Journal of Computational Mathematics
Vol.39, No.4, 2021, 633–654.
http://www.global-sci.org/jcm
doi:10.4208/jcm.2009-m2020-0138

ITERATIVE ILU PRECONDITIONERS FOR LINEAR SYSTEMS AND EIGENPROBLEMS*

Daniele Boffi
Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Kingdom of Saudi Arabia
Dipartimento di Matematica "F. Casorati" University of Pavia Via Ferrata 5, 27100 Pavia, Italy
Email: daniele.boffi@kaust.edu.sa
Zhongjie Lu[1)]
Dipartimento di Matematica, Università di Pavia, via Ferrata 5, 27100 Pavia, Italy
School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, China
Email: zhjlu@ustc.edu.cn
Luca F. Pavarino
Dipartimento di Matematica, Università di Pavia, via Ferrata 5, 27100 Pavia, Italy
Email: luca.pavarino@unipv.it

Abstract

Iterative ILU factorizations are constructed, analyzed and applied as preconditioners to solve both linear systems and eigenproblems. The computational kernels of these novel Iterative ILU factorizations are sparse matrix-matrix multiplications, which are easy and efficient to implement on both serial and parallel computer architectures and can take full advantage of existing matrix-matrix multiplication codes. We also introduce level-based and threshold-based algorithms in order to enhance the accuracy of the proposed Iterative ILU factorizations. The results of several numerical experiments illustrate the efficiency of the proposed preconditioners to solve both linear systems and eigenvalue problems.

Mathematics subject classification: 65F08, 65F15, 15A23.
Key words: Iterative ILU factorization, Matrix-matrix multiplication, Fill-in, Eigenvalue problem, Parallel preconditioner.

1. Introduction

The LU factorization is an efficient direct method to solve linear systems. For sparse problems, the computed L and U factors might suffer from fill-in and lose sparsity compared with the original matrix. For large-scale problems, this fact generates factors that are both time consuming to compute and memory consuming to store. In this case, iterative methods become attractive alternatives, see, e.g., [6, 19, 32], since they usually require less memory and floating point computation. However, iterative methods might suffer from lack of robustness and slow convergence, and preconditioning techniques become necessary in order to make iterative methods faster and more reliable. While many preconditioners are designed for special problems, incomplete LU (ILU) preconditioners are relatively general, since they are based on the LU factorization.

* Received May 27, 2020 / Revised version received August 17, 2020 / Accepted September 21, 2020 / Published online June 11, 2021 /
1) Corresponding author

There are a large number of references for the analysis and applications of ILU preconditioners in the literature of mathematics, physics and computer science, see e.g., [7, 24, 35], and ILU preconditioners are implemented in many numerical libraries [5, 26]. The convergence speed of most iterative methods depends on the distribution of the spectra of the matrices involved. In [8,30], it has been proven that incomplete factorization preconditioners can essentially reduce the condition number for some matrices. There are many modifications of standard ILU based on various techniques. For example, high-level ILU(p) [32] enhances the accuracy of the factorization by allowing more fill-ins, modified ILU (MILU) [20, 27] compensates diagonal entries to make the row or column sums of the multiplication LU equal to the ones of the original matrix. Threshold ILU (ILUT(p, τ)) [31] introduces a tolerance and a number of fill-ins into the dropping rules, which provide more options to balance the accuracy and storage of the factors. There are also many references about the stability and implementation of ILUs, see, e.g., [4, 16, 33].

Gaussian elimination, the core technique of ILU factorization, is a highly sequential procedure, which is an obstacle to ILUs' parallelization. Consequently, many other techniques have been introduced in order to improve the parallelism of LU factorizations. Graph coloring techniques and domain decomposition techniques have been used to partition a matrix into several submatrices, where factorizations can then be executed on each submatrix, see, e.g., [21, 22, 28]. In [13], a factorization is designed based on residual correction, which can be implemented in parallel. Recently, a type of fine-grained parallel incomplete factorization based on fixed-point iterations, has been proposed in [1, 3, 14] and it has been successfully implemented in shared-memory environments.

In this paper, we will construct and analyze an iterative ILU factorization of a given matrix. The main procedure of our new algorithms is based on sparse matrix-matrix multiplications. This matrix operation is nowaday contained in many computing libraries and implemented in efficient codes, which can be much faster than computing the entries one by one according to the definition. If we take full advantage of these techniques and libraries, the new algorithms are easy to implement on both share-memory and distributed-memory systems without the need to enter into the complex details of optimized serial and parallel implementations, in particular interprocessor communications. We also present two modifications of the new Iterative ILU algorithms based on high-level and threshold variants aimed at improving the computational efficiency.

Throughout this paper, we will not study the parallelism of the new algorithms explicitly, since the proposed preconditioners and iterative solvers require only matrix-matrix and matrix-vector multiplications, which have intrinsically fine-grained parallelism and have been implemented efficiently on different computing architectures.

The paper is organized as follows. In Section 2, we derive the conventional LU factorization in a matrix form and recall some of its fill-in property. We then propose several new Iterative ILU preconditioners and prove some convergence results in Section 3. In Section 4, we discuss some issues involved in the new algorithms, such as matrix-matrix multiplication and solving triangular systems. In Sections 5 and 6, we apply the new preconditioner for solving linear systems and eigenvalue problems, respectively, showing the results of several numerical experiments. Some conclusions and the limitations of the proposed preconditions are reported in Section 7.