[20] V. Pan and J. Reif. Fast and Efficient Parallel Solution of Linear Systems *Proc. 17th Annual ACM Symposium on Theory of Computing*, 143–152, 1985.

[21] F. P. Preparata and D. V. Sarwate. An Improved Parallel Processor Bound in Fast Matrix Inversion *Inform. Process. Lett.*, 7:148–149, 1978.

[22] E. L. Yip. A Note on the Stability of Solving a Rank-p Modification of a Linear System by the Sherman-Morrison-Woodbury Formula. *SIAM J. Sci. Stat. Comput.*, 7:507–513, 1986.

[23] S. Zohar. Toeplitz Matrix Inversion: The Algorithm of W. F. Trench *J. Assoc. Comput. Mach.*, 16:592–601, 1969.

[6] A. K. Chandra. Maximal Parallelism in Matrix Multiplication *IBM Research Report R.C. 6193*, 1976.

[7] B. Codenotti and M. Leoncini. Incomplete Factorization Methods for Banded Toeplitz Matrices *I.E.I. Technical Report B4-18*, Istituto di Elaborazione dell'Informazione del CNR, Pisa, 1989.

[8] B. Codenotti and F. Romani. Stability and Complexity of Newton's Method Applied to Parallel Matrix Inversion *International Conference on Linear Algebra and its Applications*, Valencia, 1987.

[9] P. Concus and G. Meurant. On Computing INV Block Preconditionings for the Conjugate Gradient Method. *BIT*, 26:493–504, 1986.

[10] L. Csanky. Fast Parallel Matrix Inversion Algorithms *SIAM J. Comput.*, 5:618–623, 1976.

[11] G. J. Davis. Numerical Solution of a Quadratic Matrix Equation. *SIAM J. Sci. Stat. Comput.*, 2:164–175, 1981.

[12] D. Fischer, G. Golub, O. Hald, C. Leiva, and O. Widlund. On Fourier-Toeplitz Methods for Separable Elliptic Problems *Mathematics of Computation*, 28:349–368, 1974.

[13] S. Fortune and J. Wyllie. Parallelism in Random Access Machines *Proc. 10th Annual ACM Symposium on Theory of Computing*, 114–118, 1978.

[14] J. von zur Gathen. Parallel Arithmetic Computations: A Survey *12th Int. Symp. Math. Found. of Computer Science, Lecture Notes in Computer Science*, 233:93–112, 1986.

[15] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Md, 1989.

[16] J. Grcar and A. Sameh. On Certain Parallel Toeplitz Linear System Solvers. *SIAM J. Sci. Stat. Comput.*, 2:238–256, 1981.

[17] D. Heller. A Survey of Parallelism in Numerical Linear Algebra. *SIAM Rev.*, 20:740–777, 1978.

[18] T. Kailath, A. Vieira and M. Morf. Inverses of Toeplitz Operators, Innovations, and Orthogonal Polynomials *SIAM Rev.*, 20:106–119, 1978.

[19] W. Kratz and E. Stickel. Numerical Solution of Matrix Polynomial Equations by Newton's Method. *IMA J. of Numerical Analysis*, 7:355–369, 1987.

Example 6. A is 6×6 Toeplitz, while B and C^T are lower triangular Toeplitz such that T is banded Toeplitz:

$$A = \begin{pmatrix} 0.516679 & 0.279703 & 0.127793 & 0.0839437 & -0.11893 & 0.277015 \\ -0.164592 & 0.516679 & 0.279703 & 0.127793 & 0.0839437 & -0.11893 \\ 0.864845 & -0.164592 & 0.516679 & 0.279703 & 0.127793 & 0.0839437 \\ -0.68101 & 0.864845 & -0.164592 & 0.516679 & 0.279703 & 0.127793 \\ -0.742669 & -0.68101 & 0.864845 & -0.164592 & 0.516679 & 0.279703 \\ -0.852 & -0.742669 & -0.68101 & 0.864845 & -0.164592 & 0.516679 \end{pmatrix},$$

$$\begin{aligned} B_{*1} &= (\; 0.409388 \quad 0.277015 \quad -0.11893 \quad 0.0839437 \quad 0.127793 \quad 0.279703 \;)^T, \\ C_{1*} &= (\; -0.0525222 \quad -0.852 \quad -0.742669 \quad -0.68101 \quad 0.864845 \quad -0.164592 \;). \end{aligned}$$

With $X_0^- \approx (0.56912 + 0.249638i)I$, we obtain $X = X_r + iX_i$ where

$$X_r = \begin{pmatrix} -0.19786 & 0.04579 & 0.17548 & -0.13383 & -0.13147 & -0.03350 \\ 0.08550 & -0.18569 & 0.07363 & 0.19408 & -0.15969 & -0.11016 \\ 0.17924 & 0.11816 & -0.02339 & 0.11386 & 0.06945 & -0.09011 \\ 0.26615 & 0.21461 & 0.16794 & 0.03407 & 0.05834 & 0.12694 \\ -0.06093 & 0.23906 & -0.07352 & 0.15148 & 0.23425 & -0.02114 \\ 0.11671 & -0.04871 & 0.21307 & -0.04890 & 0.15953 & 0.24800 \end{pmatrix},$$

$$X_i = \begin{pmatrix} 0.10522 & -0.00212 & -0.07364 & 0.02842 & 0.03877 & -0.00327 \\ -0.05883 & 0.10155 & 0.04718 & -0.08562 & -0.00020 & 0.04053 \\ -0.31445 & -0.08884 & 0.21435 & -0.01865 & -0.13629 & -0.02701 \\ -0.11535 & -0.31798 & 0.06154 & 0.19150 & -0.11108 & -0.12000 \\ 0.53519 & -0.05517 & -0.37719 & 0.17518 & 0.18779 & -0.03365 \\ 0.03734 & 0.54391 & 0.00649 & -0.36847 & 0.13035 & 0.20896 \end{pmatrix},$$

in $n_0 \leq 16$ iterations, and $\|F(\tilde{X})\|_\infty 1.77824 \cdot 10^{-10}$. With $X_0^+ \approx 1.4118I$, Newton's method fails to converge. ■

References

[1] G. Ammar and W. Gragg. Superfast Solution of Real Positive Definite Toeplitz Systems *SIAM J. Matrix Anal. Appl.*, 9:61–76, 1988.

[2] O. Axelsson. Incomplete Block Matrix Factorization Preconditioning Methods. The Ultimate Answer? *J. of Computational and Applied Mathematics*, 12:3– 18, 1985.

[3] O. Axelsson and B. Polman. On Approximate Factorization Methods for Block Matrices Suitable for Vector and Parallel Processors. *Linear Algebra and its Applications*, 77:3–26, 1986.

[4] R. Bevilacqua, B. Codenotti, and F. Romani. Parallel Solution of Block Tridiagonal Linear Systems *Linear Algebra Appl.*, 104:391–57, 1988.

[5] R. H. Chan and G. Strang. Toeplitz Equations by Conjugate Gradients with Circulant Preconditioner *SIAM J. Sci. Stat. Comput.*, 10:104–119, 1989.

$$B = \begin{pmatrix} 0.517239 & 0.542287 & -0.617065 & -0.627036 \\ 0.992267 & -0.835852 & -0.110293 & 0.0754547 \\ -0.949065 & 0.276742 & 0.481544 & 0.681173 \\ -0.978031 & 0.396513 & 0.220374 & 0.359957 \end{pmatrix}.$$

With $X_0^- \approx (0.437063 + 0.70993i)I$, Newton's method converges to

$$\tilde{X} = 10^{-12} \cdot \begin{pmatrix} 0.580130 + 5.34110i & -0.540627 - 1.07642i & -0.060226 - 1.28542i & -0.125883 - 2.73068i \\ 0.101667 - 0.40692i & -0.426473 + 3.11874i & -0.304544 - 2.38783i & -0.301464 - 4.03469i \\ 0.269826 + 5.47303i & -0.464609 - 3.32431i & 0.021286 + 0.37304i & 0.013383 + 0.18637i \\ 0.238554 - 6.02057i & 0.106400 + 1.69698i & -0.025332 + 0.99756i & -0.484515 + 2.34629i \end{pmatrix}$$

in $n_0 \leq 14$ iterations, with $\|F(\tilde{X})\|_\infty \approx 7.123 \cdot 10^{-11}$. Starting from $X_0^+ \approx 1.37836I$, we obtain

$$\bar{X} = \begin{pmatrix} 0.58013 & -0.540627 & -0.0602264 & -0.125883 \\ 1.01667 & -0.426473 & -0.304544 & -0.301464 \\ 0.269826 & -0.464609 & 0.21286 & 0.133826 \\ 0.238554 & 0.1064 & -0.253317 & -0.484515 \end{pmatrix}$$

in $n_1 \leq 12$ iterations, with $\|F(\bar{X})\|_\infty \approx 3.38192 \cdot 10^{-15}$. ∎

Example 5. A is 4×4 Toeplitz, while B and C^T are lower triangular Toeplitz such that T is banded Toeplitz:

$$A = \begin{pmatrix} 0.461566 & 0.858435 & -0.490227 & 0.707031 \\ 0.22595 & 0.461566 & 0.858435 & -0.490227 \\ -0.414279 & 0.22595 & 0.461566 & 0.858435 \\ 0.736799 & -0.414279 & 0.22595 & 0.461566 \end{pmatrix},$$

$$B_{*1} = \begin{pmatrix} -0.906584 \\ 0.707031 \\ -0.490227 \\ 0.858435 \end{pmatrix}, \quad C_{1*} = \begin{pmatrix} 0.974593 & 0.736799 & -0.414279 & 0.22595 \end{pmatrix}.$$

Starting from $X_0^- \approx (0.535218 + 0.986517i)I$, Newton's method converges to

$$\tilde{X} = 10^{-17} \cdot \begin{pmatrix} 0.127893 - 0.53558i & -0.092363 + 1.14785i & 0.079576 + 2.64198i & 0.200933 - 3.85174i \\ -0.015835 - 0.30269i & 0.127074 + 0.50505i & -0.079558 + 1.24412i & 0.106000 - 1.23528i \\ -0.000835 - 0.06315i & -0.001627 - 0.06941i & 0.012775 - 0.03769i & -0.006562 + 0.79251i \\ 0.005171 - 0.62644i & -0.080865 + 1.31353i & -0.166859 + 3.04535i & 1.268860 - 4.40729i \end{pmatrix}$$

in $n_0 \leq 14$ iterations, giving $\|F(\tilde{X})\|_\infty \approx 3.84185 \cdot 10^{-16}$. With $X_0^+ \approx 1.77865I$, we obtain

$$\bar{X} = \begin{pmatrix} 1.27893 & -0.0923633 & 0.0795764 & 0.200933 \\ -0.158357 & 1.27074 & -0.0795578 & 0.106 \\ -0.083539 & -0.162677 & 1.27749 & -0.0656186 \\ 0.0517145 & -0.0808646 & -0.166859 & 1.26886 \end{pmatrix},$$

in $n_1 \leq 12$ iterations, and $\|F(\bar{X})\|_\infty \approx 3.76787 \cdot 10^{-16}$. Note that both $\tilde{X}$ and $\bar{X}$ approximate the same (real) solvent. ∎

$$X_i = \begin{pmatrix} 0.41713 & 0.06611 & 0.24949 & 1.18596 & -0.16937 & -0.15012 \\ -0.10747 & 0.95104 & -0.04612 & 0.47739 & -0.15629 & 0.20858 \\ 0.66944 & 0.07775 & 0.64334 & -0.43617 & -0.00145 & 0.62218 \\ 0.13185 & 0.03048 & -0.13581 & 0.28319 & 0.20814 & -0.13234 \\ 0.56233 & 0.05984 & -0.30451 & -1.13265 & 1.12261 & 0.29882 \\ -0.09639 & -0.06417 & 0.01498 & -1.12158 & 0.15423 & 0.32263 \end{pmatrix},$$

in $n_0 \leq 16$ iterations, with $\|F(\tilde{X})\|_\infty \approx 5.53918 \cdot 10^{-16}$. Starting from $X_0^+ \approx 1.14178I$ Newton's method converges to the real solvent

$$\bar{X} = \begin{pmatrix} 0.620434 & -0.75959 & -0.109756 & -0.563788 & -0.178393 & 0.831679 \\ 0.125155 & 0.0939193 & 0.0621351 & 0.577561 & -0.766806 & 0.470617 \\ -0.461337 & 0.785254 & -0.0549352 & -0.10672 & -0.287831 & 0.0880682 \\ -0.235339 & -0.358634 & 0.401347 & -0.0833514 & 0.0544886 & -0.547929 \\ 0.215709 & 0.614566 & 0.652756 & 0.277448 & 0.23819 & -0.25092 \\ -1.1711 & 0.846046 & 0.31629 & 0.665308 & 0.0193916 & -1.17968 \end{pmatrix}$$

in $n_1 \leq 16$ iterations, and $\|F(\bar{X})\|_\infty \approx 1.28591 \cdot 10^{-9}$. ■

Example 3. $A = I_4$, $C = -B$:

$$B = \begin{pmatrix} 0.00516658 & 0.733489 & 0.428635 & 0.531839 \\ -0.753396 & 0.606722 & 0.49199 & 0.213826 \\ 0.495661 & -0.202535 & 0.924642 & -0.0790184 \\ 0.259717 & -0.753891 & -0.404637 & -0.280178 \end{pmatrix}.$$

With $X_0^- \approx (0.242021 + 0.970271i)I$, Newton's method converges to the solvent

$$\tilde{X} = 10^{-11} \cdot \begin{pmatrix} 0.056420 - 0.233377i & -0.792390 - 3.667700i & -0.004604 + 0.923050i & -0.534639 - 2.670220i \\ 1.201400 + 2.676740i & -0.138980 - 4.035120i & -0.054107 - 0.992251i & 0.144338 - 2.769190i \\ -0.005061 - 0.056853i & 0.218162 + 1.416670i & -0.046156 + 0.129123i & 0.110317 + 1.028050i \\ -0.081672 + 0.331851i & 0.830910 + 3.702880i & 0.045067 - 0.125420i & 0.312903 + 2.701530i \end{pmatrix}$$

in $n_0 \leq 10$ iterations, giving $\|F(\tilde{X})\|_\infty \approx 1.15583 \cdot 10^{-10}$. With $X_0^+ \approx 1.27089I$, we obtain

$$\bar{X} = \begin{pmatrix} 0.894743 & -0.74742 & 0.28599 & -0.442651 \\ 1.41626 & -0.109748 & -0.0558818 & 0.204132 \\ 0.346077 & 0.334101 & 1.46281 & 0.347477 \\ -1.12301 & 0.78924 & -0.240963 & 0.227666 \end{pmatrix}$$

in 7 iterations, and $\|F(\bar{X})\|_\infty \approx 3.81937 \cdot 10^{-16}$. ■

Example 4. $A = A^T$ (but not positive definite), and $C = B^T$:

$$A = \begin{pmatrix} -0.504133 & 0.916786 & -0.941018 & 0.642083 \\ 0.916786 & 0.581326 & -0.296251 & -0.968074 \\ -0.941018 & -0.296251 & 0.530552 & -0.756876 \\ 0.642083 & -0.968074 & -0.756876 & -0.549169 \end{pmatrix};$$

to develop fast and efficient sequential and parallel algorithms (see section 4). Furthermore, the matrices of the factorizations could be computed with cost depending only on the bandwidth (i.e. not on the size of the system). The LL^T factorization can be always applied. We have studied the existence of these factorizations and shown how this problem is related to the existence of a solution of certain nonlinear matrix equations. In particular we have stated the relation between the factorization $T = LL^T + K$ and the equation $ZZ^T + FZ^T + G = O$. We have determined the general form of a solution of this equation and found conditions (either necessary or sufficient) for the existence of a solution. Finally we have performed a number of computational experiments which show that the matrix equation related to the LU incomplete factorization does have a solution in many cases not covered by the theory.

A Appendix

In the following we list six examples of resolution of quadratic matrix equations.
Example 1. $A = I_4$, $C = B$:

$$B = \begin{pmatrix} 0.185442 & 0.415787 & -0.568279 & -0.762019 \\ 0.812455 & 0.465055 & -0.656925 & -0.0738415 \\ 0.991509 & -0.968499 & -0.0970314 & -0.392705 \\ 0.131298 & 0.51184 & -0.910668 & 0.771585 \end{pmatrix}.$$

With $X_0^- \approx (0.204103 + 0.978949i)I$, we obtain

$$\tilde{X} = \begin{pmatrix} -0.370603 + 0.940136i & 0.707384 - 0.102185i & 0.031346 + 0.162928i & -0.282357 + 0.101734i \\ -0.147816 - 0.160286i & 0.62605 + 0.982378i & -0.444024 + 0.085956i & -0.31206 - 0.015953i \\ -0.513037 - 0.244091i & 0.544864 + 0.100227i & -0.271994 + 1.0937i & -0.592966 + 0.005366i \\ -0.444395 - 0.026003i & 0.107409 + 0.124542i & -0.031808 + 0.026261i & 0.203222 + 0.923239i \end{pmatrix}$$

in 7 iterations, with $\|F(\tilde{X})\|_\infty \approx 8.05067 \cdot 10^{-15}$. Starting from $X_0^+ \approx 1.22472I$ Newton's method fails to converge. ■

Example 2. $A = I_6$, $C = B$:

$$B = \begin{pmatrix} -0.117961 & -0.722142 & -0.41316 & -0.788115 & -0.268241 & 0.56837 \\ 0.221061 & 0.210068 & 0.334215 & 0.39728 & -0.87456 & 0.732023 \\ 0.326577 & 0.999912 & -0.0197027 & -0.35755 & 0.329825 & 0.470401 \\ -0.414453 & -0.52622 & 0.262314 & -0.320514 & 0.00550452 & -0.621506 \\ 0.979834 & 0.0592741 & 0.794505 & -0.123669 & 0.684797 & -0.0894222 \\ -0.352167 & 0.87444 & 0.573304 & 0.735454 & 0.314397 & -0.910233 \end{pmatrix}.$$

With $X_0^- \approx (0.132979 + 0.991119i)I$, we obtain $X = X_r + iX_i$ where

$$X_r = \begin{pmatrix} 0.16675 & 0.20072 & -0.75241 & -0.29353+ & 0.06194 & 0.73149 \\ 0.47776 & 0.15227 & -0.06188 & -0.36409 & -0.08412 & 0.77690 \\ 0.60712 & 0.22441 & 0.14749 & -0.20536 & 0.49309 & 0.14988 \\ -0.72694 & -0.14407 & 0.06448 & -0.12861 & -0.01095 & -0.69993 \\ -0.05172 & -0.55553 & 0.47431 & -0.38670 & 0.34290 & -0.67070 \\ -0.47075 & -0.12145 & 0.89753 & 0.44735 & 0.06330 & -1.09691 \end{pmatrix},$$

where $a = \|A\|_\infty / \|C\|_\infty$ and $b = \|B\|_\infty / \|C\|_\infty$. In the latter case we always obtain real iterations, while in the former we can also look for complex solutions. The actual entries of the matrices were chosen at random, uniformly in the interval $(-1, 1)$. Six numerical examples are reported in the appendix.

We have also tested several types of structured A, B, C matrices of different size, and performed hundreds of random tests for each combination. A summary of some of the outcomes of this analysis is depicted in Figure 1. We found that often a solvent exists in many cases not satisfying the sufficient conditions of section 6.

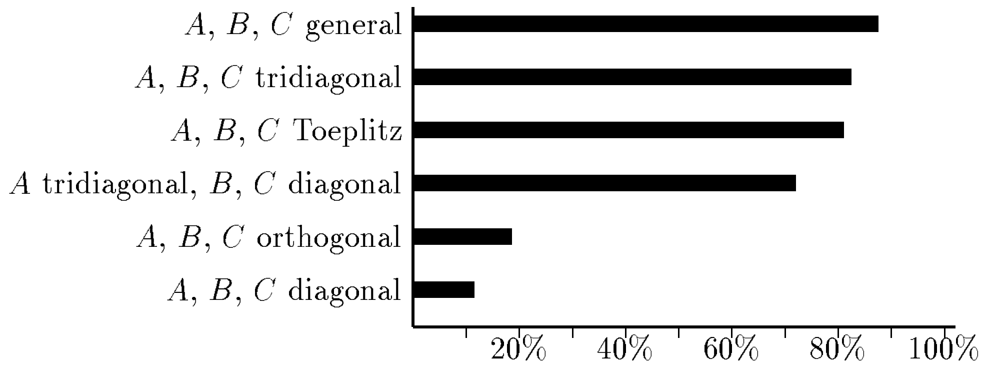

Figure 1: Percentages of solvent existence for some types of A, B and C.

(2) We have computed the L factor both in the element case (i.e. L lower bidiagonal) and in the block case (i.e. L banded lower triangular). In the former case, Newton's iterative method has been used to solve a system of two nonlinear equations. In all the experiments, we have been able to compute the solution. In the latter case, we have developed a LISP program for symbolically determining the nonlinear equations needed to minimize the Frobenius norm of the matrix $LL^T - T$ (see section 3.2), and for symbolically computing the partial derivatives needed to implement Newton's method. Such a program directly produces the FORTRAN code for actual (numeric) computations. No significant differences were observed with respect to the scalar case.

(3),(4) We evaluated the spectral radius of the iteration matrices. In all experiments, we found convergent iterative methods, and the LL^T factorization has shown itself to be more suited to generate a fast iterative method. This fact is quite natural, since this factorization is obtained corresponding to the minimization of the Frobenius norm of a matrix with spectral properties close to the ones of the iteration matrix.

8 Conclusions

In this paper we have presented a framework for preconditioning and solving banded Toeplitz linear systems by using incomplete factorizations. These techniques allowed

It is interesting to compare this approach with an incomplete Cholesky factorization of the matrix (26). In the latter case we maintain the block pentadiagonal view of the matrix and write

$$T = LL^H + H,$$

where

$$L = \begin{pmatrix} I & & & & \\ B & I & & & \\ I & B & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & B & I \end{pmatrix},$$

and

$$H = \begin{pmatrix} I+\hat{E} & & & & \\ & \hat{E} & & & \\ & & \ddots & & \\ & & & \hat{E} & \\ & & & & I+\hat{E} \end{pmatrix} + \begin{pmatrix} I_{2k} \\ O \\ \vdots \\ \vdots \\ O \end{pmatrix} \begin{pmatrix} B^2+I & B \\ B & I \end{pmatrix} \begin{pmatrix} I_{2k} & O & \dots & \dots & O \end{pmatrix},$$

where $\hat{E} = 2EE^H$. Thus the incomplete factorization does exist, and the matrix H has rank $2m + 3k - 6$.

7.3 Numerical experiments

We now report on several results concerning the practical implementation of the factorization techniques introduced in the previous sections. The numerical experiments have been performed on an IBM 3033 computer by using the FORTRAN 77 programming language. Newton's iterations relative to case (1) below were conducted on a Macintosh IIcx using Mathematica™.

This section is organized in four parts (1)-(4).

1. solution of the matrix equation $CX^2 - AX + B = O$;

2. solution of the nonlinear system arising to determine the LL^T factorization with minimum $\|LL^T - T\|_F$;

3. evaluation of the number of steps of the iterative methods corresponding to the incomplete LU factorization;

4. evaluation of the number of steps of the iterative methods corresponding to the incomplete LL^T factorization with minimum $\|LL^T - T\|_F$.

(1) In all the cases considered the matrix T is either unsymmetric or symmetric sign indefinite. For what concerns the starting point, we chose either

$$X_0^- = \frac{a + \sqrt{a^2 - 4b}}{2}, \qquad \text{or} \qquad X_0^+ = \frac{a + \sqrt{a^2 + 4b}}{2},$$

where $f_\perp(\mathbf{u})$ is the normal (outward) derivative, leads to the following coefficient matrix:

$$T = \begin{pmatrix} C+I & 2B & I & & & & \\ 2B & C & 2B & I & & & \\ I & 2B & C & 2B & I & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & I & 2B & C & 2B & I \\ & & & I & 2B & C & 2B \\ & & & & I & 2B & C+I \end{pmatrix}, \tag{26}$$

where $C = B^2 + 2I + 2EE^H$, $B = -A$, A is the same as for the Poisson equation, and

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

We can apply the incomplete factorization (2) to (26) by working on the block tridiagonal matrix

$$\hat{T} = \begin{pmatrix} M & \hat{L} & & & \\ \hat{L}^H & M & \hat{L} & & \\ & \ddots & \ddots & \ddots & \\ & & \hat{L}^H & M & \hat{L} \\ & & & \hat{L}^H & M \end{pmatrix},$$

where

$$\hat{L} = \begin{pmatrix} I_k & O \\ 2B & I_k \end{pmatrix}, \qquad \text{and} \qquad M = \begin{pmatrix} C & 2B \\ 2B & C \end{pmatrix}$$

are $2k \times 2k$ matrices. Observe that

$$T = \hat{T} + \begin{pmatrix} I_k & O & \ldots & O & O \\ O & O & \ldots & O & I_k \end{pmatrix}^H \begin{pmatrix} I_k & O & \ldots & O & O \\ O & O & \ldots & O & I_k \end{pmatrix}.$$

The incomplete factorization (2) will exist if and only if the matrix equation

$$\hat{L}^H X^2 - MX + \hat{L} = 0$$

has a solution. In this case the difference matrix H will be an $(m/2) \times (m/2)$ block matrix (with block size $2k \times 2k$), whose elements are all zero but the following

$$H_{11} = \begin{pmatrix} I_k & O \\ O & O \end{pmatrix} + \hat{L}^H X, \qquad \text{and} \qquad H_{\frac{m}{2}\frac{m}{2}} = \begin{pmatrix} O & O \\ O & I_k \end{pmatrix}$$

In this case H has rank $3k$.

has a solution. Note, in fact, that A and I commute and the matrix $A^2 - 4I$ being positive definite, since all the eigenvalues of A are greater than 2 (in fact $\lambda_i(A) = 4 - 2\cos(i\pi/(n+1))$, $i = 1, \ldots, n$). Then the hypotheses of the Theorem 11 apply, and the blocks are easily found in

$$\begin{aligned} X &= \frac{1}{2}A + \frac{1}{2}\left(A^2 - 4I\right)^{1/2} \\ Y &= \frac{1}{2}A - \frac{1}{2}\left(A^2 - 4I\right)^{1/2}. \end{aligned}$$

The matrix $B^{-1}A = -A$ have all the eigenvalues less then -2, hence from the results of section 6 we can see that the iterative method (12) will converge or not depending on the actual value of k. Experimental results have shown that the method convergers only if $k \leq 7$.

The matrices A, $A + 2B = A - 2I$ and $A - 2B = A + 2I$ are symmetric and nonnegative definite, hence, for Theorem 9, the factorization $T = LL^T + K$ exists. The blocks of the matrix L are:

$$\begin{aligned} Z &= \frac{1}{2}(A - 2I)^{1/2} + \frac{1}{2}(A + 2I)^{1/2}, \\ Y &= \frac{1}{2}(A - 2I)^{1/2} - \frac{1}{2}(A + 2I)^{1/2}, \end{aligned}$$

because A and $B = -I$ commute, and thus it holds $V = I$. From

$$G = Y^{-1}Z = -\frac{1}{2}A - \frac{1}{2}\left(A^2 - 4I\right)^{1/2},$$

and from the fact that the eigenvalues of A are all greater than 2, it follows that the iterative method (15) is not convergent.

7.2 Biharmonic equation

Another major problem that leads to a banded (almost) Toeplitz linear system is the solution of the biharmonic equation via difference approximation. The biharmonic equation is

$$\nabla^4 f(\mathbf{u}) = g(\mathbf{u}), \qquad \mathbf{u} \in U = \{(x, y) : 0 \leq x, y \leq 1\}.$$

The finite difference approximation over U using a mesh size of $h = 1/(1 + \sqrt{n})$ and the following conditions on the boundary of U,

$$f(\mathbf{u}) = 0, \qquad \text{and} \qquad f_{\perp}(\mathbf{u}) = 0,$$

The conditions given are certainly too restrictive to have a lot of applications that satisfy them; however, in Section 7 we will see an important special case for which the conditions does hold.

In [11] Davis reports on a procedure for solving $F(X) = AX^2 + BX + C = O$ using Newtons triangularizations. The basic step of such a procedure is accomplished by a modified version of the QZ algorithm. Experimental results are also reported which show that such a modified procedure finds a solvent of $F(X) = O$ in many cases where the QZ approach fails. The sequential cost of the method, for $k \times k$ matrices, is dominated by the quantity $9\sigma k^3$, where σ is the number of iterations for the QZ algorithm to reduce a subdiagonal element to 0. In view of applying our incomplete LU factorization, the method can be usefully adopted in case the order k of the blocks is much lesser than n.

7 Special cases and experimental results

In this section we take into account two very important special cases, namely linear systems arising from the discrete approximation by finite differences of Poisson and biharmonic equations. Then we present a number of experimental results obtained by applying Newton's method to the solution of $CX^2 - AX + B = O$ for various choices of A, B, and C. These confirm that a solution to the quadratic matrix equation problem can be found in many cases not covered by the theory.

7.1 Poisson equation

The solution of the Poisson equation is an example of a problem whose finite difference approximation gives rise to a banded Toeplitz linear system. Poisson equation is

$$\nabla^2 f(\mathbf{u}) = g(\mathbf{u}), \qquad \mathbf{u} \in U = \{(x,y) : 0 \leq x, y \leq 1\}.$$

The finite difference approximation over U using a mesh size of $h = 1/\sqrt{n}$ and the Dirichlet conditions,

$$f(\mathbf{u}) = g(\mathbf{u})$$

on the boundary of U, leads to a linear system of equations whose coefficient matrix is an $n \times n$ banded Toeplitz as (1), in which $k = \sqrt{n}$, both B and C are equal to $-I$, and A is

$$\begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix}$$

The incomplete factorization (2) can be applied to the coefficient matrix because the matrix equation

$$X^2 + AX + I = 0,$$

Theorem 11 *Let $A, B, C \in \mathbf{R}^{k\times k}$, with C nonsingular. If the matrix $C^{-1}A$ and $C^{-1}B$ are symmetric, and the matrix $(C^{-1}A)^2 - 4C^{-1}B$ is symmetric, nonnegative definite and commutes with the matrix $C^{-1}A$, then the matrix*

$$X = 1/2C^{-1}A + 1/2\left[\left(C^{-1}A\right)^2 - 4C^{-1}B\right]^{1/2}$$

is a solvent of the quadratic equation.

Proof If C is nonsingular, then, multiplying per C^{-1}, we obtain the equation

$$X^2 - C^{-1}AX + C^{-1}B = O. \tag{24}$$

Let us consider the equation

$$XX^T - C^{-1}AX^T + C^{-1}B = O. \tag{25}$$

By Theorem 7 the matrix

$$X = \frac{1}{2}C^{-1}A + \frac{1}{2}\left[\left(C^{-1}A\right)^2 - 4C^{-1}B\right]^{1/2}$$

is a solvent of the equation (25). The matrix X is symmetric, hence X is a solvent of the equation (24). ∎

In this case one of the possible factorizations and a has the following blocks

$$\begin{aligned} X &= \frac{1}{2}C^{-1}A + \frac{1}{2}\left[\left(C^{-1}A\right)^2 - 4C^{-1}B\right]^{1/2}, \\ Y &= \frac{1}{2}A - \frac{1}{2}C\left[\left(C^{-1}A\right)^2 - 4C^{-1}B\right]^{1/2}, \\ Z &= C. \end{aligned}$$

In particular, if T is block symmetric, the hypotheses of the Theorem 11 turn into $B^{-1}A$ symmetric and $(B^{-1}A)^2 - 4I$ nonnegative definite, i.e. the eigenvalues of $B^{-1}A$ must be in module great or equal to 2.

If A and B commute, then $G = Y^{-1}Z = X$ and

$$XG = X^2 = \frac{1}{2}\left(B^{-1}A\right)^2 - I + \frac{1}{2}B^{-1}A\left[\left(B^{-1}A\right)^2 - 4I\right]^{1/2}.$$

Let $\lambda \in \mathbf{R}$ be an eigenvalue of XG (the matrix XG is symmetric), then

$$\lambda = \frac{1}{2}\eta^2 - 1 + \frac{1}{2}\eta\sqrt{\eta^2 - 4},$$

where η is an eigenvalue of $B^{-1}A$. If $\eta < -3\sqrt{2}/2$, then $0 < \lambda < 1/2$ and the iterative method (12) is convergent.

(symbolic) polynomial computations (sum, product and partial derivatives). Finally, there are Lisp functions that generate Fortran code for the (numeric) computation of the produced formulas. For example, given a function $F(\mathbf{x}) = [F_1(\mathbf{x}), \ldots, F_n(\mathbf{x})]$, where $\mathbf{x} = [x_1, \ldots, x_n]$, Newton's method applied to compute a solution of $F(\mathbf{x}) = 0$ requires, at the $k-$th step, the computation of the vector $F(\mathbf{x}^{(k)})$ and of the matrix $DF(\mathbf{x}^{(k)})$ whose i, j-th element is $\partial F_i(\mathbf{x}^{(k)})/\partial x_j$. Given F, there a Lisp function NEWTON that produces just the Fortran code to compute $F(\mathbf{x}^{(k)})$ and $DF(\mathbf{x}^{(k)})$.

6.2 Existence of the LU factorization

Let us consider now the question of the existence of the LU incomplete factorization. A sufficient condition for the existence of a solvent of the quadratic equation is stated in the following proposition, which immediately leads to a solution algorithm.

Proposition 10 *Let A, B and C be real $k \times k$ matrices. If there exists a orthogonal matrix Q such that $A = QT_AQ^T$, $B = QT_BQ^T$ and $C = QT_CQ^T$ (i.e. A, B and C share the same set of eigenvectors), where T_A, T_B and T_C are upper triangular matrices, then the matrix equation $CX^2 - AX + B = O$ has a solution X^* if the triangular matrix equation $T_CY^2 - T_AY + T_B = O$ has a solution Y^*, and $X^* = QY^*Q^T$.*

Proof ¿From the hypotheses it follows that

$$\begin{aligned} CX^2 - AX + B &= QT_CQ^TX^2 - QT_AQ^TX + QT_BQ^T \\ &= Q(T_CQ^TX^2Q - T_AQ^TXQ + T_B)Q^T. \end{aligned}$$

Now, let $Y = Q^TXQ$ be an upper triangular matrix. (Note that this position corresponds to a transformation of unknowns.) It is not hard to see that the upper triangular matrix equation

$$T_CY^2 - T_AY + T_B = O$$

gives rise to a system of $k \times (k+1)/2$ equations and $k \times (k+1)/2$ unknowns that can be easily solved provided that no equation reduces to the constant=0 form. One of the possible solution path is depicted below:

$$\begin{pmatrix} \frac{(k-1)k}{2}+1 & \ldots & \ldots & \frac{k(k+1)}{2} \\ \ddots & & & \vdots \\ & 4 & 5 & 6 \\ & & 2 & 3 \\ & & & 1 \end{pmatrix}$$

where the $[(k-i)(k-i+1)/2 + j - i + 1]$-th equation is solved with respect to y_{ij}, $j \geq i$. ∎

In the following we give another sufficient condition for the existence of a solvent of the quadratic matrix equation, together with the explicit solution.

Note that if there exists always a matrix V such that MVN is symmetric, then the nonnegative definiteness of $A \pm 2B$ is a necessary and sufficient condition for the existence of the factorization (7).

If the hypotheses of the Corollary 9 are verified and A and B commute, we have $V = I$ and

$$G = Y^{-1}Z = \frac{1}{2}B^{-1}\left[A + \left(A^2 - 4B\right)^{1/2}\right].$$

Let λ be an eigenvalue of $B^{-1}A$. If B is nonnegative definite, then

$$\lambda(G) = \frac{1}{2}\lambda + \frac{1}{2}\sqrt{\lambda^2 - 4},$$

and the iterative method (15) is convergent, if $\lambda \leq -3\sqrt{2}/2$.

Consider now the problem of actually computing the matrix L which minimizes the functional $\|LL^T - T\|_F$. We can look for a minimum of the convex functional $f(L) = \|LL^T - T\|_F^2$, which is equivalent to the previous one. The minimum can be obtained by equating to zero the partial derivatives of $f(L)$. In the following, we will first consider the case in which $k = 1$ (e.g. we assume L to be element bidiagonal, and T to be symmetric tridiagonal, with i,i-th entry denoted by a and $i, i+1$-th entry by b). Then we will deal with the general case.

Let x, and y be complex numbers. Then the minimum of $f(L)$ can be computed by solving the system of two nonlinear equations

$$\left\{ \begin{array}{r} nx^3 - nax + 2(n-1)xy^2 - (n-1)by = 0 \\ y^3 - ay + 2x^2y - bx = 0 \end{array} \right. .$$

The solution of such system can be computed by Newton's method, and as starting points we suggest the values $x_0 = (\sqrt{a+2b} + \sqrt{a-2b})/2$, and $y_0 = (\sqrt{a+2b} - \sqrt{a-2b})/2$, which are the solution of the system

$$\left\{ \begin{array}{r} x^2 + y^2 = a \\ xy = b \end{array} \right. ,$$

i.e. x_0 and y_0 give rise to an incomplete LL^T factorization themselves.

The block extension of this approach leads to a system of $2k^2$ nonlinear equations. Though not difficult, the process of obtaining such equations and developing code for its solution is time consuming. Thus, we have implemented a tool for the symbolic computation of these equations and the automatic generation of Fortran code for their numeric evaluation. The complete code listing is available with the Technical Report [7]. A brief description follows.

The tool consists of Lisp functions to symbolically perform simple matrix operations and functions, such as matrix sum, row by column product and the square of the Frobenius norm of a matrix. Such computations require the ability to handle multivariate polynomials; hence the most basic functions of the package are related to

Corollary 8 *Let T be the matrix (1) with B symmetric. If the factorization (7) exists then:*

1. *A is symmetric and nonnegative definite,*

2. *$C = B$,*

3. *$A + 2B$ and $A - 2B$ are symmetric and nonnegative definite.*

The blocks Y and Z are of the form

$$\begin{aligned} Z &= \frac{1}{2}\left[(A-2B)^{1/2}V + (A+2B)^{1/2}\right], \\ Y &= \frac{1}{2}\left[(A+2B)^{1/2} - (A-2B)^{1/2}V\right], \end{aligned}$$

where V is an orthogonal matrix. ■

Corollary 9 *Let $T \in \mathbf{R}^{n\times n}$ be a symmetric and block symmetric tridiagonal Toeplitz matrix*

$$T = \begin{pmatrix} A & B & & & \\ B & A & B & & \\ & \ddots & \ddots & \ddots & \\ & & B & A & B \\ & & & B & A \end{pmatrix} \qquad (23)$$

with A symmetric and nonnegative definite and B symmetric. If $A+2B$ and $A-2B$ are symmetric and nonnegative definite and one of the following conditions holds:

1. *$A + 2B$ and $A - 2B$ share the same set of eigenvectors,*

2. *$\mathrm{rank}(F) = \max\{\mathrm{rank}(A+2B), \mathrm{rank}(A-2B)\}$, where F is the $n \times 2n$ matrix given by*

$$F = (A + 2B | A - 2B),$$

the factorization (7) of the matrix (23) exists, and it exists an orthogonal matrix V such that the matrices

$$\begin{aligned} Z &= \frac{1}{2}\left[(A-2B)^{1/2}V + (A+2B)^{1/2}\right], \\ Y &= \frac{1}{2}\left[(A+2B)^{1/2} - (A-2B)^{1/2}V\right], \end{aligned}$$

are the blocks of the factorization (7). ■

Lemma 6 *Let $M, N \in \mathbf{R}^{n \times n}$. If one of the following conditions hold:*

1. *the left singular vectors of M are equal to the right singular vectors of N,*
2. $\operatorname{rank}(W) = \max\{\operatorname{rank}(M), \operatorname{rank}(N)\}$, *where $W \in \mathbf{R}^{n \times 2n}$, and $W = (M \ N^T)$,*

then there exists an orthogonal matrix such that MVN is symmetric, and the system

$$\begin{cases} MVN - N^T V^T M^T = & O \\ V^T V = & I \end{cases} \tag{21}$$

has a solution.

Proof Let $M = R_1 \Sigma_1 S_1^T$ and $N = R_2 \Sigma_2 S_2^T$ the SVD of M and N. Then using the substitution $Q = S_1^T V R_2$, we obtain

$$\begin{cases} R_1 \Sigma_1 Q \Sigma_2 S_2^T - S_2 \Sigma_2 Q^T \Sigma_1 R_1^T = & O \\ Q^T Q = & I \end{cases} . \tag{22}$$

If the condition (1) holds, the $R_1 = S_2$, and $Q = I$ is a solution of (22).

If the condition 2 holds and $\operatorname{rank}(M) = \operatorname{rank}(W)$, then there exists a solution Y of $MY = N^T$. Let $Y = \tilde{R}\tilde{\Sigma}\tilde{S}^T$ be the SVD of Y, thus $V = \tilde{R}\tilde{S}^T$ is an orthogonal matrix and satisfy the system (21). If condition 2 holds and $\operatorname{rank}(N^T) = \operatorname{rank}(W)$, we show the existence of V in the same way. ∎

Note that if one of the matrix M and N is nonsingular then the condition 2 holds. In particular, if M and N are symmetric and normal, then $V = I$ is a solution of the system (21). Moreover, a number of experimental results, leads to conjecture that the system (21) has always at least a solution V.

From Lemma 6 it descends the following theorem.

Theorem 7 *Let G be a symmetric matrix. If $FF^T - 4G$ is symmetric and nonnegative definite and one of the following conditions hold:*

1. *the left singular vectors of F are equal to the eigenvectors of $FF^T - 4G$,*
2. $\operatorname{rank}(W) = \max\{\operatorname{rank}(F), \operatorname{rank}(FF^T - 4G)\}$, *where $W = (F \ FF^T - 4G)$,*

then there exists an orthogonal matrix V such that

$$Z = \frac{1}{2}\left[\left(FF^T - 4G\right)^{1/2} V - F\right]$$

is a solvent of the equation (18). ∎

We summarize the above results with the following Corollaries.

6 On the matrix equations related to the existence of the factorizations

We state here some formal propositions about the existence of the incomplete LU and LL^T factorizations. No satisfying result is known about even the mere existence of a solution for the general case of quadratic matrix equations

$$CX^2 - AX + B = O, \tag{17}$$

and for the matrix equations

$$ZZ^T + FZ^T + G = O. \tag{18}$$

Some iterative algorithms for the solution of (17) and (18), have been proposed (see [11, 19]), which are all based on Newton's method. Let us analyze in more details the equations (17) and (18).

6.1 Existence of the LL^T factorizations

We have analyzed the equation (18) when G is a symmetric matrix, and we state a necessary (sufficient) condition for the existence of a solvent in Theorem 5 (Theorem 7).

Theorem 5 *Let G be an $n \times n$ symmetric matrix. If Z is a solution of (18) then $FF^T - 4G$ is symmetric and nonnegative definite, and X is of the form*

$$Z = \frac{1}{2}\left[\left(FF^T - 4G\right)^{1/2} V - F\right], \tag{19}$$

where V is an orthogonal matrix.

Proof If G is symmetric and Z is a solution of (18), then $FZ^T = -ZZ^T - G = ZF^T$. Then (18) can be rewritten as

$$\left(Z + \frac{1}{2}F\right)\left(Z + \frac{1}{2}F\right)^T = \frac{1}{4}\left(FF^T - 4G\right) \tag{20}$$

and using Lemma 2, we have that $FF^T - 4G$ is symmetric and nonnegative definite and $Z = \frac{1}{2}\left[\left(FF^T - 4G\right)^{1/2} V - F\right]$. ∎

On the other hand, assume that $FF^T - 4G$ is symmetric and nonnegative definite. If Z is of the form (19) then Z satisfy $FZ^T = ZF^T$. Thus, using (20), we have that Z is a solvent of (18). We want to show that it exists an orthogonal matrix V so that $Z = \frac{1}{2}\left[\left(FF^T - 4G\right)^{1/2} V - F\right]$ satisfy $FZ^T = ZF^T$, i.e. that $FV^T(FF^T - 4G)^{1/2}$ is symmetric.

The following lemma studies the existence of the matrix V and gives a method to find it when the matrices F and $(FF^T - 4G)^{1/2}$ satisfy some particular conditions.

ple

$$
\begin{array}{rrcll}
(1) & X_{i+1} & = & X_i(-TX_i+2I), & i=0,1,2,\ldots \\
 & X_0 & = & U^{-1}L^{-1}; & \\
 & & & & \\
(2) & X_{i+1} & = & 2X_i - X_iTX_i, & i=0,1,2,\ldots \\
 & X_0 & = & U^{-1}L^{-1}; & \\
 & & & & \\
(3) & X_{i+1} & = & X_i(R_i+I), & i=0,1,2,\ldots \\
 & R_{i+1} & = & R_iR_i, & i=0,1,2,\ldots \\
 & X_0 & = & I, & \\
 & R_0 & = & -U^{-1}L^{-1}H. &
\end{array}
$$

The third formulation is well suited to be implemented in a parallel environment (see [8, 20]). The computational cost of this approach will be analyzed next. Note that all these schemes have quadratic convergence [15]. Analogous results could be obtained for iterative methods derived from the incomplete LL^T factorization, which is even more suitable for the application of iterative methods. Experimental results on this will be reported in section 7.

We now analyze the cost of the classical iterative method (12) derived from the incomplete factorization (2). The computation of the iteration matrix (13) for the sequential case is different from that for the parallel case. While in parallel we can efficiently compute matrix powers and products, in sequential it can be better to solve triangular (matrix) equations. Thus, our sequential method is as follows:

1. Solve $LV = -H$ and $UW = V$. Observing that only the first k columns of V and W are nonzero, this can be done with $O(mk^3)$ arithmetic operations (see steps i and ii of the direct sequential algorithm).

2. Solve $L\mathbf{y} = \mathbf{b}$ and then $U\mathbf{x} = \mathbf{y}$. This costs $O(mk^2)$ time. (The method is $\mathbf{w}_{i+1} = W\mathbf{w}_i + \mathbf{x}$.)

The cost of sequentially setting up the method is then dominated by step 1) and therefore is $O(mk^3)$. In parallel we can take advantage of the knowledge of (the form of) the iteration matrix, and it turns out that its computation can be done in $PT(mk^\alpha/\log k, (2\log k)(\log m))$.

For what concerns the execution of the method, it is known that the number of steps required by an iterative method to compute the solution of a linear system $\mathbf{x} = P\mathbf{x}+\mathbf{q}$ with relative analytic error of $O(2^{-t})$ is $O(-t/\log\rho(P))$. Each step requires an $n\times n$ matrix by vector multiplication and the sum of two n-vectors. In sequential this costs $O(n^2)$ arithmetic operations, and the overall execution costs $O(n^2(-t/\log\rho(I-U^{-1}L^{-1}T)))$. In parallel, each step can be carried out in $PT(mk^2/\log k, 2\log k)$, and the whole circuit has $PT(mk^\alpha/\log k, \max\{\log m, (-t/\log\rho(I-U^{-1}L^{-1}T))\}2\log k)$.

after having used $U^{-1}L^{-1}$ as a preconditioning matrix, conjugate gradient methods are supposed to work very quickly (about k iterations) when applied to solve systems with $U^{-1}L^{-1}T$ as a coefficient matrix. Furthermore, note that $(U^{-1}L^{-1}T)^{-1}$ is given by

$$(U^{-1}L^{-1}T)^{-1} = \begin{pmatrix} (I + M_m GX)^{-1} & & & & \\ -M_{m-1}G^2X(I + M_m GX)^{-1} & I & & & \\ M_{m-2}G^3X(I + M_m GX)^{-1} & 0 & I & & \\ \vdots & \vdots & \ddots & \ddots & \\ (-1)^{m-1}G^m X(I + M_m GX)^{-1} & 0 & \dots & 0 & I \end{pmatrix},$$

so that an upper bound to the condition number $\mu(U^{-1}L^{-1}T)$ can be easily computed, with respect either to the spectral norm or the max norm. The evaluation of an upper bound to the condition number with respect to the max norm is pretty easy; for what concerns the spectral norm, one can proceed as follows:

1. $U^{-1}L^{-1}T$ can be written as $U^{-1}L^{-1}T = I + VE_1^T$, where V is an $n \times k$ matrix.

2. The spectral norm of $U^{-1}L^{-1}T$ is given by

   $$\|U^{-1}L^{-1}T\|_2 = \rho((U^{-1}L^{-1}T)^T U^{-1}L^{-1}T) = \rho((I + E_1V^T)(I + VE_1^T)),$$

   from which it readily follows that the knowledge of the eigenvalues of the $k \times k$ matrix $E_1V^TVE_1^T$ suffices to determine $\|U^{-1}L^{-1}T\|_2$.

3. Use the previous procedure to evaluate $\|(U^{-1}L^{-1}T)^{-1}\|_2$, and therefore

   $$\mu_2(U^{-1}L^{-1}T) = \|U^{-1}L^{-1}T\|_2 \cdot \|(U^{-1}L^{-1}T)^{-1}\|_2.$$

It is worth pointing out that $\mu_2(U^{-1}L^{-1}T)$ can be evaluated in terms of operations involving only a $k \times k$ matrix.

(iii) Consider now the matrix equation $TX - I = O$, which has solvent $X = T^{-1}$. This equation can be solved by using Newton's method [8, 20], which leads to the iterative expression

$$\begin{aligned} X_{i+1} &= X_i - X_i(TX_i - I), \quad i = 0, 1, 2, \dots \\ X_0 &= U^{-1}L^{-1}. \end{aligned}$$

The above iterations can be carried out by means of different formulations, for examis an eigenvalue of $M_m GX$. If XG is symmetric, then $\lambda \in \mathbf{R}$, and the iterative method (12) is convergent if the following constraints hold:

$$-1 < \frac{1-\lambda^{m+1}}{1-\lambda} - 1 < 1. \tag{14}$$

The inequality (14) is satisfied for $\alpha_m \leq -1 < \lambda < 1/2 \leq \beta_m$, where $\lim_{m\to\infty} \alpha_m = -1$, and $\lim_{m\to\infty} \beta_m = 1/2$. Due to the vanishing term λ^{m+1}, experimental results have shown that μ can be considered a constant even for small values of m. Hence, in practice, the convergence of the iterative method (12) is quite independent from m, the number of the blocks of the matrix T.

The iterative method corresponding to the LL^T factorization has the form

$$\begin{cases} \mathbf{w}_0 & = \mathbf{q} \\ \mathbf{w}_{i+1} & = (I - L^{-T}L^{-1}T)\mathbf{w}_i + L^{-T}L^{-1}\mathbf{b}, \end{cases} \tag{15}$$

where $\mathbf{q}$ is an arbitrary n-vector. The iterative method (15) converge if and only if

$$\rho\left(I - L^{-T}L^{-1}T\right) = \rho\left(Y^{-T}N_m GZ^T\right) < 1.$$

If Y, Z e B are symmetric, then $C = B = YZ = ZY$ (i.e. Y and Z commute) and G is symmetric. Thus, we have

$$\rho\left(I - L^{-T}L^{-1}T\right) = \rho\left(\sum_{q=1}^{m} G^{2q}\right) = \sum_{q=1}^{m} (\lambda_{max})^{2q}, \tag{16}$$

where λ_{max} is the eigenvalue of $G = Y^{-1}Z$ with maximum module ($\rho(G) = |\lambda_{max}|$). From (16), it follows that if $|\lambda_{max}| < 1/\sqrt{2}$, the iterative method (15) converges.

(ii) Now we turn our attention to the possible use of the incomplete factorization (2) as a preconditioning for the conjugate gradient method. It is well known that such a technique results in a faster algorithm as long as the preconditioned coefficient matrix is "close" to the identity matrix. If we multiply both sides of the equation $T\mathbf{w} = \mathbf{b}$ by $U^{-1}L^{-1}$, we get the system $U^{-1}L^{-1}T\mathbf{w} = U^{-1}L^{-1}\mathbf{b}$, and our interest is in studying how close is the matrix $U^{-1}L^{-1}T$ to the identity matrix. It turns out immediately that

$$U^{-1}L^{-1}T = \begin{pmatrix} I + M_m GX & & & & \\ -M_{m-1}G^2X & I & & & \\ M_{m-2}G^3X & 0 & I & & \\ \vdots & \vdots & \ddots & \ddots & \\ (-1)^{m-1}G^mX & 0 & \dots & 0 & I \end{pmatrix}.$$

First of all, note that only k eigenvalues of $U^{-1}L^{-1}T$ are different from 1, and that the minimum polynomial of $U^{-1}L^{-1}T$ has degree not greater than k. It follows that,

```
ALGORITHM DP
Step 1  Invert the matrix Y in PT(2k^{α+1/2}/log k, (7/2) log^2 k)
Step 2  Compute G = Y^{-1}Z in PT(k^α/log k, 2 log k)
Step 3  Compute the powers X^i, G^i, i = 2,...,m-1,
          in PT(mk^α/log k, (2 log k)(log m))
Step 4  Compute the products R_j = G^j Y^{-1}, j = 0...,m-1,
          in PT(mk^α/log k, 2 log k)
Step 5  Compute the matrix products P_ij = X^i R_j, i,j = 0,...,m-1,
          in PT(m^2 k^α/log k, 2 log k)
Step 6  Compute the matrix V = U^{-1}L^{-1} using the products P_ij
          according to (5) in PT(n^2 m/log m, 2 log m)
Step 7  Compute W = V E_1 Z and x = Vb in PT(n^2/log n, 2 log n)
Step 8  Compute z = X E_1^T x and S = I + X E_1^T W in PT(k^α/log k, 2 log k)
Step 9  Invert the matrix S in PT(2k^{α+1/2}/log k, (7/2) log^2 k)
Step 10 Compute s = S^{-1}z in PT(k^2/log k, 2 log k)
Step 11 Compute w = x - Ws in PT(mk^2/log k, 2 log k)
```

be the incomplete factorization (2), and consider the linear system

$$T\mathbf{w} = \mathbf{b}.$$

By using (2) as a splitting of T, one can get the equivalent system

$$\mathbf{w} = (I - U^{-1}L^{-1}T)\mathbf{w} + U^{-1}L^{-1}\mathbf{b},$$

which naturally leads to the iterative method

$$\left\{ \begin{array}{ll} \mathbf{w}_0 & = \mathbf{q} \\ \mathbf{w}_{i+1} & = (I - U^{-1}L^{-1}T)\mathbf{w}_i + U^{-1}L^{-1}\mathbf{b}, \end{array} \right. \tag{12}$$

where $\mathbf{q}$ is an arbitrary n-vector. It is a well known fact that the above iterative method is convergent, provided that $\rho(I - U^{-1}L^{-1}T) < 1$.

Now, recalling that the i,j-th block element V_{ij} of the matrix $U^{-1}L^{-1}$ can be written according to the formula (5), we have

$$I - U^{-1}L^{-1}T = -\begin{pmatrix} M_m GX & 0 & \dots & 0 \\ -M_{m-1}G^2X & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ (-1)^{m-1}G^mX & 0 & \dots & 0 \end{pmatrix}. \tag{13}$$

Thus, the iterative method will converge if the spectral radius of the $k \times k$ matrix M_mGX is less than one. Suppose that G and X commute (e.g. if T is block symmetric). Let λ be an eigenvalue of XG, then

$$\mu = \sum_{q=1}^{m} (\lambda)^q = \frac{1 - \lambda^{m+1}}{1 - \lambda} - 1$$

```
ALGORITHM DS2
Step 1  As in DS1
Step 2  As in DS1
Step 3  S = I + X E_1^T U^{-1} L^{-1} E_1 Z = sum_{i=0}^{m} X^i G^i
        3.1  Solve YG = Z
        3.2  S <- I
        3.3  For i <- 1 to m do
                S <- XSG + I
Step 4  As in DS1
Step 5  w <- x - U^{-1} L^{-1} E_1 Zs, where U^{-1} L^{-1} E_1 Z = [W_1, ..., W_m]^T
        5.1  Compute L_x U_x = X
        5.2  Solve L_x U_x W_1 = S - I
        5.3  w_1 <- x_1 - W_1 s
        5.4  For i <- 2 to m do
        5.5     Solve L_x U_x W_i = (-1)^i G^{i-1} - W_{i-1}
                5.5.1  Q <- QG  (Remark: Q is initially set to I)
                5.5.2  T <- (-1)^i Q - W_{i-1}
                5.5.3  L_x U_x W_i <- T
        5.6     w_i <- x_i - W_i s
```

can be implemented by a PRAM with $\max\{4k^{1/2+\alpha}/\log k, m^2k^\alpha/\log k, n^2m/\log m\}$ processors with running time bounded by $\max\{7\log k, 2\log m\}\log k)$. Depending upon the actual value of k such algorithm can be significantly better than the general $O(\log^2 n)$ time algorithm for matrix inversion. We finally observe that Toeplitz matrix by vector products could also be computed by FFT circuits. However this would not improve the asymptotic performance of the algorithm.

5 Iterative methods derived form the incomplete factorizations

In this section we analyze the suitability of the incomplete factorizations (2) and (7) to generate rapidly convergent iterative methods. We consider three possibilities. (i) Use the incomplete factorization as a splitting of T to generate a classical iterative method $\mathbf{w}_{i+1} = P\mathbf{w}_i + \mathbf{q}$; in this case we are interested in studying the spectral radius of the iteration matrix P. (ii) Use the matrix $U^{-1}L^{-1}$ as a preconditioner to obtain a system which is well suited to the application of conjugate gradient methods; in this case our interest is in the evaluation of the condition number of the preconditioned coefficient matrix. (iii) Use the matrix $U^{-1}L^{-1}$ as a starting point for the application of Newton's method to the inversion of T; in this case our interest is in the evaluation of the difference between T^{-1} and $U^{-1}L^{-1}$.

(i) As in section 3, let

$$T = LU + H, \qquad H = E_1 Z X E_1^T$$

```
ALGORITHM DS1
Step 1  x ← U^{-1}L^{-1}b
        1.1  Solve Ly = b,   y = [y_1,...,y_m]^T,  b = [b_1,...,b_m]^T
             1.1.1  Compute L_yU_y = Y
             1.1.2  Solve L_yU_yy_1 = b_1
             1.1.3  For i ← 2 to m do
                      Solve L_yU_yy_i = b_i − Zy_{i−1}
        1.2  Solve Ux = y,   x = [x_1,...,x_m]^T
             1.2.1  x_m ← y_m
             1.2.2  For i ← m − 1 downto 1 do
                      x_i ← y_i − Xx_{i+1}
Step 2  z ← XE_1^T x(= Xx_1)
Step 3  Compute S = I + XE_1^T U^{-1}L^{-1}E_1Z = Σ_{i=0}^{m} X^iG^i
          and [W_1,...,W_m]^T = U^{-1}L^{-1}E_1Z
        3.1  Solve YG = Z
        3.2  Compute and store the matrix powers G^2,...,G^m
        3.3  W_m ← (−1)^{m−1}G^m
        3.4  For i ← m − 1 downto 1 do
                W_i ← (−1)^{i−1}G^i − XW_{i+1}
        3.5  S ← I + XW_1
Step 4  s ← S^{-1}z
        4.1  Compute L_sU_s = S
        4.2  Solve L_st = z
        4.3  Solve U_ss = t
Step 5  For i ← 1 to m do
          w_i ← x_i − W_is
```

It is not difficult to see that the cost of the above algorithm is dominated by step 3, which requires approximately $2mk^3$ arithmetic operations and mk^2 storage (note that the storage used for the G^is can be overwritten with the W^is). In step 3 of algorithm DS1 we used the recurrence $W_i \leftarrow (-1)^{i-1}G^i - XW_{i+1}$ ($W_m \leftarrow (-1)^{m-1}G^m$) and the equality $S = I + XW_1$; if we first compute $S = \sum_i X^iG^i$ and then $Wi = (-1)^iG^{i-1} - W_{i-1}$, then we succeed in reducing the amount of temporary storage to $n + O(k^2)$. This is done in algorithm DS2. It is interesting to note that, besides the storage for X, Y, Z and $\mathbf{b}$, algorithm DS2 only uses small additional work space, i.e. $O(k^2)$. In fact, the storage for W_{i+1} can be the same as for W_i. Of course the total work space is $n + O(k^2)$. On the other hand, the time cost of DS2 is approximately $4mk^3$ since both step 3 and step 5 requires roughly $2mk^3$ arithmetic operations.

Consider now the parallel implementation of factorization (2). Recall that a common feature of fast parallel algorithms for matrix inversion is the need of high matrix powers, whose computation dominates the overall parallel cost. By using formula (4) we can take advantage of the knowledge of the form of L^{-1} and U^{-1}. Even if the explicit computation of L^{-1} and U^{-1} still requires powering, the matrix involved are only of order k. Thus an $O((\log n)(\log k))$ time bounded PRAM algorithm seems within reach. The algorithm DP does in fact achieve such a performance. Algorithm DP shown that the Sherman-Morrison-Woodbury formula applied to a splitting of the type $A = B - C$ can be implemented by a numerically stable algorithm, provided that A and B are well- conditioned. The analysis to be performed here starts assuming T (the original band Toeplitz matrix) to be well-conditioned, and gives conditions for LU being well-conditioned. Following Yip, one can then easily see that under these conditions our factorizations lead to stable algorithms.

¿From the fact that $LU = T - E_1 ZXE_1^T$, it follows that

$$\|LU\| \leq \|T\| + \|CX\| = \|T\|(1 + \frac{\|CX\|}{\|T\|}).$$

Moreover, $(LU)^{-1}$ can be written as $T^{-1} - T^{-1}E_1RE_1^TT^{-1}$, where the expression for the $k \times k$ matrix R can be derived from section 3.1, so that

$$\|(LU)^{-1}\| \leq \|T^{-1}\| + \|T^{-1}\|^2\|R\| = \|T^{-1}\|(1 + \|T^{-1}\|\|R\|).$$

We have then

$$\text{cond}(LU) \leq \text{cond}(T)(1 + \frac{\|CX\|}{\|T\|})(1 + \|T^{-1}\|\|R\|),$$

which can be easily evaluated only in terms of the $k \times k$ matrices X and R.

4 Algorithms and computational cost

In this section we restrict ourselves to the evaluation of the performance of factorization (2). Similar results could be derived for the incomplete LL^T factorization as well. We present both sequential and parallel algorithms for the computation of $T^{-1}\mathbf{b}$ by means of (4). In [22] Yip analyzes one sequential implementation of (4) that is suitable for the general case, i.e. when $T - LU$ is a generic rank k matrix; the cost of his method is $O(kn^2)$. By exploiting the block Toeplitz structure of T, L and U, our first sequential implementation (algorithm DS1) attains a performance of $O(nk^2)$ arithmetic operations and $O(nk)$ storage. Alternatively (in algorithm DS2) we can reduce the storage usage to $O(n + k^2)$ at the price of twice as much time. In both cases the hidden constants are small.

In a parallel computation setting we can take further advantage of the knowledge of L^{-1} and U^{-1}. We present an asymptotically fast algorithm for solving $T^{-1}\mathbf{b}$ which is either in NC_F^2 or NC_F^1 depending on whether k is a constant value or grows with the size n of the problem, and F is a field with characteristic zero.

Let us first consider the sequential algorithms. It can be easily verified that the algorithm DS1 produce, in absence of roundoff errors, the exact solution of the original system $T\mathbf{w} = \mathbf{b}$. (Note: if a step has substeps, then the latter are implementation of the former.)

where

$$S = I + Z^T E_1^T L^{-T} L^{-1} E_1 Z = \sum_{i=0}^{m} \left(G^T\right)^i G^i.$$

The i-th vector component $\mathbf{w}_i$, $i = 1, \ldots, m$, of $\mathbf{w} = T^{-1}\mathbf{b}$ is then

$$\mathbf{w}_i = \left[T^{-1}\mathbf{b}\right]_i = \sum_{j=1}^{m} \left(V_{ij} - V_{i1} Z \left(\sum_{r=0}^{m} \left(G^T\right)^r G^r \right)^{-1} Z^T V_{1j} \right) \mathbf{b}_j.$$

Even in this case we have found an explicit expression of the solution of the original system in terms of Y, Z and $\mathbf{b}$.

We consider here also another interesting factorization. We look for a matrix L of the form

$$L = \begin{pmatrix} Y & & & \\ Z & Y & & \\ & \ddots & \ddots & \\ & & Z & Y \end{pmatrix},$$

where Y and Z are $k \times k$ matrices, and such that its entries are computed in order to minimize the Frobenius norm of the matrix $LL^T - T$, i.e. we seek a matrix L for which

$$\|LL^T - T\|_F$$

attains its minimum.

Proposition 4 *The LL^T incomplete factorization of the block tridiagonal Toeplitz matrix (1) is possible since the functional $\|LL^T - T\|_F$ has a minimum.*

■

We will deal with the problem of actually finding a minimum for the above described functional in section 6.

Let now consider the problem of solving $T\mathbf{w} = \mathbf{b}$ using a direct method. Let $T = LL^T + K$ be the incomplete LL^T factorization. We get

$$T^{-1} = (LL^T + K)^{-1} = [LL^T(I - L^{-T}L^{-1}K)]^{-1} = (I - L^{-T}L^{-1}K)^{-1}L^{-T}L^{-1},$$

provided that no eigenvalue of $L^{-T}L^{-1}K$ is equal to 1. Note that this situation is not likely to occur because of the choice of K.

3.3 Numerical stability of the Sherman-Morrison-Woodbury formula

In this section, we briefly look at numerical stability issues. We restrict ourselves to analyze the application of the Sherman- Morrison-Woodbury formula to (2), i.e. to the splitting $T = LU + H$. Analogous results can be obtained for the Cholesky incomplete factorization. We follow the approach presented in [22] by Yip. He has

Lemma 2 *Let A be an $n \times n$ nonnegative definite symmetric matrix with normal Schur decomposition $A = UDU^T$. Then the $n \times n$ matrix X satisfies $XX^T = A$ if and only if it holds $X = A^{1/2}UV^T$, where V is an orthogonal matrix.* ∎

Theorem 3 *Let T be the matrix (1). The factorization $T = LL^T + K$ exists if and only if:*

1. *A is symmetric and nonnegative definite,*
2. *$C = B^T$,*
3. *$A + B + B^T$ is symmetric and nonnegative definite,*

and the equation

$$ZZ^T - \left(A + B + B^T\right)^{1/2} Z^T + B = O \tag{9}$$

has a solution. In particular, if Z is solution of (9), then it holds $Y = -Z + (A + B + B^T)^{1/2}$.

Proof Properties 1, 2, and 3 follow immediately from the constraints (8). Then, by Lemma 2, the equation $(Y + Z)(Y + Z)^T = A + B + B^T$ becomes $Y + Z = (A + B + B^T)^{1/2}$, from which we obtain (9). ∎

In section 6 we will study the equation (9) in its more general form, $ZZ^T + FZ^T + G = O$. Assuming that a solution Z of (9) has been found, we use the Sherman-Morrison-Woodbury updating formula to obtain an explicit representation of T^{-1}. Repeating the same method used for the LU factorization, we obtain the following analogous result

$$\begin{aligned} T^{-1} &= (LL^T + E_1 ZZ^T E_1^T)^{-1} \\ &= L^{-T}L^{-1} - L^{-T}L^{-1}E_1 Z(I + Z^T E_1^T L^{-T} L^{-1} E_1 Z)^{-1} Z^T E_1^T L^{-T} L^{-1}. \end{aligned} \tag{10}$$

The i,j-th block element V_{ij} of the matrix $V = L^{-T}L^{-1}$ can then be written as

$$V_{ij} = (-1)^{i-j} Y^{-T} \left(G^T\right)^{j\dot{-}i} N_{m-\max(i,j)+1} G^{i\dot{-}j} Y^{-1}, \tag{11}$$

where

$$N_r = \sum_{q=0}^{r-1} \left(G^T\right)^q G^q,$$

and $G = Y^{-1}Z$. Using (11) into (10) leads to the following expression of the block element $(T^{-1})_{ij}$ of the matrix T^{-1}

$$(T^{-1})_{ij} = V_{ij} - V_{i1} Z S^{-1} Z^T V_{1j}$$

The i,j-th block element V_{ij} of the matrix $V = U^{-1}L^{-1}$ can then be written as

$$V_{ij} = (-1)^{i-j} X^{j \dot{-} i} M_{m-\max(i,j)+1} G^{i \dot{-} j} Y^{-1}, \tag{5}$$

where

$$M_r = \sum_{q=0}^{r-1} X^q G^q, \qquad \text{and} \qquad i \dot{-} j = \begin{cases} i-j & \text{if } i > j \\ 0 & \text{otherwise.} \end{cases}$$

Using (5) into (4) leads to the following expression of the block element $(T^{-1})_{ij}$ of the matrix T^{-1}:

$$(T^{-1})_{ij} = V_{ij} - V_{i1} Z S^{-1} X V_{1j}$$

with $S = (I + X E_1^T U^{-1} L^{-1} E_1 Z) = \sum_{i=0}^{m} X^i G^i$. The i-th vector component $\mathbf{w}_i$, $i = 1, \ldots, m$, of $\mathbf{w} = T^{-1}\mathbf{b}$ is then

$$\mathbf{w}_i = \left[T^{-1}\mathbf{b}\right]_i = \sum_{j=1}^{m} \left(V_{ij} - V_{i1} Z \left(\sum_{r=0}^{m} X^r G^r \right)^{-1} X V_{1j} \right) \mathbf{b}_j. \tag{6}$$

Formula (6) is an explicit expression of the solution of the original system in terms of X, Y, Z and $\mathbf{b}$.

In section 4 we will make use of (4), (5) and (6) to formulate both sequential and parallel algorithms for the solution of $T\mathbf{w} = \mathbf{b}$.

3.2 Incomplete LL^T factorizations

The approach of section 3.1 could also be applied to an incomplete Cholesky factorization. We look for the matrices L e K such that:

$$\begin{aligned} T &= \begin{pmatrix} Y & & & \\ Z & Y & & \\ & \ddots & \ddots & \\ & & Z & Y \end{pmatrix} \begin{pmatrix} Y^T & Z^T & & \\ & \ddots & \ddots & \\ & & Y^T & Z^T \\ & & & Y^T \end{pmatrix} + E_1 Z Z^T E_1^T \\ &= LL^T + K, \end{aligned} \tag{7}$$

where Y e Z are $k \times k$ matrices, $E_1 = [I, O, \ldots, O]^T$, and $I = I_k$. This holds if and only if the following constraints are satisfied:

$$\begin{cases} (Y+Z)(Y+Z)^T &= A + B + C \\ YZ^T &= B \\ ZY^T &= C. \end{cases} \tag{8}$$

¿From the following Lemma 2 and Theorem 3, it descends that also the existence of the incomplete factorization $T = LL^T + K$ is related to the existence of a solution of a particular matrix equation.

In turn, this means that the quadratic matrix equation

$$CX^2 - AX + B = O \tag{3}$$

must have a solution.

The following Theorem holds.

Theorem 1 *The factorization (2) exists if and only if the matrix equation $CX^2 - AX + B = O$ has a solution.* ∎

We will deal with the question of the existence of the LU incomplete factorization of nonsymmetric Toeplitz matrices in section 6, where we will give sufficient conditions for the existence of a solvent for (3), and in section 7, where we will present experimental results, which make it evident that the incomplete factorization (2) exists in many cases not covered by the theory.

Let assume that a solution X to (3) has been found and consider the incomplete factorization $T = LU + H$, $H = E_1 ZXE_1^T$. If the $k \times k$ matrix $I + XE_1^T U^{-1}L^{-1}E_1 Z$ is nonsingular, then by the Sherman-Morrison-Woodbury updating formula [15, 16, 22] we obtain an explicit representation of T^{-1}, namely

$$\begin{aligned} T^{-1} &= (LU + E_1 ZXE_1^T)^{-1} \\ &= U^{-1}L^{-1} - U^{-1}L^{-1}E_1 Z(I + XE_1^T U^{-1}L^{-1}E_1 Z)^{-1}XE_1^T U^{-1}L^{-1}. \end{aligned} \tag{4}$$

The fact that the matrix H has low rank (i.e. $\text{rank}(H) \leq k \ll n$), and that L and U are still block Toeplitz will allow us to obtain good performance algorithms for the solution of $T\mathbf{w} = \mathbf{b}$ by the use of (4). It is easy to see that, since

$$L = \begin{pmatrix} Y & & & \\ & Y & & \\ & & \ddots & \\ & & & Y \end{pmatrix} \begin{pmatrix} I & & & \\ Y^{-1}Z & I & & \\ & \ddots & \ddots & \\ & & Y^{-1}Z & I \end{pmatrix},$$

then it holds

$$L^{-1} = \begin{pmatrix} Y^{-1} & & & & \\ -GY^{-1} & Y^{-1} & & & \\ G^2Y^{-1} & -GY^{-1} & Y^{-1} & & \\ \vdots & & & \ddots & \\ (-1)^{m-1}G^{m-1}Y^{-1} & \dots & \dots & -GY^{-1} & Y^{-1} \end{pmatrix},$$

where $G = Y^{-1}Z$. Similarly, we have

$$U^{-1} = \begin{pmatrix} I & -X & \dots & \dots & (-1)^{m-1}X^{m-1} \\ & \ddots & & & \vdots \\ & & I & -X & X^2 \\ & & & I & -X \\ & & & & I \end{pmatrix}.$$

- $\log x$ denotes the logarithm to the base 2 of $x > 0$.
- $\rho(A)$ denotes the spectral radius of the matrix A.
- $\|A\|_F$ denotes the Frobenius norm of the matrix A, i.e.

$$\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}.$$

3 The incomplete factorizations

Let T be an $m \times m$ block tridiagonal Toeplitz matrix, i.e.

$$T = \begin{pmatrix} A & B & & & \\ C & A & B & & \\ & \ddots & \ddots & \ddots & \\ & & C & A & B \\ & & & C & A \end{pmatrix}, \tag{1}$$

where A, B and C are $k \times k$ matrices (i.e. T is $n \times n$ and $n = mk$). If $C^T = B$ and A is symmetric, then T is symmetric, and if $C = B$, then T is block symmetric. Note also that a banded Toeplitz matrix T of order $n = mk$ and bandwidth $2k + 1$ can be viewed as a block tridiagonal Toeplitz matrix.

Our goal is to develop efficient algorithms based on two types of factorization, namely

1. $T = LU + H$, (Incomplete LU factorization)
2. $T = LL^T + K$, (Incomplete LL^T factorization)

where L (U) is a block lower (upper) bidiagonal Toeplitz matrix.

3.1 Incomplete LU factorization

We look for matrices L, U and H such that

$$T = \begin{pmatrix} Y & & & & \\ Z & Y & & & \\ & \ddots & \ddots & & \\ & & Z & Y & \\ & & & Z & Y \end{pmatrix} \begin{pmatrix} I & X & & & \\ & I & X & & \\ & & \ddots & \ddots & \\ & & & I & X \\ & & & & I \end{pmatrix} + E_1 Z X E_1^T = LU + H, \tag{2}$$

where X, Y and Z are $k \times k$ matrices, $E_1 = [I, O, \ldots, O]^T$, and $I = I_k$. This holds if and only if the following constraints are satisfied:

$$\begin{cases} Z = C \\ YX = B \\ ZX + Y = A \end{cases}.$$

For the case of parallel algorithms, we make use of the parallel random access machine (from now on simply called PRAM) [13], with an extended instruction set including the arithmetic operations and tests over a ground field F. The usual cost measures for PRAMs are parallel time and the maximum number of processors operating in parallel. The notation $PT(a, b)$ will be used to denote an $O(b)$ time bounded PRAM with $O(a)$ processors.

Given a field F, one important class of numerical and algebraic problems over F is the class NC_F (see e.g. [14]). Problems in NC_F can be solved by PRAMs with polynomially (i.e. $n^{O(1)}$) many processors running in polylogarithmic (i.e. $O(\log^{O(1)} n)$) time with respect to the size n of the problem. Also, for any given $k \geq 1$, the subclass NC_F^k of NC_F is defined as the set of problems solvable in time $O(\log^k n)$ by PRAMs with a polynomial number of processors. We remark that the NC classes are mainly a tool used by computational complexity theorists to classify fast parallel algorithms. In other words, NC algorithms are not always intented to be of practical interest. Our goal is here to show that, using our incomplete factorization method, very fast parallel algorithms can be obtained for block Toeplitz nonsymmetric problems, thus extending previous results which apply to the symmetric case.

To assess the cost of the parallel algorithms of Section 3, we use the following known results.

1. If $\mathbf{u} = [u_1, \ldots, u_n]^T$ and $\mathbf{v} = [v_1, \ldots, v_n]^T$ are two n-vectors, then the inner product $\mathbf{u}^T\mathbf{v}$ is in NC_F^1, for it can be computed by the naive fan-in algorithm in $PT(n, \log n)$. The number of processors can be reduced of a factor $\log n$ in the following way. First, partition the two vectors in $n/\log n$ vectors, $\mathbf{u}_i$ and $\mathbf{v}_i$, $1 \leq i \leq n/\log n$, of $\log n$ elements each. Then compute simultaneously in $PT(n/\log n, 2\log n)$ the $n/\log n$ scalar products $\mathbf{u}_i^T\mathbf{v}_i$ by using the sequential algorithm. Finally, using the fan-in algorithm, add the $n/\log n$ partial results in $PT(n/(2\log n), \log(n/\log n))$.

2. Matrix by vector products can be computed in $PT(n^2/\log n, \log n)$, as a consequence of the latter result, thus proving that this problem is in NC_F^1.

3. The matrix product is NC_F^1, for it can be computed in $PT(n^\alpha/\log n, 2\log n)$, where $2 \leq \alpha \leq 3$. Such a result, with $\alpha = \log 7$, was first due to Chandra [6] and then improved by several authors. (The current value of α is approximately 2.38.)

4. Finally, the inversion of an $n \times n$ matrix with elements in a field F of characteristic zero is in NC_F^2, for it can be carried in $PT(n^{1/2+\alpha}/\log n, (7/2)\log^2 n)$ by a result of Preparata and Sarwate [21], who in turn improved the fundamental work of Csanky ($PT(n^4/2, (3/2)\log^2 n)$) [10].

Throughout the paper, we will view banded matrices of size $n = km$ as block tridiagonal ones, as is done in [4] for generic banded matrices, and use the following notations and/or definitions.

in an incomplete Cholesky factorization of positive definite band Toeplitz matrices. We generalize Grcar and Sameh's incomplete factorization, and introduce incomplete factorization methods also for nonsymmetric banded Toeplitz matrices.

We describe two strategies which allow to compute both an LU and two LL^T incomplete factorizations. For the sake of brevity, we show how to compute the LU and the LL^T factorizations by using one method, and then how to compute the LL^T factorization by using the other method, even if both methods could be used to derive both factorizations.

Our approach allows to overcome some drawbacks of both superfast methods, which are not suitable for parallel implementation, and parallel band solvers, which can only be applied to systems enjoying special properties (e.g. positive definiteness). On the other hand, the algorithms based on our incomplete factorizations have good performances both in sequential and in parallel, and can be applied to solve general banded Toeplitz systems. Related work in the field can be found in [5], where circulant matrices are used as preconditioners of Toeplitz ones to accelerate conjugate gradient methods.

The rest of this paper is organized as follows. In the next section, we give some preliminary results of parallel complexity analysis and we introduce the main notations used in the paper; the incomplete factorization methods are described in section 3; more precisely, section 3.1 contains the description of an LU factorization, and section 3.2 of two LL^T factorizations. In section 4 we present the algorithms derived by the factorization methods and analyze their demand on computational resources; in section 5 we study some iterative methods obtained by viewing the factorizations as splittings; in section 6 the existence of the incomplete factorizations is taken into account. We find a necessary condition and a sufficient condition for the existence of the LL^T incomplete factorization, and a sufficient condition for the existence of the LU incomplete factorization. Section 7 deals with the application of incomplete factorizations to the matrices obtained from the discrete approximation of Poisson and biharmonic problems and presents some experimental results on the existence of the LU incomplete factorization. Finally we report in the appendix some numerical examples.

2 Preliminaries and notations

In this section we briefly describe the models of computation used for cost analysis. We then recall some known results concerning the arithmetic parallel complexity of basic linear algebra operations (to be used in Section 4), and finally introduce notations adopted throughout the rest of the paper.

We measure the cost of running a sequential algorithm, operating over a field F, by counting the total number of arithmetic operations over F and the number of memory cells used as the workspace. This is often referred to as the "straight-line programs" model.

1 Introduction

Several authors have considered incomplete factorization methods for either solving or preconditioning linear systems (see for example [2, 3, 9]). As it is well known, the interest in incomplete factorization techniques arises because direct methods (as well as the corresponding factorizations) applied to the solution of sparse or structured linear systems are likely to destroy the sparsity or the structure, respectively. The case of sparse matrices has been investigated more intensively, and Cholesky incomplete factorization has been suggested as a major technique [2, 3] to avoid (at least partially) the fill-in caused by those methods.

Let us recall, in more detail, some basic ideas besides the notion of incomplete factorization. If one wants to compute a factorization, say LU, of a matrix A, and if A has some properties, like sparsity, it generally happens that its factors L and U loose all or some of these properties. In fact, for example, a sparse matrix does not have, in general, sparse factors. For these reasons, it has been suggested to consider, for a matrix A with some properties, a factorization LU such that (i) the factors have the same properties of A; (ii) the matrix $A - LU = H$ is *small* with respect to a given measure (e.g. norm, rank). Thus we have a decomposition of the type $A = LU + H$, from which the term incomplete factorization comes out.

In this paper, we investigate a special case of structured linear systems, namely that of banded, and/or block tridiagonal, Toeplitz linear systems. Banded Toeplitz linear systems of bandwidth much less than the order of the coefficient matrix arise in many problems of mathematical physics and statistics, such as least squares approximations by polynomials, stationary time series, and problems involving convolutions.

Given a block tridiagonal Toeplitz matrix T, we present an incomplete LU factorization, where L and U are lower and upper block bidiagonal Toeplitz matrices, respectively, and two incomplete LL^T factorizations, where L is a lower block bidiagonal Toeplitz matrix. Both methods preserve the Toeplitz structure of the factors, and this allows to develop good performance algorithms for the solution of linear systems associated with T. Indeed, preserving the Toeplitz structure will allow us to use Fast Fourier Transform (FFT) algorithms to perform matrix-vector multiplication.

The existence of these incomplete factorizations is related to the existence of a solvent of certain matrix equations. In this paper, we investigate about the existence of a solvent of these equations. We give some conditions that assure the existence of a solvent and show a possible form of it.

Several efficient sequential and parallel algorithms for the solution of Toeplitz linear systems have been developed over the last decades (see [16, 17, 18, 23]). In particular, several superfast (sequential) methods for the solution of dense Toeplitz systems have been developed, which perform $O(n \log^2 n)$ arithmetic operations (see [1] for an overview and for efficient implementations). On the side of parallelism, Grcar and Sameh [16] have presented three parallel algorithms for the solution of positive definite banded Toeplitz linear systems (see also [12]). One of their algorithms results

Incomplete Factorizations for Certain Toeplitz Matrices *

C. Bernini† B. Codenotti‡ M. Leoncini§

G. Resta¶

TR-91-075

May 1992

Abstract

Let T be a block tridiagonal Toeplitz matrix. We introduce two incomplete factorizations of T, namely $T = LU + H$, and $T = LL^T + K$, where L (U) is a block lower (upper) bidiagonal Toeplitz matrix. We first address the question of the existence of the proposed factorizations, which is related to the existence of solutions to suitable matrix equations of size dependent on the bandwidth of T, and then we analyze the efficiency of the factorizations. We also report on a number of computational experiments which show that the quadratic matrix equations involved do have a solvent in many cases not covered by the theory. We finally consider applications of these techniques to the solution of linear systems arising from the discrete approximation of the Poisson and biharmonic equations.

*This work has been partly supported by the Italian National Research Council, under the "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo", subproject 2 "Processori dedicati". Part of this work was done while the second author was with the Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa (Italy).

†Dipartimento di Informatica, Università di Pisa.

‡Int. Comp. Sci. Instit., Berkeley, Ca94704. E-Mail: brunoc@icsi.berkeley.edu.

§Consorzio Pisa Ricerche, Pisa (Italy).

¶Ist. di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa (Italy).