Block LU Factorization*

James W. Demmel[†] Nicholas J. Higham[‡] Robert S. Schreiber[§]

February 16, 1992

Abstract. Many of the currently popular "block algorithms" are scalar algorithms in which the operations have been grouped and reordered into matrix operations. One genuine block algorithm in practical use is block LU factorization, and this has recently been shown by Demmel and Higham to be unstable in general. It is shown here that block LU factorization is stable if A is block diagonally dominant by columns. Moreover, for a general matrix the level of instability in block LU factorization can be bounded in terms of the condition number $\kappa(A)$ and the growth factor for Gaussian elimination without pivoting. A consequence is that block LU factorization is stable for a matrix A that is symmetric positive definite or point diagonally dominant by rows or columns as long as A is well-conditioned.

Key words: block algorithm, LAPACK, level 3 BLAS, iterative refinement, LU factorization, backward error analysis, block diagonal dominance.

AMS(MOS) subject classifications. primary 65F05, 65F25, 65G05.

1 Introduction

Block methods in matrix computations are widely recognised as being able to achieve high performance on modern vector and parallel computers. Their performance benefits have been investigated by various authors over the last decade (see, for example, [11, 14, 15]), and in particular by the developers of LAPACK [1]. The rise to prominence of block methods has been accompanied by the development of the level 3 Basic Linear Algebra Subprograms (BLAS3)—a set of specifications of

*This work was completed while the first two authors were visitors at the Institute for Mathematics and Its Applications, University of Minnesota.

†Computer Science Division and Mathematics Department, University of California, Berkeley, CA 94720, U.S.A. (na.demmel@na-net.ornl.gov). This work was supported in part by NSF grant ASC-9005933 and DARPA grant DAAL03-91-C-0047 via a subcontract from the University of Tennessee.

‡Nuffield Science Research Fellow. Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK (na.nhigham@na-net.ornl.gov).

§RIACS, NASA Ames Research Center, Moffett Field, CA 94035, U.S.A. (na.schreiber@na-net.ornl.gov). This author acknowledges the financial support of the Numerical Aerodynamic Simulation Systems Division of NASA via Cooperative Agreement NCC 2-387 between NASA and the Universities Space Research Association.

Fortran primitives for various types of matrix multiplication, together with solution of a triangular system with multiple right-hand sides [9, 10]. Block algorithms can be cast largely in terms of calls to the BLAS3, and it is by working with these matrix-matrix operations that they achieve high performance. (For a detailed explanation of why matrix-matrix operations lead to high efficiency see [8] or [16].)

While the performance aspects of block algorithms have been thoroughly analyzed, numerical stability issues have received relatively little attention. This is perhaps not surprising, because most block algorithms in practical use automatically have excellent numerical stability properties. Indeed, Demmel and Higham [7] show that all the block algorithms in LAPACK are as stable as their point counterparts. However, stability cannot be taken for granted. LAPACK includes a block algorithm for inverting a triangular matrix that is a generalization of a standard point algorithm. During the development of LAPACK another, equally plausible block generalization was considered—this one was found to be unstable [12].

In this work we investigate the numerical stability of a block form of the most important of all matrix factorizations, LU factorization. What we mean by "block form" needs to be explained carefully, since the adjective "block" has more than one meaning in the literature. We will use the following terminology, which emphasises an important distinction and leads to insight in interpreting stability results.

A *partitioned algorithm* is a scalar (or point) algorithm in which the operations have been grouped and reordered into matrix operations. The partitioned form may involve some extra operations over the scalar form (as is the case with algorithms that aggregate Householder transformations using the WY technique of [4]).

A *block algorithm* is a generalization of a scalar algorithm in which the basic scalar operations become matrix operations ($\alpha + \beta$, $\alpha\beta$, α/β become $A + B$, AB and AB^{-1}), and a matrix property based on the nonzero structure becomes the corresponding property blockwise (in particular, the scalars 0 and 1 become the zero matrix and the identity matrix, respectively). A *block factorization* is defined in a similar way, and is usually what a block algorithm computes.

The distinction between a partitioned algorithm and a block algorithm is rarely made in the literature (an exception is the paper [24]). The term "block algorithm" is frequently used to describe both types of algorithm. A partitioned algorithm might also be called a "blocked algorithm" (as is done in [8]), but the similarity to "block algorithm" can cause confusion and so we do not recommend this terminology. Note that in the particular case of matrix multiplication partitioned and block algorithms are equivalent.

LAPACK contains only partitioned algorithms. A possible exception is the multi-shift Hessenberg QR iteration [2], which could be regarded a block algorithm, even though it does not work with a block Hessenberg form. As this example indicates, not all algorithms fit neatly into one class or the other, so our definitions should not be interpreted too strictly.

Block LU factorization is one of the few block factorizations in practical use. It takes the form

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} I & & \\ L_{21} & I & \\ L_{31} & L_{32} & I \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ & U_{22} & U_{23} \\ & & U_{33} \end{bmatrix} = LU, \tag{1.1}$$

where, for illustration, we are regarding A as a block 3×3 matrix. L is block lower triangular with identity matrices on the diagonal (and hence is lower triangular), and U is block upper triangular (but the diagonal blocks U_{ii} are not triangular, in general).

Block LU factorization has been discussed by various authors; see, for example, [5, 15, 23, 24]. It appears to have first been proposed for block tridiagonal matrices, which frequently arise in the discretization of partial differential equations [16, Sec. 4.5.1], [21, p. 59], [22], [26]. An attraction of block LU factorization is that one particular implementation has a greater amount of matrix multiplication than conventional LU factorization (see section 2), and this is likely to make it more efficient on high-performance computers.

By contrast with (1.1), a standard LU factorization can be written in the form

$$A = \begin{bmatrix} \overline{L}_{11} & & \\ \overline{L}_{21} & \overline{L}_{22} & \\ \overline{L}_{31} & \overline{L}_{32} & \overline{L}_{33} \end{bmatrix} \begin{bmatrix} \overline{U}_{11} & \overline{U}_{12} & \overline{U}_{13} \\ & \overline{U}_{22} & \overline{U}_{23} \\ & & \overline{U}_{33} \end{bmatrix} = \overline{L}\,\overline{U},$$

where $\overline{L}$ is unit lower triangular and $\overline{U}$ is upper triangular. A partitioned version of the outer product LU factorization algorithm (without pivoting) computes the first block column of $\overline{L}$ and the first block row of $\overline{U}$ as follows. $A_{11} = \overline{L}_{11}\overline{U}_{11}$ is computed as a point LU factorization, and the equations $\overline{L}_{i1}\overline{U}_{11} = A_{i1}$ and $\overline{L}_{11}\overline{U}_{1i} = A_{1i}$ are solved for $\overline{L}_{i1}$ and $\overline{U}_{1i}$, $1 = 2, 3$. The process is repeated on the Schur complement,

$$S = \begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix} - \begin{bmatrix} \overline{L}_{21} \\ \overline{L}_{31} \end{bmatrix} [\,\overline{U}_{12} \quad \overline{U}_{13}\,].$$

This algorithm does the same arithmetic operations as any other version of standard LU factorization, but in a different order.

Demmel and Higham [7] have recently shown that block LU factorization can be unstable, even when A is symmetric positive definite or diagonally dominant by rows. This instability had previously been identified and analysed in [3] in the special case where A is a particular row permutation of a symmetric positive definite block tridiagonal matrix. The purpose of this work is to gain further insight into the instability of block LU factorization. We also wish to emphasise that of the two classes of algorithms we have defined it is the block algorithms whose stability is most in question. We know of no examples of an unstable partitioned algorithm. (Those partitioned algorithms based on the aggregation of Householder transformations that do slightly different arithmetic to the point versions have been shown to be stable [4, 7]).

In section 2 we derive backward error bounds for block LU factorization and for the solution of a linear system $Ax = b$ using the block LU factors. In section 3 we show that block LU factorization is stable if A is block diagonally dominant by columns; this generalizes the known results that Gaussian elimination without pivoting is stable for column diagonally dominant matrices [28] and that block LU factorization is stable for block tridiagonal matrices that are block diagonally dominant by columns [26]. We also show that for a general matrix A the backward error is bounded by a product involving $\kappa(A)$ and the growth factor ρ_n for Gaussian elimination without pivoting on A. If A is (point) diagonally dominant this bound simplifies because $\rho_n \leq 2$. If A is diagonally dominant by columns we show that a potentially much smaller bound holds that depends only on the block size. In section 4 we specialize to symmetric positive definite matrices and show that the backward error can be bounded by a multiple of $\kappa_2(A)^{1/2}$. Block LU factorization is thus *conditionally stable* for symmetric positive definite and diagonally dominant matrices: it is guaranteed to be stable only if A is well-conditioned. Results of this type are rare for linear equation solvers based on factorization methods, although stability results conditional on other functions of A do hold for certain iterative linear equation solvers [20, 29].

In section 5 we present some numerical experiments that show our error bounds to be reasonably sharp and reveal some interesting numerical behaviour. Concluding remarks are given in section 6.

2 Error Analysis of Block LU factorization

We consider a block LU factorization $A = LU \in \mathbb{R}^{n\times n}$, where the diagonal blocks in the partitioning are square but do not necessarily all have the same dimension.

If $A_{11} \in \mathbb{R}^{r\times r}$ is nonsingular we can write

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & S \end{bmatrix}, \tag{2.1}$$

which describes one block step of an outer product based algorithm for computing a block LU factorization. Here, $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ is a Schur complement of A. If the $(1,1)$ block of S of appropriate dimension is nonsingular then we can factorize S in a similar manner, and this process can be continued recursively to obtain the complete block LU factorization. The overall algorithm can be expressed as follows.

Algorithm BLU.
This algorithm computes a block LU factorization $A = LU \in \mathbb{R}^{n\times n}$.
1. $U_{11} = A_{11}$, $U_{12} = A_{12}$.
2. Solve $L_{21}A_{11} = A_{21}$ for L_{21}.
3. $S = A_{22} - L_{21}A_{12}$ (Schur complement).
4. Compute the block LU factorization of S, recursively.

Given the block LU factorization of A, the solution to a system $Ax = b$ can be obtained by solving $Lx = y$ by forward substitution (since L is triangular) and solving $Ux = y$ by block back substitution. There is freedom in how step 2 of Algorithm BLU is accomplished, and how the linear systems with coefficient matrices U_{ii} that arise in the block back substitution are solved. The two main possibilities are as follows.

Implementation 1: A_{11} is factorized by Gaussian elimination with partial pivoting (GEPP). Step 2 and the solution of linear systems with U_{ii} are accomplished by substitution with the LU factors of A_{11}.

Implementation 2: A_{11}^{-1} is computed explicitly, so that step 2 becomes a matrix multiplication and $Ux = y$ is solved entirely by matrix-vector multiplications. This approach is attractive for parallel machines [15, 24].

We now give an error analysis for Algorithm BLU, under the following model of floating point arithmetic, where u is the unit roundoff:

$$\begin{aligned} fl(x \pm y) &= x(1+\alpha) \pm y(1+\beta), \qquad |\alpha|, |\beta| \le u, \\ fl(x \operatorname{op} y) &= (x \operatorname{op} y)(1+\delta), \qquad |\delta| \le u, \quad \operatorname{op} = *, /. \end{aligned}$$

It is convenient to use the matrix norm defined by

$$\|A\| = \max_{i,j} |a_{ij}|. \tag{2.2}$$

Note that if $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$ then $\|AB\| \le n\|A\|\|B\|$ is the best such bound; this inequality affects some of the constants in our analysis and will be used without comment.

We assume that the computed matrices $\widehat{L}_{21}$ from step 2 of Algorithm BLU satisfy

$$\widehat{L}_{21}A_{11} = A_{21} + E_{21}, \qquad \|E_{21}\| \le c_n u \|\widehat{L}_{21}\|\|A_{11}\| + O(u^2), \tag{2.3}$$

where c_n denotes a constant depending on n (we are not concerned with the precise values of the constants in this analysis). We also assume that when a system $U_{ii}x_i = d_i$ is solved, the computed solution $\widehat{x}_i$ satisfies

$$(U_{ii} + \Delta U_{ii})\widehat{x}_i = d_i, \qquad \|\Delta U_{ii}\| \le c_n' u \|U_{ii}\| + O(u^2). \tag{2.4}$$

The assumptions (2.3) and (2.4) are satisfied for implementation 1 and are sufficient to prove the following result.

Theorem 2.1 *Let $\widehat{L}$ and $\widehat{U}$ be the computed block LU factors of $A \in \mathbb{R}^{n\times n}$ from Algorithm BLU, and let $\widehat{x}$ be the computed solution to $Ax = b$. Under assumptions (2.3) and (2.4),*

$$\begin{aligned} \widehat{L}\widehat{U} &= A + E, \qquad \|E\| \le d_n u(\|A\| + \|\widehat{L}\|\|\widehat{U}\|) + O(u^2), & (2.5) \\ (A + \Delta A)\widehat{x} &= b, \qquad \|\Delta A\| \le d_n' u(\|A\| + \|\widehat{L}\|\|\widehat{U}\|) + O(u^2). & (2.6) \end{aligned}$$

Proof. Standard error analysis for matrix multiplication [16, p. 66] shows that in step 3 of the first block stage of the factorization,

$$\widehat{S} = A_{22} - \widehat{L}_{21}A_{12} + \Delta S, \qquad \|\Delta S\| \le c_n u(\|A_{22}\| + \|\widehat{L}_{21}\|\|A_{12}\|) + O(u^2).$$

The remaining stages of the factorization compute the block LU factorization $\widehat{S} \approx \widehat{L}_S\widehat{U}_S$, which, inductively, we can assume satisfies

$$\widehat{L}_S\widehat{U}_S = \widehat{S} + E_S, \qquad \|E_S\| \le c_n' u(\|\widehat{S}\| + \|\widehat{L}_S\|\|\widehat{U}_S\|) + O(u^2).$$

Using (2.3) we have

$$\begin{aligned} A - \widehat{L}\widehat{U} &= A - \begin{bmatrix} I_r & 0 \\ \widehat{L}_{21} & \widehat{L}_S \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & \widehat{U}_S \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 \\ -E_{21} & -\Delta S - E_S \end{bmatrix}, \end{aligned}$$

and so

$$\begin{aligned} \|A - \widehat{L}\widehat{U}\| &\le c_n'' u \max\Big(\|\widehat{L}_{21}\|\|A_{11}\|, \|A_{22}\| + \|\widehat{L}_{21}\|\|A_{12}\| + \|\widehat{S}\| + \|\widehat{L}_S\|\|\widehat{U}_S\|\Big) + O(u^2) \\ &\le d_n u(\|A\| + \|\widehat{L}\|\|\widehat{U}\|) + O(u^2). \end{aligned}$$

The system $Ax = b$ is solved via $\widehat{L}y = b$ and $\widehat{U}x = y$. Since $\widehat{L}$ is triangular we have from a standard result [16, sec. 3.1] that

$$(\widehat{L} + \Delta\widehat{L})\widehat{y} = b, \qquad \|\Delta\widehat{L}\| \le c_n u\|\widehat{L}\| + O(u^2). \tag{2.7}$$

For $\widehat{U}x = \widehat{y}$ consider the first block row, which can be written

$$\widehat{U}_{11}x_1 = \widehat{y} - \widehat{U}_{12}x_2.$$

In the last stage of block back substitution x_2 is known and this equation is solved for x_1. Accounting for the error in forming the right-hand side, and invoking (2.4), we have

$$(\widehat{U}_{11} + \Delta\widehat{U}_{11})\widehat{x}_1 = \widehat{y} - (\widehat{U}_{12} + \Delta\widehat{U}_{12})\widehat{x}_2, \qquad \|\Delta\widehat{U}_{1j}\| \le c_n' u\|\widehat{U}_{1j}\| + O(u^2).$$

Since analogous equations hold for all the block rows, we have, overall,

$$(\widehat{U} + \Delta\widehat{U})\widehat{x} = \widehat{y}, \qquad \|\Delta\widehat{U}\| \le c_n' u\|\widehat{U}\| + O(u^2). \tag{2.8}$$

Combining (2.5), (2.7) and (2.8) we have

$$\begin{aligned} b &= (\widehat{L} + \Delta\widehat{L})(\widehat{U} + \Delta\widehat{U})\widehat{x} \\ &= (A + E + \Delta\widehat{L}\widehat{U} + \widehat{L}\Delta\widehat{U} + \Delta\widehat{L}\Delta\widehat{U})\widehat{x} \\ &= (A + \Delta A)\widehat{x}, \end{aligned}$$

and $\|\Delta A\|$ is bounded as in (2.6). ∎

Theorem 2.1 shows that the stability of block LU factorization is determined by the ratio $\|\widehat{L}\|\|\widehat{U}\|/\|A\|$ (the sharpness of the bounds is demonstrated in the numerical experiments of section 5). If this ratio is reasonably bounded, by a modest function of n, say, then $\widehat{L}$ and $\widehat{U}$ are the true factors of a matrix close to A, and $\widehat{x}$ solves a slightly perturbed system. It was noted in [7] that $\|\widehat{L}\|\|\widehat{U}\|$ can exceed $\|A\|$ by an arbitrary factor, even if A is symmetric positive definite or diagonally dominant by rows. Indeed, $\|L\| \geq \|L_{21}\| = \|A_{21}A_{11}^{-1}\|$, using the partitioning (2.1), and this lower bound for $\|L\|$ can be arbitrarily large. In the following two sections we investigate this instability more closely and show that $\|L\|\|U\|$ can be bounded in a useful way for particular classes of A. Without further comment we make the reasonable assumption that $\|L\|\|U\| \approx \|\widehat{L}\|\|\widehat{U}\|$, so that these bounds may be used in Theorem 2.1.

We mention that the bounds in Theorem 2.1 are valid also for other version of block LU factorization obtained by "block loop reordering", such as a block gaxpy based algorithm [16, p. 101].

Finally, we comment on implementation 2. Suppose, for simplicity, that the inverses A_{11}^{-1} (which are used in step 2 of Algorithm BLU and in the block back substitution) are computed exactly. Then the best bounds of the forms (2.3) and (2.4) are

$$\begin{aligned} \widehat{L}_{21}A_{11} &= A_{21} + E_{21}, \qquad \|E_{21}\| \leq c_n u\kappa(A_{11})\|A_{21}\| + O(u^2), \\ (U_{ii} + \Delta U_{ii})\widehat{x}_i &= d_i, \qquad \|\Delta U_{ii}\| \leq c_n' u\kappa(U_{ii})\|U_{ii}\| + O(u^2). \end{aligned}$$

Working from these results, we find that Theorem 2.1 still holds provided the first order terms in the bounds in (2.5) and (2.6) are multiplied by $\max_i \kappa(\widehat{U}_{ii})$. This suggests that implementation 2 of Algorithm BLU can be much less stable than implementation 1 when the diagonal blocks of U are ill-conditioned, and this is confirmed by the numerical results in section 5.

3 Diagonal Dominance

One class of matrices for which block LU factorization has long been known to be stable is block tridiagonal matrices that are block diagonally dominant. A general matrix $A \in \mathbb{R}^{n\times n}$ is block diagonally dominant by columns, with respect to a given partitioning $A = (A_{ij})$ and a given norm, if, for all j,

$$\|A_{jj}^{-1}\|^{-1} - \sum_{i\neq j}\|A_{ij}\| = \gamma_j \geq 0. \tag{3.1}$$

A is block diagonally dominant by rows if A^T is block diagonally dominant by columns. For the block size 1 the usual property of point diagonal dominance is obtained. Note that for the 1 and ∞-norms diagonal dominance does not imply block diagonal dominance, nor does the reverse implication hold.

Block diagonal dominance was introduced in [13], and has mostly found use in generalizations of the Gershgorin circle theorem. However, Varah [26] proved that if A is block tridiagonal and has the block LU factorization $A = LU$ (so that L and U are block bidiagonal and $U_{i,i+1} = A_{i,i+1}$), then if A is block diagonally dominant by columns

$$\|L_{i,i-1}\| \leq 1, \qquad \|U_{ii}\| \leq \|A_{ii}\| + \|A_{i-1,i}\|, \tag{3.2}$$

while if A is block diagonally dominant by rows

$$\|L_{i,i-1}\| \leq \frac{\|A_{i,i-1}\|}{\|A_{i-1,i}\|}, \qquad \|U_{ii}\| \leq \|A_{ii}\| + \|A_{i,i-1}\|. \tag{3.3}$$

Here, the norm is assumed to be a subordinate matrix norm. For the ∞-norm the inequalities (3.2) imply that $\|L\|_\infty \le 2$ and $\|U\|_\infty \le 3\|A\|_\infty$, so block LU factorization is stable if A is block diagonally dominant by columns. Similarly, if A is block diagonally dominant by rows we have stability if $\|A_{i,i-1}\|/\|A_{i-1,i}\|$ is suitably bounded for all i.

Varah's results can be extended to full, block diagonally dominant matrices, as we now explain. First, we show that for such matrices a block LU factorization exists, using the key property that block diagonal dominance is inherited by the Schur complements obtained in the course of the factorization. In the following analysis we assume that A has m block rows and columns.

Lemma 3.1 *Suppose $A \in \mathbb{R}^{n\times n}$ is nonsingular and block diagonally dominant by rows or columns with respect to a subordinate matrix norm in (3.1). Then A has a block LU factorization, and all the Schur complements arising in Algorithm BLU have the same kind of diagonal dominance as A.*

Proof. The proof is a generalization of the corresponding result for point diagonal dominance [16, p. 20], [28]. We consider the case of block diagonal dominance by columns; the proof for row-wise diagonal dominance is analogous.

Let

$$A^{(2)} = \begin{bmatrix} U_{11} & U_{12} \\ 0 & S \end{bmatrix}$$

denote the matrix obtained from A after one step of Algorithm BLU. For $2 \le j \le n$ we have

$$\begin{aligned}
\sum_{\substack{i=2\\ i\ne j}}^m \|A_{ij}^{(2)}\| &= \sum_{\substack{i=2\\ i\ne j}}^m \|A_{ij} - A_{i1}A_{11}^{-1}A_{1j}\| \\
&\le \sum_{\substack{i=2\\ i\ne j}}^m \|A_{ij}\| + \|A_{1j}\|\|A_{11}^{-1}\| \sum_{\substack{i=2\\ i\ne j}}^m \|A_{i1}\| \\
&\le \sum_{\substack{i=2\\ i\ne j}}^m \|A_{ij}\| + \|A_{1j}\|\|A_{11}^{-1}\| \Big(\|A_{11}^{-1}\|^{-1} - \|A_{j1}\| \Big), \quad \text{using (3.1)}, \\
&= \sum_{\substack{i=2\\ i\ne j}}^m \|A_{ij}\| + \|A_{1j}\| - \|A_{1j}\|\|A_{11}^{-1}\|\|A_{j1}\| \\
&\le \|A_{jj}^{-1}\|^{-1} - \|A_{1j}\|\|A_{11}^{-1}\|\|A_{j1}\|, \quad \text{using (3.1)}, \\
&= \min_{\|x\|=1} \|A_{jj}x\| - \|A_{1j}\|\|A_{11}^{-1}\|\|A_{j1}\| \\
&\le \min_{\|x\|=1} \|(A_{jj} - A_{j1}A_{11}^{-1}A_{1j})x\| \\
&= \min_{\|x\|=1} \|A_{jj}^{(2)}x\|. \qquad (3.4)
\end{aligned}$$

Now if $A_{jj}^{(2)}$ is singular it follows that $\sum_{i=2, i\ne j}^m \|A_{ij}^{(2)}\| = 0$; therefore $A^{(2)}$, and hence also A, is singular, which is a contradiction. Thus $A_{jj}^{(2)}$ is nonsingular, and (3.4) can be rewritten

$$\sum_{\substack{i=2\\ i\ne j}}^m \|A_{ij}^{(2)}\| \le \|A_{jj}^{(2)^{-1}}\|^{-1},$$

showing that $A^{(2)}$ is block diagonally dominant by columns. The result follows by induction. ∎

The next result allows us to bound $\|U\|$ for a block diagonally dominant matrix.

Lemma 3.2 *Let A satisfy the conditions of Lemma 3.1. If $A^{(k)}$ denotes the matrix obtained after $k-1$ steps of Algorithm BLU, then*

$$\max_{k\le i,j\le m} \|A_{ij}^{(k)}\| \le 2 \max_{1\le i,j\le m} \|A_{ij}\|.$$

Proof. The proof is a straightforward generalization of Wilkinson's proof of the corresponding result for point diagonally dominant matrices [28, pp. 288–289]. Let A be block diagonally dominant by columns (the proof for row diagonal dominance is similar). Then

$$\begin{aligned}\sum_{i=2}^{m} \|A_{ij}^{(2)}\| &= \sum_{i=2}^{m} \|A_{ij} - A_{i1}A_{11}^{-1}A_{1j}\| \\ &\le \sum_{i=2}^{m} \|A_{ij}\| + \|A_{1j}\|\|A_{11}^{-1}\| \sum_{i=2}^{m} \|A_{i1}\| \\ &\le \sum_{i=1}^{m} \|A_{ij}\|,\end{aligned}$$

using (3.1). By induction, using Lemma 3.1, it follows that $\sum_{i=k}^{m} \|A_{ij}^{(k)}\| \le \sum_{i=1}^{m} \|A_{ij}\|$. This yields

$$\max_{k\le i,j\le m} \|A_{ij}^{(k)}\| \le \max_{k\le j\le m} \sum_{i=k}^{m} \|A_{ij}^{(k)}\| \le \max_{k\le j\le m} \sum_{i=1}^{m} \|A_{ij}\|.$$

¿From (3.1), $\sum_{i\ne j} \|A_{ij}\| \le \|A_{jj}^{-1}\|^{-1} \le \|A_{jj}\|$, so

$$\max_{k\le i,j\le m} \|A_{ij}^{(k)}\| \le 2 \max_{k\le j\le m} \|A_{jj}\| \le 2 \max_{1\le j\le m} \|A_{jj}\| = 2 \max_{1\le i,j\le m} \|A_{ij}\|.$$

∎

The implications of Lemmas 3.1 and 3.2 for stability are as follows. Suppose A is block diagonally dominant by columns. Also, assume that the norm has the property that

$$\max_{i,j} \|A_{ij}\| \le \|A\| \le \sum_{i,j} \|A_{ij}\|, \tag{3.5}$$

which holds for any p-norm, for example. Then Lemma 3.1 implies that $\|[L_{j+1,j}^T, \ldots, L_{mj}^T]^T\| \le 1$ for each subdiagonal block column of L, and since $U_{ij} = A_{ij}^{(i)}$ for $j \ge i$, Lemma 3.2 shows that $\|U_{ij}\| \le 2\|A\|$ for each block of U. Therefore $\|L\| \le m$ and $\|U\| \le m(m+1)\|A\|$, and so $\|L\|\|U\| \le m^2(m+1)\|A\|$. For particular norms the bounds on the blocks of L and U yield a smaller bound for $\|L\|$ and $\|U\|$. For example, for the 1-norm we have $\|L\|_1\|U\|_1 \le 2m\|A\|_1$ and for the ∞-norm $\|L\|_\infty\|U\|_\infty \le 2m^2\|A\|_\infty$. We conclude that block LU factorization is stable if A is block diagonally dominant by columns with respect to any subordinate matrix norm satisfying (3.5).

Unfortunately, block LU factorization can be unstable when A is block diagonally dominant by rows. For although Lemma 3.2 guarantees that $\|U_{ij}\| \le 2\|A\|$, $\|L\|$ can be arbitrarily large. This can be seen from the example

$$A = \begin{bmatrix} A_{11} & 0 \\ \frac{1}{2}I & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ \frac{1}{2}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & I \end{bmatrix} = LU,$$

where A is block diagonally dominant by rows in any subordinate norm for any nonsingular matrix A_{11}. It is easy to confirm numerically that block LU factorization can be unstable on matrices of this form. Note that if the block size is 1 then we do have stability, since block LU factorization is equivalent to Gaussian elimination (GE) and the growth factor is bounded by 2 [28] (see Lemma 3.2).

It is also of interest to bound $\|L\|\|U\|$ for a point diagonally dominant matrix, since this property is much easier to check than block diagonal dominance. We will derive a bound for $\|L\|\|U\|$ for a general matrix and then specialise to diagonal dominance. We partition A according to

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \qquad A_{11} \in \mathbb{R}^{r\times r}. \tag{3.6}$$

In the rest of this section we use the norm (2.2) and ρ_n denotes the growth factor for Gaussian elimination (GE) without pivoting, that is, $\rho_n = \max_{i,j,k} |a_{ij}^{(k)}|/|a_{ij}|$ in the usual notation. We assume that GE applied to A succeeds.

Lemma 3.3 *If $A \in \mathbb{R}^{n\times n}$ then $\|A_{21}A_{11}^{-1}\| \le n\rho_n\kappa(A)$.*

Proof. ¿From (2.1) it can be seen that $(A^{-1})_{21} = -S^{-1}A_{21}A_{11}^{-1}$, where the Schur complement $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$. Hence

$$\|A_{21}A_{11}^{-1}\| \le n\|S\|\|(A^{-1})_{21}\| \le n\|S\|\|A^{-1}\|.$$

S is the trailing submatrix that would be obtained after $r-1$ steps of GE. It follows immediately that $\|S\| \le \rho_n\|A\|$. ∎

Lemma 3.4 *If $A \in \mathbb{R}^{n\times n}$ then the Schur complement $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ satisfies $\kappa(S) \le \rho_n\kappa(A)$.*

Proof. $\|S\| \le \rho_n\|A\|$, as noted in the proof of Lemma 3.3, and $\|S^{-1}\| \le \|A^{-1}\|$ because S^{-1} is the $(2,2)$ block of A^{-1}, as is easily seen from (2.1). ∎

To bound $\|L\|$ note that, under the partitioning (3.6), for the first block stage of Algorithm BLU we have $\|L_{21}\| = \|A_{21}A_{11}^{-1}\| \le n\rho_n\kappa(A)$ by Lemma 3.3. Since the algorithm works recursively with the Schur complement S, and since $\kappa(S) \le \rho_n\kappa(A)$ (by Lemma 3.4), each subsequently computed subdiagonal block of L has norm at most $n\rho_n^2\kappa(A)$. Since U is composed of elements of A together with elements of Schur complements of A, $\|U\| \le \rho_n\|A\|$. Overall, for a general matrix $A \in \mathbb{R}^{n\times n}$,

$$\|L\|\|U\| \le n\rho_n^2\kappa(A)\cdot\rho_n\|A\| = n\rho_n^3\kappa(A)\|A\|.$$

Thus block LU factorization is stable for a general matrix A as long as GE is stable for A (that is, $\rho_n = O(1)$) and A is well-conditioned.

If A is diagonally dominant by rows or columns then $\rho_n \le 2$ [28], as noted above. Hence for a diagonally dominant matrix A,

$$\|L\|\|U\| \le 8n\kappa(A)\|A\|, \tag{3.7}$$

that is, stability depends only on the condition of A.

The upper bound in Lemma 3.3 gives about the best possible bound for row diagonally dominant matrices, but a potentially much smaller bound holds under column diagonal dominance, as we now explain. Consider the standard LU factorization,

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} \overline{L}_{11} & 0 \\ \overline{L}_{21} & \overline{L}_{22} \end{bmatrix} \begin{bmatrix} \overline{U}_{11} & \overline{U}_{12} \\ 0 & \overline{U}_{22} \end{bmatrix} = \overline{LU}, \tag{3.8}$$

where $A_{11} \in \mathbb{R}^{r\times r}$. Equating this factorization with (2.1), we see that

$$L_{21} = A_{21}A_{11}^{-1} = A_{21}\overline{U}_{11}^{-1}\overline{L}_{11}^{-1} = \overline{L}_{21}\overline{L}_{11}^{-1}.$$

If A is diagonally dominant by columns then the multipliers for GE are all bounded by 1 in absolute value (from Lemma 3.1 with block size 1), or, equivalently, no row interchanges are required by partial pivoting. This implies that $\|\overline{L}_{21}\| \le 1$ and $\|\overline{L}_{11}^{-1}\| \le 2^{r-2}$, and so

$$\|L_{21}\| \le n2^{r-2}. \tag{3.9}$$

The bound for $\|\overline{L}_{11}^{-1}\|$ is not attainable, for if a unit lower triangular $T \in \mathbb{R}^{n\times n}$ satisfies $\|T^{-1}\| = 2^{n-2}$ then $t_{ij} = -1$ for all $i > j$, and this implies that the first column of $A = TU$ is $u_{11}(1, -1, -1, \ldots, -1)^T$, so A is not diagonally dominant by columns. In any case, $\|\overline{L}_{11}^{-1}\|$ is typically $O(r)$ in practice, assuming only that partial pivoting requires no row interchanges ($A \equiv PA$) (and thus not fully exploiting the diagonal dominance) [25]. No such bound as (3.9) holds for row diagonal dominance, because in this case there is no a priori bound on the multipliers.

For a column diagonally dominant matrix (3.7) and (3.9) give

$$\|L\|\|U\| \le 2n\min(2^{r-2}, 4\kappa(A))\|A\|, \tag{3.10}$$

where the maximum block size is r.

To summarise, for (point) diagonally dominant matrices stability is guaranteed if A is well-conditioned. This in turn is guaranteed if the block column diagonal dominance amounts γ_j in (3.1) are sufficiently large relative to $\|A\|$, because for the ∞-norm and any block sizes in (3.1), $\|A^{-1}\|_1 \le (\min_j \gamma_j)^{-1}$ [27]. In the case of column diagonal dominance, stability is guaranteed for small block sizes r irrespective of $\kappa(A)$, by (3.10).

4 Symmetric Positive Definite Matrices

Let A be a symmetric positive definite matrix, partitioned as

$$A = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix}, \qquad A_{11} \in \mathbb{R}^{r\times r}.$$

The definiteness implies certain relations among the submatrices A_{ij} that can be used to obtain a stronger bound for $\|L\|_2$ than can be deduced from Lemma 3.3.

Lemma 4.1 *If A is symmetric positive definite then $\|A_{21}A_{11}^{-1}\|_2 \le \kappa_2(A)^{1/2}$.*

Proof. Let A have the Cholesky factorization

$$A = \begin{bmatrix} R_{11}^T & 0 \\ R_{12}^T & R_{22}^T \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}, \qquad R_{11} \in \mathbb{R}^{r\times r}.$$

Then $A_{21}A_{11}^{-1} = R_{12}^T R_{11} \cdot R_{11}^{-1} R_{11}^{-T} = R_{12}^T R_{11}^{-T}$, so

$$\|A_{21}A_{11}^{-1}\|_2 \le \|R_{12}\|_2 \|R_{11}^{-1}\|_2 \le \|R\|_2 \|R^{-1}\|_2 = \kappa_2(R) = \kappa_2(A)^{1/2}.$$

∎

Note: At the cost of a much more difficult proof, Lemma 4.1 can be strengthened to the attainable bound $\|A_{21}A_{11}^{-1}\|_2 \le (\kappa_2(A)^{1/2} - \kappa_2(A)^{-1/2})/2$, as shown in [6, Theorem 4], but the weaker bound is sufficient for our purposes.

The proof of the following lemma is similar to that of Lemma 3.4.

Lemma 4.2 *If A is symmetric positive definite then the Schur complement $S = A_{22} - A_{21}A_{11}^{-1}A_{21}^T$ satisfies $\kappa_2(S) \le \kappa_2(A)$.*

Using the same reasoning as in the last section, we find that each subdiagonal block of L is bounded in 2-norm by $\kappa_2(A)^{1/2}$. Therefore $\|L\|_2 \le 1 + m\kappa_2(A)^{1/2}$, where there are m block stages in the algorithm. Also, it can be shown that $\|U\|_2 \le \sqrt{m}\|A\|_2$. Hence

$$\|L\|_2\|U\|_2 \le \sqrt{m}(1 + m\kappa_2(A)^{1/2})\|A\|_2. \tag{4.1}$$

It follows from Theorem 2.1 that when Algorithm BLU is applied to a symmetric positive definite matrix A, the backward errors for the LU factorization and the subsequent solution of a linear system are both bounded by

$$c_n\sqrt{m}u\|A\|_2(2 + m\kappa_2(A)^{1/2}) + O(u^2). \tag{4.2}$$

Any resulting bound for $\|x - \widehat{x}\|_2/\|x\|_2$ will be proportional to $\kappa_2(A)^{3/2}$, rather than $\kappa_2(A)$ as for a stable method. This suggests that block LU factorization can lose up to 50% more digits of accuracy in x than a stable method for solving symmetric positive definite linear systems.

Note that the $\kappa_2(A)^{1/2}$ term in (4.2) can be pessimistic, because it is clear from the proof of Lemma 4.1 that it is terms $\|U_{ii}^{-1}\|_2^{1/2}$, where U_{ii} is a diagonal block of U, that influence the error bounds, and $\|U_{ii}^{-1}\|_2^{1/2} \le \|A^{-1}\|_2^{1/2}$. One would expect the backward error to increase with the block size, with a backward error of size (4.2) being nearly attainable for a sufficiently large block size.

Our main conclusion is that block LU factorization is guaranteed to be stable for a symmetric positive definite matrix A if A is well-conditioned.

One might wonder whether block LDL^T and block Cholesky factorizations have better stability properties than block LU factorization. A genuine block Cholesky factorization $A = R^T R$ would use matrix square roots ($R_{11} = A_{11}^{1/2}$, etc.), which makes this factorization too expensive, whereas a partitioned Cholesky factorization is numerically equivalent to the point case. A block LDL^T factorization, where $D = \mathrm{diag}(D_{11}, \ldots, D_{mm})$, is feasible to compute, but it is easily shown to have analogous stability properties to block LU factorization.

5 Numerical Experiments

We describe some numerical experiments that give further insight into the analysis presented above. The computations were performed in MATLAB, which has unit roundoff $u = 2^{-53} \approx 1.1 \times 10^{-16}$.

We use the following two matrices, which were also used in [7]. The symmetric positive definite Moler matrix [19] is defined by $A_n(\alpha) = R_n(\alpha)^T R_n(\alpha)$, where $R_n(\alpha)$ is unit upper triangular with all the off-diagonal elements equal to α. The Dorr matrix $D_n(\alpha)$ [19], is an unsymmetric, row diagonally dominant tridiagonal matrix. $D_n(\alpha)$ has row diagonal dominance factors $\gamma_i := |d_{ii}| - |d_{i,i-1}| - |d_{i,i+1}| = (n+1)^2\alpha$ for $i = 1, n$ and $\gamma_i = 0$ otherwise, and we perturbed the diagonal elements $d_{22}, \ldots, d_{n-1,n-1}$ to ensure that $\gamma_i \geq 10^{-14}$ for the computed matrix. Neither of these two matrices is row or column block diagonally dominant for any block sizes in the 1, 2 and ∞ norms.

In the first experiment we chose $x = e = (1, 1, \ldots, 1)^T$, formed $b = Ax$ and solved for x using Algorithm BLU with implementations 1 and 2. One step of iterative refinement in fixed precision was done, yielding a corrected solution $\widehat{y}$. We report the relative residuals

$$\mathrm{res}(\widehat{L}, \widehat{U}) = \frac{\|A - \widehat{L}\widehat{U}\|_\infty}{\|A\|_\infty}, \qquad \mathrm{res}(\widehat{x}) = \frac{\|A\widehat{x} - b\|_\infty}{\|A\|_\infty \|\widehat{x}\|_\infty + \|b\|_\infty},$$

and the forward error

$$\mathrm{err}(\widehat{x}) = \frac{\|\widehat{x} - x\|_\infty}{\|x\|_\infty}.$$

Note that $\mathrm{res}(\widehat{x})$ is the normwise backward error of $\widehat{x}$ (see, e.g., [17]), and that, approximately, $\mathrm{err}(\widehat{x}) \leq \kappa_\infty(A)\mathrm{res}(\widehat{x})$. We also report the upper bounds for $\mathrm{res}(\widehat{L}, \widehat{U})$ and $\mathrm{res}(\widehat{x})$ from Theorem 2.1, which modulo the constant terms are both approximately

$$\mathrm{bound}_1 = \frac{u\|\widehat{L}\|_\infty \|\widehat{U}\|_\infty}{\|A\|_\infty};$$

the corresponding bound for implementation 2 is

$$\mathrm{bound}_2 = \max_i \kappa_\infty(U_{ii}) \mathrm{bound}_1.$$

The results for the Moler matrix $A_{16}(-2)$ are shown in Tables 5.1 and 5.2. Note that we know the exact solution x because A has integer entries with $|a_{ij}| \leq 61$ and so $b = Ax$ is formed exactly. We comment on several interesting features.

(1) For implementation 1 instability is revealed in the residuals for both the factorization and for $\widehat{x}$; it increases with the block size, as is to be expected (see the discussion at the end of section 4).

The values for bound_1 show that the theoretical error bounds correctly model the variation of the residuals with the block size and are mostly within two orders of magnitude of the actual residuals.

(2) Implementation 2 is much more unstable than implementation 1 as a means of computing the block LU factorization. The residuals of the computed solutions $\widehat{x}$ are as small as for implementation 1 but the forward errors are mostly larger. The quantity bound_2 is very pessimistic as an estimate of the residuals.

(3) One step of iterative refinement works extremely well for implementation 1, but it is ineffective for most block sizes with implementation 2. Theoretical backing for iterative refinement in fixed precision can be given using Theorem 2.1 together with Theorem 2.1 of [18]; see the discussion in section 2.2 of [7]. For implementation 2 the instability is too severe for iterative refinement to work.

(4) The forward errors for $\widehat{y}$ in implementation 1 reflect the ill-condition of the problem. It is not clear why the forward errors for $\widehat{x}$ are no larger than those for the "more stable" solution $\widehat{y}$.

(5) For the block size 15 ($m = 2$) with implementation 1,

$$\|\widehat{L}\|_2\|\widehat{U}\|_2 \approx 3 \times 10^9 \approx 0.05\kappa_2(A)^{1/2}\|A\|_2,$$

which shows that (4.1) is reasonably sharp.

We solved another system with the same coefficient matrix and with $b = e$. Now x is a "large-normed" solution, that is, $\|x\|_\infty = O(\|A^{-1}\|_\infty\|b\|_\infty)$ (indeed, $\|x\|_\infty = \|A^{-1}\|_\infty\|b\|_\infty$ since $A^{-1} \geq 0$). For this right-hand side the instability in the block LU factorization does not affect $\widehat{x}$ for implementation 1: $\text{res}(\widehat{x}) \leq 5 \times 10^{-19}$ for all block sizes. In our experience this behaviour is not uncommon for large-normed solutions.

Table 5.3 reports results for the Dorr matrix $D_{16}(10^{-4})$, for implementation 1 with $x_i = i$. In computing the $\text{err}(\cdot)$ quantities for the Dorr matrix we approximated the true solution by the computed solution from Gaussian elimination with partial pivoting. The results for implementation 2 are very similar. We see more severe instability than for the Moler matrix. One step of iterative refinement is not sufficient to achieve a residual of order u. It is surprising that despite the instability evident for the block size 15, the magnitude of the error $\text{err}(\widehat{x})$ indicates that $\widehat{x}$ is about as accurate as the solution from GEPP. For the block size 15 with implementation 1,

$$\|\widehat{L}\|_\infty\|\widehat{U}\|_\infty \approx 8 \times 10^{15} \approx 0.4\kappa_\infty(A)\|A\|_\infty,$$

confirming that (3.7) is reasonably sharp.

We also solved the Dorr matrix system with $b = e$ and found the results to be very similar to those in Table 5.3. Thus, although $\widehat{x}$ is now a large-normed solution, the instability in the LU factorization is still fully reflected in $\widehat{x}$. We solved the same systems with the transpose of the Dorr matrix, which is diagonally dominant by columns. All the relative residuals for Implementation 1 were less than $3u$. Implementation 2 behaved erratically: for the system with $x_i = i$, $\text{res}(\widehat{L}, \widehat{U}) \leq 3u$ but $\text{res}(\widehat{x})$ was as large as 5×10^{-4}! In this example $\|\widehat{L}\|_\infty\|\widehat{U}\|_\infty/\|A\|_\infty$ was approximately equal to the block size r, so the 2^{r-2} bound in (3.10) is pessimistic here.

We conclude from these experiments that our backward error bounds for implementation 1 of Algorithm BLU are almost attainable and they seem to capture well the behaviour of the backward error. We have also observed some varied and interesting behaviour, all of which is within the freedom afforded by the error bounds, but not all of which is easily explained heuristically.

Table 5.1: Moler matrix. Implementation 1. $x = e$.

$\kappa_\infty(A_{16}(-2)) \approx 7 \times 10^{16}$

block size	$\mathrm{res}(\widehat{L}, \widehat{U})$	bound_1	$\mathrm{res}(\widehat{x})$	$\mathrm{res}(\widehat{y})$	$\mathrm{err}(\widehat{x})$	$\mathrm{err}(\widehat{y})$
1	0.00	2.34e-16	0.00	0.00	0.00	0.00
2	0.00	4.87e-16	0.00	0.00	0.00	0.00
3	6.64e-17	2.91e-15	6.27e-17	0.00	6.02e-2	6.03e-2
4	2.56e-16	8.41e-15	1.24e-16	0.00	2.75e-2	4.79e-2
5	3.67e-16	3.39e-14	3.14e-16	6.27e-17	3.15e-2	1.31e-1
6	1.18e-15	8.35e-14	3.14e-16	0.00	3.46e-2	4.67e-2
7	4.09e-15	2.93e-13	2.70e-15	0.00	2.24e-2	5.40e-2
8	1.66e-15	4.98e-13	2.38e-15	0.00	8.04e-3	5.89e-2
9	1.02e-14	1.65e-12	5.58e-15	6.12e-17	2.57e-2	5.02e-2
10	1.14e-13	5.35e-12	6.30e-14	0.00	1.21e-1	7.74e-3
11	1.56e-13	1.71e-11	8.65e-14	0.00	4.81e-2	1.28e-1
12	7.63e-13	5.38e-11	1.13e-13	0.00	9.31e-2	9.04e-2
13	3.89e-13	1.68e-10	1.24e-12	5.94e-17	4.03e-3	1.13e-1
14	1.71e-12	5.17e-10	3.55e-12	3.92e-18	1.64e-2	2.88e-2
15	2.95e-11	1.58e-9	1.32e-11	0.00	1.17e-1	3.20e-2

Table 5.2: Moler matrix. Implementation 2. $x = e$.

$\kappa_\infty(A_{16}(-2)) \approx 7 \times 10^{16}$

block size	res($\widehat{L}, \widehat{U}$)	bound$_2$	res($\widehat{x}$)	res($\widehat{y}$)	err($\widehat{x}$)	err($\widehat{y}$)
1	0.00	2.34e-16	0.00	0.00	0.00	0.00
2	0.00	2.39e-14	0.00	0.00	0.00	0.00
3	3.36e-16	2.31e-12	6.27e-17	6.18e-17	2.58e-2	3.10e-2
4	2.40e-15	1.06e-10	6.27e-16	0.00	3.97e-1	2.08e-1
5	1.47e-14	6.17e-9	2.57e-15	2.90e-17	1.38e0	1.63e-1
6	1.13e-13	2.04e-7	8.32e-15	9.41e-17	5.32e0	1.07e-1
7	5.95e-13	9.01e-6	3.11e-13	6.12e-17	1.72e0	4.87e-2
8	1.83e-11	1.85e-4	1.34e-13	3.78e-14	1.69e2	1.24e0
9	1.18e-10	7.07e-3	3.20e-13	1.28e-13	3.39e2	5.81e-1
10	4.82e-10	2.59e-1	1.45e-11	8.37e-14	6.52e1	9.32e-2
11	1.93e-8	9.15e0	2.82e-12	2.30e-12	6.79e3	4.67e1
12	1.11e-7	3.13e2	8.08e-12	5.89e-12	1.25e4	5.00e1
13	1.54e-6	1.04e4	2.69e-11	2.19e-11	6.32e4	3.90e2
14	8.49e-6	3.38e5	1.38e-10	2.47e-10	8.65e4	1.06e2
15	1.13e-4	1.08e7	2.55e-10	2.04e-10	4.51e5	2.39e3

Table 5.3: Dorr matrix. Implementation 1. $x = (1, 2, \ldots, n)^T$.

$\kappa_\infty(D_{16}(10^{-4})) \approx 2 \times 10^{15}$

block size	res($\widehat{L}, \widehat{U}$)	bound$_1$	res($\widehat{x}$)	res($\widehat{y}$)	err($\widehat{x}$)	err($\widehat{y}$)
1	5.89e-17	2.49e-14	6.76e-17	2.14e-17	2.10e-4	5.72e-4
2	2.92e-14	4.77e-12	5.67e-15	2.85e-17	1.84e-4	7.37e-4
3	1.06e-12	3.38e-10	7.65e-13	5.09e-16	1.49e-4	3.77e-4
4	7.11e-12	1.00e-8	6.03e-12	6.95e-15	1.91e-4	5.01e-4
5	7.59e-9	6.48e-6	8.87e-10	7.83e-13	2.37e-4	6.06e-4
6	1.51e-10	2.00e-8	8.35e-12	2.31e-15	7.18e-5	4.29e-4
7	1.42e-5	8.48e-4	3.79e-7	9.95e-11	7.64e-5	4.29e-4
8	1.28e-19	2.24e-16	2.49e-17	2.85e-17	1.39e-4	5.51e-4
9	9.98e-17	1.20e-14	2.49e-17	6.05e-17	1.47e-4	3.88e-4
10	3.67e-14	1.05e-12	1.41e-15	1.42e-17	2.37e-4	6.10e-4
11	1.54e-12	1.28e-10	7.68e-14	2.67e-17	1.03e-4	7.44e-4
12	1.52e-10	2.00e-8	8.94e-12	3.17e-15	7.68e-5	6.12e-4
13	3.10e-8	3.83e-6	2.23e-9	7.44e-13	1.00e-4	7.41e-4
14	8.98e-6	8.56e-4	4.39e-7	1.37e-10	8.84e-5	6.04e-4
15	5.44e-4	6.07e-2	2.30e-5	5.90e-9	6.51e-5	4.04e-4

6 Concluding Remarks

The main conclusions of this work are that although block LU factorization is unstable in general, it is stable for matrices that are block diagonally dominant by columns, and generally the level of instability is bounded in terms of the condition number and the growth factor for Gaussian elimination without pivoting. Therefore if the matrix is symmetric positive definite or (point) diagonally dominant, stability is assured if A is well-conditioned. These results are summarised in Table 6.1, which tabulates a bound for $\|A-\widehat{L}\widehat{U}\|/(c_n u\|A\|)$ for block and point LU factorization with the matrix properties considered in sections 3 and 4. The constant c_n incorporates any constants in the bound that depend polynomially on the dimension, so a value of 1 in the table indicates unconditional stability.

The implications for practical computation are that when using block LU factorization to solve $Ax = b$ (which we certainly do not discourage) it is vital to check the relative residual (or normwise backward error) $\|A\widehat{x} - b\|_\infty/(\|A\|_\infty\|\widehat{x}\|_\infty + \|b\|_\infty)$. If the residual is unacceptably large it is worth trying one step of iterative refinement in fixed precision, although this is not guaranteed to yield a smaller residual if the instability is severe. Note that one may be fortunate enough to obtain an acceptable $\widehat{x}$ even if $\|A - \widehat{L}\widehat{U}\|_\infty/\|A\|_\infty$ is large, as our numerical experiments illustrate.

A more general conclusion is that the stability of a block algorithm can not be taken for granted. Existing error analysis for point algorithms is not directly applicable to block algorithms; it is, however, applicable to partitioned algorithms. A complicating feature is that there may be several possible block reformulations of a basic algorithm to consider, as is the case with Algorithm BLU in section 2. Assessing the stability of other block algorithms is clearly an interesting area for further research.

Table 6.1: Stability of LU factorization.
ρ_n is the growth factor for GE without pivoting.
r is the maximum block size.

Matrix property	Block LU	Point LU
symmetric positive definite	$\kappa(A)^{1/2}$	1
block column diag. dom.	1	$\rho_n \kappa(A)$
point column diag. dom.	2^{r-2}	1
block row diag. dom.	$\rho_n^3 \kappa(A)$	$\rho_n \kappa(A)$
point row diag. dom.	$\kappa(A)$	1
arbitrary	$\rho_n^3 \kappa(A)$	$\rho_n \kappa(A)$

References

[1] E. Anderson, Z. Bai, C.H. Bischof, J.W. Demmel, J.J. Dongarra, J.J. Du Croz, A. Greenbaum, S.J. Hammarling, A. McKenney, S. Ostrouchov and D.C. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1992, to appear.

[2] Z. Bai and J.W. Demmel, On a block implementation of Hessenberg multishift QR iteration, Int. J. High Speed Computing, 1 (1989), pp. 97–121.

[3] R.E. Bank and D.J. Rose, Marching algorithms for elliptic boundary value problems. I: The constant coefficient case, SIAM J. Numer. Anal., 14 (1977), pp. 792–829.

[4] C.H. Bischof and C.F. Van Loan, The WY representation for products of Householder matrices, SIAM J. Sci. Stat. Comput., 8 (1987), pp. s2–s13.

[5] J.R. Bunch, Block methods for solving sparse linear systems, in *Sparse Matrix Computations*, J.R. Bunch and D.J. Rose, eds., Academic Press, New York, 1976, pp. 39–58.

[6] J.W. Demmel, The condition number of equivalence transformations that block diagonalize matrix pencils, SIAM J. Numer. Anal., 20 (1983), pp. 599–610.

[7] J.W. Demmel and N.J. Higham, Stability of block algorithms with fast level 3 BLAS, LAPACK Working Note #22 and Numerical Analysis Report No. 188, University of Manchester, England, 1990; to appear in ACM Math. Trans. Soft.

[8] J.J. Dongarra, I.S. Duff, D.C. Sorensen and H.A. van der Vorst, *Solving Linear Systems on Vector and Shared Memory Computers*, Society for Industrial and Applied Mathematics, Philadelphia, 1991.

[9] J.J. Dongarra, J.J. Du Croz, I.S. Duff and S.J. Hammarling, A set of Level 3 basic linear algebra subprograms, ACM Trans. Math. Soft., 16 (1990), pp. 1–17.

[10] J.J. Dongarra, J.J. Du Croz, I.S. Duff and S.J. Hammarling, Algorithm 679. A set of Level 3 basic linear algebra subprograms: Model implementation and test programs, ACM Trans. Math. Soft., 16 (1990), pp. 18–28.

[11] J.J. Dongarra, S.J. Hammarling and D.C. Sorensen, Block reduction of matrices to condensed forms for eigenvalue computations, J. Comp. Appl. Math., 27 (1989), pp. 215–227.

[12] J.J. DuCroz and N.J. Higham, Stability of methods for matrix inversion, LAPACK Working Note #27 and Numerical Analysis Report No. 190, University of Manchester, England, 1990; to appear in IMA J. Numerical Analysis, 12 (1992).

[13] D.G. Feingold and R.S. Varga, Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem, Pacific J. Math., 12 (1962), pp. 1241–1250.

[14] K. Gallivan, W. Jalby, U. Meier and A.H. Sameh, Impact of hierarchical memory systems on linear algebra algorithm design, Int. J. of Supercomputer Applics., 2 (1988), pp. 12–48.

[15] K.A. Gallivan, R.J. Plemmons and A.H. Sameh, Parallel algorithms for dense linear algebra computations, SIAM Review, 32 (1990), pp. 54–135.

[16] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Second Edition, Johns Hopkins University Press, Baltimore, Maryland, 1989.

[17] N.J. Higham, How accurate is Gaussian elimination?, in *Numerical Analysis 1989, Proceedings of the 13th Dundee Conference*, Pitman Research Notes in Mathematics 228, D.F. Griffiths and G.A. Watson, eds., Longman Scientific and Technical, 1990, pp. 137–154.

[18] N.J. Higham, Iterative refinement enhances the stability of QR factorization methods for solving linear equations, BIT, 31 (1991), pp. 447–468.

[19] N.J. Higham, Algorithm 694: A collection of test matrices in MATLAB, ACM Trans. Math. Soft., 17 (1991), pp. 289–305.

[20] N.J. Higham and P.A. Knight, Componentwise error analysis for stationary iterative methods, Numerical Analysis Report No. 206, University of Manchester, December 1991; to be submitted to Proceedings of Workshop on Linear Algebra, Markov Chains, and Queuing Models, C.D. Meyer, Jr. and R.J. Plemmons, eds, IMA Volumes in Mathematics and Its Applications, Springer-Verlag.

[21] E. Isaacson and H.B. Keller, *Analysis of Numerical Methods*, John Wiley, New York, 1966.

[22] R.M.M. Mattheij, The stability of LU-decompositions of block tridiagonal matrices, Bull. Austral. Math. Soc., 29 (1984), pp. 177–205.

[23] R.M.M. Mattheij, Stability of block LU-decompositions of matrices arising from BVP, SIAM J. Alg. Disc. Meth., 5 (1984), pp. 314–331.

[24] R.S. Schreiber, Block algorithms for parallel machines, in *Numerical Algorithms for Modern Parallel Computer Architectures*, M.H. Schultz, ed., IMA Volumes In Mathematics and Its Applications 13, Springer-Verlag, Berlin, 1988, pp. 197–207.

[25] L.N. Trefethen and R.S. Schreiber, Average-case stability of Gaussian elimination, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 335–360.

[26] J.M. Varah, On the solution of block-tridiagonal systems arising from certain finite-difference equations, Math. Comp., 26 (1972), pp. 859–868.

[27] J.M. Varah, A lower bound for the smallest singular value of a matrix, Linear Algebra and Appl., 11 (1975), pp. 3–5.

[28] J.H. Wilkinson, Error analysis of direct methods of matrix inversion, J. Assoc. Comput. Mach., 8 (1961), pp. 281–330.

[29] H. Woźniakowski, Roundoff-error analysis of iterations for large linear systems, Numer. Math., 30 (1978), pp. 301–314.