ParCFD'2020
32^{nd} International Conference on Parallel Computational Fluid Dynamics
May-11-13 2020, Nice FRANCE

A HIERARCHICAL WAVEFRONT METHOD FOR LU-SGS ON MODERN MULTI-CORE VECTOR PROCESSORS

Yuta Hougi*, **Kazuhiko Komatsu***, **Osamu Watanabe***, **Masayuki Sato***, **Hiroaki Kobayashi***

* Tohoku University
Sendai, Miyagi, Japan
e-mail: {yuta.hougi.t8@dc.,komatsu@,osamu.watanabe.t5@dc.,masa@,koba@}tohoku.ac.jp

Key words: LU-SGS, Wavefront, Hyperplane, Multi-core vector processors

Abstract. Modern processors tend to employ a multi-core architecture with a longer vector processing capability. To take advantage of these processors' potential, the parallelization and vectorization are mandatory. Although the lower-upper Symmetric-Gauss-Seidel method (LU-SGS) is widely used in the field of CFD, its strong data-dependencies among grid points prevent from parallelization and vectorization. This paper proposes a hierarchical wavefront method that combines the wavefront and the hyperplane methods to effectively perform both parallel and vector processing of 3D LU-SGS. The evaluation results show that the hierarchical wavefront method is 6.9 times faster than the conventional wavefront method at maximum.

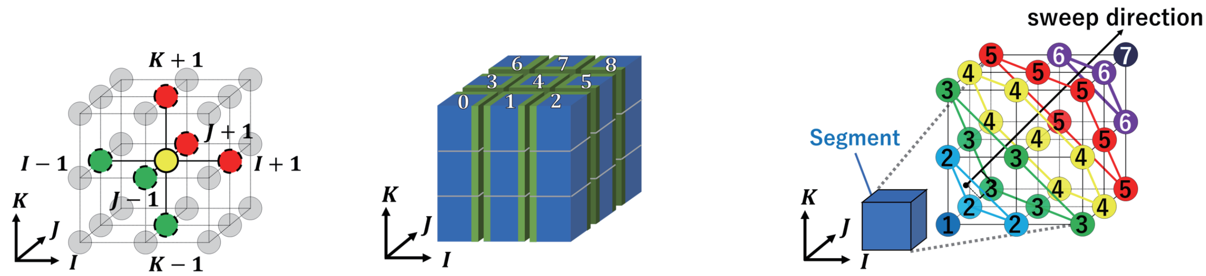
1 INTRODUCTION

Since the performance requirement of CFD dramatically increases, it is necessary to exploit the potential of modern processors that employ a multi-core architecture with a longer vector processing capability. The new generation vector supercomputer SX-Aurora TSUBASA [1] that can handle a very long vector of 256 elements at a time is expected for its high sustained performance by parallel execution with multiple cores and vector computation.

The goal of this paper is to accelerate a CFD program by taking account of the modern computer architecture. To this end, this research aims to parallelize and vectorize an application based on the LU-SGS [2] that is widely used because of its good convergence. The obstacle for its parallelization and vectorization is its strong data-dependencies among adjacent grid points. To enhance the performance of 3D LU-SGS, this paper proposes a hierarchical wavefront method so as to exploit the potential of modern processors. The proposed method hierarchically adapts the wavefront method [3] for parallelization and the hyperplane method [4] for vectorization.

2 THE LU-SGS METHOD

The LU-SGS method is one of the typical implicit methods to solve the Navier-Stokes equation with a high convergence and reliability. LU-SGS uses unknown variables of adjacent grid points at the current time step to obtain the physical quantities at a grid



(a) Referenced neighbors. (b) Domain decomposition. (c) Inner wavefront.

Figure 1: The LU-SGS dependencies and the proposed method.

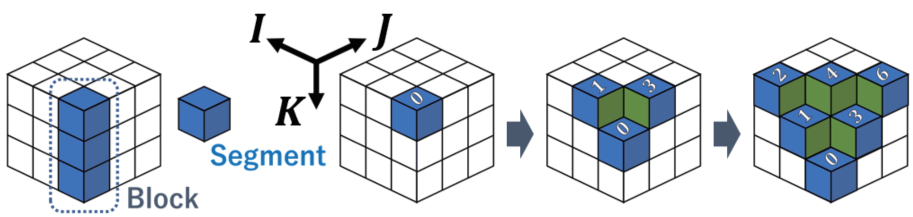


Figure 2: Outer wavefront.

point. By repeatedly updating the values of the current time step, the implicit method can solve the problem faster than the explicit method.

However, there are strong data-dependencies among adjacent grid points to guarantee the convergence stability. Figure 1 (a) shows the data-dependencies of LU-SGS. The calculations of physical quantities at the yellow point require those of the green points in the same sweep of the current time step and red points of the previous time step. The most naive method of the sweep is to calculate the values of grid points sequentially in the I direction. Although this method does not violate the data-dependencies shown in Figure 1 (a), it is difficult to parallelize and vectorize this method.

3 HIERARCHICAL WAVEFRONT METHOD

This paper proposes a hierarchical wavefront method that can parallelize and vectorize LU-SGS by hierarchically applying the wavefront method and the hyperplane method. Both basic ideas of these two methods are the same in which they calculate simultaneously multiple elements on a special plane called wavefront or hyperplane. Therefore, the wavefront and the hyperplane are hereinafter referred to as the outer wavefront and the inner wavefront, respectively. The proposed method uses the outer wavefront for coarse-grained parallel processing and the inner wavefront for fine-grained vector processing.

Figure 1 (b) shows the domain decomposition of the proposed method. The large cubic area is the entire flow field of LU-SGS. In Figure 1 (b), the I-J plane is divided into nine blocks. Each block is assigned to each core for the parallel processing. In the boundary area between blocks shown in the green part, boundary data in adjacent blocks need to be exchanged. The blocks are further divided into *segments* in the K direction. In Figure 1 (b), each block is divided into three segments.

The outer wavefront performs parallel processing among these segments. Assuming the flat MPI, Figure 2 shows that only one segment in rank 0 is calculated at first, and

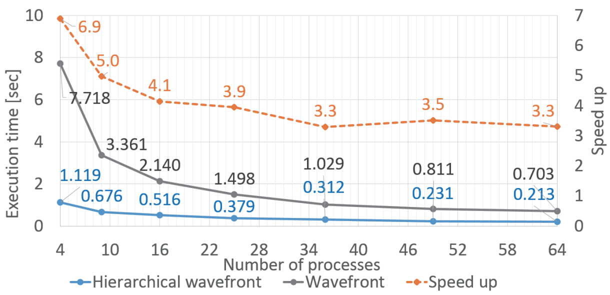


Figure 3: The execution time using the different number of processes.

Table 1: The profiled data of the proposed method.

$N_I \times N_J$	N_K	P_e	L_v	Hit_{cache}
2 × 2	70	100	203.4	91.7
3 × 3	40	73.6	205.4	96.6
4 × 4	60	54.2	176.0	94.9
5 × 5	30	47.2	187.2	98.1
6 × 6	40	39.9	168.5	98.6
7 × 7	30	39.6	167.7	97.6
8 × 8	30	32.9	160.0	97.8

the others are idle in the forward sweep. After the calculation of the first segment in rank 0, the process begins to send the boundary data of the calculated segment to the adjacent segments in ranks 1 and 3. Ranks 1 and 3 start calculating the first segments in each block after they receive the boundary data from rank 0. The second segment in rank 0 can be calculated as soon as the calculation of the first segment is finished, since the two segments belong to the same block. After the calculations of these three segments in parallel, these boundary data are sent to five segments that are shown in the green part of Figure 2. Next, six processes can calculate six segments in parallel. In the case of executing $N_I \times N_J = 3 \times 3 = 9$ processes, it can be executed in parallel with 9 processes when $N_K \geq 5$. In order to keep the parallel execution efficiency at a certain level, it is necessary to divide the blocks into many segments along the K direction.

Figure 1 (c) shows how a segment is vectorized by using the inner wavefront. The numbers in the grid points indicate the order of a sweep. Groups of the grid points with the same numbers/colors are called *hyperplanes*. In the calculations of the grid points in the m-th hyperplane, their adjacent points exist in the $(m-1)$-th and $(m+1)$-th hyperplanes. This avoids neighboring grid points from being calculated at the same time. Figure 1 (c) shows that the inner wavefront does not violate the data-dependencies among the hyperplanes of the sweep direction. Thus, the inner wavefront allows vector processing in the same hyperplanes. The maximum size of the hyperplanes is the product of the segment sizes in the I and J directions $(n_I \times n_J)$ and the vector length increases gradually like 1,3,6,7, ...,$n_I \times n_J$,...,$n_I \times n_J$... . Then, it decreases at the end of the segment. In order to maintain long vector lengths, sufficient segment sizes are required.

4 EVALUATIONS AND DISCUSSIONS

NEC SX-Aurora TSUBASA Vector Engine Type 10B and its Fortran compiler version 2.5.1 with an optimization option -O3 are used. The kernel used for this evaluation is the LU-SGS part of Numerical Turbine [5] that simulates the fluid around steam turbines. The problem size is fixed to $300 \times 300 \times 400$.

Figure 3 shows the execution time of the conventional wavefront method and the proposed hierarchical wavefront method using the different numbers of processes. The execution time and speed up in this figure are obtained by using the appropriate segment size found by exhaustively searching the various numbers of segments. *Wavefront* and *Hierar-*

chical wavefront indicate the conventional method and the proposed method, respectively. Figure 3 shows that the proposed method achieves 3.3 to 6.9 times faster computation than the conventional wavefront method. This is because the parallelization of segments and the vectorization inside of the segments by the proposed method can effectively exploit the potential of the multi-core vector processor. Table 1 shows the evaluation results when the number of divisions $N_I \times N_J$ and the number of segments N_K are changed. In this table, P_e, L_v, and Hit_{cache} mean parallelization efficiency compared with 4 processes, average vector length, and cache hit rate, respectively. The cache hit rates of the proposed method are high because of the high spatial locality in data reference.

However, the scalability of the proposed method decreases as the number of parallel processes increases. This is because the vector lengths become short as the size of the segments becomes small in the fixed size problem. Therefore, the segment size should be appropriately selected by carefully considering the trade-off between the degrees of parallel processing and vector processing to maximize its sustained performance.

5 CONCLUSIONS AND FUTURE WORK

Considering the trend in the number of cores and vector length of modern microprocessors, this paper proposes the hierarchical wavefront method for a 3D LU-SGS method by combining parallelization in the outer wavefront and vectorization in the inner wavefront. The evaluation results clarify that the proposed method can achieve 6.9 times faster computation than the conventional wavefront method.

As a part of future work, first, it is required to determine appropriate segment sizes for acquiring good performances. It is also necessary to clarify whether the proposed method is effective for larger problem sizes than that used in the evaluation and weak scaling problems that are expected to obtain higher scalability. Moreover, the proposed method should be examined regarding further improvement by reducing memory access time. This can be realized by a different data layout that can exploit the data access locality of LU-SGS.

REFERENCES

[1] K. Komatsu et al., Performance Evaluation of a Vector Supercomputer SX-Aurora TSUBASA, In *Proc. SC18*, pp. 685-696.

[2] Yoon, S. and Jameson, Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations, A., AIAA Journal, 26 (1988), p.1025.

[3] C. Gong et al., An efficient wavefront parallel algorithm for structured three dimensional LU-SGS, Computers & Fluids, 134-135 (2016), pp. 23-30.

[4] H. Matsuoka et al., Program optimization of Numerical Turbine for vector supercomputer SX-ACE, In *Proc. Parallel CFD2016*, p. 8, 2017.

[5] S. Yamamoto et al., Parallel computation of condensate flows through 2-d and 3-d multistage turbine cascades, In *Proc. Intl. Gas Turbine Cong., 2007.*