Fast Solution of a Magnetostatic Problem with Parallel ILU

M. Bollhoefer[1], R. Bianchetti[2], J. Ostrowski[2], and D. Pusch[2]

[1] Institute of Computational Mathematics, TU Braunschweig, D-38106 Braunschweig, Germany
m.bollhoefer@tu-bs.de
[2] ABB Switzerland, Corporate Research, 5405 Baden-Dättwil, Segelhofstrasse, Switzerland
Romeo.Bianchetti@ch.abb.com, Joerg.Ostrowski@ch.abb.com, David.Pusch@ch.abb.com

Summary. In this contribution we investigate the performance of a parallel ILU preconditioner for the iterative solution of a magnetostatic model problem. Using the magnetic vector potential **A** and conformal FEM-discretization results in a singular system matrix. We construct the preconditioner for the CG-solver by applying a shift for regularization, see [2, 3, 6]. The resulting regular matrix is then decomposed by the ILUPACK[1] library and is used for preconditioning. ILUPACK is a MPI-parallelized implementation of the inverse-based multilevel block ILU, see [4].

1 Introduction

The magnetostatic problem under consideration writes

$$\mathbf{curl}\,\frac{1}{\mu}\,\mathbf{curl}\,\mathbf{A} = \mathbf{j}. \tag{1}$$

Herein **A** is the magnetic vector potential, **j** is a prescribed divergence free current density, and μ is the possibly nonlinear magnetic permeability. Using conformal finite elements for the discretization leads to the following problem that is to be solved in weak formulation

$$(\frac{1}{\mu}\,\mathbf{curl}\,\mathbf{A}, \mathbf{curl}\,\mathbf{A}') = (\mathbf{j}, \mathbf{A}'). \tag{2}$$

The solution of magnetostatic problems in presence of nonlinear magnetic material can be time consuming. The change in the material parameters during the nonlinear iteration results in a change in the system matrix $\mathbf{M} := (\frac{1}{\mu}\,\mathbf{curl}\,\mathbf{A}, \mathbf{curl}\,\mathbf{A}')$. Magnetostatic problems are typically solved by preconditioned iterative solvers [2, 5, 7]. Jumps in the magnetic permeability deteriorate the condition of the problem, and preconditioning is mandatory. The preconditioner $\mathbf{P}^{-1}$ has to be updated many times if **M** changes too much during the outer nonlinear iteration, see Fig. 1. Therefore it is essential to provide a fast way of updating the preconditioner.

The system matrix **M** is singular. In this paper we regularize **M** by using a small shift $\beta \in \mathbb{R}$. This yields the preconditioner

$$\mathbf{P} := (\frac{1}{\mu}\,\mathbf{curl}\,\mathbf{A}, \mathbf{curl}\,\mathbf{A}') + (\frac{\beta}{\mu}\mathbf{A}, \mathbf{A}'). \tag{3}$$

[1] http://ilupack.tu-bs.de

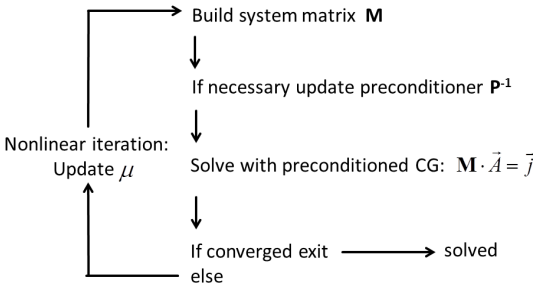

Fig. 1. Algorithm to solve a nonlinear magnetostatic problem.

The decomposition of **P** is then accomplished by using the mentioned MPI-parallelized ILUPACK library. In the next Section 2 we introduce the theoretical background of ILUPACK. In the numerical experiments of the final Section 3 we show the parallel scaling performance of the ILU-preconditioner as well as the preconditioning behavior itself.

2 Inverse-Based Multilevel Block ILU

For preconditioning the conjugate gradient method, ILUPACK computes an incomplete Cholesky-like factorization $\mathbf{P} \approx LDL^T$, where L is unit lower triangular and entries of small modulus are dropped. ILUPACK's hallmark is to keep $\|L^{-1}\|$ below a given bound κ during the factorization [4]. To do so, at each step l of the decomposition we either pursue the factorization whenever $\|L^{-1}\| \leqslant \kappa$, or we postpone a step, otherwise (cf. Figure 2). The block of postponed updates S_C (known as Schur complement) becomes the starting matrix of the next level. Using this inverse-based strategy and a moderate value of κ (e.g. $\kappa = 5$) it can be shown that small eigenvalues of **P** are revealed by S_C. Thus S_C serves as some kind of coarse grid system.

The parallelization of ILUPACK mainly consists of a nested dissection partitioning of the graph of **P**. This yields a hierarchy of subsystems which can be represented by an incomplete binary task tree. Starting with the leaves, the multilevel ILU is applied to all subsystems concurrently until these join the same parent task. Separators are factorized at last [1].

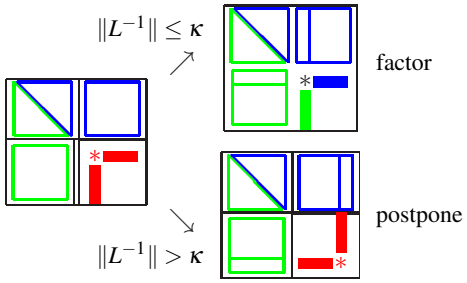

Fig. 2. ILUPACK pivoting strategy.

3 Numerical Experiments

Numerical experiments were made on the model problem of Fig. 3. It consists of a copper coil, and a non-conductive high permeable core. The tests were

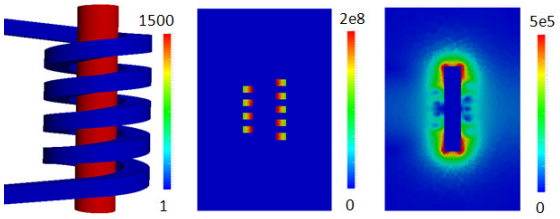

Fig. 3. Results of the magnetostatic field computations. The left picture shows the setting and the relative magnetic permeability, the central picture shows the exciting current density (1kA total current), and the right picture shows the resulting magnetic field $\mathbf{H} = \frac{1}{\mu_r \cdot \mu_0} \cdot \mathbf{curl}\,\mathbf{A}$.

carried out on a 12-core INTEL-Westmere workstation with 3.06GHz and activated hyper-threading. We chose $\beta = 0.01$ for the regularization parameter in (3). In our experiments we are testing the parallel performance up to 8 cores. The results of the CPU times of the solver are shown in Fig. 4. Therein, the key values of two different meshes for 1.2E+6 unknowns and for 1.02E+7 unknowns are drawn. One can observe a perfectly parallel scaling for the pure ILU factorization part. A slightly worse behavior can be seen for the preconditioned conjugate gradient (PCG) solver. Moreover, the PCG needs more time than the factorization for greater problem dimensions. This is due to the fact, that the number of iterations is also increasing with the number of unknowns, see Fig. 5.

Finally, it can be concluded that ILUPACK seems to be a very promising library for the fast parallel solution of magnetostatic problems. This needs to be confirmed for more complex geometries.

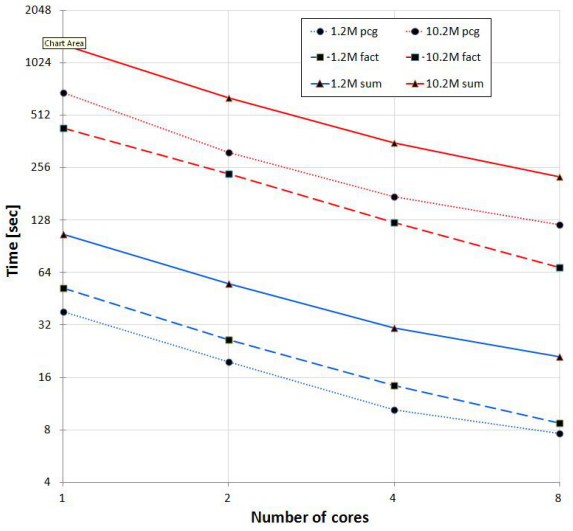

Fig. 4. CPU times for ILUPACK.

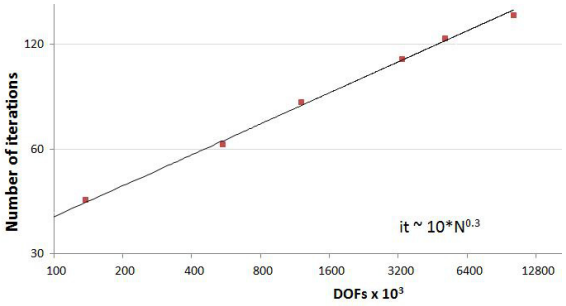

Fig. 5. Number of iteration steps.

References

1. J.I. Aliaga, M. Bollhöfer, A.F. Martín, and E.S. Quintana-Ortí. Exploiting thread–level parallelism in the iterative solution of sparse linear systems. *Parallel Comput.*, 37(3):183–202, 2011.
2. M. Bebendorf and J. Ostrowski. Parallel hierarchical matrix preconditioners for the curl-curl operator. *Journal of Computational Mathematics*, 2009, Volume 27, No. 5, pp. 624-641.
3. M. Bollhöfer und S. Lanteri. Block Preconditioning Strategies for High Order Finite Element Discretization of the Time-Harmonic Maxwell Equations *Springer Book, SCEE 2010 Mathematics in Industry*, 2012, Volume 16,
4. M. Bollhöfer and Y. Saad. Multilevel preconditioners constructed from inverse-based ILUs. *SIAM J. Sci. Comput.*, 27(5), 2006.
5. R. Hiptmair and J. Xu. Nodal auxiliary space preconditioning in H(curl) and H(div) spaces. *SIAM J. Numer. Anal.*, 45(6):2483-2509 (electronic), 2007.
6. J. Ostrowski, M .Bebendorf, F. Kraemer and R. Hiptmair. H-Matrix based operator preconditioning for full Maxwell at low frequencies. *IEEE Transactions on Magnetics*, Volume 46, Number 8, pp. 3193-3196, Aug 2010, ISSN 0018-9464
7. S. Reitzinger and J. Schoeberl. An algebraic multigrid method for finite element discretizations with edge elements. *Numer. Linear Algebra Appl.*, 9(3):223-238, 2002.