

A Rayleigh Wave Globally Optimal Full Waveform Inversion Framework Based on GPU Parallel Computing

Zhao Le1*, Wei Zhang2, Xin Rong3, Yiming Wang1, Wentao Jin2, Zhengxuan Cao3

¹School of Geophysics and Geomatics, China University of Geosciences, Wuhan, China ²Wuhan Geo-Detection Technology Co., Ltd., Wuhan, China ³Zhejiang Design Institute of Water Conservancy & Hydro-Electric Power Co., Ltd., Hangzhou, China Email: *lezhao hm@sina.com

How to cite this paper: Le, Z., Zhang, W., Rong, X., Wang, Y. M., Jin, W. T., & Cao, Z. X. (2023). A Rayleigh Wave Globally Optimal Full Waveform Inversion Framework Based on GPU Parallel Computing. *Journal of Geoscience and Environment Protection, 11*, 327-338. https://doi.org/10.4236/gep.2023.113019

Received: March 8, 2023 **Accepted:** March 28, 2023 **Published:** March 31, 2023

Abstract

Conventional gradient-based full waveform inversion (FWI) is a local optimization, which is highly dependent on the initial model and prone to trapping in local minima. Globally optimal FWI that can overcome this limitation is particularly attractive, but is currently limited by the huge amount of calculation. In this paper, we propose a globally optimal FWI framework based on GPU parallel computing, which greatly improves the efficiency, and is expected to make globally optimal FWI more widely used. In this framework, we simplify and recombine the model parameters, and optimize the model iteratively. Each iteration contains hundreds of individuals, each individual is independent of the other, and each individual contains forward modeling and cost function calculation. The framework is suitable for a variety of globally optimal algorithms, and we test the framework with particle swarm optimization algorithm for example. Both the synthetic and field examples achieve good results, indicating the effectiveness of the framework.

Keywords

Full Waveform Inversion, Finite-Difference Method, Globally Optimal Framework, GPU Parallel Computing, Particle Swarm Optimization

1. Introduction

Rayleigh wave exploration is a very useful geophysical method, it has very high resolution in near-surface exploration (Socco et al., 2010). Multi-channel analysis of surface waves (MASW) is the most widely used method in surface wave exploration (Xia et al., 1999). The main idea of MASW is extracting dispersion curve manually, and getting the local 1D S-wave velocity profiles by dispersion

curve inversion. The dispersion curve extracting is highly dependent on subjective judgement and experience. When geological conditions are complex, energy mixing and pseudo multi-mode may happen (Zhang, 2011), it's difficult to extract accurate dispersion curve manually. In addition, because the theoretical dispersion curve is based on the assumption of 1D flat layered model (Knopoff, 1964), dispersion curve inversion can only solve the problem of horizontal layered media, which is often not the case in real strata.

Full waveform inversion (FWI) in time-domain (Tarantola, 1984) and FWI in frequency-domain (Pratt, 1990) were proposed successively to solve complicated geological issues. Since FWI does not need to extract dispersion curve manually and has no restriction on the distribution of media, it has broad application prospects and developed rapidly in recent years (Romdhane et al., 2011; Pan et al., 2018).

Forward modeling is of fundamental to FWI, Rayleigh wave simulation is mainly based on the research of Virieux (1986). The finite-difference method (FDM) is the most widely used method at present for its high efficiency and accuracy (Bohlen, 2002). Due to the huge amount of calculation of FWI, the conventional gradient-based FWI is a local optimization (Liu et al., 2017). However, compared with dispersion curve inversion, FWI has more parameters, and is more nonlinear and nonunique. Locally optimal FWI is prone to trapping in local minima, and its success greatly dependent on the initial model. Thus, globally optimal FWI that can overcome this limitation is particularly attractive (O'Neill et al., 2003).

GPU parallel computing has some applications in Rayleigh wave gradientbased FWI for getting single modeling waveform or gradient (Fang et al., 2018). However, GPU parallel computing is more suitable for a large number of independent forward modeling in globally optimal FWI. In this paper, we propose a Rayleigh wave globally optimal FWI framework based on GPU parallel computing, which is globally optimal and efficient. The framework is suitable for a variety of globally optimal algorithms, and we test our framework with particle swarm optimization algorithm (PSO) for example.

2. Methodology

2.1. Forward Modeling Method

For the 2D isotropic media, the first-order linear partial differential equation of motion describing elastic wave propagation is as follows (Virieux, 1986):

$$\frac{\partial \overline{V_x}}{\partial t} = \frac{\partial \overline{\sigma_{xx}}}{\partial x} + \frac{\partial \overline{\sigma_{xz}}}{\partial z}
\frac{\partial \overline{V_z}}{\partial t} = \frac{\partial \overline{\sigma_{xx}}}{\partial x} + \frac{\partial \overline{\sigma_{zz}}}{\partial z}
\frac{\partial \overline{\sigma_{xx}}}{\partial t} = (\lambda + 2\mu) \frac{\partial V_x}{\partial x} + \lambda \frac{\partial V_z}{\partial z}
\frac{\partial \overline{\sigma_{zz}}}{\partial t} = (\lambda + 2\mu) \frac{\partial V_z}{\partial z} + \lambda \frac{\partial V_x}{\partial x}
\frac{\partial \overline{\sigma_{xz}}}{\partial t} = \mu \left(\frac{\partial V_x}{\partial z} + \frac{\partial V_z}{\partial x}\right)$$
(1)

ſ

where V_x and V_z are the particle velocity vectors of x-axis and z-axis, respectively; σ_{xx} , σ_{xz} and σ_{zz} are stress tensors; ρ is density; λ and μ are the first and second Lame coefficients, respectively.

The process of forward modeling by GPU parallel computing is shown in Figure 1.

2.2. PSO Inversion Method

PSO is a globally optimal algorithm inspired by a flock of birds searching for food (Kennedy & Eberhart, 1995). The idea of PSO is that each particle makes the misfit to minimum according to the best position of particle misfit history (*pbest*) and swarm misfit history (*gbest*). In the PSO method, a trial model will be transformed into a series of variables, the variables to be solved are called position (x), and the position increments are called velocity (v), we update the position iteratively via Equations (2)-(5).

$$pbest_i^k = \min\{\Phi(x_i^j)\}, \quad j = 1, 2, \cdots, k$$
 (2)

$$gbest^{k} = \min\{\Phi(x_{i}^{j})\}, \quad i = 1, 2, \cdots, M; \quad j = 1, 2, \cdots, k$$
 (3)

$$x_i^{k+1} = x_i^k + v_i^{k+1}$$
(4)

$$v_i^{k+1} = \omega v_i^k + a_1 r_1(pbest_i^k - x_i^k) + a_2 r_2(gbest^k - x_i^k)$$
(5)

where Φ is the objective function; *i* and *k* are the number of particle and iteration, respectively; *M* is the total number of particle; ω is the inertia weight, which increases exploration and avoids elitism; x_i^k and v_i^k are the position and velocity of the *t*th variable at the *k*th iteration, respectively; a_1 and a_2 are the local and global weights, respectively; r_i and r_2 are the random numbers between 0 and 1.



Figure 1. The flowchart of FDM by GPU parallel computing.

3. Speed-Up Analysis

Due to the huge amount of calculation of FWI, the conventional FWI is mainly gradient-based. To make globally optimal FWI more widely applied, its operational efficiency must be improved. We set up four grid models (**Table 1**) to test the efficiency of GPU parallel computing, and their parameters are the same except for the number of blocks. Where Δx and Δz are the length of blocks in x-axis and z-axis, respectively; Δt is the time interval; nx and nz are the number of blocks in x-axis, respectively; nt is the number of time; f_c is the center frequency of source; t_0 is the time shift of source. We use spatial 8th-order and temporal 2nd-order finite-difference method for all the grid models in this paper.

We wrote the code by MATLAB, C++, and CUDA, the MATLAB and C++ code only runs on CPU, the CUDA code runs on CPU and GPU, and their runtimes are shown in **Table 2**. The speed-up ratio is equal to runtime of C++ divided by runtime of CUDA, the GPU usage is the usage of GPU in CUDA computing. The results are tested on an entry-level laptop, and the CPU model is Intel Core i5-10210U, the GPU model is NVIDIA GeForce MX350, the RAM size is 16 GB.

As can be seen from the test results (**Table 2**), MATLAB is not suitable for globally optimal FWI because their runtimes are too long. GPU parallel computing can greatly improve computing efficiency, and the higher the GPU usage, the higher the speed-up ratio. To avoid running the GPU at full capacity, the grid model of #G3 is suitable for following computation, and readers can choose the appropriate grid model according to their own situation.

Grid Model	Δx (m)	Δz (m)	Δt (ms)	nx	nz	nt	f _c (Hz)	t ₀ (ms)
#G1	0.5	0.5	0.2	256	128	2048	25	80
#G2	0.5	0.5	0.2	512	256	2048	25	80
#G3	0.5	0.5	0.2	1024	512	2048	25	80
#G4	0.5	0.5	0.2	2048	1024	2048	25	80

Table 1. Parameters of grid models.

Table 2. Runtime of different languages and their speed-up effects.

Grid Model	MATLAB Time (s)	C++ Time (s)	CUDA Time (s)	Speed-up Ratio	GPU Usage
#G1	26	7.4	1.4	5.29	23%
#G2	148	26.6	4.6	5.78	73%
#G3	882	98.4	16.4	6.00	91%
#G4	-	375.1	62.4	6.01	99%

Additionally, GPU computing is a litter different from CPU computing in that GPU computing takes a lot of time to allocate and free variable memory. We record the runtime of different parts in **Table 3**, we can see that the time of memory allocation and freeing almost the same in different models. Therefore, we can further improve efficiency by allocating memory for all variables at once, and freeing memory at once after multiple forward modeling. To verify the efficiency improvement of allocating and freeing memory at once, we perform grid model of #G1 (**Table 1**) for multiple modeling. The results are shown in **Table 4**, and the efficiency improvement is obvious.

4. Parameters Optimization

4.1. Parameters Simplification

In conventional gradient-based FWI, every grid parameter is variable, namely, the number of variable parameters in model #G1 is $256 \times 128 (=nx \times nz)$. Unlike gradient-based FWI, we greatly simplify the grid parameters by introducing number of layers (n_l) and number of layer-points (n_p) . We take the model of 4 layers and 5 layer-points (**Figure 2**) for example. Each point in each layer has P_x and P_z positions, and the P_x positions of the beginning and end of each layer are fixed, namely, $2n_p - 2$ positions per layer. Thus, the parameters include n_l layer velocities and n_p positions, totally, $n_l + (2n_p - 2) \times (n_l - 1)$ parameters. Actually, the number of variable parameters is reduced from 32,768 (=256 \times 128) to 28, and the inversion efficiency is greatly improved.



Figure 2. Parameters of the 4 layers and 5 layer-points model. Where the beginning of each layer is fixed to 0, and the end of each layer is fixed to the right boundary of the model $(=nx^* \Delta x)$.

Grid Model	Memory Allocation (s)	Computing (s)	Memory Freeing (s)	Total Time (s)
#G1	0.921	0.472	0.007	1.4
#G2	0.946	3.647	0.007	4.6
#G3	0.975	15.418	0.007	16.4
#G4	0.996	61.396	0.008	62.4

Modeling Times	C++ Time (s)	CUDA Time (s)	Speed-up Ratio
1	7.4	1.4	5.29
16	116.8	16.4	7.04
128	934.0	123.2	7.58
1024	7397.3	961.0	7.70

Table 4. Speed-up effect of GPU parallel in multiple forward modeling.

4.2. Parameters Recombination

In globally optimal FWI, we perform multiple forward modeling each iteration, and parameters are recombined to further improve efficiency. For instance, we adopt #G1 model (nx = 256, nz = 128) to perform 128 modeling at one iteration, which means the length of parameter V_p is 256 * 128 * 128. As mentioned above, grid model of #G3 (nx = 1024, nz = 512) is suitable to get the maximum GPU usage in the author's computer. Thus, the V_p would be split into 8 one-dimensional vectors, and we allocate a vector of length 1024 * 512 on GPU for V_p and then perform 8 cycles of calculation.

5. Synthetic Examples

We test the globally optimal FWI framework with PSO algorithm (PSO-FWI) and we perform two synthetic examples to prove the validity of the framework. The 4 layers, 5 layer-points model (called $#M_1$) is shown in **Figure 3(a)**. The inversion parameters used in this paper are shown in **Table 5**, where *M* is the number of particles; *N* is the maximum number of iterations; #G1 is the grid model shown in **Table 1**; ω is inertia weight; a_1 and a_2 are the local and global weights, respectively; μ is the mutation rate.

In PSO-FWI, each particle corresponds to multiple parameters, for instance, the number of parameters for $\#M_1$ is 28 $(=n_l + (2n_p - 2) * (n_l - 1))$. The parameters have the corresponding value ranges, where the velocity range is from 150 to 750 m/s, the point interval (the difference from the last point) of P_z is from 1 to 10 m.

The model comparison of $\#M_1$ inversion is shown in **Figure 3**. The multi-channel record comparison of $\#M_1$ is shown in **Figure 4**, where the nearest offset is 10 m, the receiver interval is 1 m, and the channel number is 48. The single channel record comparison of $\#M_1$ is shown in **Figure 5**.

From the comparisons shown above, the inverted results are in good agreement with the true results, which proves the validity of the framework. The high efficiency is evident as the entire inversion performs 25,600 forward modeling and takes about 10 hours on a personal computer.

6. Field Data Application

We acquired the field data in Hangzhou, Zhejiang Province, China, where the



Figure 3. Model comparison in $\#M_1$ inversion. (a) is $\#M_1$ model. (b) is the comparison between $\#M_1$ and the inverted model, where the red lines and text represent the true model, and the blue ones represent the inverted model.

Table 5. Parameters of I	PSO-FWI.
--------------------------	----------

Particles M	Iterations N	Grid Model	ω	a1	a2	μ
128	200	#G1	0.5	1	2	0.1

test area had a vast undisturbed stratum, and a loess layer covered on a mudstone layer. We used 4.5 Hz vertical geophones and a 24-channel seismograph. The geophone interval was 1 m with the nearest offset of 10 m. The number of record points was 2048, and sampling interval was 0.2 ms.



Figure 4. Multi-channel records comparison in #M1 inversion. (a) is the comparison between the observed and inverted records, where the red and blue lines represent the observed and inverted records, respectively. (b) is the corresponding residual.

The comparison between field and inverted record is shown in **Figure 6(a)**, and the corresponding residual is shown in **Figure 6(b)**. We can see that the low-frequency and large-amplitude waveforms match well, while the high-frequency and small-amplitude waveforms match poorly. In field measurement, the high-frequency waves are gradually suppressed with the wave propagation, which results that the near-offset geophones have richer high-frequency components than the far-offset ones. Additionally, the high-frequency noise can't be



Figure 5. Single channel record comparison in #M1 inversion. (a), (b), (c), and (d) are the waveform comparisons of the 1st, 11th, 21st, and 31st sensors, respectively. Their corresponding offsets are 10 m, 20 m, 30 m, and 40 m, respectively. Where the red and blue lines represent the observed and inverted waveforms, respectively.



Figure 6. Multi-channel records comparison in field data inversion. (a) is the comparison between the field and inverted records, where the red and blue lines represent the observed and inverted records, respectively. (b) is the corresponding residual.

avoided, which increases the difficulty of the fitting. Thus, low-frequency source is suggested to be used in field data acquisition, and low-pass filtering is essential in data processing. The comparison of inverted model and borehole is shown in **Figure 7**. They are in good agreement which demonstrates the effectiveness of the framework.

Figure 7. Model comparison and statistics of all accept models at a distance of 15 m in field data inversion. (a) is the inverted model. (b) is the comparison between borehole and inverted model at a distance of 15 m.

7. Conclusion

In this study, we propose a globally optimal framework based on GPU parallel computing to avoid falling into local minima in Rayleigh wave FWI. We present the process of forward modeling by GPU parallel computing. The efficiency improvement of GPU parallel computing is obvious from the statistics of speed-up analysis. Parameters simplification and recombination further improve the efficiency, which is likely to make the framework more widely used. Both synthetic examples and field data application of PSO-FWI achieve good results, which demonstrate the feasibility of the framework.

Acknowledgements

We deeply appreciate the teachers and friends for their thoughtful and constructive comments, which greatly improve the quality of this paper.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- Bohlen, T. (2002). Parallel 3-D Viscoelastic Finite-Difference Seismic Modeling. Computers & Geosciences, 28, 887-899. https://doi.org/10.1016/S0098-3004(02)00006-7
- Fang, J., Zhou, H., Zhang, Q., Chen, H., Wang, N., Sun, P., & Wang, S. (2018). Effect of Surface-Related Rayleigh and Multiple Waves on Velocity Reconstruction with Time-Domain Elastic FWI. *Journal of Applied Geophysics, 148,* 33-43. https://doi.org/10.1016/j.jappgeo.2017.11.006
- Kennedy, J., & Eberhart, R. C. (1995). Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, 4, 1942-1948.
- Knopoff, L. (1964). A Matrix Method for Elastic Wave Problems. Bulletin of the Seismological Society of America, 54, 431-438. https://doi.org/10.1785/BSSA0540010431
- Liu, Y., Teng, J., Xu, T., Badal, J., Liu, Q., & Zhou, B. (2017). Effects of Conjugate Gradient Methods and Step-Length Formulas on the Multiscale Full Waveform Inversion in Time Domain: Numerical Experiments. *Pure and Applied Geophysics*, 174, 1983-2006. https://doi.org/10.1007/s00024-017-1512-3
- O'Neill, A., Dentith, M., & List, R. (2003). Full-Waveform P-SV Reflectivity Inversion of Surface Waves for Shallow Engineering Applications. *Exploration Geophysics*, 34, 158-173. <u>https://doi.org/10.1071/EG03158</u>
- Pan, Y., Gao, L., & Bohlen, T. (2018). Time-Domain Full-Waveform Inversion of Rayleigh and Love Waves in Presence of Free-Surface Topography. *Journal of Applied Geophysics*, 152, 77-85. https://doi.org/10.1016/j.jappgeo.2018.03.006
- Pratt, R. G. (1990). Seismic Waveform Inversion in the Frequency Domain, Part I: Theory and Verification in a Physical Scale Model. *Geophysics*, 64, 888-901. <u>https://doi.org/10.1190/1.1444597</u>
- Romdhane, A., Grandjean, G., Brossier, R., Rejiba, F., Operto, S., & Virieux, J. (2011). Shallow-Structure Characterization by 2D Elastic Full Waveform Inversion. *Geophysics*, 76, R81-R93. <u>https://doi.org/10.1190/1.3569798</u>
- Socco, L. V., Foti, S., & Boiero, D. (2010). Surface Wave Analysis for Building near Surface Velocity Models: Established Approaches and New Perspectives. *Geophysics*, 75, A83-A102. https://doi.org/10.1190/1.3479491
- Tarantola, A. (1984). Inversion of Seismic Reflection Data in the Acoustic Approximation. *Geophysics*, 49, 1259-1266. <u>https://doi.org/10.1190/1.1441754</u>
- Virieux, J. (1986). P-SV Wave Propagation in Heterogeneous Media: Velocity-Stress Finite-Difference Method. *Geophysics*, 51, 889-901. <u>https://doi.org/10.1190/1.1442147</u>
- Xia, J., Miller, R. D., & Park, C. B. (1999). Estimation of Near-Surface Shear-Wave Velocity by Inversion of Rayleigh Wave. *Geophysics, 64,* 691-700. https://doi.org/10.1190/1.1444578
- Zhang, S. (2011). Effective Dispersion Curve and Pseudo Multimode Dispersion Curves for Rayleigh Wave. *Journal of Earth Science*, 22, 226-230. https://doi.org/10.1007/s12583-011-0175-8