PROCEEDINGS of the
23rd International Congress on Acoustics

9 to 13 September 2019 in Aachen, Germany

Three-dimensional acoustic parabolic equation model based on GPU processing

Keunhwa Lee[1]; Woojae Seong[2]; Youngnam Na[3]

[1] Sejong University, South Korea

[2] Seoul National University, South Korea

[3] Agency for the Defense Development, South Korea

ABSTRACT

We introduce a three-dimensional acoustic parabolic equation (PE) model based on GPU processing. This model is developed by splitting the full exponential operator in the three-dimensional split-step marching solution in the Cartesian coordinate. Each exponential operator is approximated by the Taylor series and the sum representation of Padé approximation, which gives an opportunity for GPU parallel processing of the 3D PE solver. An advantage of this model is to implement higher-order cross-terms appearing to the operator approximation efficiently. In the talk, we will show the computational performance of GPU based PE code with the comparison of a traditional code for several ocean environments

Keywords: Three-dimensional parabolic acoustic equation (3DPE), GPU, Parallel algorithm

1. INTRODUCTION

Recently, some of the three-dimensional acoustic parabolic equation (3D PE) models considering the cross-terms in the two-dimensional (2D) square root operator have been presented. While these models have advantages in the accuracy and the c_0-insensitivity compared with the existing 3D PE model without the cross-terms, the total calculation time increases more due to the calculation of the cross-terms and the use of filter. Fortunately, the PE algorithm is known to be compatible with the parallel processing inherently.

In this proceeding, in order to improve the efficiency of the 3DPE model, we present a 3DPE formulation proper to the GPU processing and the computational strategy. Specifically, we introduce the 3DPE model based on Lee *et al.* approach and its parallelized formulation, known as the sum representation of the operator approximation. Moreover, we suggest a strategy for the matrix multiplication and inversion using the cuSparse library in CUDA/C. With these two approaches, the total calculation performance increases about 5 ~ 10 times.

2. METHOD

2.1 3D PE solver with cross-terms in the Cartesian coordinate

When the marching direction is set to be x-axis, the 3D PE solver with cross-terms is given by

$$u(x+\Delta x)=\prod_{i=1}^{p}\frac{1+\alpha_{i,p}Y}{1+\beta_{i,p}Y}\prod_{i=1}^{p}\frac{1+\alpha_{i,p}Z}{1+\beta_{i,p}Z}\sum_{i=0}^{p_{yz}}\Im_i u(x)\,. \tag{1}$$

Here, the variable u is defined by $u=pe^{-jk_0x}/\alpha$ with the acoustic pressure p, the reference

[1] nasalkh2@sejong.ac.kr
[2] wseong@snu.ac.kr
[3] ynna@add.re.kr

wavenumber k_0, and the energy-conserving parameter α. p is the order of Padé approximation, $\alpha_{i,p}$ and $\beta_{i,p}$ are the Padé coefficients and p_{yz} is the order of Taylor approximation. $\Im_i$ denotes the i[th] order cross-term of Taylor series. Here, Y and Z operators are defined as

$$Y = k_0^{-2}\frac{\rho}{\alpha}\frac{\partial}{\partial y}\left(\frac{1}{\rho}\frac{\partial}{\partial y}\alpha\right), \tag{2}$$

$$Z = (n^2-1) + k_0^{-2}\frac{\rho}{\alpha}\frac{\partial}{\partial z}\left(\frac{1}{\rho}\frac{\partial}{\partial z}\alpha\right), \tag{3}$$

where $n = k(y,z)/k_0$ is the refractive index where k is the complex wavenumber. These operators can be transformed into the tridiagonal matrix using the second-order finite difference scheme.

Figure 1 illustrates the marching algorithm of the 3D PE solver, regarded as the solution of the initial value problem. When the pressure field at x = x is known, the pressure field at the next step can be predicted with Eq. (1). In Eq. (1), the multiplication operator results from the Padé approximation of the exponential operator to the product representation form.

The sum representation of Eq. (1) is derived as

$$u(x+\Delta x) = \left(1+\sum_{i=1}^{n}\frac{\hat{\alpha}_{i,n}Y}{1+\hat{\beta}_{i,n}Y}\right)\left(1+\sum_{i=1}^{n}\frac{\hat{\alpha}_{i,n}Z}{1+\hat{\beta}_{i,n}Z}\right)\left(1+\sum_{i=1}^{n_{yz}}\sum_{j=1}^{n_{yz}}\hat{c}_{ij,n}Y^iZ^j\right)u(x), \tag{4}$$

where n is the order of Padé approximation in the sum representation, $\hat{\alpha}_{i,p}$ and $\hat{\beta}_{i,p}$ are the Padé coefficients and n_{yz} is the order of Taylor approximation. $\hat{c}_{ij,n}$ is the coefficient of Taylor series. In figure 2, the flow chart for the calculation of Eq. (4) is given. For convenience, the precise flow chart for the cross-term operation is omitted. Operations in each bracket of Eq. (4) can be performed in parallel. Theoretically, the calculation time in each bracket may be reduced by approximately 1/n or $1/n_{yz}^2$. This kind of parallelization concept has been already applied into several 2D PE or 3D PE solvers.

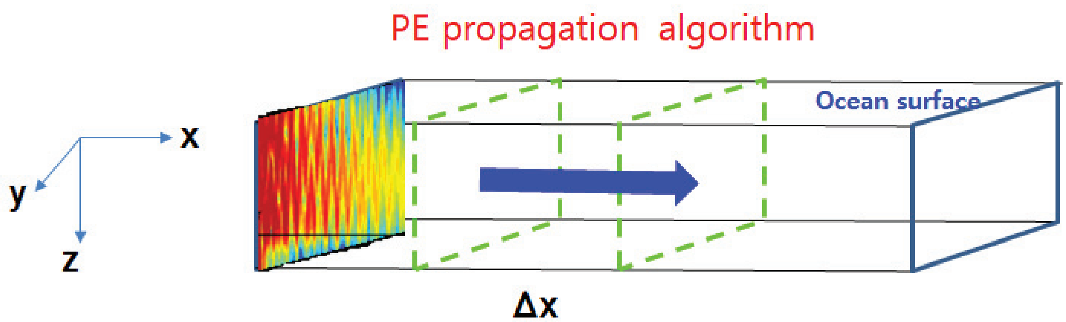

Figure 1 - Illustration of the 3D PE algorithm in the Cartesian coordinate.

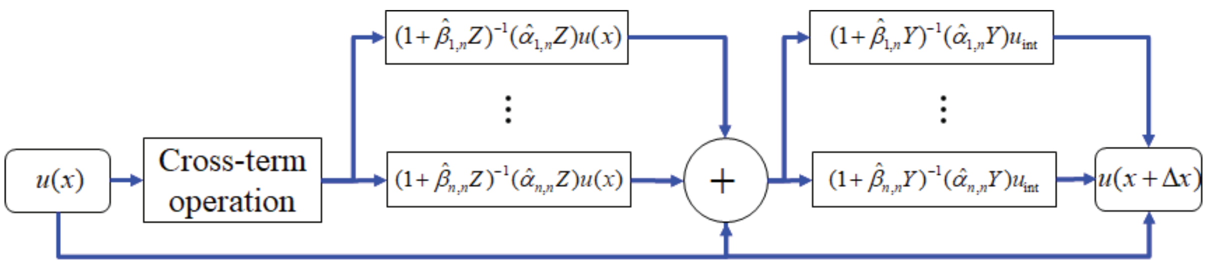

Figure 2 – Flow chart of the 3D PE algorithm in the sum representation.

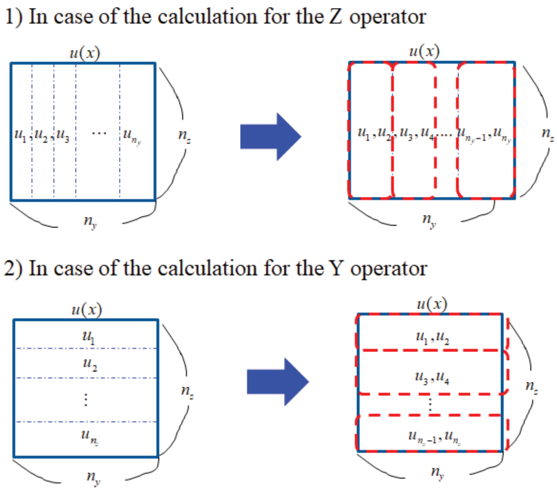

Figure 3 - Grouping of rows and columns for parallel processing.

More parallelization can be achieved for each tri-diagonal matrix multiplication and inversion in figure 2. For example, for the calculation of $(1+\hat{\beta}_{i,n}Y)^{-1}(\hat{\alpha}_{i,n}Y)u(x)$, the n_z number of tri-diagonal matrix multiplication and inversion are required when the size of the discretized u(x) is set to be $n_z \times n_y$. Since the matrix calculation for each row of u(x) is independent one another, these operations can be parallelized like those for Padé order. The same is true for the Z operator. As shown in figure 3, we assume to divide the u(x) matrix into the sub-matrices. When the numbers of the sub-matrix are set to be N_{sz} in the z-direction and N_{sy} in the y-direction, totally the $(n \times N_{sz})$ or $(n \times N_{sy})$ loops will be performed in parallel. Moreover, the inversion of a tri-diagonal matrix can be fast implemented with parallel algorithms like parallel cyclic reduction and recursive doubling.

The CUDA/C library is used to code the suggested 3DPE algorithm in parallel. The matrix inversion is performed by the function of cuSparse<t>gtsvStrideBatch in CUDA Toolkit 4.2 cuSparse library. The solver runs on a desktop with Intel i7-6700k quad-core processor, 56 GB RAM Intel and a GeForce GTX 1080 GPU with 20 multiprocessors.

3. CONCLUSIONS

In this proceeding, the parallelized 3DPE solver using GPU based on CUDA library is presented. The parallelization is carried out through two steps-the sum representation of the operator approximation and the grouping of vectors in the matrix. With the suggested algorithm, the execution time of the 3DPE is shortened to 1/5 ~ 1/10.

ACKNOWLEDGEMENTS

This work was supported by the Agency for Defense Development, South Korea, under Grant No. UD170018DD, and by the KRISO, South Korea.

REFERENCES

1. Lin Y, Collis J. M, Duda T. F. A three-dimensional parabolic equation model of sound propagation using higher-order operator splitting and Pade approximationts. J Acoust Soc Am. 2012;132(5):EL364-EL370.
2. Sturm F. Leading-order cross term correction of three-dimensional parabolic equation models. J Acoust Soc Am. 2016;139(1):263-270.
3. Castor K, Sturm F. Investigation of 3D acoustic effects using a multiprocessing parabolic equation based algorithm. J. Comput. Acoust. 2008;16(2):137-162.
4. Lee K, Seong W, Na Youngnam. Three-dimensional Cartesian parabolic equation model with higher order cross-terms using operator splitting, rational filtering, and split-step Padé algorithm. J Acoust Soc Am. 2019 (accepted).
5. NVIDIA, CUDA Toolkit 4.2 CUSPARSE Library. PG-05329-041_v01. February 2012.