Parallel Preconditioning Methods for Ill-Conditioned Problems by BILUT(*p*,*d*,*t*)

Kengo Nakajima[1]

[1] Information Technology Center, The University of Tokyo, 2-11-16 Yayoi, Bunko-ku, Tokyo 113-8658, Japan
nakajima@cc.u-tokyo.ac.jp

Abstract. We evaluated performance and robustness of parallel preconditioning methods for ill-conditioned problems based on BILUT(*p*,*d*,*t*). Two types of parallel implementations, LBJ (Localized Block Jacobi) and HID (Hierarchical Interface Decomposition), are applied. Developed methods are applied to Hetero3D code, which is a parallel finite-element benchmark program, and the code provided excellent scalability up to 240 nodes (3,840 cores) of Fujitsu PRIMEHPC FX10 (Oakleaf-FX), the University of Tokyo.

Keywords: Parallel Iterative Solvers, Preconditioning, Finite Element Method

1 Introduction

Incomplete LU factorization with threshold (ILUT) [1] is a widely-used preconditioning technique for solving various types of problems with ill-conditioned matrices. ILUT(*p*,*d*,*t*) [2] is a more practical one, where maximum fill-level *p* is specified before factorization, and *d* and *t* are parameters for dropping tolerance of non-zero off-diagonal components before/after factorization. Figure 1 describes the procedures of ILUT(*p*,*d*,*t*) method. The process (b) in Fig.1 can be substituted by other factorization methods or more powerful direct linear solvers, such as *MUMPS*, *SuperLU* and etc.

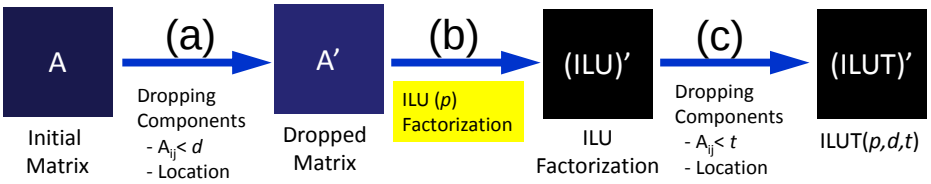

Fig.1. Procedures of ILUT(*p*,*d*,*t*) preconditioning, *p*: level of fill-in's, *d*: dropping tolerance of non-zero off-diagonal components before factorization, *t*: dropping tolerance after factorization

In the present work, parallel preconditioned iterative solvers based on ILUT(*p*,*d*,*t*) method were developed, and were implemented to *Hetero3D* code. Hetero3D is a benchmark code by parallel finite-element method (FEM) for evaluation of robustness of preconditioners, developed under JST-ANR FP3C project (Collaborative Project

between Japan and France on Framework and Programming for Post Petascale Computing) [3]. Large-scale ill-conditioned matrices derived from *Hetero3D* code were solved using up to 240 nodes (3,840 cores) of Fujitsu PRIMEHPC FX10 (Oakleaf-FX), Information Technology Center, the University of Tokyo [4].

2 Preliminary Results and Remarks

Figure 2 shows the speed-up of elapsed time of linear solvers and setting-up by strong scaling from 256 cores to 3,840 cores. Computation time for each case is normalized by that of BILUT(1,0, 2.50×10^{-2})-HID with 256 cores. Generally speaking, BILUT(2,*d*,*t*)-HID and BILUT(2,*d*,*t*)-LBJ-2 with optimum value of *t* are competitive, but BILUT(2,*d*,*t*)-HID is slightly more efficient and more robust. BILUT(2,*d*,*t*)-LBJ-2 did not converge in the case with 1,024 cores. Generally speaking, BILUT(2,0,*t*)-HID and BILUT(2,0,*t*)-LBJ-2 with optimum value of *t* are competitive, but BILUT(2,0,*t*)-HID is slightly more efficient and more robust. BILUT(2,0,*t*)-LBJ-2 did not converge in some cases.

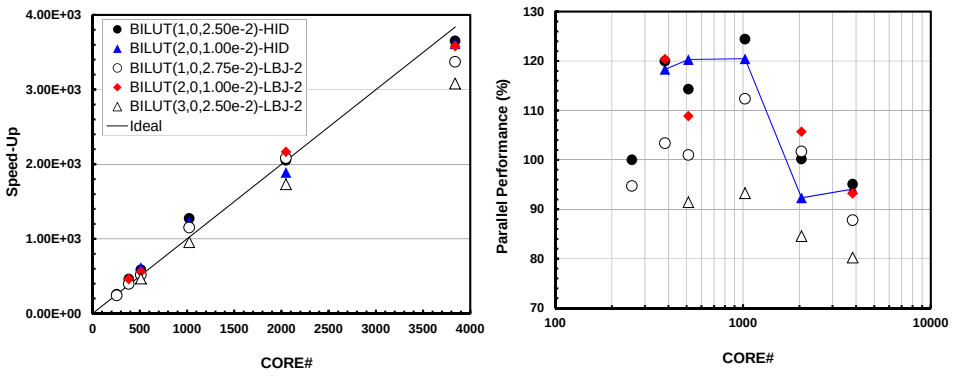

Fig.2. Strong Scalability of BILUT(*p*,*d*,*t*)-LBJ/HID-GPBi-CG with optimum *t* on the Oakleaf-FX, according to elapsed computation time (set-up+solver) for BILUT(1,0,2.5×10^{-2})-HID with 256 cores, $E_{max}=10^{-6}$, $E_{max}=10^{+6}$, d=0 for all cases

References

1. Saad, Y., Iterative Methods for Sparse Linear Systems (2nd Edition), SIAM, 2003.
2. Nakajima, K., Parallel Multistage Preconditioners by Extended Hierarchical Interface Decomposition for Ill-Conditioned Problems, Advances in Parallel ComputingVol.19, *From Multicores and GPU's to Petascale*, IOS press, 99-106, 2010.
3. JST-ANR FP3C: Collaborative Project between Japan and France on Framework and Programming for Post Petascale Computing: http://jfli.nii.ac.jp/medias/wordpress/?page_id=327
4. Information Technology Center, The University of Tokyo: http://www.cc.u-tokyo.ac.jp/