

Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems

By James R. Bunch* and Linda Kaufman**

Abstract. Several decompositions of symmetric matrices for calculating inertia and solving systems of linear equations are discussed. New partial pivoting strategies for decomposing symmetric matrices are introduced and analyzed.

1. Introduction. In [5] Bunch and Parlett present an algorithm, called the diagonal pivoting method, for calculating the inertia of real symmetric or complex Hermitian matrices, and for solving systems of linear equations when the matrix is real symmetric, complex symmetric, or complex Hermitian. Using a pivoting strategy comparable to complete pivoting for Gaussian elimination, Bunch [2] shows that the diagonal pivoting method with this complete pivoting strategy is nearly as stable as Gaussian elimination with complete pivoting. (The bound on element growth is $3nf(n)$, cf. $\sqrt{nf(n)}$, where

$$f(n) = \left(\prod_{k=2}^n k^{1/(k-1)} \right)^{1/2} < 1.8n^{(1/4)\log n},$$

the cost of stability is at least $n^3/12$ but no more than $n^3/6$ comparisons, cf. $n^3/3$ comparisons, while requiring $n^3/6$, cf. $n^3/3$, multiplications and additions.)

In [3] Bunch discusses various partial pivoting strategies for the diagonal pivoting method which require only $O(n^2)$ comparisons instead of $O(n^3)$, although these increase element growth. In this paper we shall present and analyze several good partial pivoting algorithms for the diagonal pivoting method.

In Section 2 we shall show that the diagonal pivoting method can be modified so that only n^2 comparisons are needed but element growth is now bounded by $(2.57)^{n-1}$ (cf. 2^{n-1} for Gaussian elimination with partial pivoting). Thus, the diagonal pivoting method can solve an $n \times n$ nonsingular symmetric system of linear equations with $n^3/6$ multiplications, $n^3/6$ additions, $\leq n^2$ comparisons, and $n^2/2$ storage while Gaussian elimination with partial pivoting requires $n^3/3$ multiplications, $n^3/3$ additions, $n^2/2$ comparisons, and n^2 storage.

In Section 3 other variations of the algorithm are presented and analyzed. In Section 4 the situation for symmetric band matrices is discussed. We are unable to give an algorithm which preserves the band structure for every bandwidth $2m + 1$.

Received February 25, 1975, revised March 23, 1976.

AMS (MOS) subject classifications (1970). Primary 65F05, 65F30, 15A21, 15A57, 15A63.

*Partial support was provided by NSF Grant MCS75-06510.

**Partial support was provided by NSF Grant MCS75-23333.

Current address (L. Kaufman): Bell Laboratories, Murray Hill, New Jersey.

However, we are able to give good algorithms for the important special cases when $m = 1$ and $m = 2$ (tridiagonal and five-diagonal).

We have included an appendix for those who are unfamiliar with the diagonal pivoting method. In the following sections we shall assume familiarity with the concepts in the appendix.

2. A Partial Pivoting Strategy. In this section we describe and analyze a partial pivoting strategy for transforming an $n \times n$ symmetric indefinite matrix A by stable congruences into a block diagonal matrix D , where each block is of order 1 or 2. As in Bunch and Parlett's [5] complete pivoting strategy, the algorithm generates a sequence of matrices $A^{(k)}$ of order k according to the formula

$$A^{(k-s)} = B - CE^{-1}C^t,$$

where $A^{(k)}$ or a symmetric permutation of $A^{(k)}$, $P_k A^{(k)} P_k^t$, is partitioned as

$$\left[\begin{array}{c|c} E & C^t \\ \hline C & B \end{array} \right],$$

where E is an $s \times s$ nonsingular matrix, C is a $(k-s) \times s$ matrix, and B is a $(k-s) \times (k-s)$ matrix and s is either 1 or 2. If E is $s \times s$, we say that an $s \times s$ pivot has been used. (For convenience, we shall still denote $P_k A^{(k)} P_k^t$ as $A^{(k)}$.)

Bunch and Parlett's pivoting strategy may be considered analogous to Gaussian elimination with complete pivoting. Unfortunately, there is no stable scheme exactly analogous to Gaussian elimination with partial pivoting; one cannot construct an algorithm for which there is a bound on the element growth of the sequence $A^{(k)}$ when at each stage only one column of $A^{(k)}$ is examined (see [3]). The method described in this section guarantees that the element growth in $A^{(k)}$ is bounded while searching for the largest element in at most two columns in each $A^{(k)}$. For future reference we call the strategy Algorithm A.

In Algorithm A, the matrix $A^{(k-s)}$ is determined as follows:

(1) Determine $\lambda^{(k)}$, the absolute value of the largest off-diagonal element in absolute value in the first column of $A^{(k)}$, i.e.

$$\lambda^{(k)} = \max_{2 \leq i \leq k} |A_{i1}^{(k)}|.$$

If $\lambda^{(k)} = 0$, decrease k by 1 and return to 1. Let r be the least integer such that $|A_{r1}^{(k)}| = \lambda^{(k)}$.

(2) If $|A_{11}^{(k)}| \geq \alpha \lambda^{(k)}$ where $0 < \alpha < 1$, perform a 1×1 pivot to obtain $A^{(k-1)}$, decrease k by 1 and return to (1). We will show that a good value for α is $(1 + \sqrt{17})/8$.

(3) Determine $\sigma^{(k)}$, the absolute value of the largest off-diagonal element in absolute value in the r th column of $A^{(k)}$, i.e.

$$\sigma^{(k)} = \max_{\substack{1 \leq m \leq k \\ m \neq r}} |A_{m,r}^{(k)}|.$$

(Recall that A_{r1} is the largest off-diagonal element in the first column.)

(4) If $\alpha\lambda^{(k)2} \leq |A_{11}^{(k)}|\sigma^{(k)}$, then perform a 1×1 pivot to obtain $A^{(k-1)}$, decrease k by 1, and return to (1). (We need this test to guarantee stability.)

(5) If $|A_{rr}^{(k)}| \geq \alpha\sigma^{(k)}$, then interchange the first and r th rows and columns of $A^{(k)}$, perform a 1×1 pivot with the new $A^{(k)}$, decrease k by 1, and return to (1).

(6) Interchange the second and r th rows and columns of $A^{(k)}$ so that $|A_{21}^{(k)}| = \lambda^{(k)}$, perform a 2×2 pivot to obtain $A^{(k-2)}$, decrease k by 2 and return to (1). In order to compute $A^{(k-2)}$, either E^{-1} can be formed directly (as in [2]) or $E^{-1}C^t$ can be formed by Gaussian elimination with partial pivoting. The stability analysis is the same for each here, but the operation count is slightly higher (in the lower order terms) for the former and these are the operation counts that we give here.

Step (4) of the algorithm deserves an explanation. The step was designed to screen out a pathological case with 2×2 pivoting when the largest off-diagonal element in absolute value of the r th column was larger than that of the first column, i.e. when $\sigma^{(k)} > \lambda^{(k)}$. In this case, step (4) is equivalent to:

scaling the first row and column of $A^{(k)}$ so that the absolute value of the largest element in the first column of $A^{(k)}$ is equal to $\sigma^{(k)}$ and repeating steps (1) and (2) on the scaled matrix.

In the absence of roundoff error, the reduced matrix $A^{(k-s)}$ generated by Algorithm A and the one generated by using explicit scaling would be the same. If a 2×2 pivot had been performed when the test in step (4) dictated the use of a 1×1 pivot, then the element growth of $A^{(k-2)}$ could not be bounded a priori. Whenever $\lambda^{(k)} \geq \sigma^{(k)}$, the test in step (4) cannot be passed and one proceeds with step (5).

Note that whenever a 2×2 pivot is used, after permuting $A^{(k)}$, we have $A_{11}^{(k)}A_{22}^{(k)} < \alpha^2|A_{21}^{(k)}|^2 < |A_{21}^{(k)}|^2$; thus a 2×2 block in D corresponds to a positive-negative pair of eigenvalues. This means that if A is positive definite then D will be diagonal.

We shall now analyze Algorithm A. Let $\mu = \max_{1 \leq i, j \leq n} |A_{ij}|$ and $\mu^{(k)} = \max_{1 \leq i, j \leq k} |A_{ij}^{(k)}|$ for each reduced matrix $A^{(k)}$ that exists. (If $A^{(k)}$ uses a 2×2 pivot then $A^{(k-1)}$ does not exist.) Note that both $\lambda^{(k)}$ and $\sigma^{(k)}$ are less than or equal to $\mu^{(k)}$.

If a 1×1 pivot is used, after permuting $A^{(k)}$, we have

$$A_{ij}^{(k-1)} = A_{i+1, j+1}^{(k)} - A_{i+1, 1}^{(k)}A_{j+1, 1}^{(k)}/A_{11}^{(k)}$$

so that by step (2) of Algorithm A,

$$(2.1) \quad \mu^{(k-1)} \leq \mu^{(k)} + \lambda^{(k)}/\alpha \leq \mu^{(k)}(1 + 1/\alpha);$$

by step (4),

$$(2.2) \quad \mu^{(k-1)} \leq \mu^{(k)} + \lambda^{(k)2}/|A_{11}^{(k)}| \leq \mu^{(k)} + \sigma^{(k)}/\alpha \leq \mu^{(k)}(1 + 1/\alpha);$$

and by step (5),

$$(2.3) \quad \mu^{(k-1)} \leq \mu^{(k)} + \sigma^{(k)}/\alpha \leq \mu^{(k)}(1 + 1/\alpha).$$

If a 2×2 pivot is used, after permuting $A^{(k)}$, we have

$$(2.4) \quad A_{ij}^{(k-2)} = A_{i+2, j+2}^{(k)} - \frac{[(A_{i+2,1}^{(k)}A_{22}^{(k)} - A_{i+2,2}^{(k)}A_{21}^{(k)})A_{j+2,1}^{(k)} + (A_{i+2,2}^{(k)}A_{11}^{(k)} - A_{i+2,1}^{(k)}A_{21}^{(k)})A_{j+2,2}^{(k)}]}{[(A_{11}^{(k)}A_{22}^{(k)}/A_{21}^{(k)} - A_{21}^{(k)})A_{21}^{(k)}]}$$

Since $|A_{11}^{(k)}|\sigma^{(k)} < \alpha\lambda^{(k)^2}$ by step (4) and $|A_{22}^{(k)}| < \alpha\sigma^{(k)}$ by step (5),

$$|A_{11}^{(k)}||A_{22}^{(k)}| < \alpha^2\lambda^{(k)^2},$$

which implies that

$$(2.5) \quad \begin{aligned} |A_{11}^{(k)}A_{22}^{(k)}/A_{21}^{(k)} - A_{21}^{(k)}| &> \lambda^{(k)}(1 - \alpha^2) \quad \text{or} \\ 1/|A_{11}^{(k)}A_{22}^{(k)}/A_{21}^{(k)} - A_{21}^{(k)}| &< 1/(\lambda^{(k)}(1 - \alpha^2)). \end{aligned}$$

Equations (2.4) and (2.5) together imply that

$$\mu^{(k-2)} \leq \mu^{(k)} + \frac{(\lambda^{(k)}\alpha\sigma^{(k)} + \sigma^{(k)}\lambda^{(k)})\lambda^{(k)} + (\sigma^{(k)}|A_{11}^{(k)}| + \lambda^{(k)^2})\sigma^{(k)}}{\lambda^{(k)^2}(1 - \alpha^2)},$$

Since $\sigma^{(k)}|A_{11}^{(k)}| < \alpha\lambda^{(k)^2}$,

$$(2.6) \quad \begin{aligned} \mu^{(k-2)} &\leq \mu^{(k)} + (\alpha\sigma^{(k)} + \sigma^{(k)} + \alpha\sigma^{(k)} + \sigma^{(k)})/(1 - \alpha^2) \\ &\leq \mu^{(k)}(1 + 2(1 + \alpha)/(1 - \alpha^2)) = \mu^{(k)}(1 + 2/(1 - \alpha)). \end{aligned}$$

By (2.1), (2.2), (2.3), and (2.6),

$$\max_k \mu^{(k)} \leq \max\{(1 + 1/\alpha)^{n-k}, (1 + 2/(1 - \alpha))^{(n-k)/2}\}\mu.$$

The growth is minimized when $(1 + 1/\alpha)^2 = (1 + 2/(1 - \alpha))$, i.e. when $\alpha = (1 + \sqrt{17})/8 \approx 0.6404$, in which case $\max_k \mu^{(k)} < \mu(2.57)^{n-1}$.

As noted above, Algorithm A is equivalent to one which scales the first row and column of $A^{(k)}$ at each step so that the maximum norm of the first and second columns of $A^{(k)}$ are equal. If the scaling had been done explicitly, then the algorithm would determine a permutation matrix P , a lower triangular matrix \tilde{M} , and a block diagonal matrix \tilde{D} such that $PAP^t = \tilde{M}\tilde{D}\tilde{M}^t$, where $|\tilde{M}_{ij}| \leq \max(1/\alpha, 1/(1 - \alpha))$ and $|\tilde{D}_{ij}| \leq (2.57)^{n-1}\mu$.

Algorithm A creates the same matrix P , but a unit lower triangular matrix M and a block diagonal matrix D such that $PAP^t = MDM^t$, where $M = \tilde{M}S$ and $D = S^{-1}\tilde{D}S^{-1}$, where S is a diagonal matrix given by

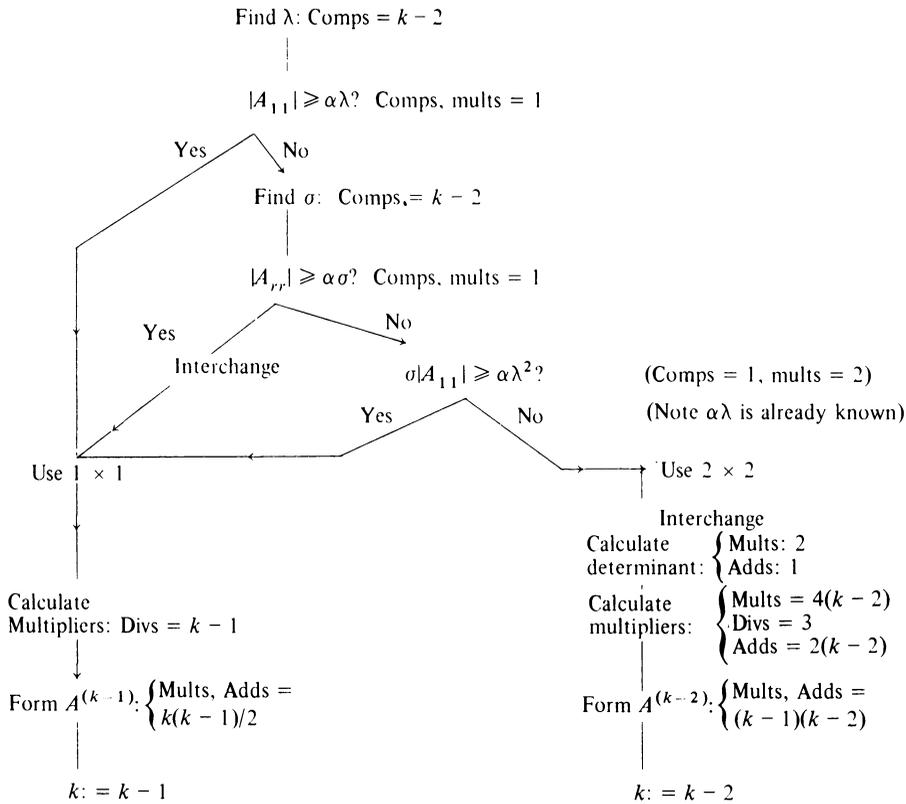
$$S_{kk} = \begin{cases} 1 & \text{if } (|A_{11}^{(k)}| \geq \alpha\lambda^{(k)}) \text{ or } (|A_{22}^{(k)}| \geq \alpha\sigma^{(k)}) \text{ or } (\lambda^{(k)} \geq \sigma^{(k)}), \\ \min(\sqrt{\alpha\sigma^{(k)}/|A_{11}^{(k)}|}, \sigma^{(k)}/\lambda^{(k)}) & \text{otherwise,} \end{cases}$$

i.e. $|S_{kk}| \geq 1$. The bound on the elemental growth of D is that of \tilde{D} .

The operation count is detailed in the following flowchart of Algorithm A. The words comps, mults, adds and divs indicate the number of comparisons, multiplications,

additions and divisions for each step of the algorithm.

Flowchart of operation count for computing $A^{(k-s)}$, $s = 1$ or 2



The comparison count is much less for Algorithm A than for the complete pivoting scheme, and in practice this fact has had a much larger impact than originally anticipated, sometimes cutting the execution time by about 40%. In the complete pivoting scheme, at each stage of the algorithm the largest element in the current submatrix is determined and the comparison count is bounded by $n^3/6$ and $n^2/2 + n/3$ (see [2]), while in Algorithm A at most two columns are searched and the comparison count is at most $n^2 - 1$.

The table given below gives upper bounds for the number of operations and storage space required for solving $Ax = b$ by Algorithm A, Gaussian elimination with partial pivoting and Aasen's tridiagonal algorithm [1]. In the table the decomposition phase for Aasen includes only the reduction to tridiagonal form. The LU decomposition of the tridiagonal matrix is included in the solution phase.

3. Variations of the Algorithm.

3.1 *Estimating $\mu^{(k)}$.* For small n , we can construct examples for which the element growth bound of $(2.57)^{n-1}\mu$ for Algorithm A of Section 2 is attained. However, we have been unable to construct an example for arbitrary n which reaches $(2.57)^{n-1}\mu$. Furthermore, as with Gaussian elimination with partial pivoting, large growth does not seem to occur in practice. Nevertheless, one would like to have a quick method for obtaining an estimate of $\mu^{(k)}$ so that whenever the element growth

	Multiplications and Divisions	Additions	Comparisons	Storage	Element Growth
Algorithm A					
decomposition	$\frac{n^3}{6} + \frac{3n^2}{4} + \frac{7}{3}n$	$\frac{n^3}{6} + \frac{n^2}{4} + \frac{7}{6}n$	$n^2 - 1$	$\frac{n^2}{2} + \frac{3}{2}n$	2.57^{n-1}
solution phase	$n^2 + 2n$	$n^2 - \frac{n}{2}$	0	$\frac{n^2}{2} + \frac{5}{2}n$	
Total	$\frac{n^3}{6} + \frac{7}{4}n^2 + \frac{13}{3}n$	$\frac{n^3}{6} + \frac{5n^2}{4} - \frac{5}{3}n$	$n^2 - 1$	$\frac{n^2}{2} + \frac{5}{2}n$	
Gaussian					
elimination					
decomposition	$\frac{n^3}{3} - \frac{n}{3}$	$\frac{n^3}{3} - \frac{n^2}{2} + \frac{n}{6}$	$\frac{n^2}{2} - \frac{n}{2}$	$n^2 + n$	2^{n-1}
solution phase	n^2	$n^2 - n$	0	$n^2 + 2n$	
Total	$\frac{n^3}{3} + n^2 - \frac{n}{3}$	$\frac{n^3}{3} + \frac{n^2}{2} - \frac{5}{6}n$	$\frac{n^2}{2} - \frac{n}{2}$	$n^2 + 2n$	
Aasen					
decomposition	$\frac{n^3}{6} + 2n^2 - \frac{49}{6}n$	$\frac{n^3}{6} + n^2 - \frac{25}{6}n$	$\frac{n^2}{2} - \frac{3n}{2}$	$\frac{n^2}{2} + \frac{n}{2}$	4^{n-1}
solution phase	$n^2 + 4n$	$n^2 + 2n$	n	$\frac{n^2}{2} + \frac{11n}{2}$	
Total	$\frac{n^3}{6} + 3n^2 - \frac{25}{6}n$	$\frac{n^3}{6} + 2n^2 - \frac{13}{6}n$	$\frac{n^2}{2} - \frac{n}{2}$	$\frac{n^2}{2} + \frac{11n}{2}$	

is excessive, a switch can be made to the slower executing complete pivoting scheme of Bunch and Parlett [5], for which the element growth is bounded by $\sqrt{n}f(n)c(\alpha)h(n, \alpha)$ where $f(n) = \prod_{k=2}^n (k^{1/(k-1)})^{1/2}$ which grows slowly like $n^{(1/4)\log(n)}$, and

$$c(\alpha)h(n, \alpha) \leq 3\sqrt{n} \quad \text{for } \alpha_0 = (1 + \sqrt{17})/8.$$

Businger [6] has presented an inexpensive algorithm for monitoring the growth in Gaussian elimination with partial pivoting. Because of the symmetry of our decomposition, Businger's idea is very satisfying when applied to Algorithm A.

According to (2.1), (2.2), and (2.3) when a 1×1 pivot is used to find $A^{(k-1)}$,

$$\mu^{(k-1)} \leq \mu^{(k)} + \beta^{(k)},$$

where

$$\beta^{(k)} = \begin{cases} \lambda^{(k)}/\alpha & \text{if } A^{(k-1)} \text{ is formed at step (2),} \\ \sigma^{(k)}/\alpha & \text{if } A^{(k-1)} \text{ is formed at steps (4) or (5).} \end{cases}$$

According to (2.6), when a 2×2 pivot is used to find $A^{(k-2)}$,

$$\mu^{(k-2)} \leq \mu^{(k)} + \beta^{(k)} + \beta^{(k-1)},$$

where $\beta^{(k)} = \beta^{(k-1)} = \sigma^{(k)}/(1 - \alpha)$. Therefore

$$\mu^{(k)} \leq \mu + \sum_{j=k+1}^n \beta^{(j)}.$$

Thus only the $n^2/2$ comparisons to determine μ , and k divisions and k additions are needed to determine a decent estimate to $\mu^{(k)}$.

We suggest that whenever $\mu^{(k)} \geq 13n\mu$ the complete pivoting strategy of Bunch and Parlett [5] should be used.

3.2 *Accumulating Inner Products: Algorithm B.* In [4] a reformulation of Algorithm A is described which permits the accumulation of inner products in multiple precision and which would probably be more suitable for electronic hand calculators. The method, called Algorithm B, relates to Algorithm A in the same way that the Crout-Doolittle algorithm for solving a system of equations with a general matrix relates to Gaussian elimination with partial pivoting; see [10]. Algorithm B was motivated by Aasen's [1] description of Parlett and Reid's [9] algorithm.

3.3. *Other Strategies.* Other partial pivoting strategies, similar to Algorithm A, can be formulated, including Algorithms C and D given below.

In Algorithm C, $A^{(k-s)}$ is determined as follows:

(1) Determine $\mu_1^{(k)} = |A_{pp}^{(k)}| = \max_{1 \leq i \leq k} |A_{ii}^{(k)}|$ and permute the first and p th rows and columns of $A^{(k)}$ so that $|A_{11}^{(k)}| = \mu_1^{(k)}$.

(2) Determine $\lambda^{(k)} = |A_{r1}^{(k)}| = \max_{2 \leq i \leq k} |A_{i1}^{(k)}|$.

(3) If $\mu_1^{(k)} \geq \alpha\lambda^{(k)}$, use $A_{11}^{(k)}$ as a 1×1 pivot to obtain $A^{(k-1)}$, decrement k by 1, and return to (1). As before, a good value for α is $(1 + \sqrt{17})/8$.

(4) Determine $\sigma^{(k)} = \max_{2 \leq m \leq k; m \neq r} |A_{m,r}^{(k)}|$.

(5) If $\alpha\lambda^{(k)^2} \leq |A_{11}^{(k)}|\sigma^{(k)}$, use $A_{11}^{(k)}$ as a 1×1 pivot to obtain $A^{(k-1)}$, decrement k by 1, and return to (1).

(6) Interchange the second and r th rows and columns of $A^{(k)}$ so that $|A_{21}^{(k)}| = \lambda^{(k)}$ and perform a 2×2 pivot to obtain $A^{(k-2)}$, decrement k by 2 and return to (1).

Because the maximum element of a positive-definite matrix is on the diagonal, when Algorithm C is applied to a positive-definite matrix A , one obtains the decomposition $PAP^t = MDM^t$ with $|M_{ij}| \leq 1$. For some applications this is very desirable. Unfortunately, on most problems, Algorithm C is more costly than Algorithm A because at each stage the diagonal is searched and extra interchanges might be required. In Algorithm A between $n^2/2 + O(n)$ and $n^2 + O(n)$ comparisons are needed to determine the pivot strategy while in Algorithm C between $3n^2/4 + O(n)$ and $3n^2/2 + O(n)$ comparisons are needed to determine the pivot strategy. The bound on element growth in $A^{(k)}$ for Algorithm C is the same as for Algorithm A. When solving a system of equations with A , Algorithm C will probably give a smaller error than Algorithm A.

In Algorithm D, $A^{(k-s)}$ is determined as follows:

(1) Determine $\lambda^{(k)} = |A_{r1}^{(k)}| = \max_{2 \leq i \leq k} |A_{i1}^{(k)}|$.

(2) If $|A_{11}^{(k)}| \geq \alpha\lambda^{(k)}$, use $A_{11}^{(k)}$ as a 1×1 pivot to obtain $A^{(k-1)}$, decrement k by 1, and return to (1). Below we shall show that a good value of α is about 0.525.

(3) Determine $\sigma^{(k)} = \max_{2 \leq m \leq k} |A_{m,r}^{(k)}|$.

(4) If $\alpha\lambda^{(k)^2} \leq |A_{11}^{(k)}|\sigma^{(k)}$, then use $A_{11}^{(k)}$ as a 1×1 pivot to obtain $A^{(k-1)}$, decrement k by 1, and return to (1).

(5) Interchange the second and r th rows and columns of $A^{(k)}$ so that $|A_{21}^{(k)}| = \lambda^{(k)}$, perform a 2×2 pivot to obtain $A^{(k-2)}$, decrement k by 2 and return to (1).

Whenever a 1×1 pivot is used in Algorithm D, no interchanges are performed, which means less bookkeeping, fewer references to memory in general, and fewer opportunities to interfere with the structure of the system. In particular, the algorithm is quite amenable to tridiagonal systems.

The disadvantage of Algorithm D is a larger bound in the element growth in the matrix. As in Section 2, let $\mu^{(k)} = \max_{1 \leq i, j \leq k} |A_{ij}^{(k)}|$. As in Algorithm A, whenever a 1×1 pivot is used, $\mu^{(k-1)} \leq \mu^{(k)}(1 + 1/\alpha)$. When a 2×2 pivot is used, after permuting $A^{(k)}$,

$$|A_{11}^{(k)}| |A_{22}^{(k)}| \leq |A_{11}^{(k)}| \sigma^{(k)} < \alpha \lambda^{(k)2},$$

so

$$|A_{11}^{(k)} A_{22}^{(k)} / A_{21}^{(k)} - A_{21}^{(k)}| > \lambda^{(k)}(1 - \alpha),$$

which is a slightly smaller bound than in Algorithm A. Because $|A_{22}^{(k)}| < \sigma^{(k)}$ and $|A_{11}^{(k)}| \sigma^{(k)} < \alpha \lambda^{(k)2}$, Eq. (2.4) implies

$$\mu^{(k-2)} \leq [1 + (3 + \alpha)/(1 - \alpha)] \mu^{(k)}.$$

Thus

$$\mu^{(k)} \leq \max\{(1 + 1/\alpha)^{n-k}, [1 + (3 + \alpha)/(1 - \alpha)]^{(n-k)/2}\} \mu,$$

which is minimized when $(1 + 1/\alpha)^2 = 1 + (3 + \alpha)/(1 - \alpha)$. This occurs when α is approximately 0.525, giving a bound of $(2.92)^{n-1} \mu$, which is larger than in Algorithm A.

4.1 Band Matrices. Many of the problems in numerical linear algebra with symmetric indefinite matrices involve band matrices. A band matrix A is said to have bandwidth m if $A_{ij} = 0$ for $|i - j| > m$. When A is band, one would like to use an algorithm, like Gaussian elimination with partial pivoting, which takes advantage of the band structure of the matrix to increase the efficiency of the algorithm.

Unfortunately, except for $m = 1$ and $m = 2$, none of the algorithms outlined in Sections 2 and 3 guarantee the preservation of the band structure of the matrix. The row and column interchanges used to guarantee stability destroy the band structure of the system.

Algorithm D does the least damage of all the algorithms A–D, since interchanges only occur when a 2×2 pivot is used, and hence only in this case is the bandwidth increased. Let m_k be the bandwidth of the matrix $A^{(k)}$ generated by Algorithm D. When a 1×1 pivot is used,

$$m_{k-1} = \begin{cases} m_k - 1 & \text{if } m_k > m, \\ m_k & \text{otherwise.} \end{cases}$$

When a 2×2 pivot is used, $m_{k-2} = \max(m_k - 2, m, j + m - 2)$ where the j th and 2nd columns are interchanged before the creation of $A^{(k-2)}$. Since $j \leq m_k$, one is assured that

$$m_{k-2} \leq m_k + m - 2 \quad \text{and} \quad m_k \leq m + \frac{n-k}{2} (m-2).$$

For $m > 2$, one must concede that the band structure might be ruined.

In Section 4.2 we discuss the tridiagonal case ($m = 1$) and in Section 4.3 we present an algorithm for the five-diagonal case ($m = 2$).

4.2 *Tridiagonal Matrices.* Let T be a symmetric tridiagonal matrix, i.e. $T_{ij} = 0$ for $|i - j| > 1$. Of the many algorithms that have been proposed to solve $Tx = b$, Gaussian elimination with partial pivoting has proved the least time-consuming. However, Gaussian elimination with partial pivoting does not preserve symmetry. In [3] Bunch has proposed a symmetry preserving algorithm which can be used to determine the inertia of T as well as solve a system of equations. Like those given in Sections 2 and 3, the algorithm finds the MDM^t decomposition of (1.1) by generating a sequence of tridiagonal matrices $T^{(k)}$ of order k . We show the first step which is typical:

Let α be a fixed number such that $0 < \alpha < 1$.

(1) If $|T_{11}| \geq \alpha |T_{21}|^2$, then use a 1×1 pivot to generate $T^{(n-1)}$.

(2) If $|T_{11}| < \alpha |T_{21}|^2$, then use a 2×2 pivot to generate $T^{(n-2)}$.

Bunch [3] shows that the bound on element growth is minimized when $\alpha = (\sqrt{5} - 1)/(2\mu)$ where $\mu = \max_{1 \leq i, j \leq n} |T_{ij}|$. With this value of α ,

$$\max_{1 \leq i, j \leq k} |T_{ij}^{(k)}| \leq \frac{(3 + \sqrt{5})}{2} \mu.$$

Table 4.2 gives the operation counts and storage requirements for Bunch's algorithm [3] and Gaussian elimination with partial pivoting. When storage is crucial, Bunch's algorithm [3] is preferable to Gaussian elimination with partial pivoting.

TABLE 4.1. *Operation Count: Tridiagonal Case*

	Bunch's Original Algorithm [3]		Modified Algorithm		Gaussian Elimination with Partial Pivoting	
	Decomposition Only	Solving $Tx = b$	Decomposition Only	Solving $Tx = b$	Decomposition Only	Solving $Tx = b$
Multiplications	$\frac{7}{2}n + \frac{1}{2}p$	$\frac{17}{2}n - \frac{3}{2}p$	$\frac{7}{2}n + \frac{1}{2}p$	$7n$	$3n$	$7n$
Additions	n	$4n - p$	n	$4n - p$	$2n$	$5n$
Comparisons	$\frac{5}{2}n + \frac{1}{2}p$	$\frac{5}{2}n + \frac{1}{2}p$	$\frac{5}{2}n + \frac{1}{2}p$	$\frac{5}{2}n + \frac{1}{2}p$	n	n
Storage Required	$3n$	$4n$	$3n$	$5n$	$5n$	$6n$

(p represents the number of 1×1 pivots)

Bunch's algorithm entails examining the whole matrix to compute μ before determining α . When the whole matrix cannot fit in main storage or when it is not necessary to obtain the complete decomposition, searching through the whole matrix

beforehand can be expensive. This problem can be remedied by changing the test in step (1) to:

(1) If $\max(|A_{21}|, |A_{22}|, |A_{32}|) \times |A_{11}| \geq \alpha A_{21}^2$, then use a 1×1 pivot. Here α is simply $((\sqrt{5} - 1)/2)$.

The bound on element growth with this modification is the same, but the decomposition now requires $4n + p$ multiplications and $3n/2 + 3p/2$ comparisons.

Bunch's original algorithm can be modified slightly to obtain an operation count closer to that of Gaussian elimination when solving linear equations. The modification involves realizing that one need not construct the MDM^t decomposition explicitly but only that part of the decomposition which is useful in solving linear equations.

To solve $Ax = b$, one solves $Mc = b$ for c , $Dy = c$ for y and $M^t x = y$ for x . Let us assume that the first block of D is 2×2 and hence one may write

$$y_2 = (\beta c_2 - c_1)/\delta \quad \text{and} \quad y_1 = (c_2 - D_{22}y_2)/D_{21},$$

where $\beta = D_{11}/D_{21}$ and $\delta = D_{22}\beta - D_{21}$. The quantities β and δ are also needed in the decomposition phase since $M_{31} = -T_{32}/\delta$ and $M_{32} = \beta M_{31}$. To decrease the operation count we suggest saving β in place of D_{11} and storing δ separately.

The computation requirements for this modified algorithm is given in the second column of Table 4.1.

4.3 Five-Diagonal Matrices. In this section we consider two methods for a symmetric indefinite five-diagonal matrix F , i.e. $F_{ij} = 0$ for $|i - j| > 2$. Such a matrix arises during the solution of partial differential equations with periodic boundary conditions. As in the case of tridiagonal matrices, Gaussian elimination with partial pivoting is still the least time-consuming stable algorithm for solving $Fx = b$, but it destroys symmetry. In this section we describe two symmetry-preserving algorithms, E and F, which, for an irreducible matrix F , determine matrices P , M , and D such that

$$(4.1) \quad PFP^t = MDM^t$$

as in (1.1). Here $M_{ij} = 0$ for $i > j + 3$. With decomposition (4.1) one can solve $Fx = b$ with less storage than Gaussian elimination with partial pivoting but with a slightly higher operation count.

The algorithms follow the ideas used in Sections 2 and 3 and generate a sequence of five-diagonal matrices $F^{(k)}$ of order k . They were designed so that the bound on the element growth of $F^{(k)}$ is independent of k and the operation count is kept low.

Both algorithms have the same bound on element growth. The bound on the operation count for Algorithm E is slightly higher than that of Algorithm F, but in Algorithm E the probability of attaining the bound is less.

The first step of each algorithm is typical.

Algorithm E. (1) If $|F_{21}| \geq |F_{31}|$, then let $\sigma = \max(|F_{21}|, |F_{32}|, |F_{42}|)$.

(a) If $\sigma|F_{11}| \geq \alpha|F_{21}|^2$, generate $F^{(n-1)}$ using a 1×1 pivot.

(b) If $|F_{22}| \geq \sigma$, then interchange the first and second rows and columns of F and perform a 1×1 pivot on the new F to generate $F^{(n-1)}$.

TABLE 4.2. Operation Count: Five-Diagonal Matrices

	Algorithm E		Algorithm F		Gaussian Elimination with Partial Pivoting	
	Decomposition Only	Solving $Fx = b$	Decomposition Only	Solving $Fx = b$	Decomposition Only	Solving $Fx = b$
Multiplications	$\frac{23}{2}n + \frac{1}{2}p$	$\frac{39}{2}n - \frac{1}{2}p$	$\frac{23}{2}n - \frac{3}{2}p$	$\frac{39}{2}n - \frac{5}{2}p$	$10n$	$17n$
Additions	$\frac{11}{2}n - \frac{1}{2}p$	$\frac{25}{2}n - \frac{3}{2}p$	$\frac{11}{2}n - \frac{1}{2}p$	$\frac{25}{2}n - \frac{3}{2}p$	$8n$	$14n$
Comparisons	$\frac{5}{2}(n + p)$	$\frac{5}{2}(n + p)$	$3(n + p)$	$3(n + p)$	$2n$	$2n$
Storage	$4n$	$5n$	$4n$	$5n$	$8n$	$9n$

- (c) Use a 2×2 pivot to generate $F^{(n-2)}$.
- (2) If $|F_{21}| < |F_{31}|$, then let $\sigma = \max_{2 \leq i \leq 5} |F_{i3}|$.
 - (a) If $\sigma |F_{11}| \geq \alpha |F_{31}|^2$, then perform a 1×1 pivot to generate $F^{(n-1)}$.
 - (b) Interchange the second and third rows and columns of F and perform a 2×2 pivot step to generate $F^{(n-2)}$.

Algorithm F. (1) If $|F_{21}| \geq |F_{31}|$, then

- (a) if $|F_{11}| \geq \alpha |F_{21}|$, then generate $F^{(n-1)}$ using a 1×1 pivot.
- (b) Let $\sigma = \max(|F_{21}|, |F_{32}|, |F_{42}|)$.
 If $|F_{22}| \geq \sigma$, then interchange the first and second rows and columns of F and perform a 1×1 pivot on the new F .
- (c) If $\sigma |F_{11}| \geq \alpha |F_{21}|^2$, generate $F^{(n-1)}$ using a 1×1 pivot.
- (d) Use a 2×2 pivot to generate $F^{(n-2)}$.
- (2) If $|F_{21}| < |F_{31}|$ then do the same as (2) in Algorithm E.

Step (1b) in each algorithm was included to ensure that the bound on $|F_{22}^{(k)}|$ would be independent of k . In Algorithm F step (1a) was included so that the bound on $|F_{11}^{(k)}|$ would be independent of k .

The bound on element growth for each algorithm is minimized for $\alpha = 0.52542$ (see [4]), and then $|F_{ij}^{(k)}| \leq 23.88 \max_{1 \leq i, j \leq n} |F_{ij}|$.

The operation count for each algorithm is largest when rows and columns are interchanged. The bound on the operation count is slightly higher for Algorithm E since more checking is done before the algorithm concedes that one must interchange rows and columns before performing a 1×1 pivot. But because of the extra checks, the bound will not be attained as often as it would be in Algorithm F.

In Table 4.2, p is the number of 1×1 pivots. If storage is crucial, Algorithm E or F should be used rather than Gaussian elimination with partial pivoting.

For Algorithm F, the bounds on the multiplication, addition, and comparison count cannot all be attained simultaneously. The bounds on multiplications and additions are attained only if all 1×1 pivots are done in (1b) and all the 2×2 's in (2b). In this case at most $5(n + p)/2$ comparisons can be done. The bound on the comparison count is attained only if all 1×1 pivots are done in (1c) and all the 2×2 's in (1d). In this case $9n - 2p$ additions and $15n - p$ multiplications are needed to solve a system of equations.

Appendix. There are several decompositions of symmetric matrices, e.g. symmetric triangular factorization (the LDL^t decomposition) [10], the Cholesky decomposition [10], the diagonal pivoting decomposition [2], [3], [5], the tridiagonal decomposition [1], [9] and the orthogonal decomposition [10]; there are analogous decompositions of Hermitian matrices. The decomposition used depends on the problem to be solved, e.g. solving systems of linear equations, calculating inertia, or finding eigenvalues. In the following, we will, in general, discuss the symmetric case only; the Hermitian case follows by replacing t (transpose) by $*$ (complex conjugate) throughout. (Note that $|x|$ represents absolute value of a real number x in the real symmetric case and modulus of a complex number x in the complex symmetric or Hermitian case.)

When solving systems of linear equations where the coefficient matrix A is nonsingular and symmetric, we may always neglect the symmetry of A and use Gaussian elimination (triangular factorization). This requires $n^3/3$ multiplications, $n^3/3$ additions, $\leq n^2/2$ comparisons, and $n^2 + n$ storage to obtain the triangular factorization of a permutation of A , i.e. $PA = LU$ where L is unit lower triangular, U is upper triangular, and P is a permutation matrix. Thus, if we want to solve $Ax = b$, we solve $Ly = Pb$ for y and then $Ux = y$ for x , each requiring $n^2/2$ multiplications and $n^2/2$ additions.

Can we take advantage of the symmetry of A to solve $Ax = b$ in $n^3/6$ multiplications and $n^3/6$ additions?

If the LU decomposition of A exists when A is symmetric, then $U = DL^t$, where D is diagonal, and $A = LDL^t$ can be computed with $n^3/6$ multiplications, $n^3/6$ additions, and $n^2/2$ storage. However, the LDL^t decomposition of A need not exist, e.g. $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$. In fact, the LDL^t decomposition of PAP^t need not exist for any permutation matrix P , e.g. $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

But if A is also positive definite or negative definite ($x^tAx > 0$, or $x^tAx < 0$, respectively, for all $x \neq 0$), then the LDL^t decomposition of A exists. If A is positive definite, then $D_{ii} > 0$ for each i , and $A = \tilde{L}\tilde{L}^t$ where $\tilde{L} = LD^{1/2}$ and $D^{1/2} = \text{diag}\{\sqrt{D_{11}}, \dots, \sqrt{D_{nn}}\}$; this is the Cholesky decomposition.

If A is real symmetric indefinite (A has at least one positive and one negative eigenvalue, i.e. there exist x and y such that $x^tAx > 0$ and $y^tAy < 0$), then these methods can fail (and can be unstable in finite precision arithmetic [5, pp. 643–645]).

Since an $n \times n$ real symmetric (or complex Hermitian) matrix A has only real eigenvalues [10], let u , v , w be the number of positive, negative, zero eigenvalues, respectively. The triple (u, v, w) is called the *inertia* of A , while $s \equiv u - v$ is called the *signature* of A . But $n = u + v + w$ is the *order* of A and $r \equiv u + v$ is the *rank* of A . Thus, $u = \frac{1}{2}(r + s)$, $v = \frac{1}{2}(r - s)$, and $w = n - r$. Knowing the order, rank, and signature of a real symmetric (or complex Hermitian) matrix A is equivalent to knowing the inertia of A . If A is nonsingular, then $w = 0$ and $r = n$, so knowing the inertia is equivalent to knowing the signature. Note that in the inertia problem we are seeking only the signs of the eigenvalues, not the eigenvalues themselves, and hence we seek some method that would be faster than calculating all the eigenvalues (cf. Cottle [7]). (If A is complex symmetric, then its eigenvalues are not necessarily real, so we do not have the concept of inertia in this case.)

Suppose $A = LDL^t$, where L is unit lower triangular and D is diagonal. We shall show below that u, v, w are equal to the number of positive, negative, and zero elements, respectively, on the diagonal of D . Since it requires only $n^3/6$ multiplications to compute the LDL^t decomposition, this is much less work than calculating the eigenvalues.

The theoretical foundation for calculating inertia is provided by Sylvester's Inertia Theorem [8, pp. 371–372]; the inertia of a real symmetric (or complex Hermitian) matrix is invariant under nonsingular congruences, i.e. if A is real symmetric (or complex Hermitian) and S is nonsingular, then A and SAS^t (SAS^*) have the same inertia, and hence the same rank and signature.

The classical method for calculating the inertia of a real symmetric matrix is based on Lagrange's method for the reduction of a quadratic form to a diagonal form.

If A is an $n \times n$ matrix and x is an n -vector, then we say that $\varphi(x) \equiv x^tAx = \sum_{i,j=1}^n A_{ij}x_i x_j$ is a *quadratic form*. If A is of rank r , then we say that $\varphi(x)$ is a quadratic form of rank r .

Note that $B \equiv \frac{1}{2}(A + A^t)$ is symmetric and $x^tBx = x^tAx$. Hence, without loss of generality, we may assume that A is symmetric.

Lagrange's method (1759) reduces by nonsingular congruence transformations a quadratic form $\varphi(x) = x^tAx$ to a diagonal form z^tDz , where D is a diagonal matrix with exactly $r = \text{rank}(A)$ nonzero elements, i.e. $x^tAx = z^tDz$ where $z = Sx$, $\det S \neq 0$. Since $A = S^tDS$, by Sylvester's Inertia Theorem, A and D have the same inertia.

We shall now relate Lagrange's method to Gaussian elimination. If $A_{11} \neq 0$, then

$$\begin{aligned} \varphi(x) &= x^tAx = \sum_{i,j=1}^n A_{ij}x_i x_j \\ &= A_{11}x_1^2 + 2A_{12}x_1x_2 + \cdots + 2A_{1n}x_1x_n + \sum_{i,j=2}^n A_{ij}x_i x_j \\ &= A_{11} \left(x_1^2 + 2 \frac{A_{12}}{A_{11}} x_1x_2 + \cdots + 2 \frac{A_{1n}}{A_{11}} x_1x_n \right) + \sum_{i,j=2}^n A_{ij}x_i x_j \\ &= A_{11} \left(x_1 + \frac{A_{12}}{A_{11}}x_2 + \cdots + \frac{A_{1n}}{A_{11}}x_n \right)^2 + \sum_{i,j=2}^n \left(A_{ij} - \frac{A_{1i}A_{1j}}{A_{11}} \right) x_i x_j. \end{aligned}$$

Thus, take

$$D_{11} = A_{11} \quad \text{and} \quad z_1 = x_1 + \frac{A_{12}}{A_{11}}x_2 + \cdots + \frac{A_{1n}}{A_{11}}x_n.$$

(This is also called the method of completing the square.)

Note that this is identical to the first step of the LDL^t decomposition of A , i.e. the symmetric form of Gaussian elimination. Let

$$A = \begin{bmatrix} E & C^t \\ C & B \end{bmatrix},$$

where $E = A_{11} \neq 0$, C is $(n - 1) \times 1$ and B is $(n - 1) \times (n - 1)$. Then $A = L_1A_1L_1^t$, where

$$L_1 = \begin{bmatrix} 1 & 0 \\ CE^{-1} & I_{n-1} \end{bmatrix} \text{ and } A_1 = \begin{bmatrix} E & 0 \\ 0 & B - CE^{-1}C^t \end{bmatrix}.$$

Now $z_1 = L_1^t x$ and

$$B - CE^{-1}C^t = \left[A_{ij} - \frac{A_{i1}A_{1j}}{A_{11}} \right].$$

If $A_{22} - A_{21}^2/A_{11} \neq 0$, we may continue with the process.

If $A_{11} = 0$, but $A_{kk} \neq 0$ for some k , then we may interchange the k th and first rows and columns and proceed as before. In matrix form, let P be the permutation matrix obtained from interchanging the k th and first columns of the $n \times n$ identity matrix. Then $\varphi(x) = x^t Ax = x^t P^t PAP^t Px = (Px)^t (PAP^t)(Px)$, where $Px = [x_k, x_2, \dots, x_{k-1}, x_1, x_{k+1}, \dots, x_n]^t$ and $(PAP^t)_{11} = A_{kk}$.

But, what do we do if $A_{11} = \dots = A_{nn} = 0$ (or if at some stage of the process the diagonal of the remaining submatrix is all zero)? If $A \equiv 0$, then the rank of A is zero and we take $D \equiv 0$ and $z = x$. If $A \not\equiv 0$ but $A_{11} = \dots = A_{nn} = 0$, then $A_{rs} \neq 0$ for some $r \neq s$. We can now interchange the r th and first, and the s th and second, rows and columns, and then the $(2, 1)$ element of the resulting matrix is nonzero. Without loss of generality, we may assume $A_{21} \neq 0$.

Let $y_1 = \frac{1}{2}(x_1 + x_2)$, $y_2 = \frac{1}{2}(x_1 - x_2)$, and $y_i = x_i$ for $3 \leq i \leq n$, i.e. $y = T^{-1}x$, where $T = S \oplus I_{n-2}$, $S = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and I_{n-2} is the identity matrix of order $n - 2$. Then $x^t Ax = y^t (T^t AT)y$ with $(T^t AT)_{11} = 2A_{12} \neq 0$. Thus we may proceed as before.

We shall now show that the above is equivalent to performing a block 2×2 elimination. Let

$$A = \begin{bmatrix} E & C^t \\ C & B \end{bmatrix}, \quad T = \begin{bmatrix} S & 0 \\ 0 & I \end{bmatrix},$$

$\det E \neq 0$, and $\text{order}(E) = \text{order}(S)$. Then

$$T^t AT = \begin{bmatrix} S^t ES & S^t C^t \\ CS & B \end{bmatrix}.$$

Lagrange's method is equivalent to choosing S to be of order 2 so that $S^t ES = D$ is diagonal. If we use E as a block pivot in A and perform a block decomposition, then the reduced matrix is $B - CE^{-1}C^t = B - CSD^{-1}S^t C^t$. Thus we need not find a 2×2 matrix S which diagonalizes E , but we can perform a block decomposition with the 2×2 submatrix E . If the diagonal of A is null, then there exists a nonsingular principal 2×2 submatrix unless $A \equiv 0$.

However, the above decomposition also exists for complex symmetric matrices. Hence, given any symmetric matrix A , there exists a permutation matrix P such that $PAP^t = MDM^t$, where M is unit lower triangular, D is block diagonal with blocks of order 1 or 2, and $M_{i+1,i} = 0$ whenever $D_{i+1,i} \neq 0$.

Let us look at the determinant of such a block of order 2:

$$\det \begin{bmatrix} 0 & D_{i,i+1} \\ D_{i+1,i} & 0 \end{bmatrix} = -D_{i+1,i}D_{i,i+1}.$$

If A is real symmetric, then $\det = -D_{i+1,i}^2 < 0$, and if A is complex Hermitian, then

$$\det = -D_{i+1,i} \overline{D_{i+1,i}} = -|D_{i+1,i}|^2 < 0.$$

Hence, by Sylvester's Inertia Theorem, one positive and one negative eigenvalue of A is associated with this 2×2 block.

Let p be the number of 1×1 blocks in D . Hence there are $q = \frac{1}{2}(n - p)$ blocks of order 2. Let a, b, c be the number of positive, negative, and zero 1×1 blocks. Thus the inertia of A is $(a + q, b + q, c)$, the rank of A is $n - c$, and the signature of A is $a - b$.

In finite precision arithmetic on a computer, in order to maintain stability and insure a good solution we must prevent large growth in the elements of the reduced matrices generated during the decomposition process [5], [10]. Hence, we will want to use 2×2 pivots whenever the diagonal is small as well as whenever the diagonal is null [5], [10]. Our knowledge of the inertia will be preserved as long as the determinant of each 2×2 pivot remains negative when A is real symmetric or complex Hermitian.

Based on the above method, called the diagonal pivoting method, Bunch [2] showed that inertia can be calculated and nonsingular real symmetric, complex symmetric, and complex Hermitian systems of linear equations can be solved by only $n^3/6$ multiplications, $n^3/6$ additions, and $n^2/2$ storage. The method is almost as stable as Gaussian elimination with complete pivoting. The price paid for stability is $\geq n^3/12$ but $\leq n^3/6$ comparisons.

Let us consider the first step of the algorithm. Let

$$A = \begin{bmatrix} E & C^t \\ C & B \end{bmatrix},$$

where E is of order $s = 1$ or 2 . Let

$$\mu_0 = \max_{i,j} |A_{ij}|, \quad \mu_1 = \max_i |A_{ii}|, \quad \nu = |\det E|.$$

If $s = 1$, then $\nu = |E|$. Assume $\nu \neq 0$. Then

$$\begin{bmatrix} E & C^t \\ C & B \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ C/E & I_{n-1} \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & A^{(n-1)} \end{bmatrix} \begin{bmatrix} 1 & C^t/E \\ 0 & I_{n-1} \end{bmatrix},$$

where $A^{(n-1)} = B - CC^t/E$ and I_{n-1} is the identity matrix of order $n - 1$. So,

$$\max_{i,j} |A_{ij}^{(n-1)}| \leq \mu_0 + \mu_0^2/\nu.$$

Thus large element growth will not occur for a 1×1 pivot if $\nu = |E|$ is large relative to μ_0 .

If E is of order 2 and $\nu \neq 0$, then

$$\begin{bmatrix} E & C^t \\ C & B \end{bmatrix} = \begin{bmatrix} I_2 & 0 \\ CE^{-1} & I_{n-2} \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & A^{(n-2)} \end{bmatrix} \begin{bmatrix} I_2 & E^{-1}C^t \\ 0 & I_{n-2} \end{bmatrix},$$

where $A^{(n-2)} = B - CE^{-1}C^t$. So,

$$\max_{i,j} |A_{ij}^{(n-2)}| \leq \left[1 + \frac{2\mu_0(\mu_0 + \mu_1)}{\nu} \right] \mu_0.$$

Let α be a fixed number with $0 < \alpha < 1$. We shall use a 1×1 pivot if $\mu_1 \geq \alpha\mu_0$. If so, we interchange the 1st and k th rows and columns, where $\mu_1 = \max_i |A_{ii}| = |A_{kk}|$. Without loss of generality, we may assume $|A_{11}| = \mu_1$. Hence, $\nu = \mu_1$ and $\max_{i,j} |A_{ij}^{(n-1)}| \leq (1 + 1/\alpha)\mu_0$.

If $\mu_1 < \alpha\mu_0$, where $\mu_0 = |A_{rq}|$ for $r \neq q$, then interchange the r th and 2nd rows and columns and the q th and 1st rows and columns, and use a 2×2 pivot. Without loss of generality, we may assume $\mu_0 = |A_{21}|$. Then,

$$\begin{aligned} \nu &= |A_{11}A_{22} - A_{21}A_{12}| \geq |A_{21}||A_{12}| - |A_{11}||A_{22}| \\ &= \mu_0^2 - |A_{11}||A_{22}| \geq \mu_0^2 - \mu_1^2 > (1 - \alpha^2)\mu_0^2 \end{aligned}$$

and

$$\max_{i,j} |A_{ij}^{(n-2)}| \leq [1 + 2/(1 - \alpha)]\mu_0.$$

(Note that this holds whether A is real symmetric, complex symmetric, or complex Hermitian.)

Thus, all the elements in all the reduced matrices are bounded by $B(\alpha)^{n-1}$, where $B(\alpha) = \max\{1 + 1/\alpha, [1 + 2/(1 - \alpha)]^{1/2}\}$.

Now

$$\min_{0 < \alpha < 1} B(\alpha) = B(\alpha_0) = (1 + \sqrt{17})/2 < 2.57,$$

where $\alpha_0 = (1 + \sqrt{17})/8 \doteq 0.6404$.

Since we must search for the largest element in each reduced matrix, this is a complete pivoting strategy analogous to Gaussian elimination with complete pivoting.

Furthermore, Bunch [2] proves that the element growth in the diagonal pivoting method with complete pivoting is bounded by $3nf(n)$ in comparison with $\sqrt{nf(n)}$ for Gaussian elimination with complete pivoting, where

$$f(n) = \left(\prod_{k=2}^n k^{1/(k-1)} \right)^{1/2} < 1.8n^{(1/4)\log n}.$$

Acknowledgement. We would like to thank B. N. Parlett and the referee for their helpful suggestions.

Department of Mathematics
University of California at San Diego
La Jolla, California 92093

Department of Computer Science
University of Colorado
Boulder, Colorado 80302

2. J. R. BUNCH, "Analysis of the diagonal pivoting method," *SIAM J. Numer. Anal.*, v. 8, 1971, pp. 656–680. MR 45 #1367.
3. J. R. BUNCH, "Partial pivoting strategies for symmetric matrices," *SIAM J. Numer. Anal.*, v. 11, 1974, pp. 521–528. MR 50 #15294.
4. J. R. BUNCH & L. KAUFMAN, "Some stable methods for calculating inertia and solving symmetric linear systems," Univ. of Colorado Tech. Report 63, CU:CS:06375.
5. J. R. BUNCH & B. N. PARLETT, "Direct methods for solving symmetric indefinite systems of linear equations," *SIAM J. Numer. Anal.*, v. 8, 1971, pp. 639–655. MR 46 #4694.
6. P. A. BUSINGER, "Monitoring the numerical stability of Gaussian elimination," *Numer. Math.*, v. 16, 1971, pp. 360–361.
7. R. W. COTTLE, "Manifestations of the Schur complement," *Linear Algebra and Appl.*, v. 8, 1974, pp. 189–211.
8. L. MIRSKY, *An Introduction to Linear Algebra*, Clarendon Press, Oxford, 1955. MR 17, 573.
9. B. N. PARLETT & J. K. REID, "On the solution of a system of linear equations whose matrix is symmetric but not definite," *BIT*, v. 10, 1970, pp. 386–397.
10. J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, 1965. MR 32 #1894.