Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures
Final Report for JHPCN Joint Research of FY 2018, May 2019

jh180022-NAHI

Innovative Multigrid Methods

Kengo Nakajima（Information Technology Center, The University of Tokyo, Japan）

Abstract
In the present work, we are developing robust and efficient GMG and AMG methods, where we are focusing on development of algorithms for (1) efficient and robust smoother, (2) parallel global reordering/aggregation methods for robustness, (3) utilization of near-kernel vectors for robustness, and (4) hierarchical methods for scalability. Moreover, we develop new algorithms for Parallel-in-Space/Time (PinST).

1. Basic Information

(1) Collaborating JHPCN Centers

- The University of Tokyo (Oakforest-PACS)
- Hokkaido University (New System A)
- Kyushu University (ITO Subsystem-A)

(2) Research Areas

- ■ Very large-scale numerical computation
- □ Very large-scale data processing
- □ Very large capacity network technology
- □ Very large-scale information systems

(3) Roles of Project Members

- Kengo Nakajima (The University of Tokyo) (Co-PI) Administration, Applications, GMG, AMG, PinST, Hierarchical Methods
- Matthias Bolten (University of Wuppertal) (Co-PI) GMG, AMG, Smoother, Near-Kernel Vectors
- Takeshi Iwashita (Hokkaido University) PinST
- Akihiro Fujii (Kogakuin University) AMG, PinST, Near-Kernel Vectors
- Ryo Yoda (Kogakuin University) PinST
- Akihiro Ida (The University of Tokyo) GMG, AMG
- Masatoshi Kawai (RIKEN R-CCS) GMG, AMG, Smoother, Global Reordering
- Naoya Nomura (The University of Tokyo) AMG, Near-Kernel Vectors
- Satoshi Ohshima (Kyushu University) Code Parallelization, Profiling & Optimization
- Tetsuya Hoshino (The University of Tokyo) Code Parallelization, Profiling & Optimization, SELL-C-σ
- Toshihiro Hanawa (The University of Tokyo) Code Parallelization, Profiling & Optimization, Mesh Generation
- Gerhard Wellein (Friedrich-Alexander-University (FAU) of Erlangen-Nürnberg) SELL-C-σ
- Lisa Claus (University of Wuppertal) GMG, AMG, Smoother, Near-Kernel Vectors
- Osni Marques (Lawrence Berkeley National Laboratory) GMG, AMG, Smoother, Near-Kernel Vectors

2. Purpose and Significance of the Research

A multigrid is a scalable multilevel method for solving linear equations and preconditioning Krylov iterative linear solvers, and is especially suitable for large-scale problems because of its scalable feature. Generally, number of iterations until convergence for multigrid method is kept constant as the problem size changes. The parallel multigrid method is expected to be one of the most powerful tools on exa-scale systems. There are two approaches in the multigrid method, where one is a geometrical multigrid (GMG) with explicit hierarchical meshes, and the other is an algebraic one (AMG). Although multigrid methods have been applied to rather well-conditioned problems for a long time, many sophisticated methods for robustness of multigrid have been developed for ill-conditioned problems derived from real-world scientific and engineering applications. In the present work, we are developing robust and

efficient GMG and AMG methods, where we are focusing on development of algorithms for (1) efficient and robust smoother, (2) parallel global reordering methods for robustness, (3) utilization of near-kernel vectors for robustness, and (4) hierarchical methods for scalability. Although parallel computations in science and engineering have been focusing on domain-decomposition approach in space direction, new approach with parallel computation in time direction (PinST, parallel-in-space/time) has been introduced recently. MGRIT (Multigrid-Reduction-in-Time) [R.D. Falgout et al., SIAM SISC 36(6), C635-C661, 2014] is one of the most well-known PinST methods, and it introduces idea of multigrid in time domain. MGRIT was very successful in various types of applications including nonlinear ones, but it suffers from instability at coarse-level grid in time. Recently, we proposed a new method (TSC, Time Segment Correction), where no coarse time step at coarse levels in time domain is needed [S. Kaneko, Y. Fujito, A. Fujii, T. Tanaka, T. Iwashita, IPSJ SIG Technical Report, Vol.2017-HPC-161, No.7, 2017 (in Japanese)]. Because preliminary results show better performance than MGRIT, we will develop efficient and robust TSC method for large-scale problems (PinST-TSC). PinST has been mainly applied to implicit problems due to constraint of time step. But, it is really needed in explicit methods, where many time steps are needed. In the present work, we will develop a new method for PinST with explicit time marching (PinST-Exp), where we apply coarser mesh in space at the coarser level in time. Developed methods (GMG, AMG, PinST-TSC, PinST-Exp) are parallelized by OpenMP/MPI hybrid parallel programming model, and optimized for multicore/manycore clusters in JHPCN by SELL-C-σ matrix storage format [M. Kreutzer, G. Hager, G. Wellein et al., SIAM SISC 36(5), C401-C423, 2014], where Professor G. Wellein (Friedrich-Alexander-University (FAU) of Erlangen-Nürnberg) is one of the original developers of SELL-C-σ. Performance and robustness will be evaluated by 3D FEM/FDM applications on these systems. Finally, developed programs and libraries will be deployed on the supercomputer systems in each center (Hokkaido, Tokyo and Kyushu), and released to the public.

Multigrid method is a promising approach for large-scale computing in exa-scale era. We develop robust and efficient parallel multigrid methods for both of GMG and AMG, focusing on robust and efficient smoothers, parallel reordering, utilization of near-kernel vectors, and hierarchical methods which are proposed and developed by ourselves. Both of PinST-TSC and PinST-Exp are also our original method. Developed methods will be implemented as a numerical library and it will be applied to various types of applications. This is one of the first practical library of multigrid including PinST, especially PinST-Exp. It is expected to provide outstanding performance for large-scale real-world applications.

3. Significance as a JHPCN Joint Research Project

JHPCN provides a variety of supercomputer systems. We can develop and optimize our programs on each platform very easily under collaboration with members of the JHPCN centers. We can deploy our programs and libraries on the supercomputer systems in each center, and release to the public. Improvement of such released programs are accelerated if they are used by users of supercomputer systems for practical scientific and engineering applications. This is an international joint proposal by Germany-Japan-USA including experts of multigrid method and HPC in each country. MOU (memorandum of understanding) for collaborative research has been exchanged between Lawrence Berkeley National Laboratory (LBNL) and Information Technology Center, The University of Tokyo (ITC/U.Tokyo) since September 2009, and between University of Wuppertal and ITC/U.Tokyo since July 2017. FAU and ITC/U.Tokyo have been collaborating in ESSEX-II Project of German SPPEXA Program by DFG/JST since 2016. Moreover, this proposal includes

most of the experts in multigrid method in Japan.

4. Outline of the Research Achievements up to FY 2017

This project started in FY.2018.

5. Details of FY 2018 Research Achievements

① Overview

In the first year, our plan for development in this project is as follows:

- Geometric Multigrid (GMG) for 3D Goundwater Flow
  - Further optimization on OFP by CM-RCM (cyclic multicolored RCM) and SELL-C-σ
  - Optimization of Mesh Generation by IME (Fast File Cache on OFP) on OFP
  - Evaluation by 3D GW Simulations
- Algebraic Multigrid (AMG, SA-AMG) for 3D Solid Mechanics
  - OpenMP/MPI Hybrid
  - Near-Kernel Vectors
  - Evaluation by 3D FEM
- PinST-TSC
  - Development of Prototype
  - Preliminary Evaluation by Supercomputers
- PinST-Explicit
  - Development of Prototype
  - Preliminary Evaluation by Supercomputers
- Parallel Algorithms
  - MS-BMC-GS
  - Parallel Reordering/Coloring
  - Parallel Aggregation

In the following part, we will describe the activities on these issues.

① GMG for 3D Groundwater Flow

The parallel multigrid method is expected to play an important role in large-scale scientific computing on post-peta/exa-scale supercomputer systems, and it also includes serial and parallel communication processes which are generally expensive. In the previous work [K. Nakajima, ICPADS 2014], new format for sparse matrix storage based on sliced Ellpack-Itpack (ELL) format was proposed for optimization of serial communication in data transfer through memories, and hierarchical coarse grid aggregation (hCGA) was introduced for optimization of parallel communication by message passing. The proposed methods are implemented for pGW3D-FVM, a parallel code for 3D groundwater flow simulations using the multigrid method (Fig.1), and the robustness and performance of the code was evaluated on up to 4,096 nodes (65,536 cores) of the Fujistu FX10 supercomputer system at the University of Tokyo. The parallel multigrid solver using the sliced ELL format provided performance improvement in both weak scaling (25%–31%) and strong scaling (9%–22%) compared to the code using the original ELL format. Moreover, hCGA provided excellent performance improvement in both weak scaling (1.61 times) and strong scaling (6.27 times) for flat MPI parallel programming model. In the present work, we evaluated the performance of the hCGA on the Oakforest-PACS (OFP) system at JCAHPC/The University of Tokyo. Figure 2 describes the results of evaluation up to 8,192 nodes (524,288 cores, because 64 of 68 cores were used on each node). hCGA was also very effective on OFP.

(a) (b)

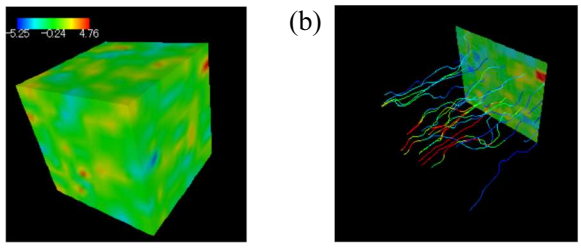

Fig.1 Example of groundwater flow through heterogeneous porous media. (a) Distribution of water conductivity; (b) Streamlines

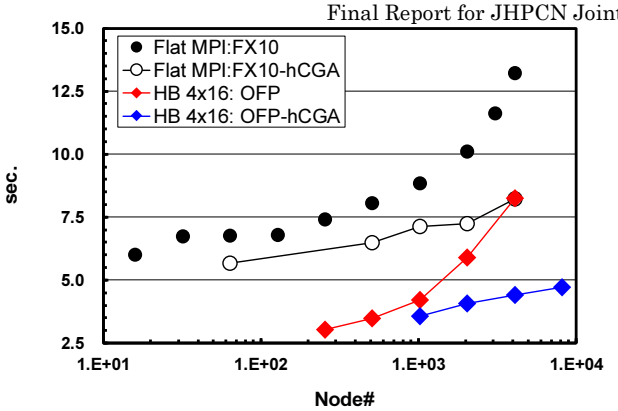

Fig.2 Performance of MGCG solver on Fujitsu FX10 using up to 4,096 nodes (65,536 cores), and OFP up to 8,192 nodes (524,288 cores), weak scaling: max. total problem size: 17B meshes on FX10, and 35B on OFP, RCM reordering

Because the hCGA can only handle 2-hierarchica-levels, we are developing AM-hCGA (Adaptive Multilevel hCGA) for multiple hierarchical-levels (more than three) (Fig.3). The prototype of AM-hCGA is now developed and evaluated on OFP system, although that was originally planned to be developed in FY.2019.

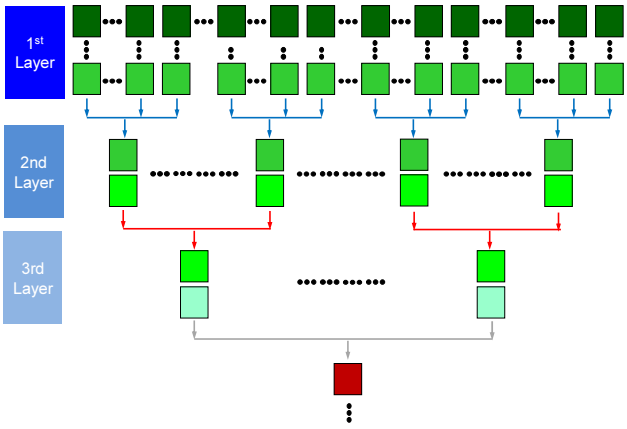

Fig.3 AM-hCGA (Adaptive Multilevel hCGA)

In the present work, we also evaluated the performance of the GMG-MGCG solver with single-precision and double-precision arithmetic, where the original version is based on the double-precision. Figure 4 shows the comparisons of single/double precision for various value of $\lambda_{max}/\lambda_{min}$. on the Reedbush-U (Intel Xeon/Broadwell-EP) and OFP (Intel Xeon Phi/Knights Landing). This graph shows the ratio of iterations and computation time for MGCG solvers normalized by results of the code with double precision. Ratio of iterations until convergence is equal to one for a wide range of $\lambda_{max}/\lambda_{min}$ up to 10^{10}. Ratio of computation time is about 0.60 on Reedbush-U and 0.85-0.90 on OFP. The MGCG solver with single precision arithmetic is stable, but computation time is not so much improved on OFP, where SIMD optimization is not sufficient. This is because that RCM reordering and Row-Based Sliced-ELL are adopted in the MGCG solver. Generally, RCM provides better convergence, but large synchronization overhead occurs due to large number of colors/levels. In order to improve performance on OFP, we need to introduce CM-RCM with fewer colors (less than 10) and SELL-C-σ for SIMD optimization. We also develop these methods in the 2nd half of FY.2018.

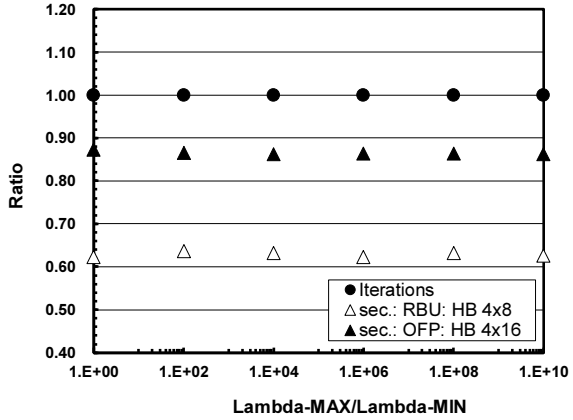

Fig.4 Performance of MGCG solver for single/double precision arithmetic on Reedbush-U and OFP for a wide range of $\lambda_{max}/\lambda_{min}$, Results are normalized by those with double-precision, Sliced-ELL, RCM reordering

② Optimization of Mesh Generaion

In regard to optimization of the mesh generation process on OFP, first, we investigated the bottleneck of the existing code under typical conditions. Originally, the program consists of the parallel mesh generator which generates mesh data to multiple files for each MPI process and the solver which reads each file as the mesh data. Both the mesh generator and the solver were written by Fortran with POSIX IO in naïve manner. In the case of 128^3 mesh with 1.024 MPI process, the file output process of mesh data consumed 352 sec. while the main part of the computation was done only in 3 sec (Fig.5).

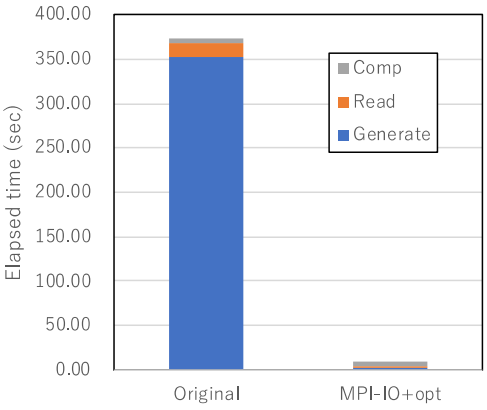

Fig.5 Time for Parallel Mesh Generation for MGCG Solvers on OFP (128^3 mesh with 1.024 MPI process)

In addition, multiple write operations and misaligned access to data significantly tend to degrade the performance. Thus, as well as being modified to MPI-IO manner in order to utilize IME effectively, a buffer with enough size for buffering are prepared in advance, and the buffer management subroutine is used instead of "MPI_File_write" API. When the buffer exceeds the pre-defined IO unit size, the buffer is written only with IO unit size. In terms of the performance, the IO unit size should be same as the transfer unit size of Lustre file system and IME. As the result, we obtained 243 times better performance in maximum than the original version with the File Per Process manner. Moreover, the case of the Single Shared File with IME is 31 times faster than the original one with the File Per Process.

In a future, we are going to develop a wrapper library for applying buffering control to the target code easily.

③ AMG for 3D Solid Mechanics: OpenMP/MPI Hybrid

In the first half of the 1st year, we investigate the execution time end parallel efficiency of SA-AMG method by applying "Parallelization by OpenMP/MPI Hybrid" on OFP (JCAHPC) and K (RIKEN) super-computer system.

Cluster architecture is one of the main-stream in recent HPC computing system. Hybrid MPI/OpenMP is suitable for such architecture (Fig.6). Both supercomputer system uses cluster architecture by Intel® Xeon Phi™ (68 cores/CPU) and SPARC64 VIIIfx(8 cores/CPU), respectively. We expect that SA-AMG is performed efficiently in massively parallel environment by Hybrid.

As the first step, we investigate the execution time and parallel efficiency by applying Hybrid parallelization. In this experiment, the problem size is set to 60^3. The setting of node, process and thread is changed in each supercomputer (Table 1). The termination criterion for the 2-norm of the relative residuals was set to 1.0×10^{-7}. The relaxation of the solution part was performed twice per level by the Symmetric Gauss-Seidel. Moreover, CM-RCM was applied at each level.

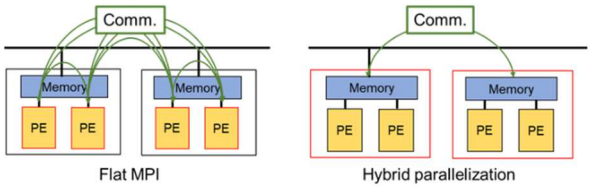

Fig.6 The difference between Flat MPI and Hybrid

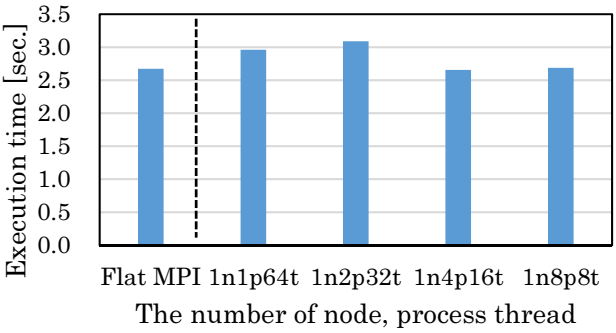

Fig.7 The execution time on OFP

Figures 7~9 are the result of execution time on OFP. From these figures, Hybrid (1n4p16) is almost the same as Flat MPI (Fig.7). Moreover, the calculation time of V-cycle, especially smoother, accounts for the majority of whole execution time. On the other hand, the communication cost is small (Fig.8 and 9). This is because the number of nodes is small in this experiment. We will investigate in highly parallel environment. Figures 10 and 11 are the result of execution time in K. The execution time of Hybrid is bad compared to Flat

MPI (Fig.10). This is due to the calculation cost of smoother (Fig.11). We must investigate about this problem.

Table 1 The setting of node, process, thread

	Candidates	Details
OFP	Flat MPI	1 node, 64 MPI proc. per node
	1n1p64t	1 node, 1 proc. per node and 64 OpenMP thread per proc.
	1n2p32t	〃 , 2 proc. per node and 32 thread per proc.
	1n4p16t	〃 , 4 proc. 〃 16 thread 〃
	1n8p8t	〃 , 8 proc. 〃 8 thread 〃
K	Flat MPI	8 node, 64 MPI proc. per node
	8n8p8t	〃 , 1 proc. 〃 8 thread 〃
	8n16p4t	〃 , 2 proc. 〃 4 thread 〃

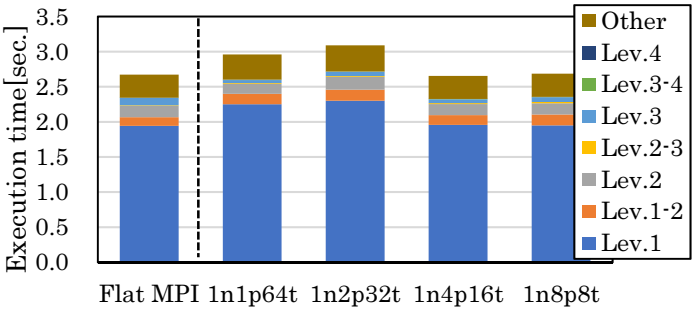

Fig.8 The ratio of V-cycle in Fig.2 (OFP)

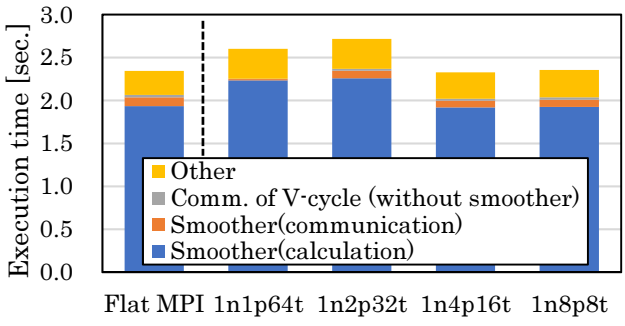

Fig.9 The details of V-cycle time (OFP)

We will improve the execution time and parallel efficiency of Hybrid parallelization. In K, the calculation cost of smoother accounts for the majority of whole execution time. At first, we need to investigate and improve this problem. Moreover, we will evaluate in high parallel environment. In this experiment, the number of nodes is used only 1 (OFP) and 8 (K) node. We will investigate the effectiveness of parallel efficiency in highly parallel environment.

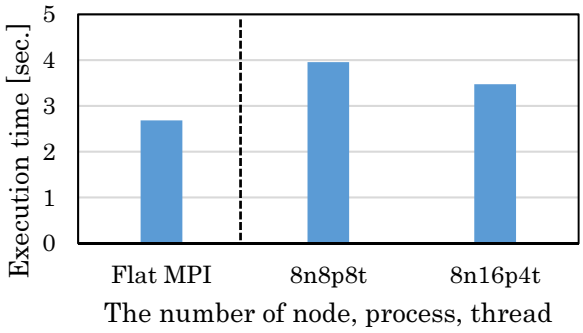

Fig.10 The execution time on K

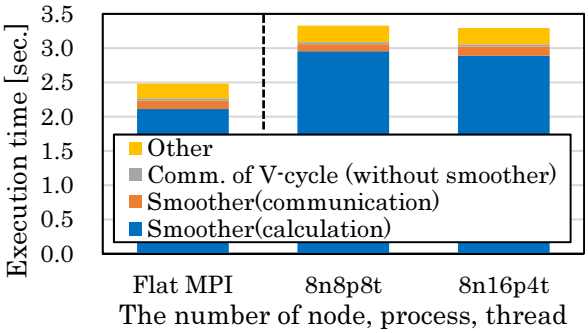

Fig.11 The details of V-cycle time (K)

④ AMG for 3D Solid Mechanics: Near Kernel Vectors

In SA-AMG, the way how to set the near-kernel vector is very important things to achieve good convergence and scalability. We propose and investigate the extraction method of near-kernel vectors.

SA-AMG method can incorporate error components that are difficult to correct in ordinary relaxation. These components correspond to near-kernel vectors defined as nonzero vector $\boldsymbol{p}$ satisfying $A\boldsymbol{p} \approx 0$. SA-AMG uses the near-kernel vectors when multi-level matrices are created to dump these components efficiently.

To extract near-kernel vectors efficiently, we propose an extraction method. We can obtain good convergence by setting optimum number of near-kernel vectors [N. Nomura, K. Nakajima, A. Fujii, SC17 Poster]. But, to set optimum number, we need to verify all patterns. This cost is very large. Therefore, we investigate the prediction method of optimum number

easily. The details are shown in Fig.12.

In our method (Fig.12), we need to set the parameter ϵ. In this experiment, we try to set 2 patterns for ϵ. The setting of ϵ is shown in Table 2 and Table 3. Fig.13 and Fig.14 show the execution time by using prediction method. The setting of ϵ is Table 2 for Fig.13 and Table 3 for Fig.14, respectively. (min) is the minimum time of all patterns (ideal one). (max) and (ave) are the maximum and average time. From this, the execution time is close to (min) by using our prediction method. From Fig.13, the whole execution time of "Our method" is close to "Previous(min)". From Fig.14, the execution time of "Our method" is bad. But, the time of only solution part is good compared to Previous method. From these results, if the same matrix A is used many times, we set ϵ to low value (like Table 3) to stabilize the convergence. If we must consider the whole execution time (setup part + solution part), we need to set high value (like Table 2). Fig.15 shows the verification time to set the optimum number. The verification time is reduced by using prediction method.

From these results, our method can predict the optimum number of near-kernel vectors easily. But we must set ϵ in our prediction method. In this experiment, we set ϵ from our experience. Therefore, we must establish the way to set ϵ.

1 $For\ level = 1\ to\ max\_level - 1$
2 $\quad For\ n = 1\ to\ extract\_number$
3 $\quad\quad Random(\boldsymbol{x})$
4 $\quad\quad V\_cycle^{\mu}\ (A_{level}\boldsymbol{x} = 0)$
5 $\quad\quad If\ \left(\frac{(A\boldsymbol{x}_1, \boldsymbol{x}_1)}{(A\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_1)}\right)^{\frac{1}{\mu}} < \epsilon\ goto\ 1$
6 $\quad\quad B_{level} \leftarrow \boldsymbol{x}$
7 $\quad\quad Multilevel\_creation(B_{level})$

Prediction part
ϵ: Input parameter

- $extract\_number$: Number of extracted vectors
- A_{level}: Matrix A of level $level$
- V_cycle^{μ}: Iterate the V-cycle μ times

Fig.12 The outline of the extraction method

Table 2 The setting of the value ϵ (for overall time)

Max level	The value of ϵ			
	Level 1	Level 2	Level 3	Level 4
3	0.250	0.156	—	—
4	0.400	0.250	0.156	—
5	0.640	0.400	0.250	0.136

Table 3 The setting of the value ϵ (for convergence)

Max level	The value of ϵ			
	Level 1	Level 2	Level 3	Level 4
3	0.100	0.070	—	—
4	0.143	0.100	0.070	—
5	0.204	0.143	0.100	0.070

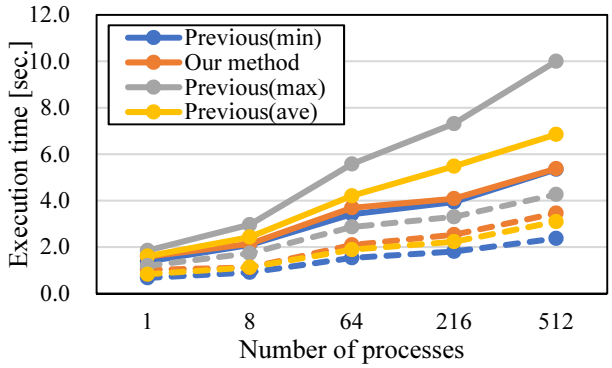

Fig.13 The result of execution time (Using ϵ of Table 2. The solid lines (round marker) show the whole execution time (setup + solution part), and the dashed lines (triangle marker) show the execution time of only solution part)

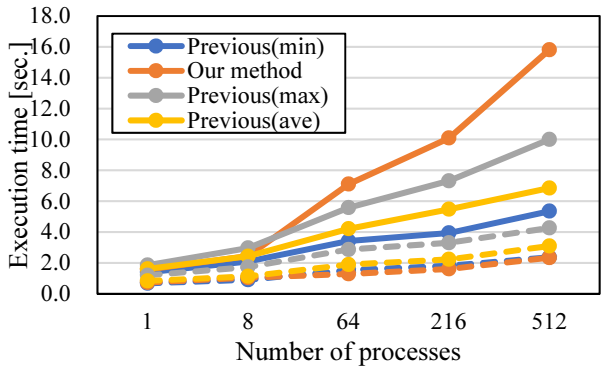

Fig.14 The result of execution time (Using ϵ of Table 3. The details are the same as Fig.14)

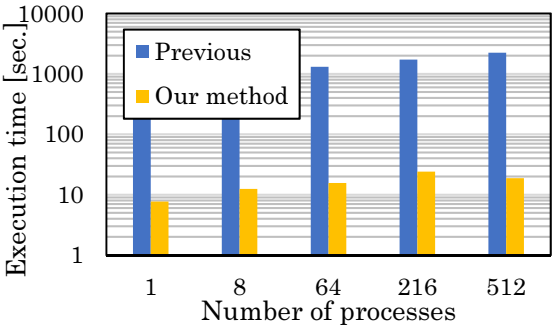

Fig.15 The result of time to verify the optimum number of near-kernel vectors

We will investigate the way to set ϵ. This is important because this parameter influence the convergence in our method. Moreover, we will investigate our method on other problems. The goal of our study is improvement of the convergence for any problems. Therefore, we consider that we need to verify the effectiveness.

⑤ PinST-TSC

Parallel time integration is an important research area, because more and more cores are available in a computer system. Parareal and MGRIT are known as such methods. This research project considers a new method, Time segmented correction (TSC) method, and aims to evaluate the effectiveness of the method with space-time parallelization to large sized problems.

We are now preparing data for papers and presentations, and plan to publish the result in near future. The contents include TSC method, its optimized implementation method, and evaluation of them.

TSC method is a parallelization method for general non-linear time evolution problem, based on TP-EEC method that is for non-linear periodic electro-magnetic problems. The performance results of the TSC method is not known yet. It has coarse grid correction, and we propose an implementation method for pipelined coarse grid corrections. The effectiveness of TSC method with our implementation method was evaluated for small sized one-dimensional non-linear heat diffusion problems. The results are shown in Fig.16. MGRIT and TSC methods are executed with 64 processes in Flat MPI model, and TSC loop1,2,3 represents the methods with coarse grid correction once, twice or three times in one TSC cycle respectively.

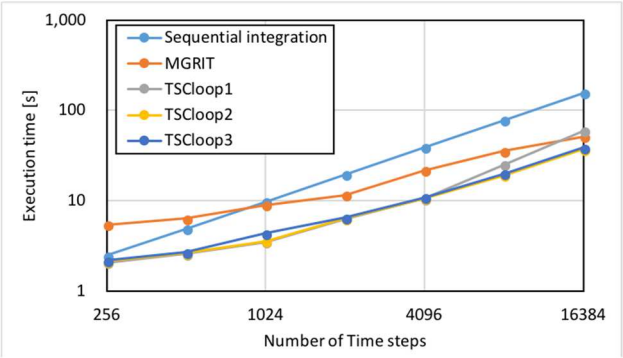

Fig.16 Time for Convergence

Figure 16 shows the convergence time. Horizontal axis means the number of time steps of heat diffusion problems and it's from 256 time steps to 16384 time steps. Although sequential time integration took time in proportion to the number of time steps, MGRIT reached convergence faster than sequential method for problems with a certain number of time steps or more. Among sequential, TSC, and MGRIT methods, TSC loop2 was the fastest for almost all problems in the test.

From the experimental result, it seems a promising method. In the near future, we would like to apply it to large sized problems, and to consider how to optimize it on super-computers.

⑥ PinST-Explicit

Explicit methods are often used for time evolution problems. It tends to require small time step width, and thus, parallelization in time direction becomes important. We studies parallelization in time direction especially for explicit methods.

As a first method, we took up MGRIT method for parallelization of explicit time evolution. Coarse grid problem with ~~large~~ enlarged time step width was observed to be unstable especially with an explicit method. This coarse grid problem appears commonly in

parallelization of time integration.

This research proposed to use MGRIT as a preconditioner of Krylov subspace method, and investigated its effectiveness in two-dimensional heat diffusion problems.

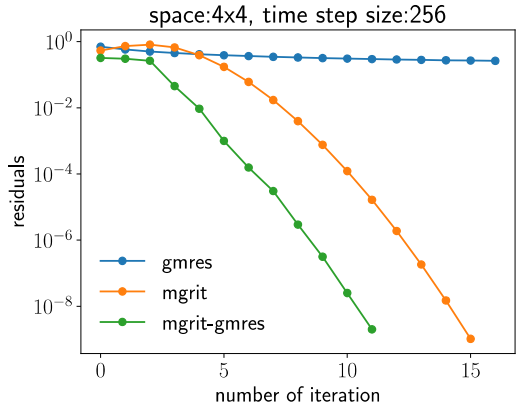

Fig.17 History of Convergence

Figure 17 shows the residual history of the proposed method, MGRIT, and non-preconditioned GMRES. Horizontal axis is the iteration number, and vertical axis is the 2-norm of the relative residual vector. From the figure, MGRIT preconditioned GMRES reached convergence faster than the other 2 methods, and it exemplified that the coarse grid problems' instability can be soothed by using MGRIT as a preconditioner.

We also proposed PinST-Exp/Imp, where explicit time marching is applied to the finest level in time direction, and implicit ones for other coarser levels.

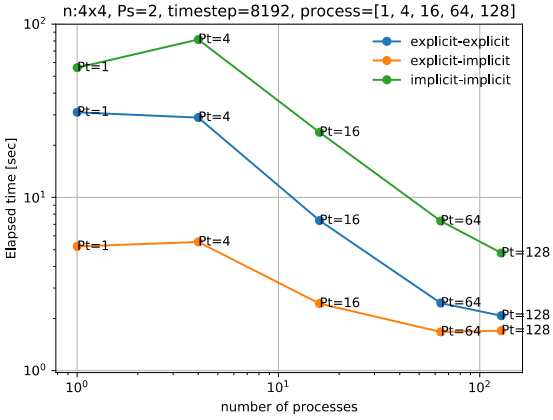

Fig.18 Strong Scaling performance

Figure 18 shows the strong scaling performance of two-dimensional heat diffusion problem using 4 node of OFP. The elapsed time of Exp/Imp method was faster than that of Exp/Exp and Imp/Imp. This experiments exemplified that when explicit method with coarser level is unstable, the implicit method with coarser level becomes more stable and faster.

We would like to explore other methods of parallel time integration with large sized and various kinds of problems.

⑦ MS-BMC-GS

Multiplicative Schwartz type block multi-coloring Gauss-Seidel (MS-BMC-GS) smoother is the new one archived high cache hit ratio, increasing convergence, and reducing communications. To support exa-scale systems, these properties are significant. We expect that the AMG and GMG method with this smoother shows good performance on the massively parallel systems.

The basic research of MS-BMC-GS have already finished. To support larger problems with AMG applied MS-BMC-GS on massively parallel systems, we have to parallelize any other parts in the AMG method. The parallel aggregation based on the multi-coloring is one of them.

MS-BMC-GS is one of the modified version based on a block multi-coloring Gauss-Seidel (BMC) smoother. With the BMC-GS smoother, the target elements are separated into some blocks. By applying the multi-coloring to these blocks, BMC-GS achieved higher cache hit ratio compared with the existing smoother such as a multi-coloring Gauss-Seidel or a hybrid smoother. In addition, with the MS-BMC-GS, we applied Gauss-Seidel smoother to each block, repeatedly. By this approach, we achieved the higher cache hit ratio, increasing convergence ratio and reducing communications. Table2 and 3 show the effect of the MS-BMC-GS on the Oakforest-PACS and the ITO, respectively. A target problem is a Poisson equation discretized with the FDM method. Sizes of the target problem are 256^3 on the Oakforest-PACS and 512^3 on the ITO. The tables show computational time and number of iterations while changing the number of Gauss-Seidel iterations in each block of MS-BMC-GS.

Results of the first left top show a performance of BMC-GS, and the other sells show performances of MS-GMB-GS. On both systems, the MS-BMC-GS is faster than the BMC-GS. In detail, MS-BMC-GS is 29.4% faster on the Oakforest-PACS and 39.4% faster on the ITO, respectively.

Now we are parallelizing all part of the AMG, and we will evaluate the effect of MS-BMC-GS with the larger problems on the massively parallel systems.

Table2 Effect of MS-BMC-GS on Oakforest-PACS

Computational time [s] / Number of iterations		Post-smoothing									
		1	2	3	4	5	6	7	8	9	10
Pre-smoothing	1	0.51/10	0.43/9	0.41/8	0.45/8	0.43/7	0.45/7	0.48/7	0.5/7	0.54/7	0.56/7
	2	0.48/9	0.37/7	0.39/7	0.37/6	0.39/6	0.41/6	0.44/6	0.46/6	0.49/6	0.51/6
	3	0.41/8	0.39/7	0.37/6	0.39/6	0.41/6	0.44/6	0.46/6	0.48/6	0.51/6	0.53/6
	4	0.45/8	0.36/6	0.38/6	0.42/6	0.43/6	0.39/5	0.5/5	0.43/5	0.45/5	0.47/5
	5	0.51/7	0.39/6	0.41/6	0.37/5	0.39/5	0.41/5	0.47/5	0.44/5	0.65/5	0.48/5
	6	0.51/7	0.41/6	0.43/6	0.39/5	0.41/5	0.43/5	0.45/5	0.47/5	0.49/5	0.5/5
	7	0.48/7	0.44/6	0.46/6	0.4/5	0.43/5	0.45/5	0.47/5	0.49/5	0.51/5	0.53/5
	8	0.51/7	0.46/6	0.48/5	0.43/5	0.45/5	0.46/5	0.49/5	0.5/5	0.53/5	0.54/5
	9	0.53/7	0.49/6	0.5/5	0.45/5	0.47/5	0.56/5	0.5/5	0.53/5	0.55/5	0.57/5
	10	0.56/7	0.51/6	0.45/5	0.51/5	0.49/5	0.5/5	0.52/5	0.54/5	0.56/5	0.58/5

fast ↕ slow

Table3 Effect of MS-BMC-GS on ITO

Computational time [s] / Number of iterations		Post-smoothing									
		1	2	3	4	5	6	7	8	9	10
Pre-smoothing	1	2.74/10	2.25/8	2.3/8	2.39/8	2.21/7	2.29/7	2.38/7	2.48/7	2.59/7	2.72/7
	2	2.25/8	2.01/8	2.11/7	1.82/6	1.9/6	1.99/6	2.08/7	2.17/6	2.28/6	2.36/6
	3	2.02/7	2.08/7	1.82/6	1.87/6	1.95/6	2.03/6	2.12/6	2.17/6	2.28/6	2.36/6
	4	2.08/7	1.82/6	1.87/6	1.9/6	1.67/5	1.74/5	1.82/5	1.89/5	1.96/5	2.03/5
	5	2.15/7	1.89/6	1.97/6	1.66/5	1.72/5	1.74/5	1.84/5	1.95/5	2/5	2.09/5
	6	2.25/7	1.97/6	1.68/5	1.66/5	1.79/5	1.85/5	1.84/5	1.95/5	2.08/5	2.16/5
	7	2.36/7	2.07/6	1.77/5	1.79/5	1.85/5	1.93/5	1.99/5	2.07/5	2.08/5	2.24/5
	8	2.1/6	2.15/6	1.82/5	1.88/5	1.94/5	2.01/5	2.08/5	2.17/5	2.24/5	2.33/5
	9	2.22/6	2.15/6	1.82/5	1.96/5	2.02/5	2.08/5	2.17/5	2.17/5	2.32/5	2.41/5
	10	2.31/6	2.36/6	2.01/5	2.04/5	2.1/5	2.18/5	2.24/5	2.33/5	2.4/5	2.47/5

fast ↕ slow

⑧ Parallel Reordering/Coloring

In this study, we applied multi-coloring algorithms parallelized with a hierarchical approach to the domain decomposed smoother for increasing convergence. The localized smoother is often used because of easy to implement. The problem of this smoother is decreasing the convergence if we increase a degree of parallelism. To avoid this problem, we applied the multi-coloring to the smoother. We expect that this approach keeps convergence ratio and is useful on the massively parallel systems.

Now, we are preparing an interface between SA-AMG program and hierarchical parallelization routine. The implemented SA-AMG for the study of the near kernel vector has a Gauss-Seidel smoother which is parallelized focusing on an only local part. In some situations, we validated decreasing of convergence. Then we will implement a proper parallelized Gauss-Seidel smoother based on the multi-coloring. The multi-coloring algorithms which we applied are parallelized with the hierarchical approach. In this study, we have to parallelize all part in the SA-AMG because of targeting massive problems.

On the study of near kernel vector, the program was modified to support the hybrid parallelism. Now we are preparing the interface between the SA-AMG and the multi-coloring. We will implement the parallelized Gauss-Seidel smoother based on the multi-coloring and evaluate performance on the massively parallel systems.

⑨ Parallel Aggregation

In this study, we proposed a parallelization method with multi-coloring parallelization for aggregation in the AMG.

Parallelization of the aggregation process in the AMG is necessary to support massively parallel systems. Then, the performance and convergence of the AMG strongly depend on the results of aggregation. We need an approach to parallelize the aggregation without changing the properties of sequential processes.

In this year, we proposed a parallelization method of the aggregation process with the multi-coloring method. By this approach, compared with the parallel and sequential aggregation, we achieved constant convergence.

The result of aggregation in the AMG is important to achieve constant convergence. In the aggregation process, we separate whole elements to subdomains. Then, we expect that more strong couplings between elements are included in each sub-domain, and the coupling between each sub-domain is weak. There are some approaches to achieve this requirement. If we parallelize these approaches easily, we apply domain decomposition and decouple among decomposed

domains. The aggregation process is applied to each domain. This parallelization approach is easy to implement. However, the strong couplings among the decomposed domains remain on a coarse grid. It causes decreasing convergence and increasing the amount of computation on the coarser grids.

To keep the properties of sequential aggregation, we proposed the parallelization method based on the multi-coloring. In this approach, we colored all elements. The elements colored with the same color are handled, parallelly. The strong coupling among decomposed domain is included in proper sub-domain. We expect that the aggregations are parallelized without changing the properties. Then, the coloring algorithm has a unique requirement compared with the general method. By the general multi-coloring algorithms, the elements which have relationships are colored with different colors. If we parallelize aggregation based on the general multi-colorings, as we shown in Fig19, The same element is included in both sub-domains. To avoid this situation, any elements which have distance 2 relationships have to color with different colors (Fig. 20). To support this requirement, we modified the general multi-coloring algorithms.

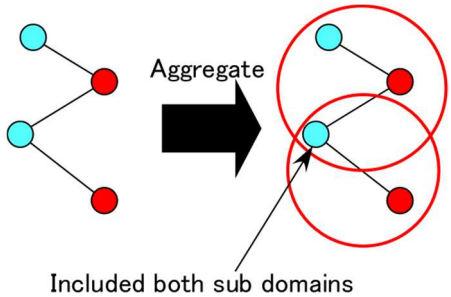

Fig.19 Missing parallelization of the aggregation

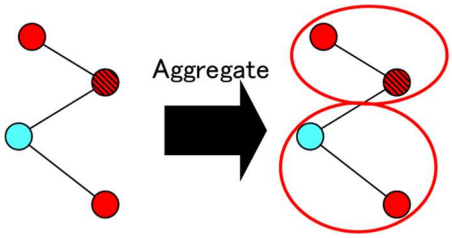

Fig.20 Parallelization of the aggregation with 2-distance multi-coloring

In the numerical evaluations, we compared with the convergence of the parallel and the sequential aggregation. Figures 21-24 show the comparison with some problems which downloaded from "SuiteSparse Matrix Collection". If we applied the decoupling aggregation, the convergences changed depends on the degree of parallelisms. As we expected, the parallel aggregation with multi-coloring showed almost constant convergence.

In the future, we will evaluate the performance of the AMG with the parallel aggregation and MS-BMC-GS smoother on the massively parallel systems.

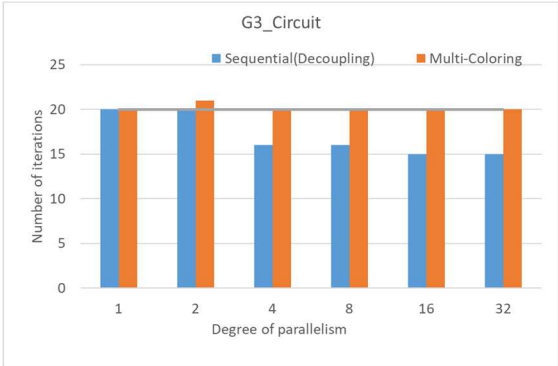

Fig.21 Convergence with the G3_Circuit

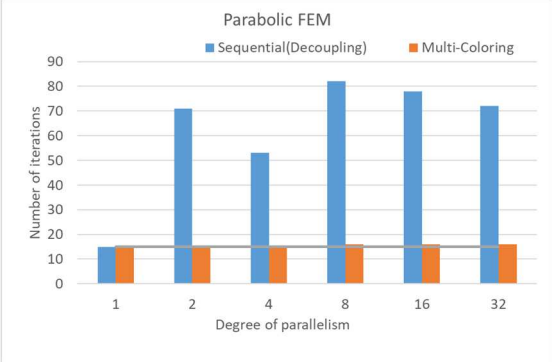

Fig.22 Convergence with the Parabolic_FEM

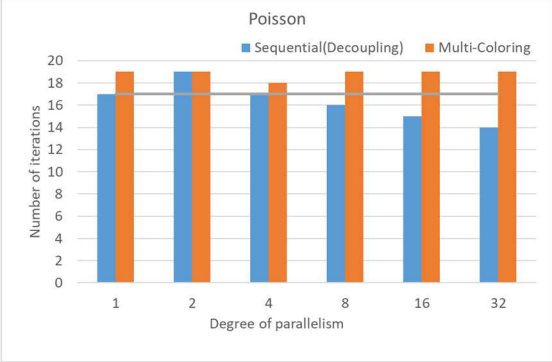

Fig.23 Convergence with a Poisson problem

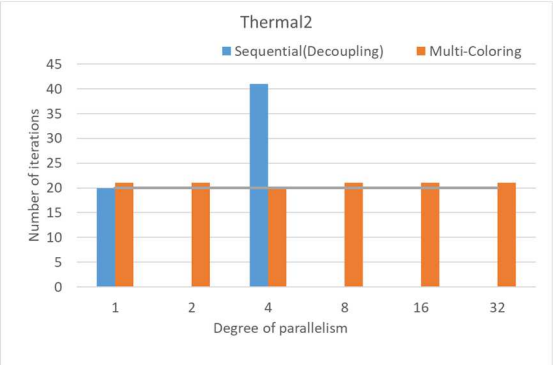

Fig.24 Convergence with the Thermal2

6. Progress of FY 2018 and Future Prospects

This is a 2-year project starting in FY.2018, where we do fundamental research & development, code optimization and preliminary evaluation of performance and robustness in the first year, and performance evaluation using real-world large-scale applications in the second year. Generally speaking, things are going well, and we have already done or have been doing most of the works. We have many excellent developments which were not planned in the beginning of the project, such as (1) Parallel Aggregation, (2) Iterative Solver with MGRIT Preconditioning, (3) PinST-Exp/Imp Method, (4) Evaluation of Efficiency and Accuracy by Computations with Single Precision. We have also done the preliminary works on AM-hCGA, which were originally planned in FY.2019. Research plan in FY.2019 is as follows:

GMG:

- Further Optimization of the pGW3D-FVM on OFP (CM-RCM, SELL-C-σ, MS-BMC-GS) (Hoshino, Hanawa, Nakajima, Wellein, Kawai, Ida, Bolten, Claus)
- Improvement of Convergence by Parallel Reordering (Kawai, Nakajima)
- Improvement of Convergence by Utilization of Near-Kernel Vectors (Nomura, Bolten, Claus, Nakajima),
- Further Optimization of Mesh Generation by IME on OFP (Hanawa, Nakajima)
- Improvement of Scalability by Adaptive Multilevel hCGA (AM-hCGA) (Nakajima)
- Evaluation by 3D GW Code on the Supercomputers (Nakajima).

AMG:

- Optimization of GeoFEM SA-AMG on OFP using ELL, SELL-C-σ (Hoshino, Hanawa, Nakajima, Nomura, Wellein)
- Further Improvement of the Algorithm by MS-BMC-GS (Kawai, Ida, Nomura, Marques, Bolten, Claus)
- Further Improvement of the Convergence by Parallel Reordering/Aggregation (Kawai, Marques, Bolten, Claus, Nomura)
- Implementation of hCGA and MA-hCGA for Scalability (Nakajima, Nomura)
- Evaluation by Supercomputers (Nomura, Nakajima).

PinST-TSC:

- Further optimization of the pHEAT-3D (Fujii, Iwashita, Nakajima, Hoshino)
- Implementation of Multigrid Solver (SA-AMG) in Space Domain for Scalable Computation (Nomura, Fujii)
- Final Evaluations on the Supercomputers (Fujii, Iwashita).

PinST-Exp:

- Further Optimization of the MGRIT Preconditioner (Yoda, Fujii)
- Further Optimization of pHEAT-3D with PinST Exp/Inp (Yoda, Fujii, Nakajima)
- Development of hybNS for 3D Compressible Navier-Stokes Flow by FVM with PinST-Exp (Yoda, Nakajima)
- Final Evaluations on the Supercomputers (Yoda, Fujii)

Because this is an international project, international collaboration is also an important part of the project. Researchers from Japan, Germany and USA have been working closely on certain topics, such as SELL-C-σ, MS-BMC-GS and Nera Kernel Vectors. We have been holdings meetings for discussions during international conferences, such as ISC-HPC 2018 (Frankfurt, Germany, 2018.6), CoSaS 2018 (Erlangen, Germany, 2018.9), Second French-Japanese-German Workshop on Programming and Computing for Exascale and Beyond (Tokyo, Japan, 2018.10), and SC18 (Dallas, TX, USA, 2018.11). Nakajima made a presentation introducing this project in the Second French-Japanese-German Workshop in the Embassy of Germany in Tokyo.

Table 4. Program of the 39th ASE Seminar, January 8, 2019, University of Tokyo

Speaker	Title
Kengo Nakajima	Welcome & Overview of "Innovative Multigrid Methods"
Kengo Nakajima	Numerical Library with High-Performance/Adaptive-Precision/High-Reliability
Matthias Bolten	Multigrid for Structured Grids on Large-Scale Parallel Computers
Masatoshi Kawai	Multiplicative Schwartz type Block Multi-Color Gauss-Seidel Smoother for AMG
Naoya Nomura	The Evaluation of The SA-AMG Method By Applying Hybrid Parallelizaion on Cluster Computing System
Toshihiro Hanawa	Optimization of Parallel Mesh Generation via File for Multigrid Method
Akihiro Ida	Lattice H-Matrices: More Efficient Structures of H-Matrices for Arithmetic and Parallelization
Akihiro Fujii	Time Segmented Correction Method for Simple Nonlinear Heat Diffusion Problem
Ryo Yoda	The Evaluation of MGRIT Preconditioning for Spring-Mass-Damper System and Burgers Equation

In January 2019, Matthias Bolten (University of Wuppertal) (Co-PI) visited Japan under support by JHPCN for International Project, had a talk in U.Tokyo and RIKEN R-CCS in Kobe, and discussed with members of the project. On January 8, 2019, we hosted the 39th ASE Seminar (Advanced Supercomputing Environment), the University of Tokyo (https://www.cc.u-tokyo.ac.jp/events/ase/39/39.php) "Ode to Multigrid II", focusing on the progress of this project. We had 18 audience.

We also applied to the JSPS KAKENHI with our German and USA partners. Unfortunately, it was not accepted, but we are also preparing for next year.

7. List of Publications and Presentations

(1) Journal Papers

[1] 野村直也，中島研吾，藤井昭宏，高スケーラブル・安定的なSA-AMG法に向けたニアカーネルベクトル自動抽出手法に関する研究，第65回ACS論文誌，2019 (in press) (in Japanese)

(2) Conference Papers

[2] R. Yoda, A. Fujii, T. Tanaka, MGRIT Preconditioned Krylov subspace method, IEEE/ACM Proceedings of SC18 (The International Conference for High Performance Computing, Networking, Storage and Analysis) (Research Poster), Dallas, TX, USA, 2018

[3] T. Hanawa and K. Nakajima, Optimization of parallel mesh generation via file for multigrid method, Proceedings of Intel Extreme Performance Users Group (IXPUG) Asia Workshop, in conjunction with HPC Asia 2019, Guangzhou, China, 2019

(3) Oral Presentations

[4] 中島研吾，AM-hCGA法による並列多重格子法，第23回計算工学講演会（日本計算工学会）（名古屋，2018年6月7日）（in Japanese）

[5] 河合直聡，伊田明弘，代数マルチグリッド法のための並列Aggregationの評価，第23回計算工学講演会（日本計算工学会）（名古屋，2018年6月8日）（in Japanese）

[6] K. Nakajima, Numerical Library with High-Performance/Adaptive-Precision/High-Reliability Extension of ppOpen-HPC towards the Post Moore Era, Future Accelerated Math Library Design, ISC High Performance 2018 (Frankfurt, Germany, June 25, 2018) **(Invited Talk)**

[7] K. Nakajima, Innovative Method for Integration of Computational & Data Sciences in the Exa-Scale/Post-Moore Era, APSCIT 2018 Annual Meeting (Asia Pacific Society for Computing and Information Technology) (Sapporo, Hokkaido, Japan, July 21, 2018) **(Keynote Talk)**

[8] K. Nakajima, Innovative Method for Integration of Computational & Data Sciences in the Exa-Scale/Post-Moore Era, Seminar of IBM T.J. Watson Research Center (Yorktown Heights, NY, USA, July 25, 2018) **(Invited Talk)**

[9] K. Nakajima, Application Development Framework for Manycore Architectures -from Exascale to Post Moore Era-, Exa/Post-Peta Scale Computational Mechanics, The 13th World Congress on Computational Mechanics (WCCM XIII) (New York, NY, USA, July 27, 2018)

[10] 中島研吾，ポストムーア時代の並列反復法アルゴリズム，2018年並列／分散／協調処理に関する『熊本』サマー・ワークショップ（SWoPP熊本2018），日本応用数理学会「行列・固有値問題の解法とその応用」研究部会（MEPA）（熊本，2018年7月31日）（in Japanese）

[11] 野村直也，中島研吾，藤井昭宏，Oakforest-PACS上におけるSA-AMG法のHybrid並列化に関する分析，情報処理学会研究報告（2018-HPC-165-34），2018年並列／分散／協調処理に関する『熊本』サマー・ワークショップ（SWoPP熊本2018）（熊本，2018年8月1日）（in Japanese）

[12] N. Nomura, K. Nakajima, The Evaluation of the SA-AMG Method by Applying Hybrid Parallelization on Cluster Supercomputer System, Proceedings of the 3rd International Symposium on Research and Education of Computational Science (RECS2018) (poster session) (Tokyo, Japan, September 20-21, 2018)

[13] K. Nakajima, Innovative Multigrid Methods and the JHPCN Program, Second French-Japanese-German Workshop on Programming and Computing for Exascale and Beyond (Embassy of Germany in Tokyo, October 30-31, 2018) **(Invited Talk)**

[14] 野村直也，中島研吾，Oakforest-PACS上におけるSA-AMG法の高並列環境下に向けたHybrid並列化に関する分析，日本応用数理学会2018年度年会（名古屋，2018年9月3日～5日）（in Japanese）

[15] 塙敏博，中島研吾，マルチグリッド法におけるファイルを介したメッシュ生成プロセスの高速化，情報処理学会研究報告（2018-HPC-167），日本情報処理学会第167回HPC研究会（那覇，沖縄，2018年12月17-18日）（in Japanese）

[16] 野村直也，中島研吾，藤井昭宏，大規模クラスタ環境上におけるSA-AMG法のHybrid並列化に関する分析，情報処理学会研究報告（2018-HPC-167），日本情報処理学会第167回HPC研究会（那覇，沖縄，2018年12月17-18日）（in Japanese）

[17] M. Bolten, Multigrid for Structured Grids on Large-Scale Parallel Computers, The 39th ASE Seminar, Information Technology Center, The University of Tokyo (Tokyo, Japan, January 8, 2019)

[18] M. Kawai, Multiplicative Schwartz type Block Multi-Color Gauss-Seidel Smoother for AMG, The 39th ASE Seminar, Information Technology Center, The University of Tokyo (Tokyo, Japan, January 8, 2019)

[19] N. Nomura, The Evaluation of The SA-AMG Method By Applying Hybrid Parallelizaion on Cluster Computing System, The 39th ASE

Seminar, Information Technology Center, The University of Tokyo (Tokyo, Japan, January 8, 2019)

[20] T. Hanawa, Optimization of Parallel Mesh Generation via File for Multigrid Method, The 39th ASE Seminar, Information Technology Center, The University of Tokyo (Tokyo, Japan, January 8, 2019)

[21] A. Ida, Lattice H-Matrices : More Efficient Structures of H-Matrices for Arithmetic and Parallelization, The 39th ASE Seminar, Information Technology Center, The University of Tokyo (Tokyo, Japan, January 8, 2019)

[22] A. Fujii, Time Segmented Correction Method for Simple Nonlinear Heat Diffusion Problem, The 39th ASE Seminar, Information Technology Center, The University of Tokyo (Tokyo, Japan, January 8, 2019)

[23] M. Bolten, Multigrid for structured grids on large-scale parallel computers, RIKEN R-CCS Café (Kobe, Japan, January 10, 2019)

[24] M. Bolten, Blocked multigrid methods for structured matrices, Poster Session, SIAM Conference on Computational Science & Engineering (CSE19) (Spokane, WA, USA, Feb.25-Mar.1, 2019)

[25] R.Yoda, A. Fujii, T. Tanaka, Multigrid Reduction in Time preconditioning for parallel-in-time Krylov solver, 19th Copper Mountain Conference on Multigrid Methods (Copper Mountain, CO, USA, March 24-28, 2019)

[26] K. Nakajima, M. Horikoshi, T. Hanawa, H. Fujita, M. Shiryaev, Optimization of Parallel Multigrid Solvers on Manycore Clusters, 19th Copper Mountain Conference on Multigrid Methods (Copper Mountain, CO, USA, March 24-28, 2019)

[27] N. Nomura, K. Nakajima, M. Kawai and A. Fujii, The Analysis of SA-AMG Method by Applying Hybrid MPI/OpenMP Parallelization on Cluster Supercomputer System, 19th Copper Mountain Conference on Multigrid Methods (Copper Mountain, CO, USA, March 24-28, 2019)

(4) Others

None