon E5-2680. Whatever the Operational Intensity (OI), the attainable flops is under the red and blue lines. Our pressure kernel has been benchmarked at 19 Gflops when for its OI of 0.72, the attainable performance is 46.1 Gflops.

To increase the performance above the green line, vectorization has to be used. But one can see that the green line is not attainable with this very low OI. Three ways to improve performance are conceivable: maximizing in-core performance, minimizing traffic or maximizing bandwidth. This last point could notably be tackled by increasing the order of the scheme. As it is shown by figure 1, higher 8^{th} and 16^{th} order schemes would increase the OI, and thus the attainable Gflops.

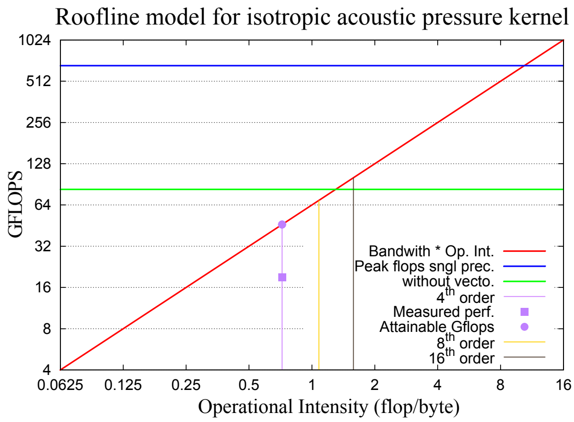

Figure 1 *Roofline model for a bi-socket Sandy Bridge XeonE5-2680 and a pressure kernel algorithm. The x-axis represents the OI, that is a ratio between the number of additions-multiplications and the number of loads-stores (full computing capabilities are reached for an OI of 10). The blue line represents the peak flops in single precision with balanced add/mul, the green line bounds the attainable flops performance when there is no vectorization, and the red line represents the boundary from the bandwidth between memory and processor (benchmarked at 64Gb/s).*

Frequency-domain modeling with direct solver

The equation (2) could be solved for each frequency using a LU decomposition. The LU factorization is computationally expensive but it is performed only once since it is independent of the right-hand side term (*i.e.* the number of sources). Then, we solve each seismic source by backward and forward substitutions. Advances in development of massively-parallel sparse direct solvers based on the multi-frontal approach (Duff et al., 1986) have made frequency-domain seismic modeling based on sparse direct solver feasible to perform seismic modeling for FWI applications at different scales, in combination with optimized compact FD stencil (Operto et al., 2007).

The two main drawbacks of direct solvers for large scale applications remain the poor scalability for taking benefit on multi-node/multi-core architectures of large scale HPC plateforms, and the required in-core memory size for factorization, which limits application to fat nodes with lots of memory.

However, recent advances using Hierarchically SemiSeparable (HSS) matrices (Wang et al., 2011) or Block Low-Rank (BLR) approximation (Amestoy et al., 2012; Weisbecker et al., 2013) allow to mitigate these two limitations with an improved scalability, lower number of operations and lower memory requirements. Figure 2 shows how the BLR format allows to reduce the computational complexity of the factorization compared to the Full-Rank (FR) for a Laplacian operator, close to the Helmholtz equation.

These developements, in combination of advances in HPC architecture including large memory nodes with high-bandwith capabilities, should allow to tackle large size FWI projects in the near future.

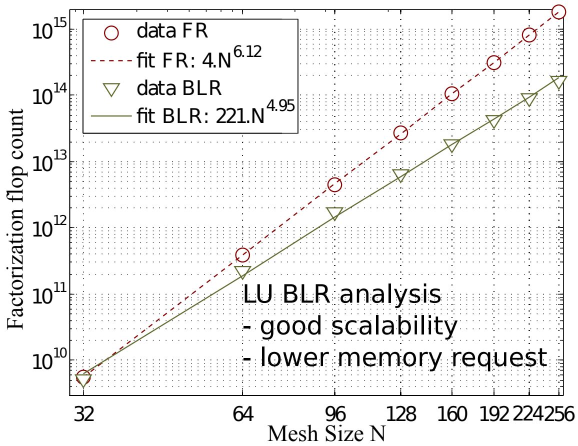

Figure 2 *Flop scalability of the BLR multifrontal factorization of the Laplacian operator discretized with a 3D 7-point stencil, revealing a $O(N^{4.95})$ time complexity for $\varepsilon = 10^{-8}$. Results are obtained with an experimental version of MUMPS embedding Block Low-Rank (BLR) technologies and the standard Full-Rank (FR). The numerical parameter ε defines how much approximation we accept to put in the factorization. A small value leads to a good accuracy but potentially less complexity improvements, while a large value leads to a loosely accurate solution but achieved through a substancially cheaper process. Very large values often lead to unacceptable solutions.*

Frequency-domain modeling with iterative solver

Iterative solvers are generally based on Krylov subspace methods such as GMRES, BiCG(STAB) or CGNR (Saad, 2003). Contrary to direct solvers, iterative solvers fully benefit from the sparsity of the system, in terms of memory requirements and scalability, as only matrix-vector products are required. However, these methods might present poor convergence properties or even fail to converge without preconditioning. Complex Laplacian shifting and multigrid techniques have been proven efficient for the Helmholtz equation (Erlangga and Nabben, 2008) as an appropriate preconditioning step.

The CARP-CG method transforms the original system into a symmetric positive semi-definite system by cyclic row-projections. This system is efficiently solved with the conjugate gradient (CG) method. The cyclic row-projection transformation can be seen as a purely algebraic preconditioning technique which is versatile and easy to implement. The algorithm can be parallelized through a row-block decomposition combined with component-averaging operations.

Recent studies have shown that the 2D and 3D Helmholtz equation in heterogeneous media can be solved efficiently with CARP-CG (van Leeuwen et al., 2012; Gordon and Gordon, 2013). Moreover, Li et al. (2014) show that the same technique is robust and scalable for the 2D elastic equation. Figure 3 shows that the scalability is preserved especially at high frequencies for which the problem size is larger.

Conclusions

We have reviewed three modeling methods allowing to perform frequency-domain FWI. For each of them, we pointed out recent developments that alleviate their respective computing issues. Now the methods are mature enough to be compared for 3D realistic common cases and one can consider a strategy bringing them to play alternately. However the issue of the scalabity to higher number of processes remain for the direct and iterative solvers and thus how to leverage recent architectures such as GPU and MIC with the two frequency-domain modeling methods is an open question.

Acknowledgements

This study was funded by the SEISCOPE II consortium (*http://seiscope2.osug.fr*), sponsored by BP, CGG, CHEVRON, EXXON-MOBIL, JGI, PETROBRAS, SAUDI ARAMCO, SHELL, SINOPEC, STATOIL, TOTAL and WESTERNGECO. This study was granted access to the HPC facilities of CIMENT

CARP-CG FOR ELASTODYNAMIC 2D EQUATIONS

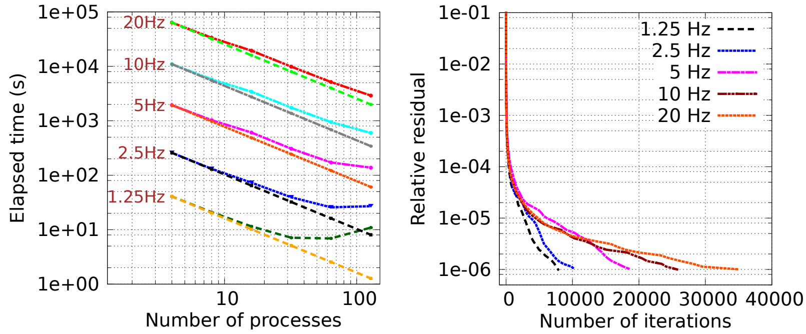

Figure 3 *CARP-CG solver. Left panel: scalability illustrated for different frequencies. The number of grid points by wavelength is kept constant (the problem size increases with the frequency). Right panel: convergence history for different frequencies. The results are obtained on the Marmousi 2 model with high Poisson ratio ($\sigma = 0.4$). 2D elastic wave-equation is discretized with a 2^{nd} order staggered-grid FD scheme.*

(Université Joseph Fourier Grenoble) and of GENCI-CINES under Grant 046091. A part of the results was obtained thanks to collaborative works with INTEL and MUMPS team.

References

Amestoy, P., Ashcraft, C., Boiteau, O., Buttari, A., L'Excellent, J.Y. and Weisbecker, C. [2012] Improving multifrontal methods by means of block low-rank representations. Rapport de recherche RT/APO/12/6, INRIA/RR-8199, Institut National Polytechnique de Toulouse, Toulouse, France, submitted for publication in SIAM Journal on Scientific Computing.

Duff, I.S., Erisman, A.M. and Reid, J.K. [1986] *Direct methods for sparse matrices*. Clarendon Press, Oxford, U. K.

Erlangga, Y.A. and Nabben, R. [2008] On a multilevel Krylov method for the Helmholtz equation preconditioned by shifted Laplacian. *Electronic Transactions on Numerical Analysis*, **31**, 403–424.

Fornberg, B. [1988] Generation of finite difference formulas on arbitrarily spaced grids. *Mathematics of Computation*, **51**, 699–706.

Gordon, D. and Gordon, R. [2013] Robust and highly scalable parallel solution of the Helmholtz equation with large wave numbers. *Journal of Computational and Applied Mathematics*, **237**(1), 182–196.

Komatitsch, D. and Martin, R. [2007] An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. *Geophysics*, **72**(5), SM155–SM167.

Li, Y., Métivier, L., Brossier, R., Han, B. and Virieux, J. [2014] A robust parallel iterative solver for frequency-domain elastic wave modeling. *Expanded Abstracts*, EAGE, A.

Marfurt, K. [1984] Accuracy of finite-difference and finite-element modeling of the scalar and elastic wave equations. *Geophysics*, **49**, 533–549.

Operto, S., Virieux, J., Amestoy, P., L'Éxcellent, J.Y., Giraud, L. and Ben Hadj Ali, H. [2007] 3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study. *Geophysics*, **72**(5), SM195–SM211.

Plessix, R.E. [2009] Three-dimensional frequency-domain full-waveform inversion with an iterative solver. *Geophysics*, **74**, WCC149–WCC157.

Saad, Y. [2003] *Iterative methods for sparse linear systems*. SIAM, Philadelphia.

Sirgue, L., Etgen, T.J., Albertin, U. and Brandsberg-Dahl, S. [2007] System and method for 3D frequency-domain waveform inversion based on 3D time-domain forward modeling. *US Patent Application Publication*, **US2007/0282535 A1**.

van Leeuwen, T., Gordon, D., Gordon, R. and Herrmann, F. [2012] Preconditioning the Helmholtz equations via row projections. *Expanded Abstracts*, EAGE, A002.

Virieux, J. and Operto, S. [2009] An overview of full waveform inversion in exploration geophysics. *Geophysics*, **74**(6), WCC1–WCC26.

Wang, S., de Hoop, M.V. and Xia, J. [2011] On 3d modeling of seismic wave propagation via a structured parallel multifrontal direct helmholtz solver. *Geophysical Prospecting*, **59**(5), 857–873.

Weisbecker, C. et al. [2013] 3D frequency-domain seismic modeling with a block low-rank algebraic multifrontal direct solver. *SEG Technical Program Expanded Abstracts*, 3411–3416.

Williams, S., Waterman, A. and Patterson, D. [2009] Rooofline: an insightful visual performance model for multicore architectures. *Communications of the ACM*, **52**(4), 65–76.