JOURNAL OF COMPUTATIONAL PHYSICS 19, 317–323 (1975)

Algorithms for the Solution of Cyclic Tridiagonal Systems

In this paper we consider several algorithms for the solution of the system $A\mathbf{x} = \mathbf{b}$, where A is an $n \times n$ "cyclic tridiagonal" matrix of the form

$$A = \begin{bmatrix} b_1 & c_1 & & & & a_1 \\ a_2 & b_2 & c_2 & & & \\ & a_3 & b_3 & c_3 & & \\ & & \cdot & \cdot & \cdot & \\ & & & \cdot & \cdot & \cdot \\ & & & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & & & & a_n & b_n \end{bmatrix}$$

and A is assumed to be diagonally dominant.

In the case where $a_1 = c_n = 0$, the matrix A is strictly tridiagonal, and algorithms for the solution of such systems are well known (e.g., [7, p. 195]). Cyclic tridiagonal systems with $a_1 \neq 0$, $c_n \neq 0$ arise in the finite-difference solution of elliptic equations over domains with periodic boundary conditions, e.g., on a cylinder or an annulus, both by direct methods [5] and by iterative methods such as SLOR and ADI [8]; and in the successive peripheral overrelaxation (SPOR) method on the topologically equivalent case of a square with a central hole [2]. They also arise in working with splines on a periodic mesh [1].

We will consider primarily the most usual case in which A is a symmetric circulant matrix. It is natural to scale the equations so that we have $a_i = c_i = 1$ and $b_i = \lambda$, $1 \leqslant i \leqslant n$, where λ is a constant with $|\lambda| > 2$. Also, we will generally assume that the system has to be solved a number of times with different right-hand sides, so that any coefficients required by the solution algorithm can be precalculated and stored. In this situation, the standard method [7] for the corresponding strictly tridiagonal system requires $2n$ additions and $2n$ multiplications per right-hand side, and the storage of n precomputed coefficients. It is of course possible to extend this method, based on Gaussian elimination, to the cyclic tridiagonal case; an example is given by Evans and Atkinson [4]. The resulting algorithm is, however, rather clumsy; even in the symmetric case it requires $4n$ additions and $4n$ multiplications per right-hand side, and the storage of $3n$ precomputed coefficients.

We describe four algorithms for the symmetric circulant case in the form

$$\begin{bmatrix} \lambda & 1 & & & & 1 \\ 1 & \lambda & 1 & & & \\ & 1 & \lambda & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & \lambda & 1 \\ 1 & & & & 1 & \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}. \tag{1}$$

Three of them have appeared in the literature before, while the fourth seems to be new.

(a) *Matrix factorization.* Evans [3] has proposed a neat algorithm based on the factorization $A = \mu QQ^T$, where $\mu = \lambda/(1 + \alpha^2)$ and α, Q are given by

$$\alpha = (-\lambda \pm (\lambda^2 - 4)^{1/2})/2, \tag{2}$$

$$Q = \begin{bmatrix} 1 & & & & & -\alpha \\ -\alpha & 1 & & & & \\ & -\alpha & 1 & & & \\ & & -\alpha & 1 & & \\ & & & \ddots & \ddots & \\ & & & & -\alpha & 1 \end{bmatrix}.$$

The solution vector $\mathbf{x}$ is thus found by first solving for an auxiliary vector $\mathbf{y}$ given by

$$Q\mathbf{y} = \mu^{-1}\mathbf{b}$$

and then solving $Q^T\mathbf{x} = \mathbf{y}$.

The first stage of the algorithm is given by

$$\mu(1 - \alpha^n)\, y_1 = b_1 + \alpha^{n-1}b_2 + \alpha^{n-2}b_3 + \cdots + \alpha^2 b_{n-1} + \alpha b_n\,, \tag{3}$$

$$y_i = \mu^{-1}b_i + \alpha y_{i-1}\,, \qquad 2 \leqslant i \leqslant n,$$

and the second stage by

$$(1 - \alpha^n)\, x_n = \alpha y_1 + \alpha^2 y_2 + \cdots + \alpha^{n-1}y_{n-1} + y_n\,, \tag{4}$$

$$x_i = y_i + \alpha x_{i+1}\,, \qquad n - 1 \geqslant i \geqslant 1.$$

Using a nesting technique to compute the right-hand sides of (3) and (4), the only coefficients which need to be precalculated are μ, α, and $(1 - \alpha^n)$, and the whole algorithm requires $4n$ additions and $5n$ multiplications for each right-hand side. In some applications it would be possible to save one multiplication by incorporating the constant multiplier μ^{-1} during the computation of the right-hand side. Numerical stability is guaranteed by choosing the positive sign in Eq. (2) for $\lambda > 2$ and the negative sign for $\lambda < -2$, so that we always have $|\alpha| < 1$.

(b) *Cyclic reduction.* In the direct method due to Hockney [5] for solving Poisson's equation on a rectangular mesh, extensive use is made of a cyclic reduction technique for solving systems of the form (1). In the following it is assumed that n is a power of 2, say $n = 2^m$; the algorithm can quite easily be extended to the case $n = 3 \times 2^m$.

Consider any three consecutive equations from the system (1):

$$\begin{aligned} x_{i-2} + \lambda x_{i-1} + x_i \qquad\qquad &= b_{i-1}, \\ x_{i-1} + \lambda x_i + x_{i+1} \qquad &= b_i, \\ x_i + \lambda x_{i+1} + x_{i+2} &= b_{i+1}, \end{aligned}$$

with the indices interpreted cyclically, i.e., $x_{n+i} = x_i$. Multiplying the second equation by $-\lambda$ and adding, we obtain

$$x_{i-2} + \lambda^{(1)} x_i + x_{i+2} = b_i^{(1)},$$

where $\lambda^{(1)} = 2 - \lambda^2$, $b_i^{(1)} = b_{i-1} - \lambda b_i + b_{i+1}$.

We thus obtain a new system, of order $n/2$, with the same cyclic tridiagonal structure as the original system. The process can now be repeated until after m levels of recursion only one equation remains which can be solved for x_1. The remainder of the solution can then be built up using a recursive back-solution process. Details of the algorithm are given by Hockney in [5] and [6] and need not be repeated here. The solution requires approximately $4n$ additions and $2n$ multiplications for each right-hand side; only m auxiliary coefficients $\lambda^{(t)}$ and their reciprocals are used, and these can be precomputed if desired. The principal disadvantages of the algorithm are that it can only handle systems of order $n = 2^m$, or at best $n = p \times 2^m$ for p small, and that its recursive structure, with indices interpreted cyclically, makes the computation of storage addresses rather complicated. As shown in [5], there is a possible compensating advantage for large n in that it may be unnecessary to carry out all m levels of reduction in order to obtain the required degree of accuracy.

(c) *The Ahlberg–Nilson–Walsh algorithm.* If we let E be the tridiagonal matrix obtained by deleting the last row and column of A, and define

$$\mathbf{f} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \qquad \hat{\mathbf{x}} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \end{bmatrix} \qquad \hat{\mathbf{b}} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{n-1} \end{bmatrix},$$

then the system (1) can be rewritten as

$$E\hat{\mathbf{x}} + \mathbf{f}x_n = \hat{\mathbf{b}}, \tag{5}$$

$$\mathbf{f}^T\hat{\mathbf{x}} + \lambda x_n = b_n. \tag{6}$$

As the matrix E is tridiagonal, we can easily compute the vector $\mathbf{v} = E^{-1}\hat{\mathbf{b}}$. Equation (5) then becomes

$$\hat{\mathbf{x}} = \mathbf{v} - E^{-1}\mathbf{f}x_n. \tag{7}$$

Substituting in (6) gives a scalar equation for x_n :

$$(\lambda - \mathbf{f}^TE^{-1}\mathbf{f})\,x_n = b_n - \mathbf{f}^T \cdot \mathbf{v} = b_n - (v_1 + v_{n-1}).$$

The remainder of the solution vector, i.e., $\hat{\mathbf{x}}$, can then be found from Eq. (7). The algorithm requires approximately $3n$ additions and $3n$ multiplications for each right-hand side, and $(2n - 1)$ precomputed coefficients, namely $(n - 1)$ for the tridiagonal system $\mathbf{v} = E^{-1}\hat{\mathbf{b}}$, the constant $(\lambda - \mathbf{f}^TE^{-1}\mathbf{f})^{-1}$, and the vector $E^{-1}\mathbf{f}$ which is needed to implement Eq. (7). Details of this algorithm are given (for a general cyclic tridiagonal matrix) by Ahlberg, Nilson, and Walsh ([1, p. 15]).

(d) *Algorithm* 4. The fourth algorithm presented here is based on a simple idea: Suppose that, in Eq. (1), we already knew the value of x_1. We could then obtain the reduced system

$$\begin{bmatrix} \lambda & 1 & & & & & \\ 1 & \lambda & 1 & & & & \\ & 1 & \lambda & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & & \\ & & & & 1 & \lambda & 1 \\ & & & & & 1 & \lambda \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_4 \\ \vdots \\ \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_2 - x_1 \\ b_3 \\ b_4 \\ \vdots \\ \\ b_{n-1} \\ b_n - x_1 \end{bmatrix},$$

which could be solved, using the standard tridiagonal algorithm, in $2(n-1)$ additions and $2(n-1)$ multiplications, using $(n-1)$ precomputed coefficients.

Let $\mathbf{z} = (z_1, z_2, \ldots, z_n)$ be the first row of the inverse matrix A^{-1}. Then x_1 is given by the scalar product $x_1 = \mathbf{z} \cdot \mathbf{b}$, i.e.,

$$x_1 = \sum_{i=1}^{n} z_i b_i, \tag{8}$$

which can be evaluated in $(n-1)$ additions and n multiplications, making a total of $(3n-3)$ additions and $(3n-2)$ multiplications. But since A is a symmetric circulant matrix, so too is A^{-1}, and hence $z_i = z_{n+2-i}$, $2 \leqslant i \leqslant n$. Let m = integer part of $(n+1)/2$. We can rewrite the summation (8) as

$$x_1 = z_1 b_1 + \sum_{i=2}^{m} z_i(b_i + b_{n+2-i}) \qquad \text{if} \quad n \text{ is odd}$$

or

$$x_1 = z_1 b_1 + \sum_{i=2}^{m} z_i(b_i + b_{n+2-i}) + z_{m+1} b_{m+1} \qquad \text{if} \quad n \text{ is even.}$$

We have thus reduced the total number of operations to approximately $3n$ additions and $2.5n$ multiplications, and the number of precomputed coefficients to approximately $1.5n$. Since the tridiagonal algorithm itself is stable with respect to rounding errors, the only problem lies in the accurate determination of the vector $\mathbf{z}$. It is quite easy to show that

$$z_{i+1} = \sigma(\alpha^i + \alpha^{n-i}),$$

where α is given by Eq. (2), and

$$\sigma = 1/[(\alpha + \alpha^{n-1}) + \lambda(1 + \alpha^n)].$$

The four algorithms described above were all programmed with comparable care in Assembler Language and run (in single precision) on an IBM 360/195. Each was timed for four values of n; these were chosen to be powers of 2 in order to accommodate the cyclic reduction algorithm, but it should be remembered that the other three methods are subject to no such restrictions on n. The results are presented in Table I, which shows that Algorithm 4 is about 25 % faster than the matrix factorization scheme, with the two remaining algorithms intermediate in speed. Cyclic reduction suffers from its more complicated structure for smaller values of n.

The accuracy of the various schemes was also compared for various values of n and λ. In each case, a hundred random vectors $\mathbf{x}$ were generated with entries in the range $0.0 \leqslant x_i \leqslant 10.0$, corresponding vectors $\mathbf{b} = A\mathbf{x}$ were computed, and the four algorithms were used to compute an approximate solution from $\mathbf{b}$. The largest error in each computed solution vector was found, and these were averaged over the 100 random vectors. The results are summarized in Table II.

TABLE I

Execution Times in Microseconds for Solution of a Cyclic Tridiagonal System of Order n

n	Matrix factorization	Cyclic reduction	Ahlberg/Nilson/Walsh	Algorithm 4
16	36	36	31	28
32	64	60	55	48
64	119	105	103	88
128	231	194	199	167

TABLE II

Mean Maximum Errors in the Solution of a Cyclic Tridiagonal System of Order n (Unit: 10^{-6})

n	Matrix factorization	Cyclic reduction	Ahlberg/Nilson/Walsh	Algorithm 4
		(a) $\lambda = -3.0$		
16	7.7	14.7	10.1	10.3
32	8.3	16.8	12.3	12.3
64	9.6	19.5	13.9	13.9
128	9.9	20.6	14.5	14.6
		(b) $\lambda = +4.0$		
16	4.4	8.1	6.8	6.7
32	5.0	9.4	7.6	7.5
64	5.5	11.1	8.0	8.0
128	5.7	12.1	8.5	8.6

It will be seen that matrix factorization gives the most accurate answers, and cyclic reduction the least, with the other two algorithms intermediate and very similar. However, all four algorithms are stable with respect to rounding errors, and for practical purposes there is little to choose between them from this point of view.

A careful analysis of the Ahlberg–Nilson–Walsh algorithm shows that the number of multiplications could be reduced to 2.5n as in the case of Algorithm 4, by noting that the elements of the vector $\mathbf{h} = E^{-1}\mathbf{f}$ are symmetric, i.e., $h_i = h_{n-i}$, $1 \leqslant i \leqslant n - 1$; hence half the multiplications in Eq. (7) are in fact redundant. This improvement should make the Ahlberg–Nilson–Walsh algorithm competitive with Algorithm 4, though the latter is conceptually a little simpler.

General cyclic tridiagonal systems in which the matrix A is neither symmetric nor circulant arise in both spline and elliptic equation applications when a nonuniform mesh is used. Neither matrix factorization nor cyclic reduction can easily be generalized to these cases. For the Ahlberg–Nilson–Walsh algorithm the generalized form is given in [1], and for Algorithm 4 the generalized form is obvious, provided that the first row of A^{-1} can be obtained in some way. Both generalized algorithms require approximately 3n additions and 4n multiplications per right-hand side, and 3n precomputed coefficients.

Finally, Algorithm 4 can easily be generalized to handle cyclic pentadiagonal systems of order n, either by computing two elements of the solution vector directly and then solving the resulting strictly pentadiagonal system of order $(n - 2)$, or by factorizing the cyclic pentadiagonal matrix into the product of two cyclic tridiagonal matrices.

References

1. H. H. Ahlberg, E. N. Nilson, and J. L. Walsh, "The Theory of Splines and Their Applications," Academic Press, New York, 1967.
2. A. Benson and D. J. Evans, *J. Inst. Math. Its Appl.* **9** (1972), 68.
3. D. J. Evans, The Numerical Solution of Elliptic and Parabolic Equations Occurring in Computational Physics, *in* "Computational Physics Conference Digest," pp. 46–56, Institute of Physics and the Physical Society, London, 1970.
4. D. J. Evans and L. V. Atkinson, *Comput. J.* **13** (1970), 323.
5. R. W. Hockney, *J. Assoc. Comput. Mach.* **12** (1965), 95.
6. R. W. Hockney, The Potential Calculation and Some Applications, *in* "Methods in Computational Physics," Vol. 9, pp. 135–211, Academic Press, New York, 1970.
7. R. S. Varga, "Matrix Iterative Analysis," Prentice-Hall, Englewood Cliffs, N.J., 1962.
8. W. L. Wood, *SIAM J. Numer. Anal.* **8** (1971), 439.

Received: March 17, 1975

Clive Temperton

Meteorological Office
Bracknell, Berkshire
England