Near kernel component setting method using iteration matrix in SA-AMG method

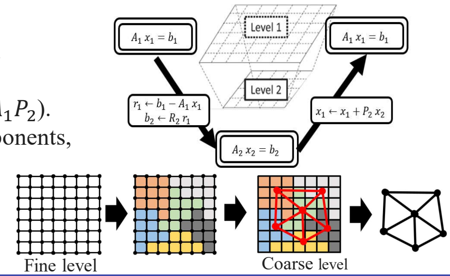
Hiromichi Sakuta , Akihiro Fujii , Teruo Tanaka
Kogakuin University

Introduction

Analysis by computer simulation is based on solving the linear equations Ax=b, which are discretized by the finite element method or other methods. When analyzing more complex phenomena, the problem to be computed becomes larger, so it is important to solve the large-scale simultaneous linear equations fast and stably in order to simulate the phenomena accurately. One of the methods to solve such problems is the SA-AMG method. This method can find a solution quickly by coarsening the problem matrix hierarchically. However, it depends on the problem setup. Conventionally, it is known that the convergence can be improved by setting the near kernel vector (0 eigenvalue component) [1].In this study, we propose applying symmetric Gauss-Seidel iterations on multiple random vectors to find components that are difficult to converge.

CG method with SA-AMG preconditioning

The SA-AMG (Smoothed Aggregation Algebraic Multigrid) method generates a coarse level matrix from an aggregate of the unknown variables in the fine level.
The product of A and the interpolation operators generates the coarser level matrix hierarchically ($A_2 = R_2 A_1 P_2$).
The coarse and fine level matrices are solved alternately(The fine lattice attenuates the high-frequency components, While the coarse lattice attenuates the low-frequency components). It applies SGS (Symmetric Gauss-Seidel) smoothing at each level. Convergence can be improved by using components that are difficult to converge when generating interpolation operators.



Hard-to-Converge Components

Set a hard-to-converge component in the interpolation operator (R, P) used to generate each hierarchy.

- The hard-to-converge component is the largest eigenvector of the matrix G applied to the SGS error vector(D, L and U are diagonal, lower and upper part of the matrix A).
  - $G = I - M^{-1}A = ((D+U)^{-1}L(D+L)^{-1}U)$
- Find multiple maximum eigenvectors simultaneously using the power method.

G:Complex nth-order square matrix to compute eigenvalues and eigenvectors
X_0: Any $n \times m$ matrix X_0 with independent random column $X_0 = I_m$
1 : Taking G and X_0 as inputs
2 : Repeat the following until X_k converges
  1 : $Y_{k+1} = GX_k$
  2 : $Y_{k+1} = \frac{Y_{k+1}}{|Y_{k+1}|}$
The column vector of X_k corresponding to eigenvectors with largest eigenvalues

exponential method

Blocking and Experiment

- Blocking functions in 2-ways. Blocking size can be arbitrarily set according to problem matrix.
  - 1. The constant vector that is used as the first near kernel vector is divided to multi-dimension constant vectors. Constant vectors in 2 and 3 Dimensional cases are illustrated in Figure 1 and 2.
  - 2. The problem matrix is treated as a blocked one. The smoother becomes blocked smoother, and it enhances the convergence.
- Since the problems are obtained from 3-dimensional settings, blocking size was set as 3, if the blocking option is used.
  - In the case of setting one hard-to-converge component with blocking option, the setting is written as [3 constant+1 NK].
- AMG library [2].
  - One SGS iteration for smoother
  - SA-AMG as a preconditioner for CG method
  - Two levels of hierarchical structure
- Convergence condition: relative residuals < 1×10^{-8}
- The problem matrices nd3k and bone010 were used from Florida university matrix collection.
- Iteration number for convergence and run time(Setup Part, Iterative Part, NK Vec Part) were measured when setting hard-to-converge components using the matrix G.
- The hard-to-converge components were calculated by applying 10 iteration of the power method of the G matrix.

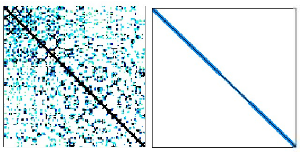
$$\begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix} \rightarrow \begin{pmatrix}1&0\\0&1\\1&0\\0&1\\1&0\\0&1\end{pmatrix}$$

Figure1

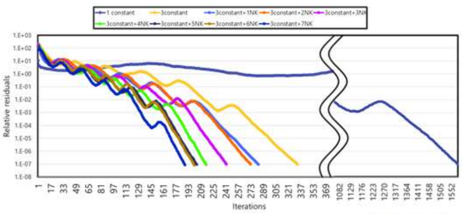
$$\begin{pmatrix}1\\1\\1\\1\\1\\1\end{pmatrix} \rightarrow \begin{pmatrix}1&0&0\\0&1&0\\0&0&1\\1&0&0\\0&1&0\\0&0&1\end{pmatrix}$$

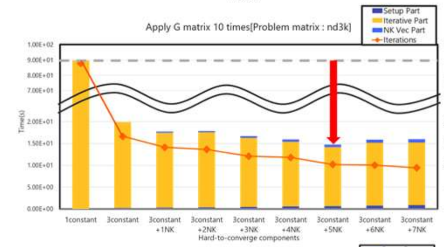
Figure2

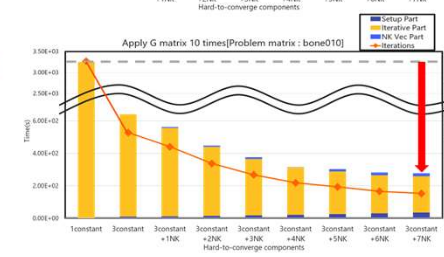
Name of the problem matrix	Rows	Nnz	Kind
Nd3k	9000	3279690	2D/3D Problem
bone010	986703	47851783	2D/3D Problem



nd3k bone010







Result

- Setting the large eigenvalue components of the matrix G as near kernel components of SA-AMG solver significantly reduced the execution time.
- The number of iterations to convergence also decreased as the number of hard-to-converge components was increased.

Conclusion

We computed hard-to-converge components by applying 10 iteration of Gauss-Seidel smoother to multiple random vectors.
Then they were set in the SA-AMG solver.
As a result, we were able to reduce the vector extraction cost and to speed up the execution time until convergence. In the future, we will verify whether it is effective for other problems.

References

[1] N. Nomura, K. Nakajima, A. Fujii. A Study on Near Kernel Vector Extraction Method for Robust SA-AMG Method. IPSJ Research Report. 2020.Vol2020-HPC-173.No.2
[2]High Performance Computing Laboratory, Department of Computer Science, School of Informatics, Kogakuin University. AMGS Manual. 2015.May 25.