Performance evalution of multi-level parareal method

Koki Hiromi
Kogakuin University, Japan
jx20227@ns.kogakuin.ac.jp

Akihiro Fujii
Kogakuin University, Japan
fujii@cc.kogakuin.ac.jp

Teruo Tanaka
Kogakuin University, Japan
teru@cc.kogakuin.ac.jp

1 Introduction

Simulation is generally parallelized in the spatial direction, but this has limitations and does not take full advantage of computer performance. Parallelism in time direction for computer simulation has attained research attention, and the research area is studied actively. Parareal is a popular algorithm, and it is usually implemented as a 2-level algorithm. In this study, we implemented the 3-level Parareal algorithm[1] with recursive call and nested OpenMP parallelism. The performance is evaluated by each execution time.

2 Parareal method

The Parareal method first performs coarse solver to calculate each initial point of a time step segment, and then calculates time step segments in parallel with a fine solver. Next, it corrects the solution by differences between fine and coarse solutions. It repeats the iteration until convergence. Error correction and coarse solver must be done sequentially. This sequential part becomes an overhead in parallel execution time.

3 multi-level Parareal method

We investigate the effectiveness of 3-level implementation in comparison with 2-level Parareal method. Since 3-level method has a coarser and smaller level than 2-level method, it is supposed to have less sequential time. Our 3-level method combines a 2-level Parareal algorithm and the coarser level. 3-level method regards the 2-level Parareal solution as a fine solution and it will be corrected by the coarser level. A multi-level version can be implemented recursive calls. Also, parallelization in this study used OpenMP. Nested parallelism is activated in the OpenMP.

4 Experiments and results

In this study, the 2-level and 3-level Parareal method are implemented for a one-dimensional thermal diffusion problem based on an explicit method. The problem has 10000 DOF in spatial direction and 1000 time steps. Supercomputer Wisteria-O at Tokyo University was used. 2-level Parareal method has 1000 time steps on the fine level, and 100 time steps on the coarse level. 3-level Parareal method has 1000 time steps at the finest level, and 100 time steps at second coarser level, and 10 time steps at the coarsest level. Parallel threads were set to 1, 10, 20, 30, 40, 50. In 2-level method, fine level is parallelized by the number of threads, and coarse level is calculated sequentially. As for 3-level Parareal method, finest level calculation is parallelized by all threads. Second coarse level is parallelized by 10 threads. Third level is calculated sequentially.

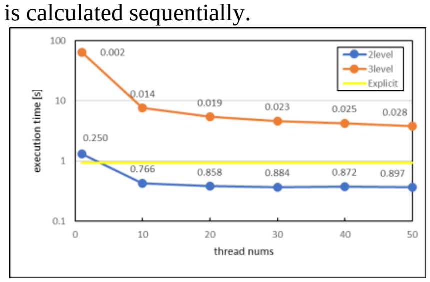

Figure 1. Execution time with changing number of threads

Figure 1 shows the execution times for each of the explicit method, 2-level, and 3-level Parareal methods. The numbers above the 2-level and 3-level graphs represent the ratio of the execution time of the sequential part to the overall execution time of each. Both the 2-level and 3-level methods were able to accelerate up to 10 threads in parallel, but beyond that, there was little change. This is because coarser level parallelism shrinks, and because the coarser levels cost is more expensive than fine levels. Coarser levels uses implicit method and the fine level uses explicit method. 3-level method was slower than 2-level method, because of the iteration number. The number of iterations increased to for the Parareal method was 2 for 2-level and 4 for 3-level. The percentage of execution time for the sequential part was reduced as targeted. The sequential execution time per iteration was 0.15 sec for 2-level and 0.02 sec for 3-level at 50 threads. The ratio is close to an one-tenth, which is the ratio between the coarsest level sizes in 2-level and 3-level methods.

5 Conclusion

The three-level implementation of the Parareal method with recursive calls was effective in reducing sequential time. However, it was not faster than the 2-level Parareal method. We will proceed with performance analysis for problems with a larger number of time steps and different size ratios between levels, and present the results in a poster.

ACKNOWLEDGEMENTS

This work is supported by JSPS KAKENHI Grant Number JP22K12063, "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures" and "High Performance Computing Infrastructure" in Japan (Project ID: jh230060-NAH).

REFERENCES

[1] Rosemeier, Juliane, Terry Haut, and Beth Wingate. "Multi-level Parareal algorithm with Averaging." arXiv preprint arXiv:2211.17239 (2022).