PAPER • OPEN ACCESS

A GPU implementation of least-squares reverse time migration

A GPU implementation of least-squares reverse time migration

Phudit Sombutsirinun* and Chaiwoot Boonyasiriwat

Department of Physics, Faculty of Science, Mahidol University, 272 Rama VI Road, Ratchathewi, Bangkok 10400, Thailand

*E-mail: s.phudit1995@gmail.com

Abstract. Least-squares reverse time migration (LSRTM) is a seismic imaging method that can provide higher-resolution image of the subsurface structures compared to other methods. However, LSRTM is computationally expensive. To reduce the computational time of LSRTM, GPU can be utilized. This leads to the objective of this work which is to develop a GPU implementation of LSRTM. In this work, the two-dimensional first-order acoustic wave equations are solved using the second-order finite difference on a staggered grid and a perfectly matched layer is used as an absorbing boundary condition. The adjoint-state method is used to compute the gradient of the objective function. A linear conjugate gradient method is used to minimize the objective function. Both forward- and backward-propagation of wavefields using the finite-difference method are performed on a single GPU using the NVIDIA CUDA library. For a verification purpose, the GPU program of LSRTM was applied to a synthetic data set generated from the Marmousi model. Numerical results show that LSRTM can provide an image with a higher resolution of subsurface structure compared to a conventional RTM image. For a computational cost issue, the GPU-version of LSRTM is significantly faster than the serial CPU-version of LSRTM.

1. Introduction

The implementation of least-squares method on reverse time migration method (RTM) has been done since [1]. A linearized inversion which can be seen as an improvement of the pre-stack RTM of [2] was proposed. The least-squares reverse time migration (LSRTM) is promising because RTM can be view as a computation of gradient in full waveform inversion. Hence, making it calculate iteratively should improve the result. Despite the better resolution of the image, LSRTM was not commonly used due to its high requirement of computational storage and time. With the introduction of finite-different calculation on GPUs using CUDA [3], seismic imaging and inverse problems are mostly applied using this API for the significantly improved speedup [4]. The objective of this work is to apply LSRTM to CUDA in order to decrease its computational time.

2. Least-squares reverse time migration

In this work, the two-dimensional first-order acoustic wave equations are solved using the second-order finite difference (FD) on a staggered grid and a perfectly matched layer is used as an absorbing boundary condition. This step will be described as applying the Born modeling operator $\mathbf{L}$ that will grant the seismogram $\mathbf{d}$ from the geophysical model $\mathbf{m}$

$$\mathbf{d}_i = \mathbf{L}_i \mathbf{m}_i \tag{1}$$

also the imaging condition that give the reflectivity model from the provided data can be recognized as a transpose operator of the Born modeling since it is the conversion of seismic data back to geological model.

$$\mathbf{m}_i = \mathbf{L}_i^T \mathbf{d}_i \tag{2}$$

A derivation of receiver wavefield propagation and imaging condition that lead to operator $\mathbf{L}^T$ is well defined by [5]. After obtaining the simulated data ($\mathbf{d}$), the data will be compared with the observed data ($\mathbf{D}$) at every sources (N_s). The goal of inversion scheme is to fine the model $\mathbf{m}$ that minimize the difference between these two parameters. To clarify that, the model $\mathbf{m}$ is sought to minimize the objective function

$$f(\mathbf{m}) = \frac{1}{2} \sum_i^{N_s} \|\mathbf{L}_i \mathbf{m} - \mathbf{D}_i\|^2 \tag{3}$$

and the way to seek that value can be done by the gradient descent method such as conjugate gradient method (CG) which is implemented in this work. The main idea of gradient descent method is to update the model by the gradient of objective function (equation (3)) that can be calculated by migrating the residual data, iteratively.

$$\mathbf{g}^{k+1} = \mathbf{L}^T(\mathbf{L}\mathbf{m}^k - \mathbf{D}) \tag{4}$$

CG algorithm will adjust this gradient $\mathbf{g}$ to speed up the convergent rate and use it to update the model by these following algorithm, while k is an iteration number.

$$\beta = \frac{\left[\mathbf{g}^{k+1}\right]^T \mathbf{g}^{k+1}}{\left[\mathbf{g}^{k}\right]^T \mathbf{g}^{k}} \tag{5}$$

$$\mathbf{z}^{k+1} = -\mathbf{g}^{k+1} + \beta \mathbf{z}^k \tag{6}$$

$$\alpha = \frac{\left[\mathbf{z}^{k+1}\right]^T \mathbf{g}^{k+1}}{\left[\mathbf{L}\mathbf{z}^{k+1}\right]^T \mathbf{L}\mathbf{z}^{k+1}} \tag{7}$$

$$\mathbf{m}^{k+1} = \mathbf{m}^k + \alpha \mathbf{z}^{k+1} \tag{8}$$

LSRTM algorithm can be summed up by the flowchart in figure 1.

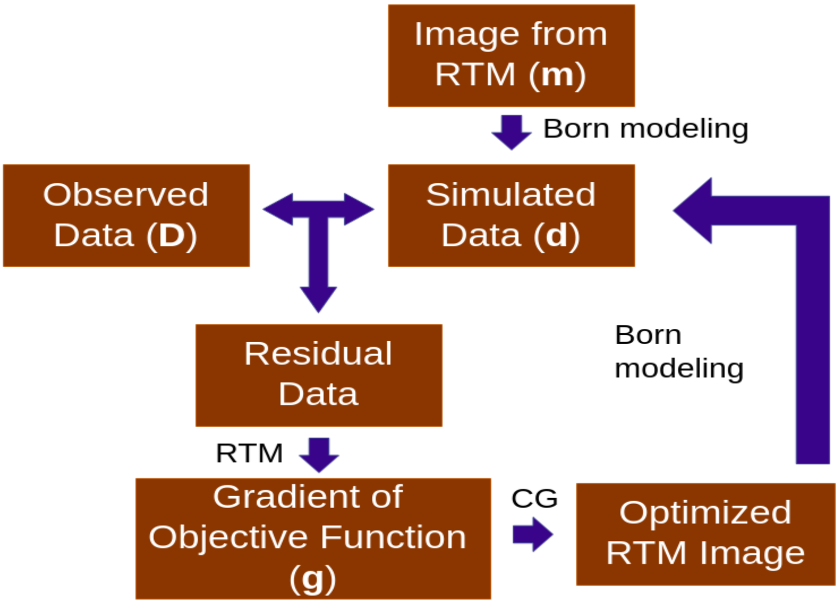


Figure 1. Flowchart of LSRTM algorithm

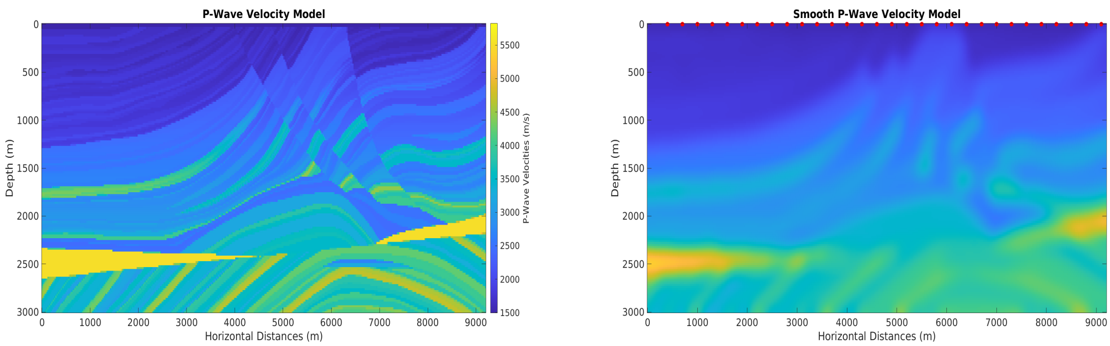
3. Single GPU implementation

GPU architecture consists of a set of multiprocessors with their own shared memory, and each multiprocessor is able to access to global memory of the device. Comparing to a CPU architecture, it is vivid that GPU is more capable of executing a pile of data at the same time. To utilize the GPU capability, the GPU kernel function from CUDA API is need to be called first. By assigning threads into block and grid, a C-language (or C++) function is executed in parallel. In addition, by using shared memory, the calculation efficiency is boosted up because it lessen the data transfer between CPU and GPU. LSRTM algorithm consists of a heavily-loaded FD calculation which are forward propagation of source wavefield and backward propagation of source and receiver wavefields during Reverse time migration method (equations (1) and (2)). Moreover, this set of algorithm is carry on during least-squares inversion plus another forward modeling in step length calculation (equation (7)). Therefore, the main target of GPU implementation of LSRTM in this work is to decrease the FD computation time by create the FD calculation kernel function.

4. Numerical results

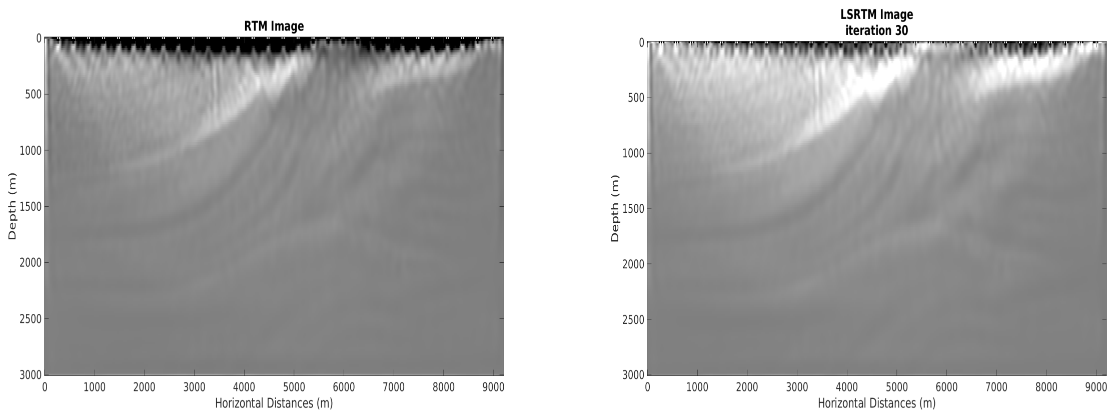
For verification, our developed LSRTM algorithm was tested on Marmousi model (figure 2). The actual model will be smoothed by Guassian filter foremost in order to use it as an initial model. A simulation was done by using 30 sources of 5-Hz Ricker wavelet, 461 receivers with 20-meters apart, and a model mesh of 461×151 with the grid size of 20 meters. Both Ricker wavelet frequency and grid size are picked in order to prevent numerical dispersion [6]. The migrated image should be comparable with the updated velocity model in [4] which produced by 21 shot of 10-Hz Ricker wavelet in the same size of Marmousi model. The image from RTM and 30-iterations LSRTM are presented in figure 3. The optimized image on the right is entirely illuminated by the least-squares method leading to a more vivid structure of Marmousi model. Furthermore, the noise that accumulate on the sources position on the top of the surface (depth 0 m) is diluted. However, another noise below the surface which came from the unwanted crosscorrelation is boomed up which can be clearly seen at the depth between 0-1000 meters and 0-2000 horizontal distances. This noise can be lessen by applying up-down separation technique

[7] which can be done in future work. The value of objective function during the inversion scheme is shown in figure 4. An exponentially-decrease curve depict a success of linear inversion by CG method.The objective function ratio between the last iteration and the first iteration which came from RTM method is 0.0023. Lastly, a total run time of LSRTM algorithm on C-language and CUDA API were measured (table 1). The acquired speed up from GPU implementation is about 7 times faster. The simulation were done using 20 sources and 15 iterations. There are lots of rooms for improvement including implement parallel reduction in β and α in equations (5) and (7), and also implement this algorithm on multiple GPUs.



(a) **(b)**

Figure 2. (a) P-wave velocity and (b) Smooth P-wave velocity of Marmousi model with red dots represent sources coordinate.



(a) **(b)**

Figure 3. (a) RTM and (b) LSRTM image of Marmousi model.

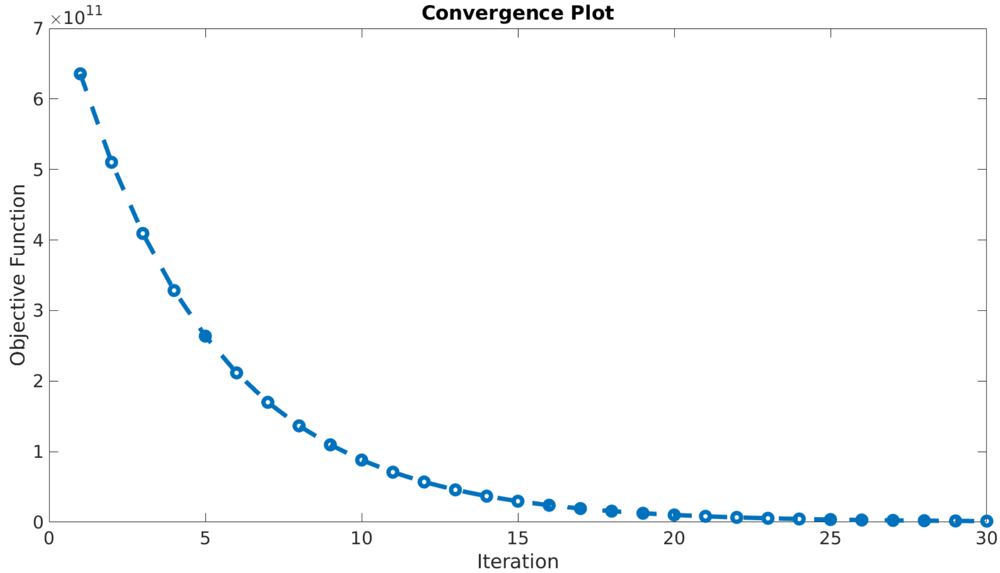


Figure 4. Convergence plot of objective function decrease at each iteration.

Table 1. Speedup performance

CPU	GPU	Speedup
444 m 22.969 s	65 m 1.915 s	6.833303

5. Conclusion

LSRTM is one of the reflection seismic imaging method that improve the migrated image by using least-squares method. Our developed RTM algorithm is capable of imaging a complex subsurface structure. The extra implementation of up-down separation should diminish the noise below the surface which may lead to the improvement of image quality. The inversion scheme in LSRTM algorithm reduced the value of objective function exponentially, and made the geologic structure more vivid. With the additional of kernel function in GPU implementation, the total run time of GPU version is about 7-times faster than that of C version.

References

[1] Bourgeois A, Jiang B F and Lailly P 1989 *Geophysics* **99** 435–45
[2] Chang W F and McMechan G A 1986 *Geophysics* **51** 67–84
[3] Micikevicius P 2009 *3D Finite Difference Computation on GPUs using CUDA* (Washington D.C., US.: GPGPU-2) p 79–84
[4] Yang P, Gao J and Wang B 2015 *Geophysics* **80** F31–9
[5] Plessix R E 2006 *Geophys. J. int.* **167** 495–503
[6] Levander A R 1988 *Geophysics* **53** 1425–36
[7] Liu F, Zhang G, Morton S A and Leveille J P 2011 *Geophysics* **76** S19–39