Robust SA-AMG Solver by Extraction of Near-Kernel Vectors

Naoya Nomura†, Kengo Nakajima†, Akihiro Fujii‡
†The University of Tokyo, ‡Kogakuin University

Introduction

The smoothed aggregation algebraic multigrid (SA-AMG) method is among the fastest solvers for large-scale linear equations [1]. This method achieves good convergence by generating small matrices from the original matrix problem. Moreover, the convergence of the method can be further improved by setting near-kernel vectors. Generally, the same number of near-kernel vectors are used at each level. In the present work, we propose and investigate a method that extracts and adds near-kernel vectors at coarser levels.

The SA-AMG method

The SA-AMG method creates multiple small matrices and uses them to solve the original matrix.
- It has two part.

- The setup part: Creates coarser matrices
  - In this part, we can set the near-kernel vectors when the coarser level matrices are created.
- The solution part (Fig. 1): Solves the original problem by using the coarser matrices
  - This process is called the V-cycle.

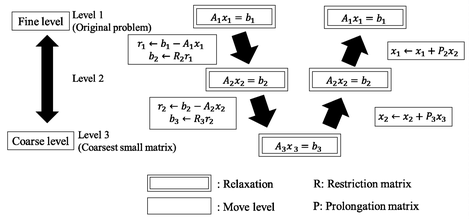

Figure 1: Solution part (V-cycle)

Near-kernel vector extraction

Almost all of the related works extract the near-kernel vectors at only level 1.
(see, e.g., our previous research [2])
- We consider varying the number near-kernel vectors by matrix level.

⇒In this research, we use a method that extracts the near-kernel vectors at all levels (not only level 1).
(Details of the proposed method are shown in Fig. 2)

```
1  For level = 1 to max_level − 1
2      For n = 1 to extract_number
3          Random(x)
4          V_cycle^μ (A_level x = 0)
5          B_level ← x
6          Multilevel_creation(B_level)
7      End for
8  End for
9  Output B as the near_kernel vectors
```

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

Figure 2: Details of method

Numerical tests

In the present work, we evaluate the convergence and the execution time of SA-AMG by setting the extracted near-kernel vectors at coarser levels.
<Experimental environment and evaluation criteria>
- FX10(Oakleaf-fx) SPARC64^{TM}IXfx (16-core CPU per 1 Node) [3]
- Problem : three-dimensional elastic problem (15^3 what per 1 proc.)
  - Near-kernel vector from problem settings
    ⇒Parallel translation, rotation
- Linear solver : AMG preconditioned GPBiCG method
- Parallel model : Flat MPI model (Max: 512 proc.)
- Termination Condition : The 2-norm of relative residual reaches 10^{-7}
- Target cases of this experiment: Shown in Table 1
- We vary the number of near-kernel vectors at level 1 (details are shown in Table 2)

Table 1: Experimental cases

Candidates	Details
Case 1	No extraction at coarser levels
Case 2, Case 3	Number of additional near-kernel vectors by each coarser level: 1 (Case 2), 5 (Case 3) (proposed method)

Table 2: Near-kernel vector settings at level 1

Candidates	Details
3p	Parallel translation(3 vectors)
6p	3p(3 vectors)+rotation(3 vectors)
3p+1,...	3p(3 vectors)+extracted near-kernel vectors at level 1(≤ 7 vectors)

e.g.: The number of near-kernel vectors at Case 3 and 3p+1
⇒level 1:4, level 2:9, level 3:14

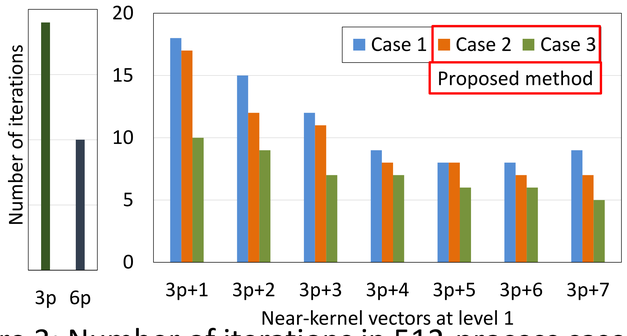

Figure 3: Number of iterations in 512-process case

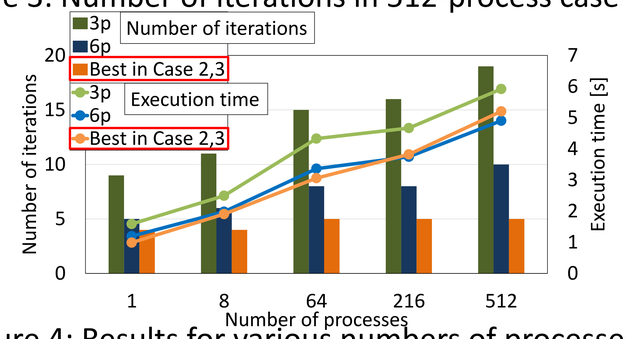

Figure 4: Results for various numbers of processes

- The convergence is improved by setting a suitable number of near-kernel vectors at coarser levels (Fig. 3).
- "Best in Case2,3" achieves good convergence, but, the execution time does not improve after 216 process case (Fig. 4).
  - This is due to a trade-off between the increasing cost and the improvement of the convergence by using near-kernel vectors.

Conclusion

In the present work, we propose a method that extracts near-kernel vectors at coarser levels, and investigate the performance for the SA-AMG method. The convergence is improved by setting a suitable number of near-kernel vectors at coarser levels, but the calculation cost of per iteration is increased by setting many near-kernel vectors. Therefore, we must find how to decide the optimal number of near-kernels as further research.

References

[1] Vanek, P., et al.: Convergence of Algebraic Multigrid Based on Smoothed Aggregation, Numerische Mathematik, vol 88, pp.559-579 (2001)
[2] Nomura, N., et al.: Performance Analysis of SA-AMG Method by Setting Extracted Near-kernel Vectors, VECPAR2016 (2016)
[3] Information Technology Center, The University of Tokyo: http://www.cc.u-tokyo.ac.jp/