

MGRIT preconditioned Krylov subspace method

Ryo Yoda[†], Akihiro Fujii[†], Teruo Tanaka[†]

[†]Kogakuin University



Introduction

- In massively parallel environment, **MGRIT (Multigrid Reduction in Time)** [1][2][5] gained attentions as a **scalable solver of the time evolution problem**.
 - Space direction parallelism is quickly depleted. Need to extract time direction parallelism [7].
 - Fig 1 shows an example where MGRIT is the scalable solver in the time direction.
 - MGRIT constructs a multi-level structure in time direction by expanding time step width.
 - Coarse-level problems with large time step width tend to be unstable.**

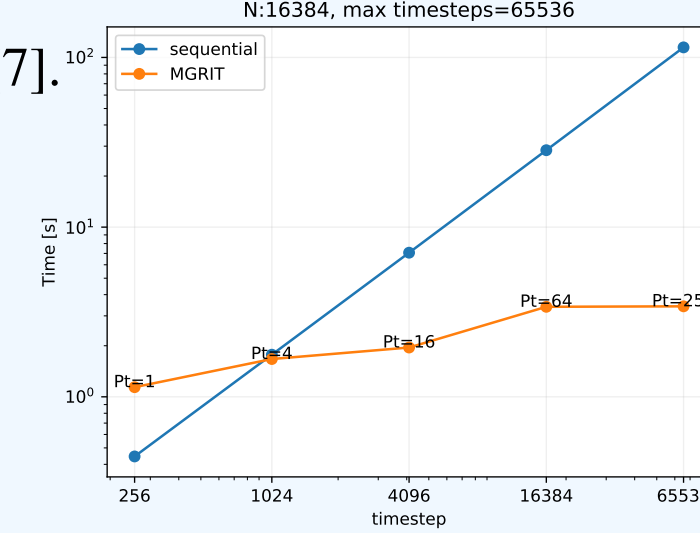
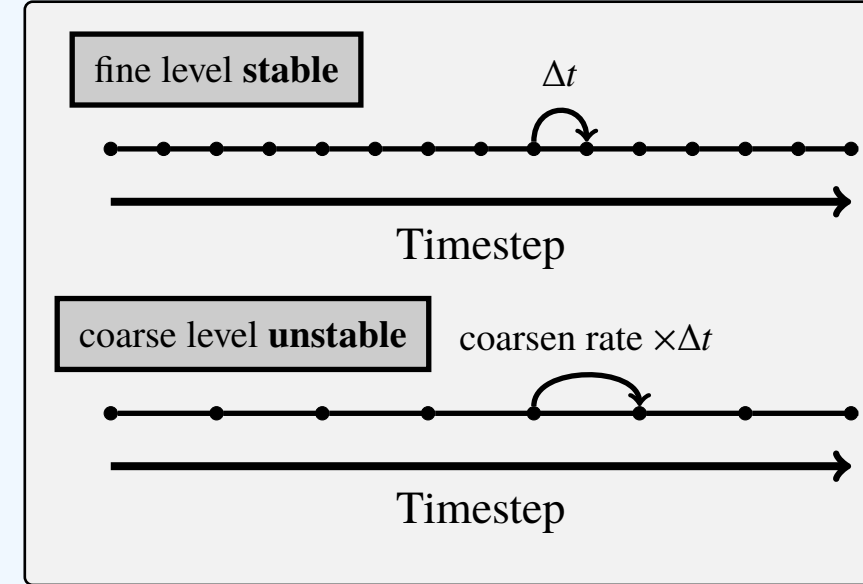


Figure 1: scalable solver in time direction



- Application with **MGRIT** as the preconditioner of **Krylov subspace method**, to enhance the stability of the solver
 - Implementation
 - We used **GMRES** (Generalized Minimum RESidual) [8][10][11][12].
 - It can realize parallelization in both space direction and time direction.
 - We used a cluster based on the Xeon Phi 7200-series, Oakforest-PACS (JCAHPC) [17].
 - Analysis
 - Same computation complexity order as **MGRIT**.
 - Additional memory cost is needed for Krylov subspace method.
 - It inherits the scalability of **MGRIT**.
 - MGRIT** (preconditioning) + **GMRES** (solver) monotonically decreasing residual norm.
- Our contribution
 - We propose **MGRIT preconditioned Krylov subspace method**.
 - We showed the matrix form of MGRIT operation and how MGRIT preconditioning improved eigenvalue distribution.
 - MGRIT-GMRES** achieves higher stability than **MGRIT** in our small and easy problems.

MGRIT

- the time evolution problem
 - In implicit method, step-by-step equation and time evolution matrix A are written as in next equations.

$$\begin{cases} g^0 = u^0 \\ \Phi u^{i+1} = u^i + g^{i+1} \end{cases} \quad (i = 0, 1, \dots, T-1) \quad (1)$$

$$Au = \begin{bmatrix} I & & \\ -I & \Phi & \\ & \ddots & \\ & & -I & \Phi \end{bmatrix} \begin{bmatrix} u^0 \\ u^1 \\ \vdots \\ u^T \end{bmatrix} = \begin{bmatrix} g^0 \\ g^1 \\ \vdots \\ g^T \end{bmatrix} = g \quad (2)$$
 - It solves $Au = g$ by incorporating coarse-grid correction in time direction.

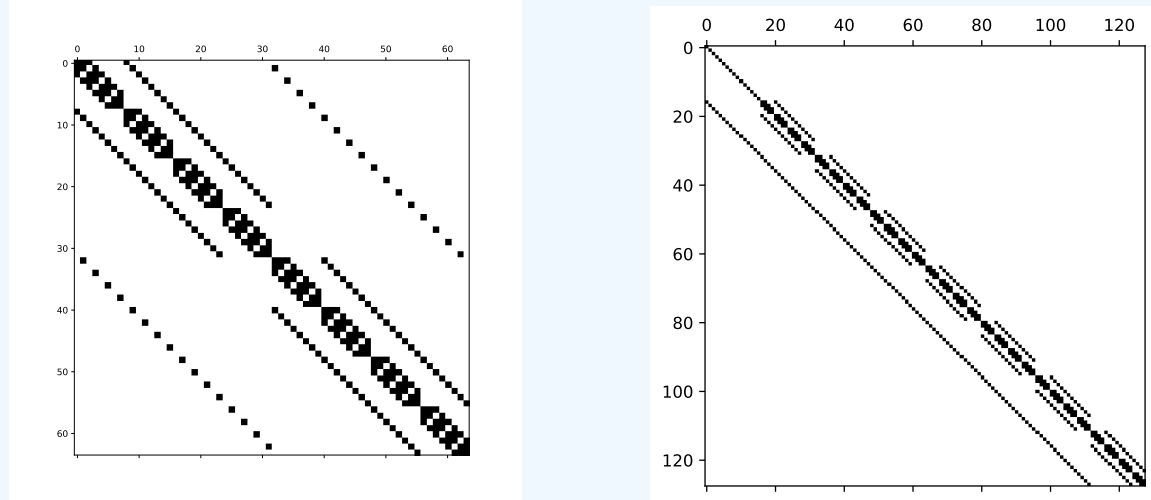


Figure 2: nonzero component of Φ

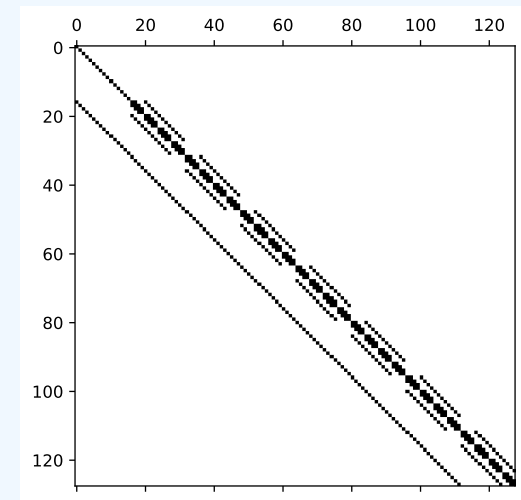
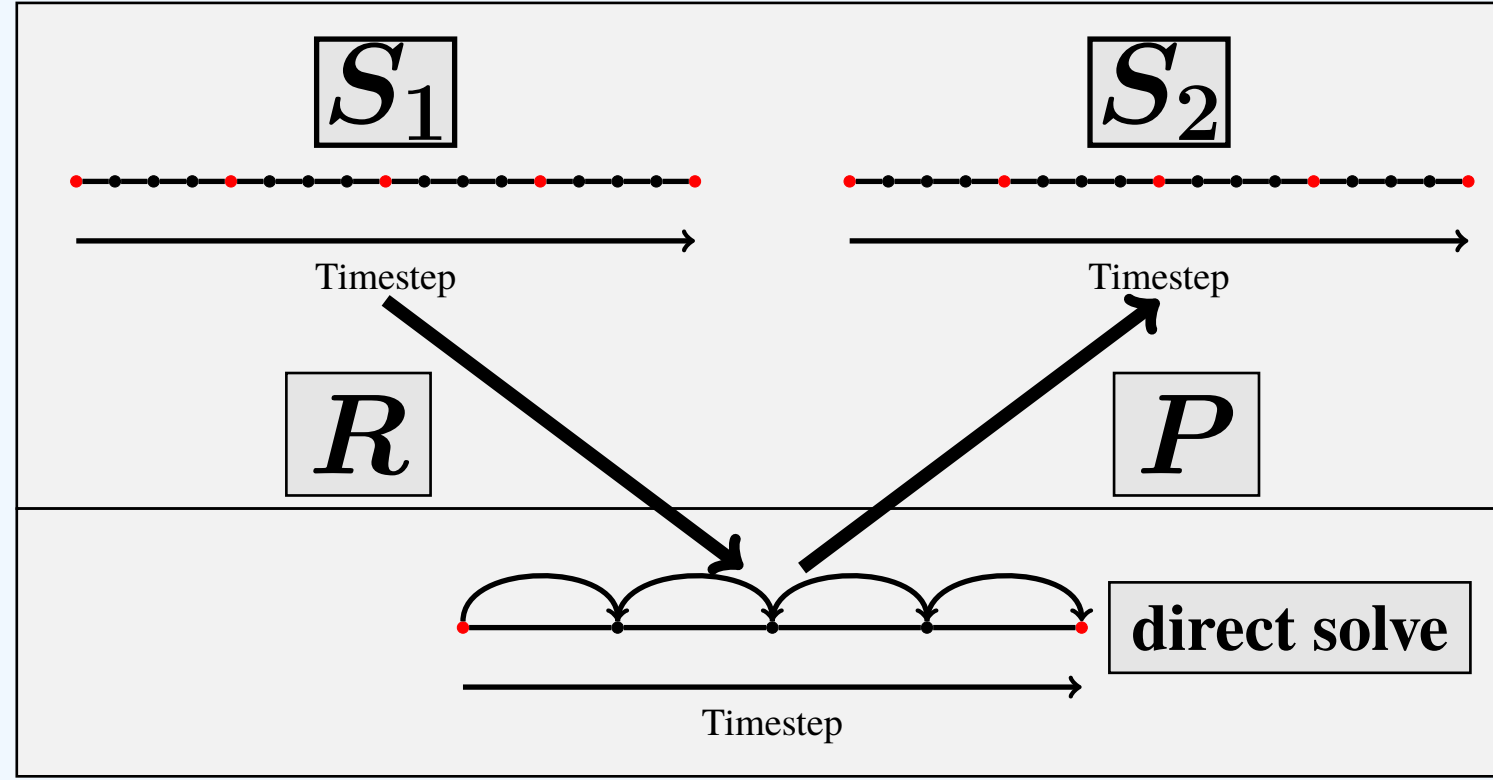


Figure 3: nonzero component of A

- Component
 - MGRIT divides time-steps to C-point (red) and F-point (black)
 - coarsening ratio: the number of time-steps between C-point to C-point
 - S_1 : pre-smoother, FCF-relaxation
 - S_2 : post-smoother, F-relaxation
 - R : extract C-points, restrict to coarse-level
 - P : prolongate coarse-level vector to fine-level
 - direct solve**: sequentially time integration with parallel execution in space direction



MGRIT as a preconditioner (1)

- We showed a **iterative matrix of MGRIT**.
 - In order to use as a preconditioner, the preconditioning must be stationary.
 - All operations of **MGRIT** are described in a matrix form.

- Introduction of symbols
 - I_c : extract only the C-points.
 $I_c u$ has values of u_i at C-points and 0 at F-points.
 - I_f : extract only the F-points.
 $I_f u$ has values of u_i at F-points and 0 at C-points.

$$\begin{pmatrix} I & I_c \\ & I \end{pmatrix} \begin{pmatrix} u \\ u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} I_c u \\ u_0 \\ 0 \\ 0 \\ u_3 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} I_f \\ I \end{pmatrix} \begin{pmatrix} u \\ u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} u \\ u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix}$$

$$\begin{pmatrix} I & I_c \\ I & I \end{pmatrix} \begin{pmatrix} u \\ u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} I_f \\ I \end{pmatrix} \begin{pmatrix} u \\ u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} + \begin{pmatrix} I_c \\ I \end{pmatrix} \begin{pmatrix} g \\ g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \end{pmatrix} \Rightarrow u = M_c^{-1} I_f u + M_c^{-1} I_c g$$

Figure 4: C-relaxation

- F-relaxation: Update F-points from C-points.

$$\begin{pmatrix} I & I_c \\ -I & \Phi \end{pmatrix} \begin{pmatrix} u \\ u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} I_c \\ I \end{pmatrix} \begin{pmatrix} u \\ u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} + \begin{pmatrix} I_f \\ I \end{pmatrix} \begin{pmatrix} g \\ g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \end{pmatrix} \Rightarrow u = M_f^{-1} I_c u + M_f^{-1} I_f g$$

Figure 5: F-relaxation

- FCF-relaxation: F-relaxation $\rightarrow$ C-relaxation $\rightarrow$ F-relaxation

$$\hat{u} = M_f^{-1} I_c u + M_f^{-1} I_f g, \quad \hat{u} = M_c^{-1} I_f \hat{u} + M_c^{-1} I_c g, \quad u = M_f^{-1} I_c \hat{u} + M_f^{-1} I_f g \Rightarrow u = M_f^{-1} I_c M_c^{-1} I_f M_f^{-1} I_c u + M_f^{-1} I_c M_c^{-1} I_f M_f^{-1} I_f g + M_f^{-1} I_c M_c^{-1} I_c g + M_f^{-1} I_f g$$

MGRIT as a preconditioner (2)

- Here, we derive the 2-level **MGRIT** preconditioning matrix [15][16].
 - It is possible to extend the algorithm to multi-levels.
 - $u_{v-cycle}$ is the approximate solution $Au = g$ after applying one-cycle Two-level MGRIT method.

$$\begin{aligned} u &= S_{fcf} u + T_1 g \\ r_\Delta &= R(g - Au) \\ A_\Delta u_\Delta &= r_\Delta \\ u &= u + P u_\Delta \\ u &= S_f u + T_2 g \end{aligned} \quad u_{v-cycle} = S_f (I - P A_\Delta^{-1} R A) S_{fcf} u + (S_f T_1 + T_2 + S_f P A_\Delta^{-1} R (I - A T_1)) g \quad (3)$$

Algorithm 1: Two level method

MGRIT preconditioned GMRES

- Use GMRES [8][10][11][12] method as the Krylov subspace method.
 - $\rightarrow$ Because time evolution matrix is nonsymmetric. Please look at Figure 3.
 - $\rightarrow$ Our purpose is to enhance the stability of MGRIT.
 - $\rightarrow$ Utilize the nature of monotonic decrease of residuals in GMRES.

- GMRES allows application of the right preconditioning [11][12].
 - $\rightarrow$ The red highlight in Algorithm 2 show the preconditioning part.

- MGRIT-GMRES** uses **MGRIT** preconditioning matrix at M^{-1} in **Algorithm 2**

```

v0 = r0 / ||r0||2
for j = 0, k # k is a restart number
    zj = M^-1 vj
    vj+1 = Azj
    for i = 0, j
        h[i][j] = (vj+1, vi)
        vj+1 = vj+1 - h[i][j] * vi
    end
    h[j+1][j] = ||vj+1||2
    vj+1 = vj+1 / h[j+1][j]
end
Compute xk = x0 + Zk yk # Zk = [z0, z1, ..., zk]
    
```

Algorithm 2: preconditioned GMRES

Implementation

- Parallelism is extracted in both space direction and time direction.
- Here, the data distribution and the representative operations (inner product and SpMV) are introduced.

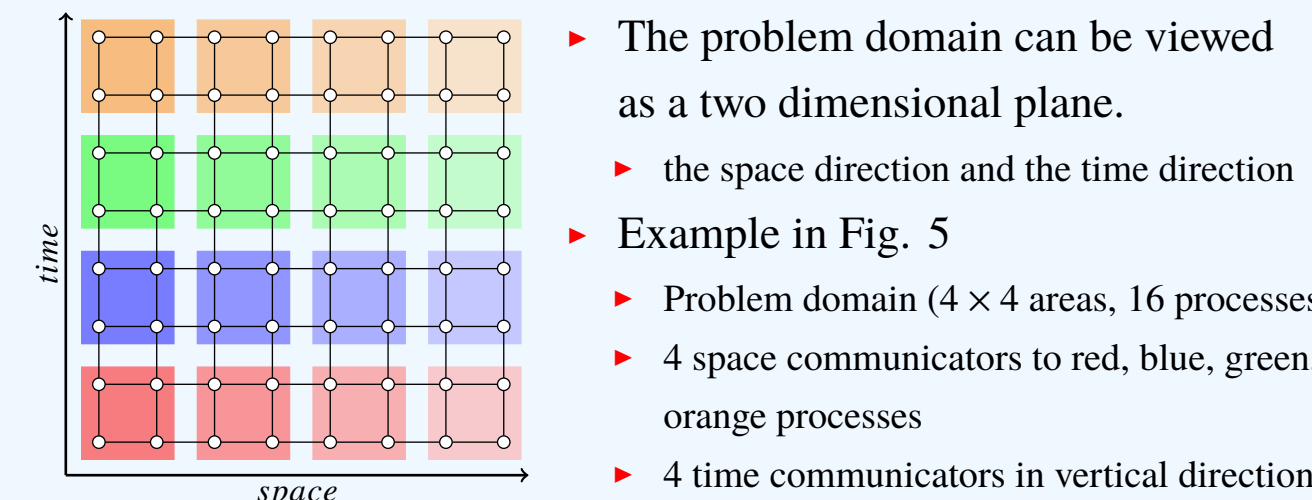


Figure 6: data distribution on 16 processes (4 in space direction and 4 in time direction)

- the inner product operation on space and time domain
 - step 1** Reduction operation for summation of local inner product is done using space communicator
 - step 2** Allreduce operation for the global inner product is done in time communicator 0
 - step 3** Broadcast operation with space communicators are done to share the global inner product among all processes
- the SpMV operation with the time evolution matrix
 - step 1** Red part: SpMV of diagonal block with space communicator, this is Φy
 - step 2** Blue part: Coefficient of sub-diagonal part is -1 . It refers to the unknown values 1 time-step before. Therefore, the calculation is done after communication using time communicator.

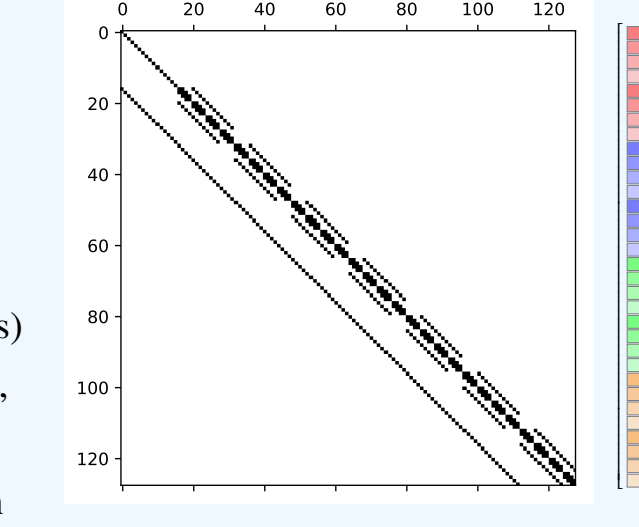
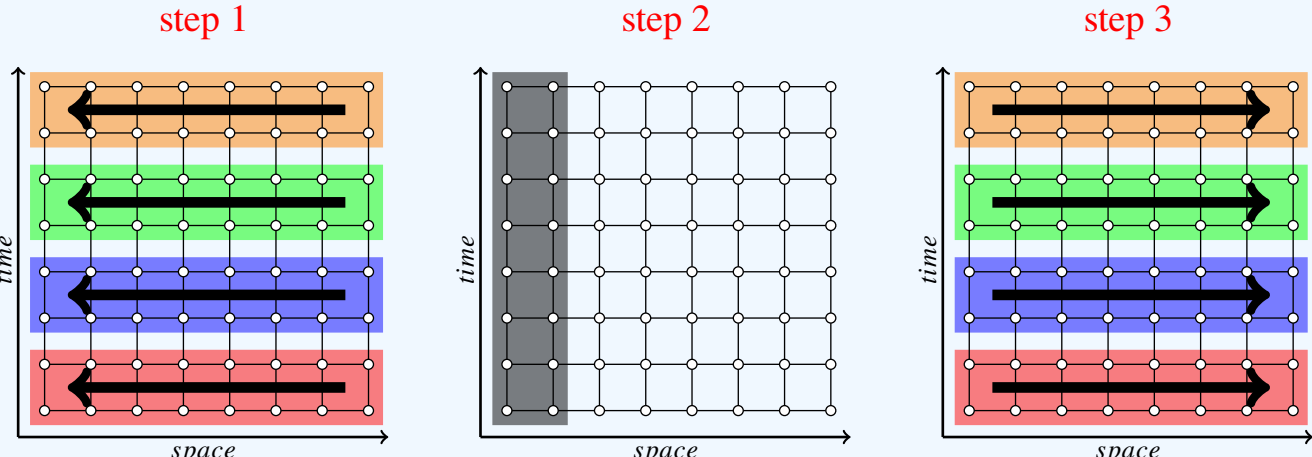


Figure 7: the relation to matrix A and an each process

- Time evolution matrix and unknown vector in the order of time steps
- Color of unknowns corresponds to decomposed domains of Fig. 5.
- It is not one-dimensional block row distribution that is a representative sparse matrix distribution as shown in Fig. 6.

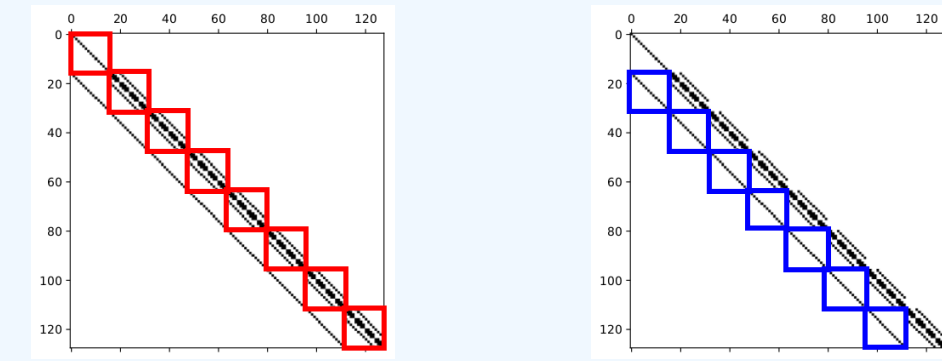


Figure 8: diagonal part of SpMV

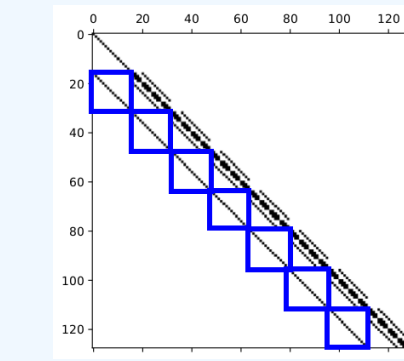


Figure 9: sub-diagonal part of SpMV

Numerical Experiment (1) eigenvalue and singular-value distribution

- Experimental problems: 2D linear diffusion-problem using 5-points difference
- The purpose is to investigate the effects of **MGRIT** preconditioning.
 - Problem and parameter settings: $n = 4 \times 4$ (DoF in space), $timestep = 32$ (DoF in time), $\frac{k\Delta t}{(\Delta x)^2} = 0.1$, coarsening ratio= 4.
 - The problem discretized by the implicit method is regarded as a **stable problem**.
 - Since large time-step width at coarse levels influences the convergence of explicit method, the problem discretized by the explicit method is regarded as a **unstable problem**.
- The convergence of Krylov subspace methods depends on the eigenvalue distribution of A .
 - In the implicit method, we examined whether eigenvalue distribution of A improved by MGRIT preconditioning.
 - In the explicit method, all diagonal elements of A are 1, therefore all eigenvalues are 1. $\rightarrow$ we examined whether singular value distribution improved by MGRIT preconditioning.
 - The eigenvalue distribution of A is blue points(-), and that of AM^{-1} is orange points(+).

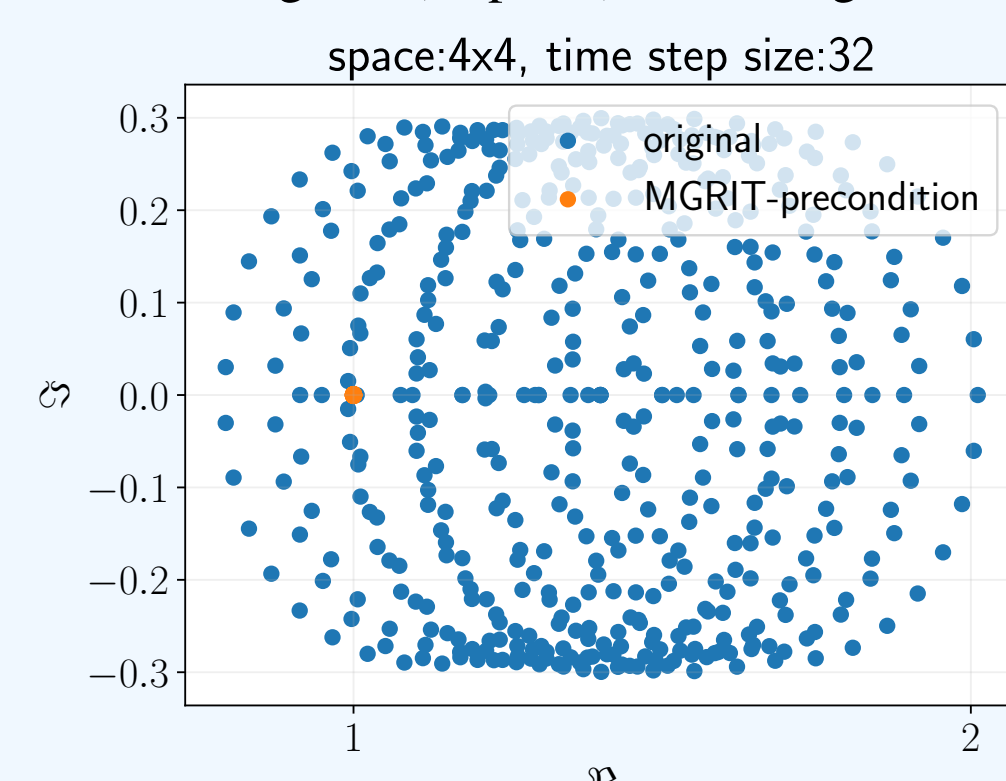


Figure 10: the eigenvalue distribution of A and AM^{-1}

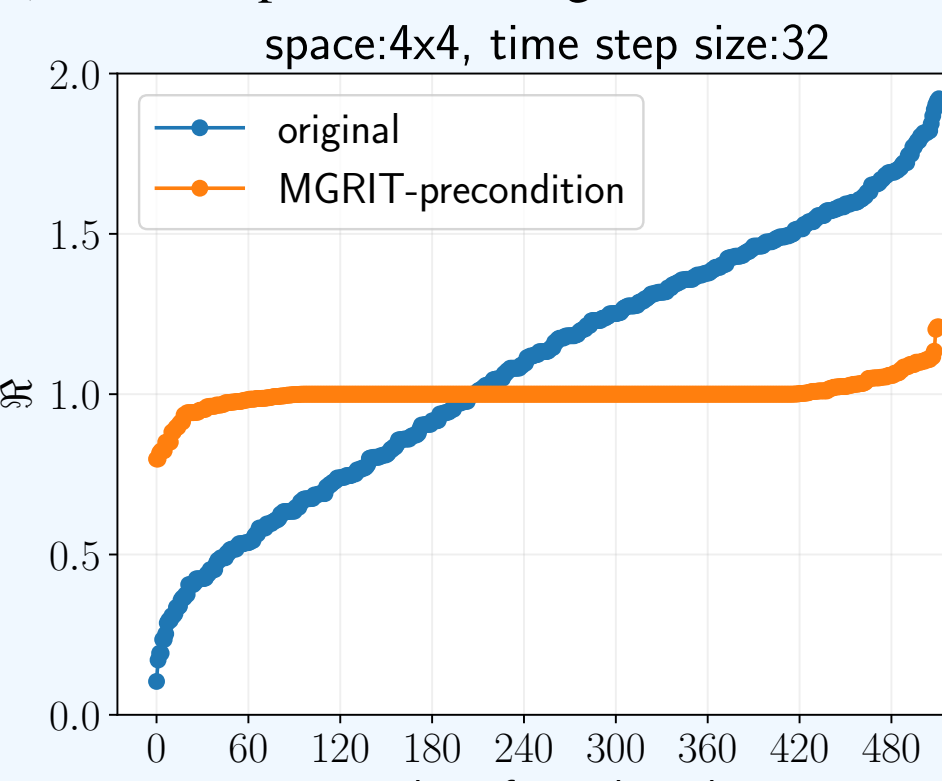


Figure 11: the singular-value distribution of A and AM^{-1}

Numerical Experiment (2) the number of iterations

- Experimental problems: 2D linear diffusion-problem using 5-points difference
 - The purpose is to check the stability by number of iterations.
 - Problem and parameter settings: $n = 4 \times 4$ (DoF in space), $timestep = 256$ (DoF in time), $\frac{k\Delta t}{(\Delta x)^2} = 0.1$, coarsening ratio= 4.
 - We compared GMRES, MGRIT, and MGRIT-GMRES (the proposed method).
 - The used environment: a cluster based on the Xeon Phi 7200-series, Oakforest-PACS supercomputer system (JCAHPC) [17].

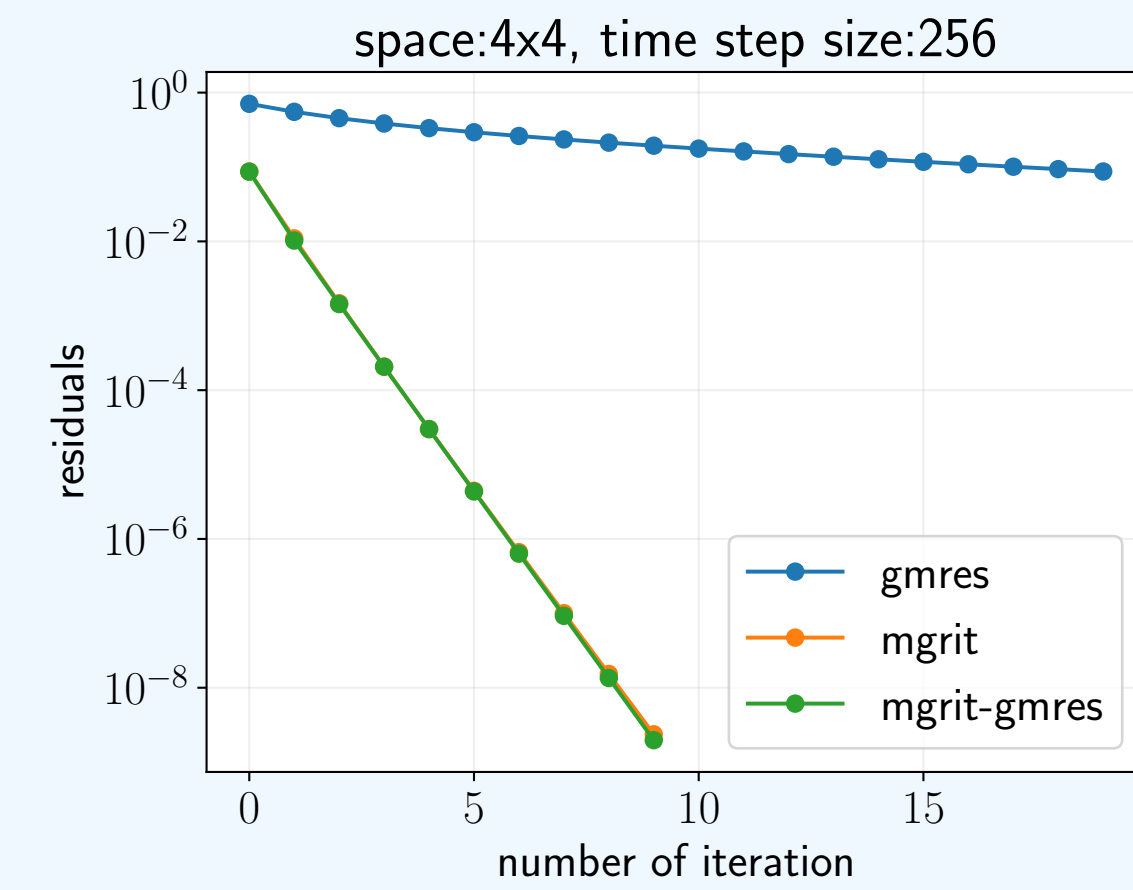


Figure 12: (Implicit) the number of iterations in GMRES, MGRIT and MGRIT-GMRES

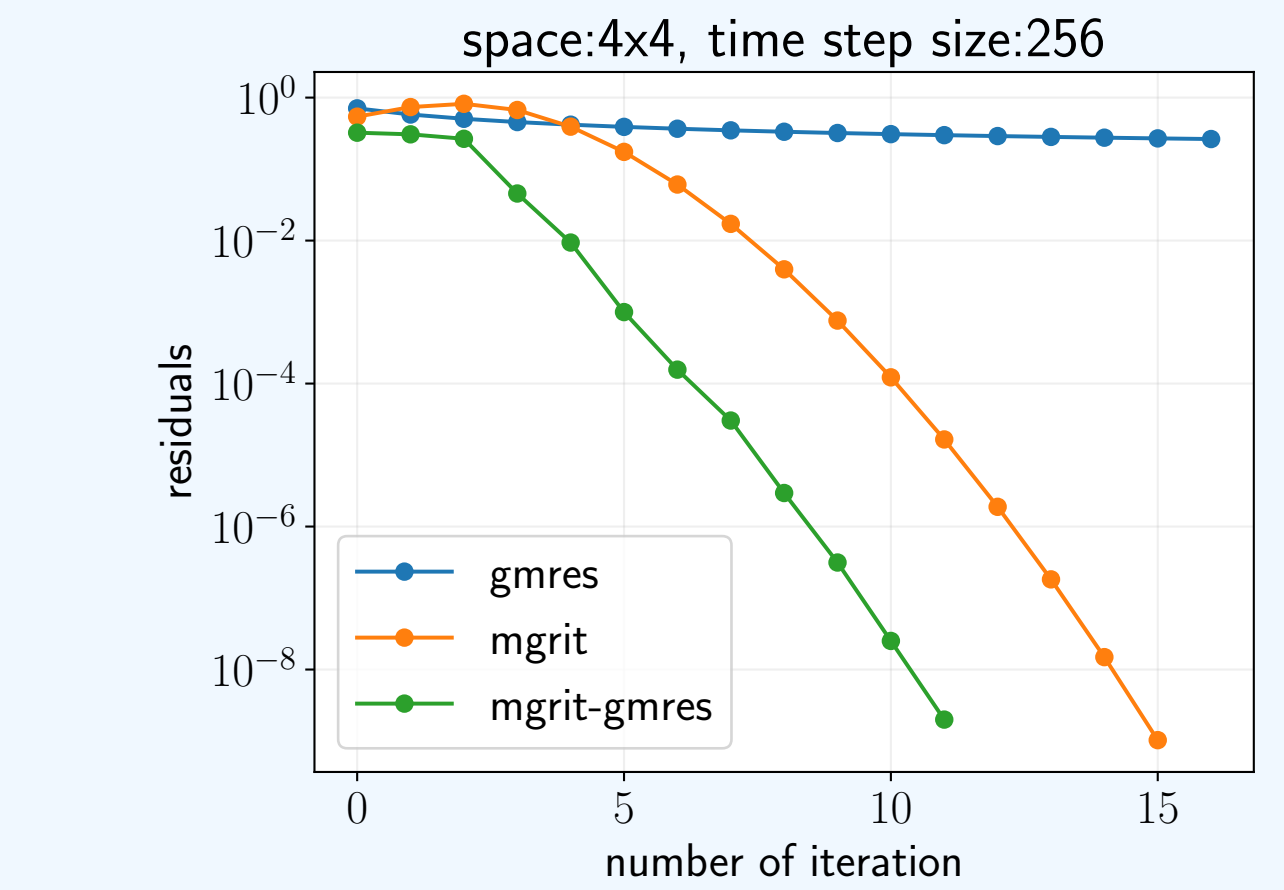


Figure 13: (Explicit) the number of iterations in GMRES, MGRIT and MGRIT-GMRES

- In implicit method, even if the time-step width is large, the coarse grid correction works efficiently.
 - As shown in Fig. 12 of the result of implicit method, The residual histories of MGRIT and MGRIT-GMRES approximately matched. $\rightarrow$ Because coarse-level problem solved efficiently
- In explicit method, if the time-step width is coarse, the problem becomes unstable.
 - As shown in Fig. 13 of the result of explicit method, The residual norm of MGRIT increased for a few initial iteration. $\rightarrow$ Unstable coarse-level problem is expected to degraded the convergence.
 - MGRIT-GMRES monotonically decreased the residual norms, and reduced the number of iterations.**

Numerical Experiment (3) the scaling performance

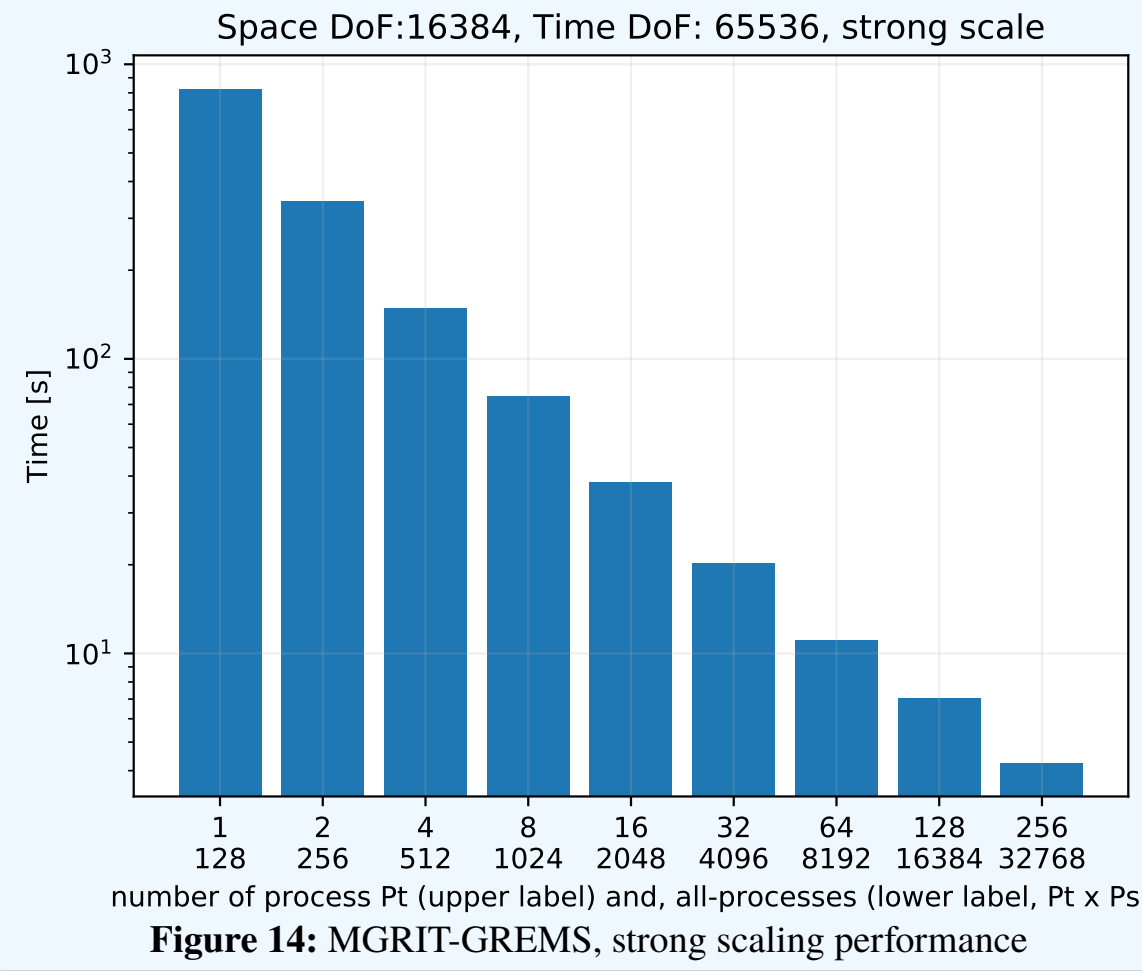


Figure 14: MGRIT-GMRES, strong scaling performance

- Experimental problems: 2D linear diffusion-problem using 5-points difference.
 - The purpose is to check the strong scaling performance.
 - We measured the execution time of MGRIT-GMRES with strong scale.
 - The used environment: 512 nodes and 32768 processor cores on Oakforest-PACS [17] (JCAHPC) supercomputer system.
 - $n = 128 \times 128$ (DoF in space), $timestep = 65536$ (DoF in time) $\rightarrow$ **DoF in problem is 16384 $\times$ 65536 = 1,073,741,824.**
 - Ps (space parallelism) is fixed at 128.
 - Pt (time parallelism, upper x-label) is [1, 2, 4, 8, 16, 32, 64, 128, 256]. $\rightarrow$ **max parallelism is 128 $\times$ 256 = 32768.**
 - In Fig 14, when the time parallelism doubled, the execution time was reduced to about 1/2. $\rightarrow$ **MGRIT-GMRES has the good scaling performance.**

Conclusion & Future work

- MGRIT attracts attentions as a scalable solver of time evolution problems.
 - Large time-step width at coarse levels make the coarse-level problems unstable especially for explicit method.
- We proposed **MGRIT preconditioned Krylov subspace method**.
 - We showed the matrix forms of all MGRIT operations, and improvement of eigenvalues distribution.
 - We evaluated **MGRIT preconditioning** for both stable and unstable problem in terms of number of iterations. $\rightarrow$ In case of unstable problem, MGRIT increased residual norm for a few initial iterations, but MGRIT-GMRES monotonically decreased it. $\rightarrow$ **MGRIT-GMRES performed better than MGRIT in terms of the number of iterations** in our easy test problems.
- Krylov subspace method enhances the stability of MGRIT method.
 - Krylov subspace method mainly consists of matrix vector product and inner product.
 - They are included in MGRIT and can be calculated effectively using time and space communicators.
 - The calculation cost of **MGRIT preconditioned GMRES** is the same order of MGRIT.
 - MGRIT preconditioned GMRES** needs the additional memory cost of k -vectors for restart GMRES(k).
- We will continue to investigate the effectiveness of **MGRIT preconditioning**
 - We will evaluate the more large timestep-width or more unstable problems, discretized from another governing equation.
 - MGRIT preconditioning doesn't depend on the Krylov subspace method. We will combine another Krylov subspace method.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Numbers JP15K15998 and "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures" and "High Performance Computing Infrastructure" in Japan (Project ID: jh180022-NAHI).

Reference

- R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel Time Integration with Multigrid*, SIAM J. Sci. Comput., 36, pp.C635-C661, (2014)
- R. D. Falgout, A. Katz, Tz. V. Kolev, J. B. Schroder, A. M. Wissink, U. M. Yan: *Parallel Time Integration with Multigrid Reduction for a Compressible Fluid Dynamics Application*, Journal of Computational Physics, (2014)
- H. Gahvari, V. A. Dobrev, R. D. Falgout, Tz. V. Kolev, J. B. Schroder, M. Schulz and U. M. Yang, *A Performance Model for Allocating the Parallelism in a Multigrid-in-Time Solver*, The 7th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS16), Supercomputing 16 (2016)
- R. D. Falgout, S. Friedhoff, Tz. V. Kolev, S. P. MacLachlan, J. B. Schroder and S. Vandewalle, *Multigrid Methods with Space-Time Concurrence*, Computing and Visualization in Science, Springer, (2017)
- V. Dobrev, Tz. Kolev, N. A. Petersson, and J. B. Schroder, *Two-level convergence theory for Multigrid Reduction in Time (MGRIT)*, SIAM J. Sci. Comput., 39, pp. 501-527, (2017)
- M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM J. Sci. Comput., 29, pp. 556 – 578, (2007)
- M. J. Gander, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition Methods, Springer, Cham, pp. 69-113, (2015)
- Y. Saad, and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., 7, pp. 856-869, (1986)
- Y. Saad, *Krylov Subspace Methods on Supercomputers*, SIAM J. Sci. Comput., 10, pp.1200 – 1232, (1989)
- H. A. van der Vorst and C. Vuik, *The superlinear convergence behaviour of GMRES*, SIAM J. Comput. Appl., 48, pp. 327-341, (1993)
- J. Baglama, D. Calvetti, G. H. Golub, and L. Reichle, *Adaptively preconditioned GMRES algorithms*, SIAM J. Sci. Comput., 20, pp. 243-279, (1998)
- Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, second edition, (2003)
- W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial Second Edition*, SIAM, (2000)
- I. Yavneh, *Why Multigrid Methods Are So Efficient*, Computing in Science and Engineering, 8, pp. 12 – 22, (2006)
- O. Tatebe, *The Multigrid Preconditioned Conjugate Gradient Method*, 6th Copper Mountain Conference on Multigrid Methods, pp. 621-634, (1993)
- K. Nakajima, *Parallel Iterative Linear Solvers with Preconditioning for Large Scale*, (2002)
- https://www.csc.tokyo.ac.jp/index-e.html (accessed on July 31, 2018)