8th Biennial International Conference
& Exposition on Petroleum Geophysics

"HYDERABAD 2010"

P-239

The GPU/CPU based Fourier integration depth migration: algorithm and implementation

Hongwei Liu* and Hong Liu
Key Laboratory of integrative researches on geophysics for petroleum, Institute of Geology and Geophysics, Chinese Academy of Science, Beijing

Summary

We develop a new one-way wave prestack depth migration method called Fourier integration (FI) method which uses the exact extrapolation operator and the exact velocity model without any approximations or assumptions. The work is performed on a CUDA (Compute Unified Device Architecture) based Graphic Processing Unit (GPU). Compared to the traditional one way wave method, the GPU version FI method is faster and more accurate in the presence of severe lateral-velocity variations.

Introduction

Since Clearbout (1985) developed the one way method in the early 1970s it has been extensively used in seismic imaging. It is crucial to find one-way propagators that can accurately model wave field propagation with wide propagation angles in the presence of severe lateral velocity variations. During the past 40 years, a number of these propagators appeared, including FX domain method, such as finite difference (FD) method (Clearbout, 1985); FK domain method, such as the Fourier method (Stolt, R. H., 1978) and the phase-shift (PS) method (Gazdag, 1978); and FKX domain method, such as the phase shift plus interpolation (PSPI) method (Gazdag and Sguazzero, 1984), split-step Fourier (SP) method (Stoffa etc., 1990), Fourier finite difference (FFD) method (Ristow and Rühl, 1994), wide-angle screen (Xie and Wu, 1998) and generalized screen (GSP) method (Le Rousseau and de Hoop, 2001). All these methods made some approximations to the exact extrapolation operator and induced some inaccuracy hereby. Margrave (1999) developed the non-stationary phase shift (NSPS) method which employed the exact operator but a practical implementation of this method is possible only when the required velocity model is made piecewise constant laterally (Robert J. Ferguson and Gary F. Margrave, 2002). In this paper, we develop the FI method which uses the exact extrapolation operator and the exact velocity model without any approximations or assumptions.

The reason why the FI method has not been used is that the computation cost is too high because of the large number of matrix multiplication. The CUDA based GPU could compute the matrix multiplication perfectly. So we perform the FI method on a CPU/GPU cooperating system architecture. The test is performed on a Tesla S1070 (NVIDIA S1070 Computing System) with the latest CUDA version 2.3.1 (NVIDIA Corporation, August 26th, 2009). The speedup ratio is so high that the speed of the FI method is faster than almost all the traditional CPU based one-way wave methods. We compare the GPU based FI method with CPU based FD, FFD, SP and PSPI methods on some numerical examples. We only discuss the two dimensional prestack depth migration cases in this article.

Theory and method

The one way wave prestack migration methods always try to solve the following equation:

$$\left(\frac{\partial}{\partial z}+i\Lambda\right)D(x,z,\omega)=0;\quad \Lambda=\sqrt{\frac{w^2}{v(x,z)^2}+\frac{\partial^2}{\partial x^2}} \qquad (1)$$

The solution of this equation in FKX domain is:

$$D(x,z+\Delta z,\omega)=\int d(k_x,z,\omega)\exp(ik_z\Delta z)\exp(ik_x x)dk_x \quad (2)$$

$$d(k_x,z,\omega)=\int D(x,z,\omega)\exp(-ik_x x)dx \quad (3)$$

$$k_z=\sqrt{\frac{w^2}{v(x,z)^2}-k_x^2} \quad (4)$$

The computation cost of equation 2 is very high because we could not use fast Fourier transform (FFT) algorithm (note that the extrapolation operator exp(ik$_z$Δz) is relevant to the space variable x). The traditional methods usually try to approximate the extrapolation operator in a way that the space and wave number variables are separable so that FFT could be used to compute equation 2. Take the simplest case as an example, the PS method just use a constant v (z) to replace v(x, z) in k$_z$ so that the extrapolation operator is irrelevant to the space variable x and FFT could be used. The approximations could reduce the computation cost notably (table 1) but induce inaccuracy meanwhile.

Figure 1 shows that the computation of equation 2 is actually a matrix multiplication and the CUDA based GPU could do this job perfectly. The 2D matrix *d(k$_x$ z, w)* should be stored in the GPU memory while extrapolation operator (3D matrix) need not to be stored because the elements could be computed simultaneously with the integration result *D(x, z+ Δz, w)*. Being different from the PS method, the FI method performs the phase shift and the inverse Fourier integration at one time while the PS method performs these operations separately and uses IFFT to perform the inverse Fourier transform. Based on the characteristic of GPU, all the elements of the result matrix *D(x, z+ Δz, w)* could be calculated at one time.

Table 1 is the comparison of the computation cost between PS and FI method. The FI (GPU) column denotes the cost of each thread. The cost of each thread of the GPU version FI method is even lower than SP method because all elements are computed simultaneously.

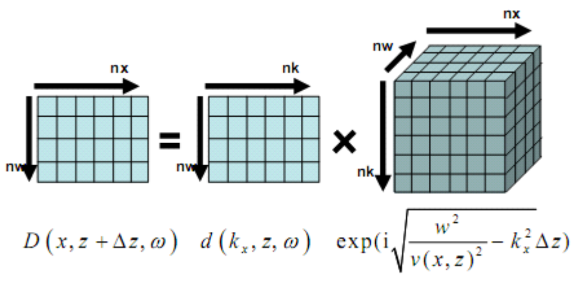


Figure 1: The computation architecture of equation 2 using FI method. The extrapolation operator is a 3D matrix and it needs not to be stored.

Table1: The computation cost of the PS method and FI method. The FI (GPU) column denotes the cost of each thread. The IFFT and phase shift are fulfilled simultaneously.

	PS	FI(CPU)	FI(GPU)
FFT	Nw*Nklog(Nk)	Nw*Nklog(Nk)	CUFFT
Imaging	Nw*Nx	Nw*Nx	Nw
Phase shift	Nw*Nk	Nw*Nk*Nx	Nk
IFFT	Nw*Nklog(Nk)		

Numerical examples

In this section, three numerical examples will be shown: impulse responses in a velocity media with linear variations in both lateral and vertical directions; the marmousi model and the sigsbee 2a synthetic model. All these examples will contain a comparison between GPU version FI method and CPU version FD, FFD, SP and PSPI methods. The GPU used is T1070 with 4 cards, each has 240 cores and 4GB memory (actually the computation time mentioned below is measured only on one card); the CUDA version is 2.3 and the CPU used is Dual Core Intel Pentium E5200 with 2GB DDR2 memory.

Impulse responses

The velocity model with linear variations in both lateral and vertical directions is used to test the impulse responses. The velocity function is as follows:

$$v=v_0+v'\sin\alpha(x-x_0)+v'\cos\alpha(z-z_0) \quad (5)$$

The position of the shot (x$_0$, z0) equals (1000m, 0m) and v$_0$ =2000m/s, *v'* is the gradient of the velocity and a is the angle between the gradient and the horizontal direction. In he test v' is 0.707m/s/m and a is 45°. We put four ricker wavelets at t=0.2s, 0.4s, 0.6s and 0.8s and the wave fronts in this media is still circles (Nolet G, 1998). The analytical solution is as follows:

$$\left(x-x_0+\frac{v_0}{v'}\sin\alpha\left(1-\mathrm{ch}(v't)\right)\right)^2+\left(z-z_0+\frac{v_0}{v'}\cos\alpha\left(1-\mathrm{ch}(v't)\right)\right)^2=\left(\frac{v_0}{v'}\mathrm{sh}(v't)\right)^2 \qquad (6)$$

Note that the variable t in the equation is the propagation time and in my test t should be 0.1s, 0.2s 0.3s and 0.4s. The analytical solution is shown in figure 2 and only the angles between -90° and 90° are shown because we just discuss the one way operator here.

The CPU version FI method takes 9388s on my computer while the GPU version only takes 0.3s on T1070. This explains why the CPU version FI method has not been used for decades of years. As a comparison, the 65° FD, 65° FFD, SP and PSPI methods take 10s, 13s, 7s and 81s on my CPU respectively. The time comparison is shown in figure 5.

Figure 2 shows the impulse responses of all the methods mentioned above. The FD and FFD methods suffer from the problem of numerical dispersion; the result of SP method is inaccurate in large propagation angles; the results of the GPU version FI and PSPI methods are similar and accurate nearly in 90° propagation angles but the computation cost and memory occupation of the PSPI method are too high.

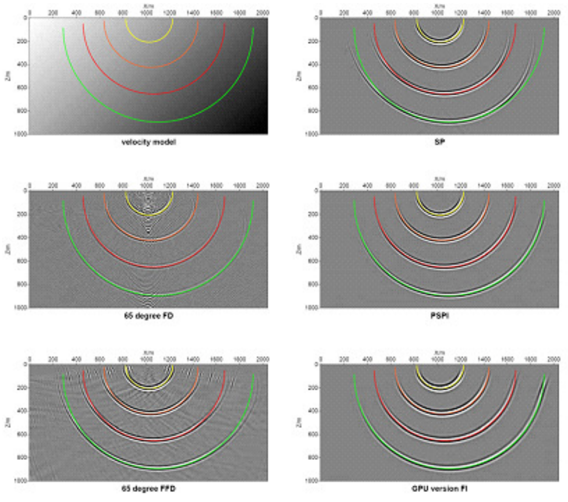
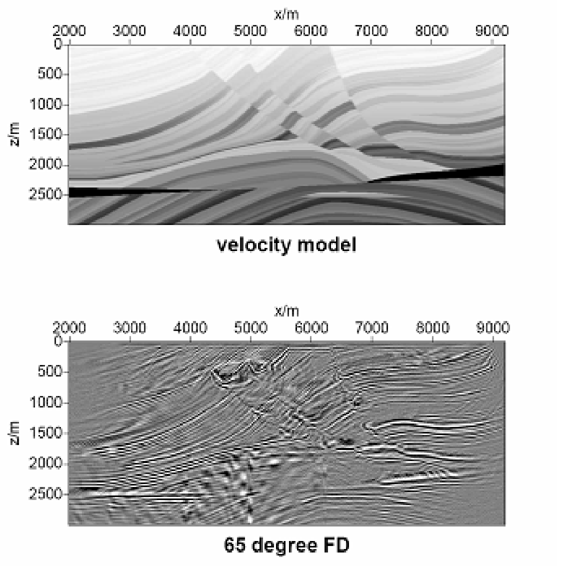


Figure 2: From top to bottom, the left column is the velocity model, the result of 65° FD and 65° FFD methods; the right column is the result of SP, PSPI and GPU version FI method. For comarision, the analytical solutions are drawn overhead in color and the yellow, orange, red and green lines are analytical wavefronts at t=0.1s, 0.2s, 0.3s and 0.4s, respectively.

Marmousi model

The marmousi model is shown in figure 3. There are 240 shots totally. I perform the GPU version FI method on T1070 and the other one way wave methods on my CPU. The computation cost is shown in figure 5 and the migration results are shown in figure 3. The result of FI method is better than the other methods because FI uses the exact extrapolation operator and the exact velocity model without any approximations or assumptions.



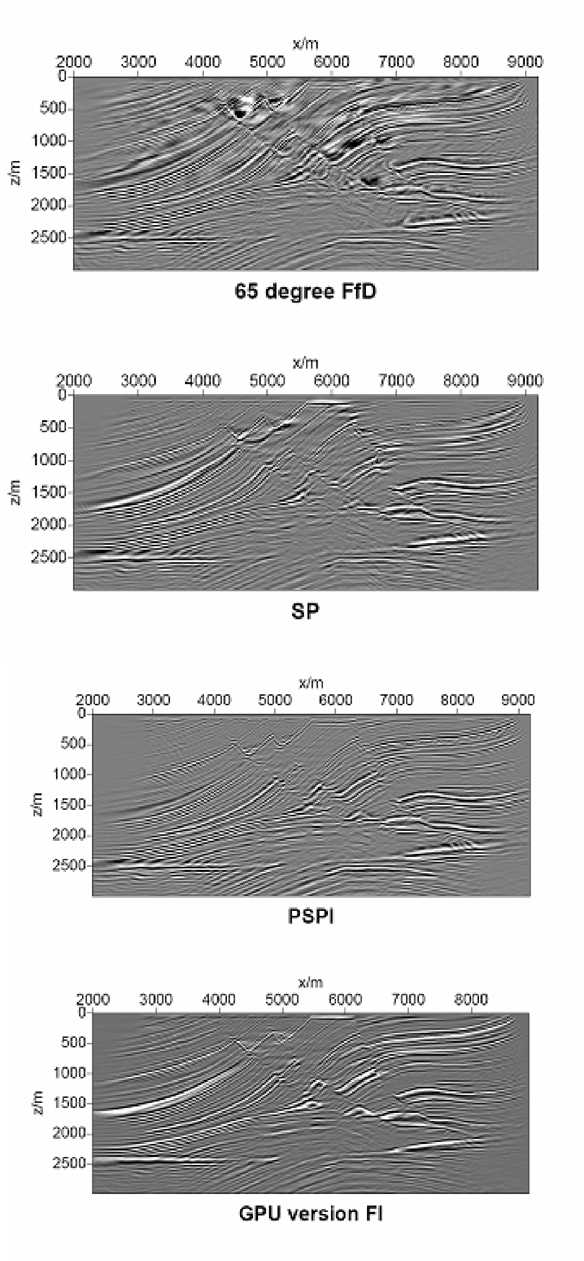
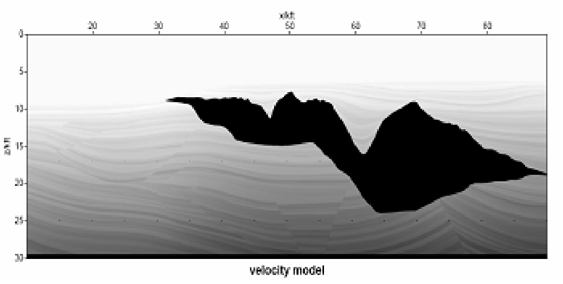


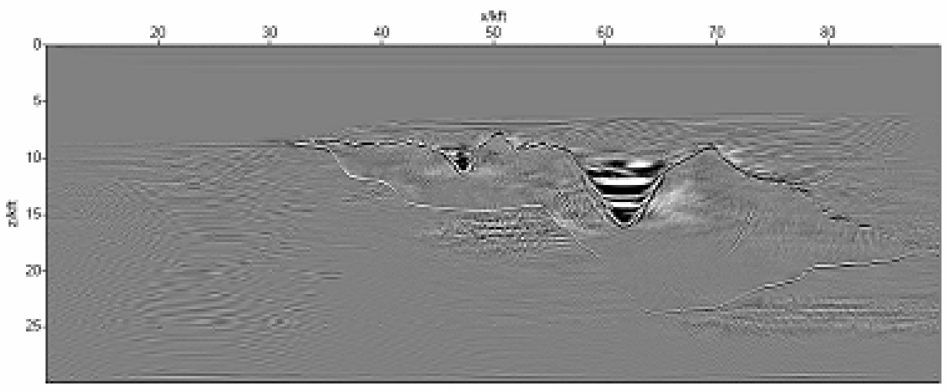
Figure 3: From top to bottom is the velocity model, the result of 65° FD, 65° FFD methods, the result of SP, PSPI and GPU version FI method. The result of FI method is better than the other one way wave methods.

I also test the CPU version of FI for only one shot and it takes 101658s (28h14m18s) while the GPU version only takes 3s and the speedup ratio is 33886!

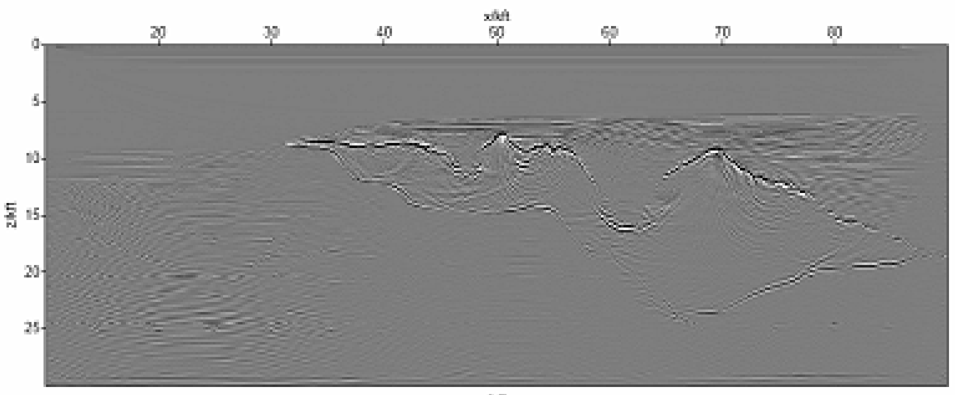
Sigsbee synthetic model

The sigsbee 2a model is shown in figure 4. There are 500 shots totally. I perform the GPU version FI method on T1070 and the other one way wave methods on my CPU. The computation cost of one shot is shown in figure 5 and the migration results are shown in figure 3 (only the SP and FFD results are shown, for comparison, the reverse time migration (RTM) result is also shown). The result of FI method is better than the other one way wave methods and even comparable to the RTM result.

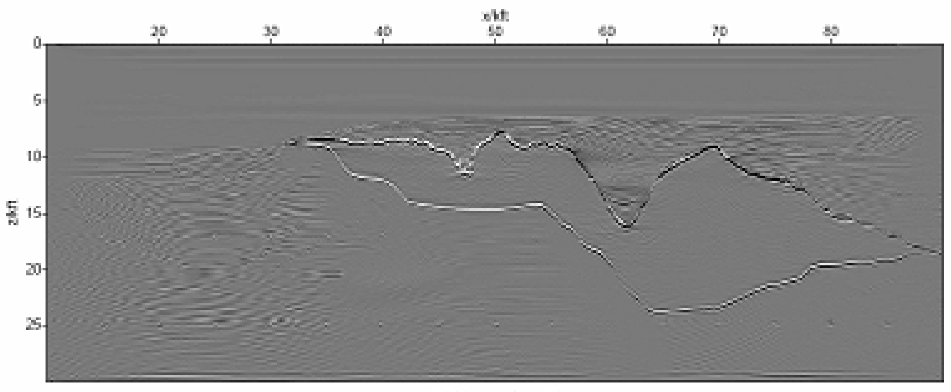




65 degree FFD



SP



GPU version FI

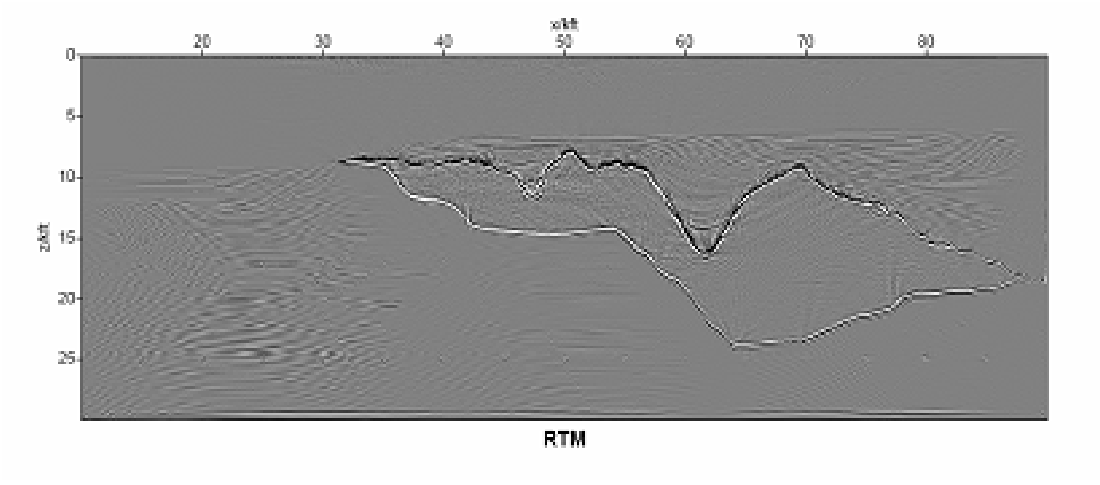


Figure 4: From top to bottom is the velocity model, the result of 65° FFD method, the result of SP method; the result of the GPU version FI method and the RTM result after denoise. The result of FI method is better than the other one way wave methods and even comparable to the RTM result.

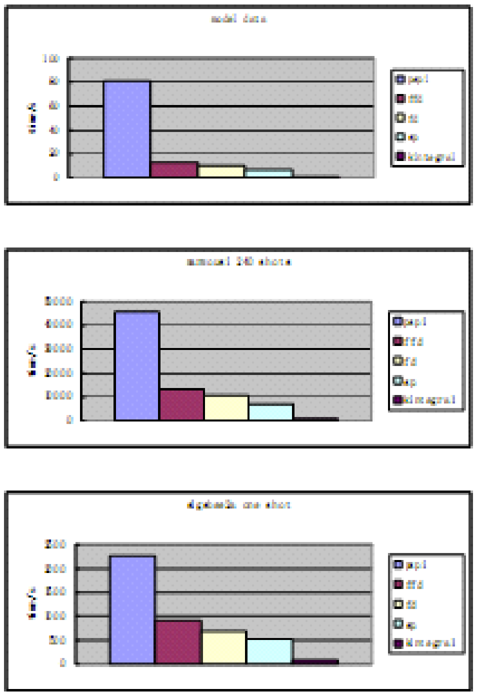


Figure 5: From top to bottom is the computation cost of the impulse responses, the marmousi model and the sigsbee model.

Conclusions

The reason why the FI method has not been used is that the computation cost is too high because of the large number of matrix multiplication. The CUDA based GPU perfectly fulfills this job and the speedup ration is tens of thousands times. As a result, the speed of the GPU version FI method is faster than almost all the traditional one way wave methods which results in a practical implementation of this method.

The impulse response example shows that the FI method can accurately model wave field propagation with nearly 90 degree propagation angles in the presence of severe lateral and vertical velocity variations. Compared to the traditional one way wave methods, the FI method does not suffer from the problem of numerical dispersion and numerical stability because we do not make use of any approximations to the extrapolation operator. The synthetic examples show that the migration effect of the FI method is better than the other one way wave methods and comparable to the RTM method even in the subsalt area. We would conclude that RTM is not the only game in town.

The FI method could easily be extended to 3D case and we will do this next. Besides, there are many other applications where the FI methods could be used, such as surfacerelated multiple elimination (SRME) algorithm and the calculation of theoretical seismogram. We will do these jobs in the future.

Acknowledgements

This research was supported by National Natural Science Foundation of China (Grant No. 40830424), National Hitech Research and Development Program of China (863 Program) (Grant No. 2006AA09A102-08), National Basic Research Program of China (the 973 Program. Grant No. 007CB209603).

We would also like to thank the SmartJV for the use of the Sigsbee synthetic data.

References

Clearbout, J. F, 1985, Imaging the Earth's interior: Oxford, England: Blackwell Science Publishers.

CUDA Programming Guide, version 2.3.1. NVIDIA Corporation, August 26th, 2009.

Gary F. Margrave and Robert J. Ferguson, 1999, Wavefield extrapolation by nonstationary phase shift: Geophysics, 64 , no. 4, 1067-1078.

Gazdag, J., 1978, Wave equation migration with the phaseshift method: Geophysics, 43, 1342–1351.

Gazdag, J., and P. Sguazzero, 1984, Migration of seismic data by phase shift plus interpolation: Geophysics, 49, 124–131.

Le Rousseau, J. H., and M. V. de Hoop, 2001, Modeling and imaging with the scalar generalized-screen algorithms in isotropic media: Geophysics, 66, 1551–1568.

Nolet G, 1998, Seismic Tomography with Applications in Global Seismology and Exploration Geophysics: Boston: D. Reidel Publishing Company, 323-337.

NVIDIA S1070 Computing System. http://www.nvidia.com/object/product_tesla_s1070_us.html

Ristow, D., and T. Rühl, 1994, Fourier finite-difference migration: Geophysics, 59, 1882–1893.

Robert J. Ferguson and Gary F. Margrave, 2002, Prestack depth migration by symmetric nonstationary phase shift: Geophysics, 67 , no. 2, 594-603.

Stoffa, P. L., J. T. Fokkema, R. M. de Luna Freire, and W. P. Kessinger, 1990, Split-step Fourier migration: Geophysics, 55, 410–421.

Stolt, R. H., 1978, Migration by Fourier transform: Geophysics, 43, no.1, 23-48.

Xie, X. B., and R. S. Wu, 1998, Improve the wide angle accuracy of screen method under large contrast: 68th Annual International Meeting, SEG, Expanded Abstracts, 1811–1814.