Stability of parallel triangular system solvers

Higham, Nicholas J.

1995

MIMS EPrint: **2006.159**

ISSN 1749-9097

SIAM J. SCI. COMPUT.
Vol. 16, No. 2, pp. 400–413, March 1995

STABILITY OF PARALLEL TRIANGULAR SYSTEM SOLVERS*

NICHOLAS J. HIGHAM[†]

Abstract. Several parallel algorithms have been proposed for the solution of triangular systems. The stability of four of them is analysed here: a fan-in algorithm, a block elimination method, a method based on a factorized power series expansion of the matrix inverse, and a method based on a divide and conquer matrix inversion technique. New forward error and residual bounds are derived, including an improvement on the bounds of Sameh and Brent for the fan-in algorithm. A forward error bound is identified that holds not only for all the methods described here, but for any triangular equation solver that does not rely on algebraic cancellation; among the implications of the bound is that any such method is extremely accurate for certain special types of triangular systems.

Key words. triangular system, matrix inversion, parallel algorithms, fan-in operation, numerical stability, rounding error analysis

AMS subject classifications. 65F05, 65G05

1. Introduction.

The standard substitution algorithm for solving triangular systems has optimal serial complexity, for each element of the $n \times n$ coefficient matrix must partake in at least one operation and so there must be $O(n^2)$ operations. However, the parallel complexity of solving a triangular system is potentially as low as $O(\log n)$ time steps[1], as can be seen by a fan-in argument. Work on parallel solution of triangular systems has proceeded in two directions. Several authors have developed parallel implementations of substitution for distributed-memory multiprocessors, some recent contributions being by Heath and Romine [9], Li and Coleman [16], Romine and Ortega [20], and Eisenstat, Heath, Henkel, and Romine [8]. On the other hand, several methods with lower parallel complexity than substitution have been proposed over the last twenty years. These methods all require $O(n^3)$ processors to achieve their minimal complexity and they perform $O(n^3)$ operations.

The numerical stability of substitution is well understood, but that of the parallel methods is not. In this work we analyse the stability of four parallel methods for solving triangular systems: a fan-in algorithm, a block elimination method, a method based on a factorized power series expansion of the matrix inverse, and a method that computes the inverse by a divide and conquer technique. Sameh and Brent [21] obtained an error bound for the fan-in algorithm, but there appears to be little or no published error analysis for the other methods. We derive informative error bounds for all the methods, obtaining, in particular, stronger bounds for the fan-in algorithm than those of Sameh and Brent. For easy reference, Table 1.1 summarizes the main bounds.

In §2 we summarize existing error analysis for substitution and a parallel method called the partitioned inverse method. We state a forward error bound (see (2.11)) that holds for a wide class of methods, including all those described here, and explore its implications.

Error analysis for the fan-in algorithm, block elimination, power series, and divide and conquer methods is presented in §§3–6, along with some numerical examples. Finally, we give some conclusions in §7.

2. Background.

For the error analysis we use the standard model of floating point arithmetic

$$fl(x \pm y) = x(1+\alpha) \pm y(1+\beta), \qquad |\alpha|, |\beta| \le u,$$
$$fl(x \text{ op } y) = (x \text{ op } y)(1+\delta), \qquad |\delta| \le u, \quad \text{op} = *, /,$$

*Received by the editors September 9, 1993; accepted for publication (in revised form) January 31, 1994.

[†]Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (na.nhigham@na-net.ornl.gov). This work was supported by Science and Engineering Research Council grant GR/H52139, and by the EEC Esprit Basic Research Action Programme, Project 6634 (APPARC).

[1]All logarithms are to the base 2.

TABLE 1.1
Main error bounds.

	Backward error/residual	Forward error
Substitution	(2.3)	(2.6)
Fan-in	(3.3)	(2.11), (3.5)
Block elimination	(4.4)	(4.3)
Power series	(2.13)	(5.2)
Divide & conquer (B)	(6.1), (6.4)	(6.6)
Divide & conquer (D)	(6.2), (6.5)	(6.6)

where u is the unit roundoff. This model is valid for machines that lack a guard digit in addition and subtraction. We place a hat over computed quantities.

Under this model it is straightforward to show that for $A, B \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$,

$$fl(Ax) = (A + \Delta A)x, \qquad |\Delta A| \le nu|A| + O(u^2), \tag{2.1}$$

$$fl(AB) = AB + E, \qquad |E| \le nu|A||B| + O(u^2). \tag{2.2}$$

(Absolute values and inequalities are interpreted componentwise for matrices.) These two results are the basis of all the analysis below. In the analysis we are not concerned with the precise values of constants and will denote by c_n or d_n a low degree polynomial in n and $\log n$.

It is useful to recall what is known about the substitution algorithm. The computed solution $\widehat{x}$ to $Lx = b$, where $L \in \mathbb{R}^{n\times n}$ is lower triangular, satisfies (see, for example, [22, p. 150] or [12])

$$(L + \Delta L)\widehat{x} = b, \qquad |\Delta L| \le \big((n+1)u + O(u^2)\big)|L|. \tag{2.3}$$

This result says that $\widehat{x}$ has a small componentwise relative backward error $\omega(\widehat{x})$, where for an approximate solution y to a general system $Ax = b$,

$$\omega(y) = \min\{\epsilon : (A + \Delta A)y = b + \Delta b, \ |\Delta A| \le \epsilon|A|, \ |\Delta b| \le \epsilon|b|\}.$$

No larger than $\omega(y)$ for the 1- and ∞-norms is the normwise relative backward error

$$\eta(y) = \min\{\epsilon : (A + \Delta A)y = b + \Delta b, \ \|\Delta A\| \le \epsilon\|A\|, \ \|\Delta b\| \le \epsilon\|b\|\}.$$

These two backward errors are easily computable for a given y, because they can be expressed in terms of the residual $r = b - Ay$ [18], [19]:

$$\omega(y) = \max_i \frac{|r_i|}{(|A||y| + |b|)_i}, \tag{2.4}$$

$$\eta(y) = \frac{\|r\|}{\|A\|\|y\| + \|b\|}. \tag{2.5}$$

The norm is any subordinate norm, and in the formula for $\omega(y)$, $\xi/0$ is interpreted as zero if $\xi = 0$ and infinity otherwise. If, in the definition of $\omega(y)$ or $\eta(y)$, we do not perturb b, then the formulas (2.4) and (2.5) remain valid when b is replaced by 0. We will use the infinity norm throughout.

Corresponding to (2.3) we have the forward error bound

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \le (n+1)u \ \mathrm{cond}(L, x) + O(u^2), \tag{2.6}$$

where

$$\mathrm{cond}(L, x) = \frac{\|\,|L^{-1}||L||x|\,\|_\infty}{\|x\|_\infty}$$

is the Bauer–Skeel condition number. The maximum value of $\operatorname{cond}(L,x)$ is $\operatorname{cond}(L) := \operatorname{cond}(L,e)$, where $e = (1,1,\dots,1)^T$.

It is also instructive to consider the partitioned inverse method for solving $Lx = b$. Any lower triangular matrix $L \in \mathbb{R}^{n\times n}$ can be factorized $L = L_1L_2\dots L_n$, where L_k differs from the identity matrix only in the kth column:

$$L_k = \begin{bmatrix} I_{k-1} & & & & \\ & l_{kk} & & & \\ & l_{k+1,k} & 1 & & \\ & \vdots & & \ddots & \\ & l_{nk} & & & 1 \end{bmatrix}. \tag{2.7}$$

The solution to a linear system $Lx = b$ may therefore be expressed as

$$x = L^{-1}b = M_nM_{n-1}\dots M_1b, \tag{2.8}$$

where $M_i = L_i^{-1}$. When evaluated in the natural right to left order, this formula yields a trivial variation of a column oriented version of substitution. The partitioned inverse method generalizes this evaluation by grouping the factors $L_1L_2\dots L_n = G_1G_2\dots G_m$, where each G_i is a product of consecutive L_j terms and $1 \le m \le n$. Then x is evaluated as

$$x = G_m^{-1}G_{m-1}^{-1}\dots G_1^{-1}b,$$

where each G_i^{-1} is formed explicitly and the product is evaluated from right to left. The partitioned inverse method is of practical interest for the parallel solution of large, sparse systems with multiple right-hand sides; in this context m is chosen as small as possible subject to the G_i^{-1} being sparse [1]. Higham and Pothen [15] show that, under a reasonable assumption on how the G_i^{-1} are formed, the computed solution $\widehat{x}$ satisfies $(L + \Delta L)\widehat{x} = b$, where

$$|\Delta L| \le c_n u\left((m-1)(|L| - I) + \sum_{i=1}^{m} |G_i||G_i^{-1}||G_i|\right) + O(u^2). \tag{2.9}$$

When $m = n$, so that $G_i = L_i$, we can use the result

$$|L_i||L_i^{-1}||L_i| \le 3|L_i| \tag{2.10}$$

to recover (2.3). Inequality (2.9) is shown in [15] to imply $\|\Delta L\|_\infty \le c_n' u\rho\|L\|_\infty + O(u^2)$, where ρ is a scalar satisfying $1 \le \rho \le m\kappa_\infty(L)$, so normwise stability is guaranteed if L is well conditioned.

An interesting theme among the methods we describe is that they all satisfy a componentwise forward error bound of the form

$$|x - \widehat{x}| \le c_n u M(L)^{-1}|b| + O(u^2), \tag{2.11}$$

where the comparison matrix $M(A) = (m_{ij})$ is defined for $A \in \mathbb{R}^{n\times n}$ by

$$m_{ij} = \begin{cases} |a_{ii}|, & i = j, \\ -|a_{ij}|, & i \ne j. \end{cases}$$

Note that $M(L)$ is an M-matrix, that is, $M(L)^{-1} \ge 0$. A bound of this form holds for any solver that does not rely on algebraic cancellation. A precise way to state this condition is that the solver computes $x_i = f_i(L,b)$ where, for all i, f_i is a multivariate rational function in which the only divisions are by diagonal elements of L and such that when $L = M(L)$ and

$b \geq 0$ there are no subtractions in the evaluation of f_i. For such a solver, it is not difficult to see that

$$|fl(f_i(L,b)) - f_i(L,b)| \leq c_n u \overline{f}_i(M(L), |b|) + O(u^2),$$

where $\overline{f}_i$ denotes f_i with all its coefficients replaced by their absolute values, and where $\overline{f}_i(M(L), |b|)$ is a rational expression consisting entirely of nonnegative terms. This bound is simply (2.11) expressed in different notation. An example (admittedly, a contrived one) of a solver that does not satisfy (2.11) is, for $n = 2$,

$$x_1 = b_1/l_{11}, \quad x_2 = \big((b_2 - b_1)(l_{11} + l_{21}) + b_1 l_{11} - b_2 l_{21}\big)/(l_{11} l_{22}).$$

We will not give a formal proof of (2.11) in the general case, but will obtain it as a consequence of the analysis for each individual method below. For substitution, (2.11) is straightforward to prove by induction on the components of x (the proof is a minor modification of Wilkinson's proof of (2.11) for the case where $L = M(L)$ and $b \geq 0$ [25, pp. 250–251]). For the partitioned inverse method, (2.11) also holds, though this is not pointed out in [15].

The upper bound in (2.11) cannot be rephrased or bounded in terms of L^{-1} or $\kappa_\infty(L)$, because, for any L, $|L^{-1}| \leq M(L)^{-1}$, and $\|M(L)^{-1}\|/\|L^{-1}\|$ can be arbitrarily large (for a parametrized example with $n = 3$, see [11]).

There are several interesting consequences of (2.11). First, if L is an M-matrix (so that $L = M(L)$) then (2.11) shows that the algorithm under consideration is componentwise forward stable, in the sense that the forward error is of the same order of magnitude as that caused by componentwise perturbations of order $c_n u$ to the vector b (and, a fortiori, a bound of the form (2.6) holds). If, in addition, $b \geq 0$, then (2.11) becomes $|x - \widehat{x}| \leq c_n u |x| + O(u^2)$, which shows that $\widehat{x}$ approximates x to almost full relative accuracy in all components (at least, modulo the $O(u^2)$ term). If $l_{ii} = 1$ and $|l_{ij}| \leq 1$ for all i and j (as holds for the matrices L from LU factorization with partial pivoting) then $M(L)^{-1}$ has elements of size at most 2^{n-2}, so $\|x - \widehat{x}\|_\infty \leq 2^{n-1} c_n u \|b\|_\infty + O(u^2)$.

For general L, (2.11) implies the normwise forward error bound

$$\frac{\|x - \widehat{x}\|_\infty}{\|x\|_\infty} \leq c_n u \frac{\| M(L)^{-1}|L||x| \|_\infty}{\|x\|_\infty} + O(u^2). \tag{2.12}$$

This bound is no smaller than (2.6), and potentially much larger, but is of comparable size if $M(L)^{-1} \leq c_n' |L|$.

Another consequence of (2.11) is the residual bound

$$|L\widehat{x} - b| \leq c_n u |L| M(L)^{-1} |b| + O(u^2). \tag{2.13}$$

This bound shows that the underlying algorithm is componentwise backward stable if L is an M-matrix and $b \geq 0$, or, more generally, whenever $M(L)^{-1}|b| \leq d_n |x|$. We have also the normwise residual bound

$$\|L\widehat{x} - b\|_\infty \leq c_n u \| \, |L| M(L)^{-1} |b| \, \|_\infty + O(u^2), \tag{2.14}$$

from which it follows that the underlying algorithm is normwise backward stable if $\mu(L, x)$ is of order 1, where

$$\mu(L, x) = \frac{\| \, |L| M(L)^{-1} |b| \, \|_\infty}{\|L\|_\infty \|x\|_\infty} \leq \frac{\| \, |L| M(L)^{-1} |L| |x| \, \|_\infty}{\|L\|_\infty \|x\|_\infty}.$$

3. Fan-in algorithm. The fan-in algorithm solves $Lx = b$ by evaluating the product (2.8) in $\lceil \log(n+1) \rceil$ steps by the fan-in operation. For example, for $n = 7$ the order of calculation is specified by

$$x = \big((M_7M_6)(M_5M_4)\big)\big((M_3M_2)(M_1b)\big),$$

where all the products appearing within a particular size of parenthesis can be evaluated in parallel. In general, the evaluation can be expressed as a binary tree of depth $\lceil \log(n+1) \rceil + 1$, with products M_1b and M_iM_{i-1} ($i = 3, 4, \dots, 2\lfloor (n-1)/2 \rfloor + 1$) at the top level and a single product yielding x at the bottom level. This algorithm has been proposed and analysed by Sameh and Brent [21], who show that it can be implemented in $\frac{1}{2}\log^2 n + O(\log n)$ time steps on $\frac{1}{68}n^3 + O(n^2)$ processors. (The algorithm requires about $n^3/10$ operations, so is of no interest for serial computation. Some interesting comments on the practical significance of $\log n$ terms in complexity results are given by Edelman [7].) Sameh and Brent also give an error analysis. They show that the computed solution $\widehat{x}$ satisfies

$$(3.1) \qquad (L + \Delta L)\widehat{x} = b, \qquad \|\Delta L\|_\infty \le \alpha_n u \kappa_\infty(L)^2 \|L\|_\infty + O(u^2),$$

where $\alpha_n = \frac{1}{4}n^2 \log n + O(n \log n)$. This bound is larger by a factor at least $\kappa_\infty(L)^2 \ge 1$ than the normwise equivalent of (2.3) for substitution.

We will derive a componentwise residual bound that is stronger than (3.1). To avoid complicated notation that obscures the simplicity of the analysis we take $n = 7$. It is not hard to see that the result we obtain is valid for all n. We assume that the inverses $M_i = L_i^{-1}$ are formed exactly, as the errors in forming them affect only the constants in the bounds. Applying (2.1) and (2.2) we find that the computed solution $\widehat{x}$ satisfies

$$(3.2) \qquad \widehat{x} = \big((M_7M_6 + \Delta_{76})(M_5M_4 + \Delta_{54}) + \Delta_{7654}\big)\big((M_3M_2 + \Delta_{32})(M_1 + \Delta_1)b\big),$$

where

$$\begin{aligned}
|\Delta_{i,i-1}| &\le c_n u |M_i||M_{i-1}| + O(u^2), \quad i = 5, 7,\\
|\Delta_{7654}| &\le c_n u (|M_7M_6||M_5M_4| + |M_7M_6M_5M_4|) + O(u^2),\\
|\Delta_{32}| &\le c_n u (|M_3||M_2| + |M_3M_2|) + O(u^2),\\
|\Delta_1| &\le c_n u |M_1| + O(u^2).
\end{aligned}$$

Premultiplying (3.2) on the left by L, we find that the residual $r = L\widehat{x} - b$ is a sum of terms of the form

$$L(M_7 \dots M_{j+1})\Delta_{j,\dots,k}M_{k-1}\dots M_1 b = L_1 \dots L_j \Delta_{j,\dots,k} L_k \dots L_7 x.$$

All these terms share the same upper bound, which we derive for just one of them. Unfortunately, it is not possible to exploit (2.10), because all but one term in the overall residual bound contains a product of the form $|M_j \dots M_k||M_{k-1} \dots M_i|$, which cannot be simplified using (2.10). For $j = 5$, $k = 4$ we have

$$\begin{aligned}
|L_1 \dots L_5 \Delta_{54} L_4 \dots L_7 x| &\le c_n u |L_1 \dots L_5||M_5||M_4||L_4 \dots L_7 x| + O(u^2)\\
&= c_n u |L_1 \dots L_5||L_6L_7L^{-1}L_1 \dots L_4|\\
&\quad \times |L_5L_6L_7L^{-1}L_1L_2L_3||L_4 \dots L_7 x| + O(u^2)\\
&\le c_n u |L||L^{-1}||L||L^{-1}||L||x| + O(u^2),
\end{aligned}$$

where we have used the property that, for any $L \in \mathbb{R}^{n \times n}$, $|L_1| \dots |L_n| = |L|$. The overall residual bound is therefore of the form

$$|L\hat{x} - b| \le d_n u |L||L^{-1}||L||L^{-1}||L||x| + O(u^2), \tag{3.3}$$

or, on taking norms,

$$\|L\hat{x} - b\|_\infty \le d_n u \| \, |L||L^{-1}||L||L^{-1}||L||x| \, \|_\infty + O(u^2). \tag{3.4}$$

By considering the binary tree associated with the fan-in algorithm, and using the fact that the matrices at the ith level of the tree have at most 2^{i-1} nontrivial columns, it is easy to see that we can take $d_n = a\, n \log n$, where a is a constant of order 1; d_n is smaller than α_n in (3.1) by a factor of order n.

Sameh and Brent's bound (3.1) can be obtained from (3.4) by using the submultiplicative property of the norm and invoking the backward error formula (2.5). However, (3.1) is a much weaker bound than (3.3) and (3.4). In particular, a diagonal scaling $Lx = b \rightarrow D_1 L D_2 \cdot D_2^{-1} x = D_1 b$ (where D_j is diagonal) leaves (3.3) (and, to a lesser extent, (3.4)) essentially unchanged, but can change the bound (3.1) by an arbitrary amount.

One special case in which (3.3) simplifies is when L is an M-matrix and $b \ge 0$: Lemma 3.4 of Higham [12] shows that in this case $|L^{-1}||L||x| \le (2n-1)|x|$, so (3.3) yields $|L\hat{x} - b| \le (2n-1)^2 d_n u |L||x| + O(u^2)$, and we have componentwise backward stability (to first order). More generally, (3.4) shows that the algorithm is normwise backward stable if

$$\theta(L, x) = \frac{\| \, |L||L^{-1}||L||L^{-1}||L||x| \, \|_\infty}{\|L\|_\infty \|x\|_\infty}$$

is of order 1, which is guaranteed if L is well conditioned.

A forward error bound can be obtained directly from (3.2). We find that

$$\begin{aligned} |x - \hat{x}| &\le d_n' u |M_7||M_6| \dots |M_1||b| + O(u^2) \\ &= d_n' u M(L)^{-1}|b| + O(u^2). \end{aligned}$$

This is of the form (2.11), so all the comments made about (2.11) apply to the fan-in algorithm. In addition to the normwise forward error bound (2.12), we have the bound

$$\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \le d_n u \frac{\| \, (|L^{-1}||L|)^3 |x| \, \|_\infty}{\|x\|_\infty} + O(u^2), \tag{3.5}$$

which is an immediate consequence of (3.3). Either bound in (2.12) and (3.5) can be arbitrarily larger than the other, for fixed n. An example where (3.5) is the better bound (for large n) is provided by the matrix with $l_{ij} \equiv 1$, for which $|L^{-1}||L|$ has maximum element 2 and $M(L)^{-1}|L|$ has maximum element 2^{n-1}.

It is easy to find numerical examples where the fan-in algorithm produces a large componentwise or normwise backward error. In one experiment in Matlab (for which $u \approx 1.1 \times 10^{-16}$) we used direct search [14] to construct such an example. The resulting system $Lx = b$ is of order 7, and has $b = fl(Le)$, where $x = e = (1, 1, \dots, 1)^T$. The results for the fan-in algorithm, substitution, and other methods to be described below, are given in Table 3.1. The exact solution, which we used to compute the forward errors, was obtained using a beta test version of Matlab 4's Symbolic Math Toolbox. In this example the new residual bound (3.4) is sharp, while Sameh and Brent's bound (3.1) is extremely pessimistic, being a factor approximately 10^{26} larger than (3.4). Since $\theta(L, x) \approx \mu(L, x)$, the normwise residual bounds (2.14) and (3.4) are of similar size. We mention that the instability of the

TABLE 3.1
Backward and forward errors for system of order 7.

$\kappa_\infty(L) = 1.23\text{e+}18$, $\operatorname{cond}(L, x) = 8.07\text{e+}14$, $\operatorname{cond}(L) = 3.02\text{e+}17$.
$\mu(L, x) = 1.74\text{e+}11$, $\theta(L, x) = 7.88\text{e+}11$.

	$\omega(\widehat{x})$	$\eta_\infty(\widehat{x})$	$\Vert x - \widehat{x}\Vert_\infty/\Vert x\Vert_\infty$
Substitution	8.29e-17	4.03e-19	1.64e-01
Fan-in	1.15e-03	7.76e-06	2.14e-01
Block elimination	3.69e-04	2.42e-06	1.88e-01
Power series	2.17e-05	1.43e-07	1.91e-01
Divide & conquer (B)	4.63e-04	2.96e-06	1.70e-01
Divide & conquer (D)	3.85e-04	2.53e-06	1.89e-01

methods can be made even worse by running direct search further in the construction of this example.

A comment is required concerning two papers by Tsao [23], [24]. In these papers Tsao compares the accuracy of substitution with that of the fan-in algorithm and concludes that "the parallel algorithm as proposed by Sameh and Brent [1][2] is essentially equivalent to the usual sequential algorithm as far as round-off error is concerned." This conclusion is incorrect. What Tsao shows in [23] is that expressions for the forward error $x - \widehat{x}$ can be obtained that are of the same form for both substitution and the fan-in algorithm. A consequence of Tsao's expressions is that both substitution and the fan-in algorithm satisfy a bound of the form (2.11), though this is not mentioned in [23]. Nevertheless, the numerical behaviour of substitution and the fan-in algorithm can be very different, as is clear from Table 3.1.

The fan-in method is topical because the fan-in operation is a special case of the parallel prefix operation and several fundamental computations in linear algebra are amenable to a parallel prefix-based implementation [5]. Indeed, parallel prefix has now made its way into undergraduate numerical analysis textbooks; see [3, §13.2], where a particularly clear explanation is given. The important open question of the stability of the parallel prefix implementation of Sturm sequence evaluation for the symmetric tridiagonal eigenproblem has recently been answered by Mathias [17]. Mathias shows that for positive definite matrices the relative error in a computed minor can be as large as a multiple of λ_n^{-3}, where λ_n is the smallest eigenvalue of the matrix; the corresponding bound for serial evaluation involves λ_n^{-1}. This condition cubing effect is analogous to what we see in (3.5).

4. Block elimination algorithm. In addition to the fan-in algorithm, Sameh and Brent [21] describe a parallel block elimination algorithm. It requires the same number of steps as the fan-in algorithm but roughly twice the number of processors. Its advantage is that it can be adapted to take advantage of band structure [4], [21].

The algorithm is best understood by considering the case $n = 8$. There is a preprocessing step in which L is made unit triangular: $L \leftarrow D^{-1}L$, $b \leftarrow D^{-1}b$ where $D = \operatorname{diag}(L)$. The first stage of the algorithm forms, with $L_1 = L$, $b_1 = b$,

$$L_2 = N_1 L_1 = \operatorname{diag}\left(\begin{bmatrix} 1 & \\ -l_{21} & 1 \end{bmatrix}, \begin{bmatrix} 1 & \\ -l_{43} & 1 \end{bmatrix}, \begin{bmatrix} 1 & \\ -l_{65} & 1 \end{bmatrix}, \begin{bmatrix} 1 & \\ -l_{87} & 1 \end{bmatrix}\right) L_1$$

$$= \begin{bmatrix} I_2 & & & \\ L_{21}^{(2)} & I_2 & & \\ L_{31}^{(2)} & L_{32}^{(2)} & I_2 & \\ L_{41}^{(2)} & L_{42}^{(2)} & L_{43}^{(2)} & I_2 \end{bmatrix}, \qquad L_{ij} \in \mathbb{R}^{2\times 2}, \qquad b_2 = N_1 b_1.$$

The next stage forms

[2]Reference [1] in Tsao's paper is [21] in the present paper.

$$L_3 = N_2L_2 = \mathrm{diag}\left(\begin{bmatrix} I_2 & \\ -L_{21}^{(2)} & I_2 \end{bmatrix}, \begin{bmatrix} I_2 & \\ -L_{43}^{(2)} & I_2 \end{bmatrix}\right) L_2 = \begin{bmatrix} I_4 & \\ L_{21}^{(3)} & I_4 \end{bmatrix}, \qquad b_3 = N_2b_2,$$

and the final stage forms

$$x = b_4 = N_3b_3 = \begin{bmatrix} I_4 & \\ -L_{21}^{(3)} & I_4 \end{bmatrix} b_3.$$

Thus in $\lceil \log n \rceil$ stages L is reduced to the identity by row operations and b is transformed to $L^{-1}b = x$.

Error analysis for this algorithm can be expressed as follows. We take $n = 2^k$ and assume, without loss of generality, that $l_{ii} \equiv 1$. Using (2.1), we have

$$\widehat{x} = (\widehat{N}_k + \Delta_k)(\widehat{N}_{k-1} + \Delta_{k-1}) \dots (\widehat{N}_1 + \Delta_1)b, \qquad |\Delta_i| \le c_n u|\widehat{N}_i| + O(u^2),$$

which yields

$$(4.1) \qquad \widehat{x} = \widehat{N}_k\widehat{N}_{k-1} \dots \widehat{N}_1 b + \sum_{i=1}^{k} \widehat{N}_k \dots \widehat{N}_{i+1}\Delta_i\widehat{N}_{i-1} \dots \widehat{N}_1 b + O(u^2).$$

We now need to relate $\widehat{N}_i$ to N_i. Observe that, from the description of the method above, N_i has the form $N_i = (p_{rs}(L))$, where each entry $p_{rs}(L)$ is a multivariate polynomial in the elements of L. It is not hard to see that $p_{rs}(M(L))$ contains only nonpositive terms $(r > s)$ and that

$$\widehat{N}_i = N_i + E_i, \qquad |E_i| \le c_n' u\big(p_{rs}(M(L))\big) = c_n' u N_i(M(L)).$$

Thus (4.1) can be rewritten

$$(4.2) \qquad \widehat{x} - x = \sum_{i=1}^{k} N_k \dots N_{i+1}(E_i + \Delta_i)N_{i-1} \dots N_1 b + O(u^2),$$

which gives

$$\begin{aligned} |x - \widehat{x}| &\le c_n'' u \sum_{i=1}^{k} |N_k| \dots |N_{i+1}|(N_i(M(L)) + |N_i|)|N_{i-1}| \dots |N_1||b| + O(u^2) \\ &\le 2c_n'' u N_k(M(L))N_{k-1}(M(L)) \dots N_1(M(L))|b| + O(u^2) \\ (4.3) \qquad &= 2c_n'' u M(L)^{-1}|b| + O(u^2). \end{aligned}$$

This bound is of the form (2.11), so all the comments made about (2.11) apply to the block elimination method.

In [4], Chen, Kuck, and Sameh use the banded system variant of the block elimination algorithm. They explain that a forward error bound can be obtained that is proportional to $\sigma^{n/m}$, where $\sigma = \max_{i,j} |l_{ij}|$ (here $l_{ii} \equiv 1$) and m is approximately half the bandwidth. An exponential bound of this form is weaker than (4.3), and obtainable from it, because, in general, $M(L)^{-1} \le W(L)^{-1}$, where $W(L) = (w_{ij})$ is defined by $w_{ii} = |l_{ii}|$ and $|w_{ij}| = -\max_{r,s} |l_{rs}| = -\sigma$ $(i \ne j)$, and if $l_{ii} \equiv 1$ then $\max_{i,j} |W(L)^{-1}|_{ij} = \sigma(\sigma + 1)^{n-2}$.

From (4.2) the following bound for the residual can be obtained:

$$(4.4) \quad |L\widehat{x} - b| \le c_n'' u \sum_{i=1}^{k} |N_1^{-1} \dots N_i^{-1}|(N_i(M(L)) + |N_i|)|N_i^{-1} \dots N_k^{-1}||x| + O(u^2).$$

This does not lead to a residual bound any more useful than can be obtained from (4.3). The reason we can obtain more satisfactory residual bounds for the partitioned inverse method and fan-in algorithms is that they employ a factorization of L that is based on structure, and hence is little affected by replacing the factors by their absolute values. The factorization of L^{-1} used by the block elimination algorithm relies on algebraic relations that are destroyed if the absolute values of the factors are taken, and so error bounds cannot be expressed solely in terms of L and L^{-1}.

5. Power series method. Heller [10] describes the following method for solving $Lx = b$, which he credits to Heller (1974 technical report) and Orcutt (1974 dissertation). Let $L = D(I - M)$ be of order $n = 2^k$, where $D = \operatorname{diag}(L)$ and M is strictly lower triangular (hence $M^n = 0$). Then

$$\begin{aligned} x &= (I - M)^{-1}D^{-1}b \\ &= (I + M + \cdots + M^{n-1})D^{-1}b \\ &= (I + M^{2^{k-1}})(I + M^{2^{k-2}}) \dots (I + M)D^{-1}b. \end{aligned}$$

The powers M^2, M^4, $\dots$, $M^{2^{k-1}}$, are formed by repeated squaring. This method can be implemented in $\log^2 n + \log n$ time steps on $n^3 + n^2$ processors [10].

It is straightforward to show that the computed powers $\widehat{M}_i = fl(M^{2^i})$ satisfy

$$\widehat{M}_i = M^{2^i} + E_i, \qquad |E_i| \le d_n u |M|^{2^i} + O(u^2). \tag{5.1}$$

This bound also holds if M^{2^i} is evaluated as a product of 2^i terms $MM \dots M$; a bound that exploits the repeated squaring implementation is complicated to express and yields no improvement to the final bound we will derive. The computed solution $\widehat{x}$ satisfies

$$\widehat{x} = (I + M^{2^{k-1}} + \Delta_{k-1})(I + M^{2^{k-2}} + \Delta_{k-2}) \dots (I + M + \Delta_1)D^{-1}b,$$

where, using (2.1) and (5.1),

$$\begin{aligned} |\Delta_i| &\le c_n u |I + M^{2^i}| + d_n u |M|^{2^i} + O(u^2) \\ &\le c_n' u (I + |M|^{2^i}) + O(u^2). \end{aligned}$$

Therefore

$$x - \widehat{x} = -\sum_{i=1}^{k-1} (I + M^{2^{k-1}}) \dots (I + M^{2^{i+1}})\Delta_i (I + M^{2^{i-1}}) \dots (I + M)D^{-1}b,$$

and so

$$\begin{aligned} |x - \widehat{x}| &\le (k-1)c_n' u (I + |M|^{2^{k-1}})(I + |M|^{2^{k-2}}) \dots (I + |M|)|D|^{-1}|b| + O(u^2) \\ &= c_n'' u M(L)^{-1}|b| + O(u^2). \end{aligned} \tag{5.2}$$

This bound is of the form (2.11). The comments made in the last paragraph of §4 are applicable to the power series method too.

In the numerical example of Table 3.1 both the block elimination method and the power series method are unstable, and the normwise residual bound (2.14) is reasonably close to equality.

6. Matrix inversion by divide and conquer. Borodin and Munro [2] and Heller [10] discuss a divide and conquer method for inverting a triangular matrix based on the formulae

$$L = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix}, \quad X = L^{-1} = \begin{bmatrix} L_{11}^{-1} & 0 \\ -L_{22}^{-1}L_{21}L_{11}^{-1} & L_{22}^{-1} \end{bmatrix}.$$

The diagonal blocks of L_{11} and L_{22} are chosen to be of equal size (or sizes differing by 1 if the dimension is odd) and the inversion of these blocks is done by the same method recursively; the (2,1) block is evaluated by matrix multiplication. The method can be implemented in $O(\log^2 n)$ time steps on $O(n^3)$ processors. If we are prepared to give up the $O(\log^2 n)$ complexity then we can consider alternative ways to evaluate the (2,1) block of L^{-1}. But, as we now show, how this block is evaluated is critical to the stability of the algorithm. There are, in fact, *seven* ways to evaluate $X_{21} = -L_{22}^{-1}L_{21}L_{11}^{-1}$. Here we use Matlab notation: $A\backslash B$ denotes solving $AX = B$ (by substitution when A is triangular) and A/B denotes solving $A = XB$ (by substitution when B is triangular). The seven ways are as follows.

A: $X_{21} = -X_{22}L_{21}X_{11}$,
B: $X_{21} = -L_{22}\backslash(L_{21}X_{11})$,
C: $X_{21} = -(L_{22}\backslash L_{21})X_{11}$,
D: $X_{21} = -(X_{22}L_{21})/L_{11}$,
E: $X_{21} = -X_{22}(L_{21}/L_{11})$,
F: $X_{21} = -(L_{22}\backslash L_{21})/L_{11}$,
G: $X_{21} = -L_{22}\backslash(L_{21}/L_{11})$.

Methods A, B, and D correspond to Methods 2B, 1B, and 2C, respectively, of Du Croz and Higham [6]. The latter three methods are not implemented recursively, but the same error analysis applies to Methods A, B, and D here. For Methods B and D, under conditions on the bottom level of recursion that are described below, the computed $\widehat{X}$ satisfies [6]

$$\text{B:} \quad |L\widehat{X} - I| \le c_n u|L||\widehat{X}| + O(u^2), \tag{6.1}$$

$$\text{D:} \quad |\widehat{X}L - I| \le c_n u|\widehat{X}||L| + O(u^2). \tag{6.2}$$

These bounds are the best that we can expect and correspond to componentwise backward stability. In general, a componentwise stable inversion method will satisfy either a right residual bound of the form (6.1) or a left residual bound of the form (6.2), but not both bounds. We give the proof of (6.1). Let $\Delta(A, B)$ denote a matrix bounded according to

$$|\Delta(A, B)| \le c_n u|A||B| + O(u^2).$$

Note that

$$fl(AB) = AB + \Delta(A, B),$$

and if T is triangular then $\widehat{X} = fl(T\backslash B)$ implies

$$T\widehat{X} + \Delta(T, \widehat{X}) = B.$$

For Method B we have

$$L_{22}\widehat{X}_{21} + \Delta(L_{22}, \widehat{X}_{21}) = -L_{21}\widehat{X}_{11} + \Delta(L_{21}, \widehat{X}_{11}),$$

that is,

$$\begin{aligned} |L_{22}\widehat{X}_{21} + L_{21}\widehat{X}_{11}| &\le c_n u\big(|L_{22}||\widehat{X}_{21}| + |L_{21}||\widehat{X}_{11}|\big) + O(u^2) \\ &= c_n u\big(|L||\widehat{X}|\big)_{21} + O(u^2). \end{aligned}$$

Therefore $|L\widehat{X}-I| \le c_n u|L||\widehat{X}|+O(u^2)$ provided that $|L_{ii}\widehat{X}_{ii}-I| \le c_n u|L_{ii}||\widehat{X}_{ii}|+O(u^2)$ for $i = 1, 2$, which is true, by induction, if these inequalities are satisfied at the bottom level of recursion. The proof of (6.2) is entirely analogous. It is clear that the crucial part of the analysis is bounding the (2,1) block of the left or right residual.

None of Methods A, C, E, F, and G satisfies (6.1) or (6.2). For example, for Method E we have, with $Y = L_{21}/L_{11}$,

$$\begin{aligned}\widehat{Y}L_{11} &= L_{21} + \Delta(\widehat{Y}, L_{11}),\\ \widehat{X}_{21} &= -\widehat{X}_{22}\widehat{Y} + \Delta(\widehat{X}_{22}, \widehat{Y}).\end{aligned}$$

Thus

$$\widehat{X}_{21}L_{11} = -\widehat{X}_{22}\widehat{Y}L_{11} + \Delta(\widehat{X}_{22}, \widehat{Y})L_{11},$$

that is,

$$\widehat{X}_{21}L_{11} + \widehat{X}_{22}L_{21} = -\widehat{X}_{22}\Delta(\widehat{Y}, L_{11}) + \Delta(\widehat{X}_{22}, \widehat{Y})L_{11}.$$

Hence we have the bound

$$\begin{aligned}|\widehat{X}_{21}L_{11} + \widehat{X}_{22}L_{21}| &\le 2c_n u|\widehat{X}_{22}||\widehat{Y}||L_{11}| + O(u^2)\\ &= 2c_n u|\widehat{X}_{22}||L_{21}L_{11}^{-1}||L_{11}| + O(u^2),\end{aligned}$$

which has the term $|L_{21}L_{11}^{-1}||L_{11}|$ instead of the term $|L_{21}|$ that we require for a small left residual. A bound for the right residual is even less satisfactory. Similar analysis for Methods A, C, F, and G shows that for none of these methods can a small componentwise left or right residual bound be obtained. Thus Methods B and D are the only componentwise stable methods for computing L^{-1} of the seven Methods A–G.

Now we consider using the computed inverse to solve $Lx = b$. We obtain

$$\widehat{x} = fl(\widehat{X}b) = \widehat{X}b + f, \qquad |f| \le c_n u|\widehat{X}||b| + O(u^2), \tag{6.3}$$

so that

$$L\widehat{x} = L\widehat{X}b + Lf.$$

For Method B, $L\widehat{X} - I$ is bounded in (6.1), and we have

$$|L\widehat{x} - b| \le 2c_n u|L||\widehat{X}||b| + O(u^2). \tag{6.4}$$

This is the best residual bound we could expect because even if we multiply b by the exact inverse we cannot avoid the term Lf that contributes a term $c_n u|L||\widehat{X}||b|$ to the residual bound. If $|L^{-1}||b| \approx |L^{-1}b| = |x|$, inequality (6.4) implies componentwise backward stability. For Method D, which has a small left residual, as shown by (6.2), the best residual bound we can obtain for $\widehat{x}$ is (by writing $L\widehat{X}b = L(\widehat{X}L)x$)

$$|L\widehat{x} - b| \le 2c_n u|L||\widehat{X}||L||x| + O(u^2), \tag{6.5}$$

which is weaker than (6.4) to the extent that $|b| \le |L||x|$. Note that this bound is stronger than the bound (3.3) for the fan-in algorithm (which has an extra term $|L||L^{-1}|$).

Both Methods B and D satisfy the forward error bound

$$|x - \widehat{x}| \le d_n u|L^{-1}||L||L^{-1}||b| + O(u^2). \tag{6.6}$$

This follows from (6.4) for Method B, and for both methods from (6.3) together with the bound $|L^{-1} - \widehat{X}| \le c_n u |L^{-1}||L||L^{-1}| + O(u^2)$ from (6.1) and (6.2).

The inequality (2.11) can be shown to hold for each of Methods A–G, provided that the inversion method used at the bottom level of recursion itself satisfies (2.11) when used as an equation solver. The proof involves showing that $|\widehat{X} - L^{-1}| \le c_n u M(L)^{-1} + O(u^2)$ (for each method A–G) and then using (6.3).

We give a numerical example that illustrates the analysis. Here, L is the 12th power of a random 25×25 lower triangular matrix from the normal $N(0, 1)$ distribution (L is generated in Matlab 3.5 by the statements `rand('normal'); rand('seed',71); L = tril(tril(rand(25))^12)`.) This particular form of ill-conditioned and badly scaled matrix had been found in [6] (by trial and error) to cause instability in some triangular matrix inversion routines. For each of Methods A–G (all of which were recurred down to the level $n = 1$), Table 6.1 shows the left (L) and right (R) componentwise and normwise relative residuals, the left ones of which are given by

$$\min\{\epsilon : |\widehat{X}L - I| \le \epsilon |\widehat{X}||L|\} \qquad \text{and} \qquad \frac{\|\widehat{X}L - I\|_\infty}{\|\widehat{X}\|_\infty \|L\|_\infty}.$$

The underlines in Table 6.1 indicate the residuals that we know must be small, by (6.1) and (6.2). Most of the other residuals are large; those normwise ones that are not are small "by chance" and are found to be large in other examples.

We also solved two linear systems: $Lx = b$, where $b = fl(Le)$, and $Ly = e$. The componentwise and normwise backward errors are tabulated in Table 6.2. The $\omega(\widehat{x})$ and $\omega(\widehat{y})$ values illustrate clearly that Method B can be superior to all the other methods as a means for solving $Lx = b$, but that it may nevertheless be componentwise unstable (the example in Table 3.1 shows that Method B can also be unstable in the normwise sense). The small value of $\omega(\widehat{y})$ for Method B is predicted by (6.4), since $|L^{-1}||b| \approx |y|$ in this example. Similarly, (6.5) correctly predicts a much larger value of $\omega(\widehat{y})$ for Method D than for Method B, since $|b| \ll |L||y|$.

To summarize, only for Methods B and D can we guarantee a small componentwise left residual (Method D) or right residual (Method B). For solving a linear system Method B is preferable to Method D as it has a smaller residual bound. Thus the error analysis shows that how the (2,1) block of L^{-1} is computed in the divide and conquer method greatly affects the stability of the computed inverse or the solution to a linear system.

TABLE 6.1
Residuals for 25×25 *random L.*

$\kappa_\infty(L) = 6.14e{+}44$, $\mathrm{cond}(L) = 8.63e{+}43$, $\mathrm{cond}(L^{-1}) = 1.24e{+}41$.

Method	L (comp.)	R (comp.)	L (norm)	R (norm)
A	5.73e-05	8.23e-03	4.44e-06	9.01e-07
B	9.39e-01	$\underline{\text{1.18e-16}}$	1.05e-01	$\underline{\text{1.19e-20}}$
C	9.51e-01	1.16e-03	3.44e-08	5.49e-18
D	$\underline{\text{1.11e-16}}$	1.60e-02	$\underline{\text{6.68e-18}}$	3.19e-06
E	1.18e-07	6.80e-01	3.06e-11	3.34e-07
F	9.51e-01	3.73e-03	3.63e-18	1.84e-12
G	1.27e-02	6.80e-01	3.66e-04	4.27e-18

7. Conclusions.

Whereas the substitution algorithm has perfect numerical stability, all the parallel methods presented here can be unstable, depending on the data. As is often the case in numerical linear algebra, these triangular system solvers gain parallelism at the cost of

TABLE 6.2
Linear system backward errors.

$\text{cond}(L, x) = 8.63\text{e+}43$, $\text{cond}(L^{-1}, x) = 1.24\text{e+}41$.
$\text{cond}(L, y) = 1.22\text{e+}16$, $\text{cond}(L^{-1}, y) = 7.54\text{e+}28$.

Method	$\omega(\widehat{x})$	$\eta_\infty(\widehat{x})$	$\omega(\widehat{y})$	$\eta_\infty(\widehat{y})$
A	4.51e-03	9.10e-07	4.48e-03	9.02e-07
B	5.73e-08	7.74e-22	1.65e-16	2.13e-20
C	1.28e-05	8.39e-16	3.76e-08	5.49e-18
D	5.68e-01	5.85e-05	1.60e-02	3.19e-06
E	5.79e-01	3.98e-05	1.37e-01	2.78e-07
F	1.00e+00	5.54e-05	1.76e-03	9.06e-14
G	6.08e-01	2.42e-18	1.37e-01	1.64e-18

stability [5]. The divide and conquer Methods B and D satisfy the strongest residual bounds of the parallel methods we have considered, with stability for this class of methods depending on precisely how the (2,1) block of the inverse is evaluated at each level of recursion. (We omit the partitioned inverse method from this comparison since its bound (2.9) is difficult to compare with the rest and depends crucially on the parameter m.) The fan-in algorithm has the next best residual bound, (3.3), which can be of order $\kappa_\infty(L)^2\|L\|_\infty\|x\|_\infty u$. Methods B and D and the fan-in algorithm are all guaranteed to be stable when L is well conditioned. For the block elimination and power series methods we were unable to derive stronger bounds than

$$|x - \widehat{x}| \le d_n u M(L)^{-1}|b| + O(u^2) \tag{7.1}$$

and the corresponding residual bound, which hold for all the methods considered here. It has not been previously appreciated that (7.1) holds for the wide class of methods that do not rely on algebraic cancellation. It is effectively a "universal" forward error bound for triangular equation solvers.

In the numerical examples we have mainly reported backward errors and residuals. Since triangular system solvers are normally used as part of a larger computation, backward stability is probably the most important requirement. In our limited experience, the forward errors for the parallel methods reported here tend to be at most $\kappa_\infty(L)u$, even when the backward error is large. We have not found any numerical examples where the fan-in algorithm has a forward error of order $\kappa_\infty(L)^3u$, or even $\kappa_\infty(L)^2u$.

While we have not attempted to gauge the average-case stability of these parallel methods, we can offer the following summary of their behaviour. All the methods can be arbitrarily unstable, but they achieve perfect stability when L is an M-matrix and $b \ge 0$ (by virtue of (7.1)), and they often yield surprisingly stable and accurate solutions, even for very ill-conditioned problems. Therefore the parallel methods should not be ruled out for practical use purely on stability grounds, particularly as it is easy to compute the normwise or componentwise backward error a posteriori to test the stability of a computed solution. We note that iterative refinement in fixed precision is a possible means for stabilizing any of the methods (the theory of [13, §2] is applicable). However, for all except the divide and conquer methods, iterative refinement significantly increases the cost of the solution process.

To our knowledge, none of the fan-in, block elimination, and divide and conquer algorithms has been implemented on a modern parallel machine and its speed compared with that of substitution. It would be an interesting exercise to make such a comparison and therefore to determine whether the parallel methods merit serious consideration as practical algorithms.

Acknowledgments. I thank Jim Demmel and Des Higham for their helpful comments on a draft of this manuscript.

REFERENCES

[1] F. L. Alvarado, A. Pothen, and R. S. Schreiber, *Highly parallel sparse triangular solution*, in Graph Theory and Sparse Matrix Computations, J. A. George, J. R. Gilbert, and J. W. H. Liu, eds., IMA Volumes in Mathematics and its Applications, Vol. 56, Springer-Verlag, New York, 1993, pp. 141–158.

[2] A. Borodin and I. Munro, *The Computational Complexity of Algebraic and Numeric Problems*, American Elsevier, New York, 1975.

[3] J. L. Buchanan and P. R. Turner, *Numerical Methods and Analysis*, McGraw-Hill, New York, 1992.

[4] S. C. Chen, D. J. Kuck, and A. H. Sameh, *Practical parallel band triangular system solvers*, ACM Trans. Math. Software, 4 (1978), pp. 270–277.

[5] J. W. Demmel, *Trading off parallelism and numerical stability*, in Linear Algebra for Large Scale and Real-Time Applications, M. S. Moonen, G. H. Golub, and B. L. D. Moor, eds., NATO ASI Series E, Kluwer Academic Publishers, Vol. 232, Dordrecht, 1993, pp. 49–68.

[6] J. J. Du Croz and N. J. Higham, *Stability of methods for matrix inversion*, IMA J. Numer. Anal., 12 (1992), pp. 1–19.

[7] A. Edelman, *Large dense numerical linear algebra in* 1993: *The parallel computing influence*, Internat. J. Supercomput. Appl., 7 (1993), pp. 113–128.

[8] S. C. Eisenstat, M. T. Heath, C. S. Henkel, and C. H. Romine, *Modified cyclic algorithms for solving triangular systems on distributed-memory multiprocessors*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 589–600.

[9] M. T. Heath and C. H. Romine, *Parallel solution of triangular systems on distributed-memory multiprocessors*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 558–588.

[10] D. Heller, *A survey of parallel algorithms in numerical linear algebra*, SIAM Rev., 20 (1978), pp. 740–777.

[11] N. J. Higham, *A survey of condition number estimation for triangular matrices*, SIAM Rev., 29 (1987), pp. 575–596.

[12] ———, *The accuracy of solutions to triangular systems*, SIAM J. Numer. Anal., 26 (1989), pp. 1252–1265.

[13] ———, *Iterative refinement enhances the stability of QR factorization methods for solving linear equations*, BIT, 31 (1991), pp. 447–468.

[14] ———, *Optimization by direct search in matrix computations*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 317–333.

[15] N. J. Higham and A. Pothen, *Stability of the partitioned inverse method for parallel solution of sparse triangular systems*, SIAM J. Sci. Comput., 15 (1994), pp. 139–148.

[16] G. Li and T. F. Coleman, *A new method for solving triangular systems on distributed-memory message-passing multiprocessors*, SIAM J. Sci. Statist. Comput., 10 (1989), pp. 382–396.

[17] R. Mathias, *The instability of parallel prefix matrix multiplication*, SIAM J. Sci. Comput., 16 (1995).

[18] W. Oettli and W. Prager, *Compatibility of approximate solution of linear equations with given error bounds for coefficients and right-hand sides*, Numer. Math., 6 (1964), pp. 405–409.

[19] J. L. Rigal and J. Gaches, *On the compatibility of a given solution with the data of a linear system*, J. Assoc. Comput. Mach., 14 (1967), pp. 543–548.

[20] C. H. Romine and J. M. Ortega, *Parallel solution of triangular systems of equations*, Parallel Comput., 6 (1988), pp. 109–114.

[21] A. H. Sameh and R. P. Brent, *Solving triangular systems on a parallel computer*, SIAM J. Numer. Anal., 14 (1977), pp. 1101–1113.

[22] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

[23] N. Tsao, *On the accuracy of solving triangular systems in parallel*, Appl. Numer. Math., 7 (1991), pp. 207–215.

[24] ———, *On the accuracy of solving triangular systems in parallel*—II, Appl. Numer. Math., 9 (1992), pp. 73–89.

[25] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, 1965.