Developing an alternating direction explicit-implicit domain-decomposition approach to solve heat transfer equation on graphics processing unit

Ali Foadaddini[1], Seyed Alireza Zolfaghari[2*], Hossein Mahmoodi Darian[3]

[1] Ph.D. Student, Mechanical Engineering, University of Birjand, Birjand, Iran
[2] Associate Professor, Department of Mechanical Engineering, University of Birjand, Birjand, Iran
[3] Assistant Professor, School of Engineering Science, College of Engineering, University of Tehran, Tehran, Iran

ABSTRACT

In the present study, a new alternating direction explicit-implicit domain decomposition approach is proposed by combining the alternating direction implicit method with the explicit-implicit domain decomposition method. The method is used for solving the two-dimensional conduction heat transfer equation on a graphics processing unit. In this method, an explicit numerical scheme is used to predict values at the inner boundaries and an implicit scheme based on the alternating direction implicit method is used to solve the sub-domains. Then, an implicit scheme is used to correct the values on the inner boundaries. Numerical experiments are done to investigate the accuracy and speed of the method. The results show that the present method can achieve a speedup of 1.3 to 2.6 times compared to the alternating direction implicit method. Increasing the number of subdomains increases the speed and decreases the accuracy of the method. Although numerical experiments show high stability of the present method, its error is higher than the alternating direction implicit method. Furthermore, the results show that the present method is more advantageous to problems with coarse grids, such that by increasing the grid size from 256 × 256 to 512 × 512, the speedup decreases from 2.4 to 1.7.

KEYWORDS

Computational Fluid Dynamics, Parallel Processing, Graphics processing unit, Alternating direction implicit method, Corrected explicit-implicit domain decomposition algorithm

1. Introduction

The two-dimensional heat conduction equation is one of the most commonly used equations in the field of mechanical engineering. Due to the high computational complexity and hug computational cost in long term and real-scale problems, there is a pressing need to develop fast and accurate GPU-accelerated solvers for this equation. Due to their special architecture, GPUs become more effective when the given problem can be decomposed to many tasks performing same operation on multiple data simultaneously. Regarding the methods of time advancement, such a condition occurs when the explicit numerical schemes are used for solving PDEs. So, many researchers have developed GPU-accelerated solvers based on explicit numerical schemes [1, 2]. Alternating direction implicit (ADI) method is a simple method for dealing with this problem. In ADI method, solving tridiagonal matrix equation is the main building block of the algorithm. So, for the efficient implementation of the ADI solver on GPU, solution method for tridiagonal matrix equation is the key part.

For solving tri-diagonal systems on GPU, serial algorithms like Thomas [3, 4] and parallel algorithm like CR and PCR [5] have been proposed. The Thomas algorithm has a low computational complexity and it's easy to implement. But because of the serial nature of the algorithm, computation related to each tri-diagonal system is mapped to one thread which leads to a low occupancy especially in small scale problems. Unlike the Thomas algorithm, In CR and PCR algorithm each equation is mapped to one thread. So, the number of resident warp is increased and this leads to higher occupancy and efficient use of resources. But parallel algorithms suffer from high computational complexity Because of the high number of mathematical operations per equation and communications between threads.

In the present study, a corrected explicit-implicit domain decomposition (CEIDD) approach is proposed to reduce the size of the tridiagonal system of equations in the ADI method. In this method in each time step values at some points in the domain are predicted using an explicit scheme. By doing this, each tridiagonal system is partitioned to many smaller systems and this allows for partitioning the workloads. Explicit-implicit domain decomposition is a non-iterative and non-overlapping domain decomposition method so it is computationally and computationally efficient. In 2002. Du et al. [6] proposed an EIDD method which can compute the values on the inner interfaces by either a high-order explicit scheme or multistep explicit scheme. The method is conditionally stable. Sun and Zhuang [7] added a further step to EIDD method to replace the values from the explicit scheme on interfaces by values computed by an implicit scheme. This method is unconditionally stable. More related to the present study, Du and Liang [8] combined EIDD algorithm with splitting technique and proposed S-DDM method. In this method the interface values are computed using local multilevel schemes and the splitting implicit scheme is utilized to compute the interior values of the subdomains. S-DDM is conditionally stable but using an efficient local multilevel scheme at interface points relax the stability condition [9]. In the present study, a new alternating direction explicit-implicit domain decomposition approach (ADI-CEIDD) is proposed by combining the ADI method with the CEIDD domain decomposition method.

2. Methodology

The ADI-CEIDD method consists of two steps: y-sweep and x-sweep. In y-sweep step the domain is decomposed to *nos* subdomains in Y direction (Figure (1-a)). Then the solution for interface points is predicted using an explicit scheme and a combination of values from two previous time levels. Next, the heat conduction equation is solved implicitly in Y direction and explicitly in X direction in the subdomains. Finally the values in boundary points are corrected using an implicit scheme. In x-sweep the domain is decomposed to *nos* subdomains in X direction (Figure (1-b)) and the equation is solved implicitly in X direction and explicitly in Y direction. In ADI-CEIDD the number of independent systems of equations is several times the ADI method. So, the number of active threads is increased and this can lead to efficient use of GPU resources.

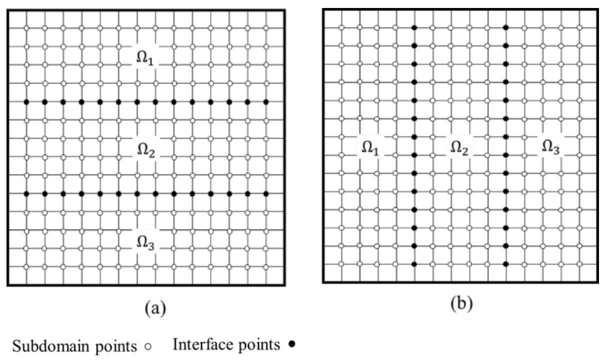

Figure 1: Domain decomposition in X and Y direction in ADI-CEIDD

3. Results

Figure (2) shows the L2-norm error for ADI-CEIDD method. The results show that decomposing the domain in ADI-CEIDD has increased the error compared to the ADI method. The minimal effect of the parameter *nos* has occurred in $\lambda = (\Delta t.k)/h^2 = 1$. As an overall trend, the error increases by increasing the λ. But in nos=4, 8, 16 the error for λ=0.5 is greater than the error for λ=1. Also, the results indicate the high stability of ADI-CEIDD method.

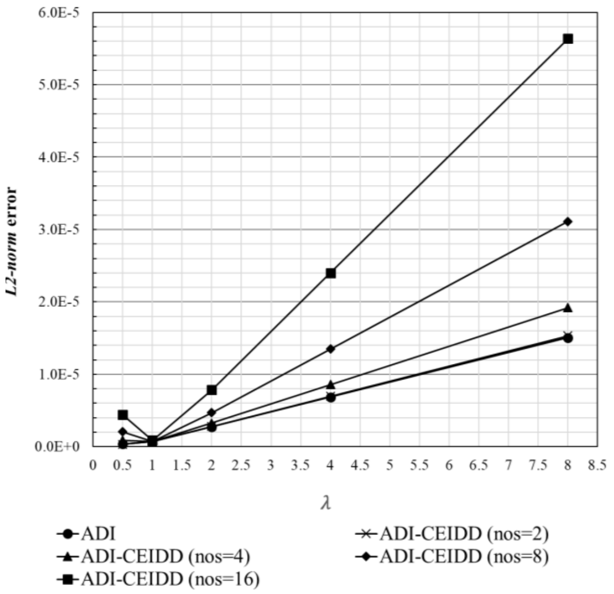

Figure 2: $L_2 - norm$ error for a 256*256 grid

For further optimization of the algorithm, coalesced and uncoalesced version is investigated. Results show that the coalesced version is more efficient especially for big-scale problems.

Speedup of the coalesced version of ADI-CEIDD method versus the ADI method is presented in Figure (3).

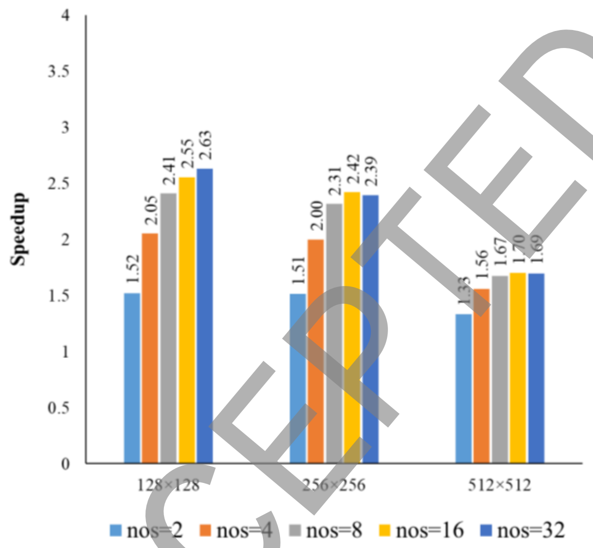

Figure 3: speedup of the ADI-CEIDD algorithm vs ADI

As the results show, the ADI-CEIDD can achieve a speedup of 1.3 to 2.6 times compared to the ADI method by increasing the occupancy. By increasing the number of subdomains from 2 to 32, the speed of the proposed method is increased up to 1.6 times. Furthermore, the results show that the ADI-CEIDD method is more advantageous to problems with coarse grids, such that by increasing the grid size from 256 × 256 to 512 × 512, the speedup decreases from 2.4 to 1.7.

4. Conclusion

In the present study, a new ADI-CEIDD approach is proposed by combining the ADI method with the explicit-implicit domain decomposition method. ADI-CEIDD can achieve a speedup of 1.3 to 2.6 times compared with the ADI method by increasing the occupancy. Increasing the number of subdomains improves the performance of the ADI-CEIDD. Although numerical experiments show high stability of the ADI-CEIDD, its error is higher than the ADI. Furthermore, the results show that the ADI-CEIDD method is more advantageous to problems with coarse grids

5. References

[1] F. Salvadore, M. Bernardini, M. Botti, GPU accelerated flow solver for direct numerical simulation of turbulent flows, *Journal of Computational Physics*, 235 (2013) 129-142.

[2] S. Vanka, A.F. Shinn, K.C. Sahu, Computational Fluid Dynamics Using Graphics Processing Units: Challenges and Opportunities, Fluids and Thermal Systems; *Advances for Process Industries*, (2011) 429-437.

[3] G. Alfonsi, S.A. Ciliberti, M. Mancini, L. Primavera, GPGPU implementation of mixed spectral-finite difference computational code for the numerical integration of the three-dimensional time-dependent incompressible Navier–Stokes equations, *Computers & Fluids*, 102 (2014) 237-249.

[4] N. Sakharnykh, Tridiagonal solvers on the GPU and applications to fluid simulation, in: *NVIDIA GPU Technology Conference*, San Jose, California, USA, 2009, pp. 17-19.

[5] S. Ha, J. Park, D. You, A GPU-accelerated semi-implicit fractional-step method for numerical solutions of incompressible Navier–Stokes equations, *Journal of Computational Physics*, 352 (2018) 246-264.

[6] Q. Du, Mu, M, Wu, Z N, Efficient parallel algorithms for parabolic problems, *SIAM Journal on Numerical Analysis*, 39(5) (2002) 1469-1487.

[7] X.-h. Sun, Y. Zhuang, stablized explici-implicit domain decomposition method for the numerical solution of finit difference equation, *SIAM Journal on Numerical Analysis*, 24(1) (2002) 335-358.

[8] C. Du, D. Liang, An efficient S-DDM iterative approach for compressible contamination fluid flows in porous media, *Journal of Computational Physics*, 229(12) (2010) 4501-4521.

[9] Z. Zhou, D. Liang, Y. Wong, The new mass-conserving S-DDM scheme for two-dimensional parabolic equations with variable coefficients, *Applied Mathematics and Computation*, 338 (2018) 882-902.