

A novel approach to level-based preconditioning

List of authors:

Jennifer Scott ¹

Miroslav Tůma ²

Incomplete Cholesky factorizations are an important tool in the solution of large sparse symmetric linear systems of equations $Ax = b$. Preconditioners based on an incomplete factorization of A (that is, a factorization in which some of the fill entries and possibly some of the entries of A are ignored) fall into three main classes:

- (i) Threshold-based $IC(\tau)$ methods in which the locations of permissible fill entries are determined in conjunction with the numerical factorization of A ; entries of the computed factors that exceed a prescribed threshold τ are dropped. Success of this approach depends on being able to choose a suitable τ and this is highly problem dependent.
- (ii) Memory-based $IC(m)$ methods in which the amount of memory available for the incomplete factorization is prescribed and only the largest entries in each column are retained.
- (iii) Structure-based $IC(\ell)$ methods in which an initial symbolic factorization phase determines the location of permissible fill entries using only the sparsity pattern of A . Each potential fill entry is assigned a level and an entry is only permitted in the factor if its level is at most ℓ . This kind of fixed fill strategy allows the memory requirements to be determined before the second phase that performs an incomplete numerical factorization.

During the last three decades, numerous refinements and variants have been proposed and used to solve problems from a wide range of application areas. Our interest is in structure-based incomplete factorization preconditioners that have both predictable memory requirements and depend on the entries of A . We propose [2] a general class of methods based on computing an incomplete LDL^T factorization $IC(\ell, \tau, m)$, where the factor L is unit lower triangular and D is diagonal, $\ell \geq 0$ is the target number of levels of fill, τ is a drop tolerance and m controls the maximum number of entries allowed in the factor.

A key feature of our approach is to allow individual entries of the system matrix A to contribute to a variable number of levels of fill. Rather than all the non-zero entries of A contributing to ℓ levels of fill, we restrict small entries to contributing to fewer levels than ℓ levels while at the same time allowing the largest entries to contribute to more than ℓ levels. We explain how this variable level approach can be implemented using a minor change to the algorithm of Hysom and Pothen [1] for computing a symbolic incomplete factorization. Using symmetric positive-definite problems arising from a wide range of practical applications, we show that the

¹Computational Science and Engineering Department, Atlas Centre, Rutherford Appleton Laboratory, Oxfordshire OX11 0QX, England.

²Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 2, 18207 Praha 8, Libeň.

use of variable levels of fill can yield incomplete Cholesky factorization preconditioners that are more efficient than those resulting from the standard level-based approach.

The concept of level-based preconditioning, which is based on the structural properties of the system matrix, is then transferred to the numerical incomplete decomposition. In particular, the structure of the incomplete factorization determined in the symbolic factorization phase is explicitly used in the numerical factorization phase. Further numerical results demonstrate that our novel level-based approach can lead to much sparser but efficient incomplete factorization preconditioners, confirming that the proposed approach is viable and can be regarded as one step in improving basic incomplete factorization preconditioning strategies.

References

- [1] Hysom, D., Pothen, A.: A scalable parallel algorithm for incomplete factor preconditioning. *SIAM J. Sci. Comput.* **22**, 2194–2215 (2001).
- [2] Scott, J., Tuma, M.: The importance of structure in incomplete factorization preconditioners. *BIT*. To appear (2011).