Solving Sparse Linear Systems with Sparse Backward Error

by

M. Arioli*, J. W. Demmel**, I. S. Duff

ABSTRACT

When solving sparse linear systems, it is desirable to produce the solution of a nearby sparse problem with the same sparsity structure. This kind of backward stability helps guarantee, for example, that one has solved a problem with the same physical connectivity as the original problem. Theorems of Oettli, Prager and Skeel show that one step of iterative refinement, even with single precision accumulation of residuals, guarantees such a small backward error if the final matrix is not too ill-conditioned and the solution components do not vary too much in magnitude. We incorporate these results into the stopping criterion of the iterative refinement step of a direct sparse matrix solver and verify by numerical experiments that the algorithm frequently stops after one step of iterative refinement with a componentwise relative backward error at the level of the machine precision. Furthermore, calculating this stopping criterion is very inexpensive. We also discuss a condition estimator corresponding to this new backward error which provides an error estimate for the computed solution. This error estimate is generally tighter than estimates provided by standard condition estimators. We also consider the effects of using a drop tolerance during the LU decomposition.

---

* This author was visiting Harwell and was funded by a grant from the Italian National Council of Research (CNR). Istituto di Elaborazione dell'Informazione – CNR, via S. Maria 46, 56100 Pisa, Italy.

** Computer Science Department, Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10012, USA

Computer Science and Systems Division,
Harwell Laboratory,
Didcot,
Oxon OX11 0RA.

February 1988.

CONTENTS

1 Introduction

When solving systems of n linear equations $\mathbf{Ax}=\mathbf{b}$ by means of Gaussian elimination with pivoting, a classical analysis, (Wilkinson 1961), shows that we should expect to get the exact solution $\hat{\mathbf{x}}$ of a slightly different linear system $(\mathbf{A}+\delta\mathbf{A})\hat{\mathbf{x}}=\mathbf{b}+\delta\mathbf{b}$ where $\delta\mathbf{A}$ and $\delta\mathbf{b}$ are both small with respect to $\mathbf{A}$ and $\mathbf{b}$. By small we mean small in norm, i.e. $\|\delta\mathbf{A}\| \le k\varepsilon\|\mathbf{A}\|$ and $\|\delta\mathbf{b}\| \le k\varepsilon\|\mathbf{b}\|$ where $\|\,.\,\|$ is a matrix norm, ε is the machine precision (that is, the greatest positive number such that $fl(1+\varepsilon)$, the floating-point representation of $(1+\varepsilon)$, equals 1) and k is the product of the pivot growth factor and a modestly growing function of the dimension n. This classical view permits any entry of $\delta\mathbf{A}$ or $\delta\mathbf{b}$ to be equally large, and in particular $\mathbf{A}+\delta\mathbf{A}$ may be dense even if $\mathbf{A}$ is quite sparse. This is unsatisfactory because zero entries of $\mathbf{A}$ may represent nonexistent physical connections in a system being modelled, and so may be known exactly.

A more satisfying approach to backward error than merely bounding $\|\delta\mathbf{A}\|$ and $\|\delta\mathbf{b}\|$ would permit the user to specify scaling factors $e_{ij} \ge 0$ and $f_i \ge 0$ for each entry of $\delta\mathbf{A}$ and $\delta\mathbf{b}$, and would compute the smallest $\omega \ge 0$ such that

$$|\delta a_{ij}| \le \omega e_{ij}\ ,\ |\delta b_i| \le \omega f_i\ . \tag{1}$$

By setting some e_{ij} to zero, we can insist that, if $\omega<\infty$, the corresponding a_{ij} are known exactly. For example, if $e_{ij}=|a_{ij}|$ and $f_i=|b_i|$, ω bounds the relative perturbation in each component of $\mathbf{A}$ and $\mathbf{b}$ needed to make $\hat{\mathbf{x}}$ an exact solution, and, in particular, $\delta\mathbf{A}$ and $\delta\mathbf{b}$ have the same sparsity structures as $\mathbf{A}$ and $\mathbf{b}$. We will call this ω the *componentwise relative backward error*. It is important to use this different error estimate when considering these restricted perturbations, since Gear (1975) has shown that the conventional error bounds are not appropriate in this case. It turns out that the backward error ω is quite easy to compute, and in fact costs as little as two matrix-vector multiplications.

In the following, if $\mathbf{u}$ and $\mathbf{v}$ are vectors of entries u_i and v_i and $\mathbf{Q}$ and $\mathbf{P}$ are matrices of entries q_{ij} and p_{ij}, $|\mathbf{u}|$ is the vector of entries $|u_i|$, $|\mathbf{Q}|$ is the matrix of entries $|q_{ij}|$. $\mathbf{u} \le \mathbf{v}$ means $u_i \le v_i$ for all i, and $\mathbf{Q} \le \mathbf{P}$ means $q_{ij} \le p_{ij}$ for all i and j.

Theorem 1: [Oettli and Prager 1964] . The smallest ω satisfying (1) is given by

$$\omega = \max_i \frac{|\mathbf{A}\hat{\mathbf{x}}-\mathbf{b}|_i}{(\mathbf{E}|\hat{\mathbf{x}}|+\mathbf{f})_i}\ . \tag{2}$$

In this expression, 0/0 should be interpreted as 0 and $\zeta/0$ $(\zeta \ne 0)$ as infinity. $\omega=\infty$ implies that no ω satisfying (1) exists. In particular, the smallest componentwise relative perturbation of $\mathbf{A}$ and $\mathbf{b}$ that makes $\hat{\mathbf{x}}$ an exact solution is

$$\omega = \max_i \frac{|\mathbf{A}\hat{\mathbf{x}}-\mathbf{b}|_i}{(|\mathbf{A}|\,|\hat{\mathbf{x}}|+|\mathbf{b}|)_i}\ . \tag{3}$$

Thus, this theorem gives an a posteriori measure of the backward error that is cheap to compute.

Gaussian elimination with pivoting does not guarantee that the backward error ω will be small

for all possible E and f. However, a theorem of Skeel (1980) shows that as long as A is not too ill-conditioned, and as long as the quantities $(|A||\hat{x}|)_i$ in the denominator of (3) do not vary too much in magnitude, then one step of iterative refinement is enough to guarantee that ω will be small for the componentwise relative backward error in (3). This is true even if the residual $r = A\hat{x} - b$ is computed in the same arithmetic precision as used for the Gaussian elimination. The actual conditions under which the following theorem is true are quite complicated, and we refer for details to Skeel (1980, Theorem 5.1)

Theorem 2: [Skeel 1980] Let ε be the machine precision, and let the arithmetic be such that the floating-point result $fl(a \Diamond b)$ of the operation $a \Diamond b, (\Diamond \in \{+,-,\times,/\})$ satisfies $fl(a \Diamond b) = (a \Diamond b)(1+e)$, with $|e| \le \varepsilon$. There is a function $f(A,b)$, typically behaving as $O(n)$, such that when the product of $\hat{\kappa}(A) \equiv \| |A| |A^{-1}| \|$ and $\sigma(A,x) \equiv \max_i (|A||x|)_i / \min_i (|A||x|)_i$ is less than $(f(A,b)\,\varepsilon)^{-1}$, and there is no overflow or underflow, the following iterative refinement algorithm will converge after one update of $\hat{x}$:

Solve $Ax = b$ using Gaussian elimination, obtaining solution $\hat{x}$ and saving the LU factors;

Compute the residual $r = A\hat{x} - b$ (using arithmetic of machine precision ε);

while $\omega = \max_i |r_i| / (|A||\hat{x}| + |b|)_i > (n+1)\varepsilon$ do

begin

Solve $Ad = r$ for d using the saved LU factors of A;

Update $\hat{x} = \hat{x} - d$;

end;

This theorem may also be extended to take into account underflow and the possibility that, for lack of a guard digit in the hardware, we can only assert that

$$fl(a \pm b) = a(1+e_1) \pm b(1+e_2),$$

where $|e_i| \le \varepsilon$, (Demmel 1984).

For sparse systems, it is also possible to improve the stopping criterion of Theorem 2 by changing n to γ, the maximum number of nonzero entries in one row of A.

Note that this theorem contradicts the usual advice that iterative refinement is not worth doing unless the residual $r = A\hat{x} - b$ is computed using arithmetic of machine precision ε^2. Note also that the theorem does not say that the refined solution will be more accurate, just that it reflects the structure of the original problem more closely than the unrefined solution. If each of the nonzero entries of the original A is uncertain in its least significant bit and if $\omega \approx \varepsilon$, then one could say that one has computed the solution as accurately as the data warrants, since the answer is exact for a problem indistinguishable from the problem one really wanted to solve.

To use Theorem 2 as the basis of a practical scheme for solving sparse linear systems, some

modifications are necessary. In particular, when solving sparse linear systems where both A and b are sparse (or b has components of widely varying magnitude), it often happens that the quantity $\sigma(A,x)$ in Theorem 2 is huge, and convergence does not occur. Therefore, we must make another choice for f, taking less account of the smaller components b_i. This can be done quite easily using a modification of Theorem 1, and is discussed in Section 2.2.

There is a new condition number corresponding to the new definition of backward error in (1). In the case of $E=|A|$ and $f=|b|$, this condition number is just $\| |A^{-1}||A| \|$. This new condition number is no larger than the traditional condition number $\| A^{-1} \| \, \| A \|$. In fact, it may be much smaller than $\| A^{-1} \| \, \| A \|$ if the rows of A are badly scaled. Thus, combining the componentwise relative backward error with the new condition number, we obtain bounds for the real error which are independent of row scaling. We discuss this further in Section 2.1.

It has become common to use inexpensive estimators for the usual condition number $\| A^{-1} \| \, \| A \|$ to estimate a bound for the error in the computed solution of $Ax=b$ (Cline *et al.* 1979, Higham 1987a, Dongarra *et al.* 1979). In Section 4, we present an inexpensive and accurate condition estimator for the new condition number $\| |A^{-1}||A| \|$ (and its variations). The new condition estimator is based on recent work by Hager (1984) and Higham (1987).

Finally, we tested our algorithm and associated condition estimator in a modified version of the sparse linear system solver MA28 (Duff 1977) from the Harwell Subroutine Library, which uses the pivotal strategy of Markowitz (1957) and a relative pivot test

$$|a_{kk}^{(k)}| \geq u \max_{j>k} |a_{kj}^{(k)}|$$

on the elements $a_{kj}^{(k)}$ of the k-th pivot row. Here u (the threshold parameter) is a preassigned factor, usually set to 0.1. MA28 can also drop entries of L and U that fall below a *'drop tolerance'* in order to further decrease fill-in. The L and U factors are used to solve $Ax=b$ for x by forward and back substitution in the usual way, followed by some steps of iterative refinement. We report on the details of the experiments in Section 5. Our conclusion is that a stopping criterion like the one in Theorem 2 (but suitably modified as discussed in Section 5) is a reliable and inexpensive stopping criterion for iterative refinement, often stopping after one or no update of x. When drop tolerances are used and we have convergence, the rate of convergence degrades slightly but is still quite good. The new condition estimator of Section 4 also proves to be inexpensive to calculate and is an accurate estimate on our test matrices, usually providing good accuracy for the cost of a few forward and back substitutions with the LU factors of A.

The rest of this paper is organized as follows. Section 2 discusses the componentwise backward error further and also the conditioning of $Ax=b$ with respect to this backward error measure. Section 3 examines how the statement of Theorem 2 must change when either the floating-point arithmetic has no guard digit (such as on the CRAY) or underflow occurs. Section 4 presents a condition estimator corresponding to componentwise relative backward error. Section 5 discusses the numerical experiments. Section 6 has conclusions.

2 Backward error and conditioning

2.1 Condition number

The condition number of a problem is the least upper bound of the ratio of the norm of perturbation in the solution to the norm of the perturbation in the input data, in the limit as the perturbation in the input data goes to zero. To compute it, we need a norm for the perturbation $\Delta \mathbf{x}$ in the solution as well as a norm for the perturbations $\Delta \mathbf{A}$ and $\Delta \mathbf{b}$ in the input data. The norm for the input data will depend on $\mathbf{E}$ and $\mathbf{f}$ as described above: $\| (\Delta \mathbf{A}, \Delta \mathbf{b}) \|_{\mathbf{E},\mathbf{f}}$ is defined as the smallest ω such that $|\Delta \mathbf{A}| \le \omega \mathbf{E}$ and $|\Delta \mathbf{b}| \le \omega \mathbf{f}$. For the norm of the output, we choose the usual sup norm $\| \mathbf{x} \|_\infty \equiv \max_i |x_i|$, in order to cater for zero components in $\mathbf{x}$. With this notation we can write

$$\kappa_{\mathbf{E},\mathbf{f}}(\mathbf{A},\mathbf{b}) \equiv \limsup_{\substack{\Delta \mathbf{A} \to 0 \\ \Delta \mathbf{b} \to 0}} \frac{\| \Delta \mathbf{x} \|_\infty / \| \mathbf{x} \|_\infty}{\| (\Delta \mathbf{A}, \Delta \mathbf{b}) \|_{\mathbf{E},\mathbf{f}}} \tag{4}$$

where $\mathbf{x} + \Delta \mathbf{x} = (\mathbf{A} + \Delta \mathbf{A})^{-1} (\mathbf{b} + \Delta \mathbf{b})$. Following Skeel (1979), this may be easily evaluated as

$$\kappa_{\mathbf{E},\mathbf{f}}(\mathbf{A},\mathbf{b}) \equiv \frac{\| |\mathbf{A}^{-1}| \, \mathbf{E} \, |\mathbf{x}| + |\mathbf{A}^{-1}| \, \mathbf{f} \|_\infty}{\| \mathbf{x} \|_\infty} \, . \tag{5}$$

For example, if we choose $\mathbf{E} = |\mathbf{A}|$ and $\mathbf{f} = |\mathbf{b}|$ for the componentwise relative error,

$$\kappa_{|\mathbf{A}|,|\mathbf{b}|}(\mathbf{A},\mathbf{b}) = \frac{\| |\mathbf{A}^{-1}| \, |\mathbf{A}| \, |\mathbf{x}| + |\mathbf{A}^{-1}| \, |\mathbf{b}| \|_\infty}{\| \mathbf{x} \|_\infty} \, . \tag{6}$$

Sometimes it is convenient to have a condition number which is independent of the right-hand side $\mathbf{b}$. Since

$$\frac{\| |\mathbf{A}^{-1}| \, |\mathbf{A}| \, |\mathbf{x}| \|_\infty}{\| \mathbf{x} \|_\infty} \le \kappa_{|\mathbf{A}|,|\mathbf{b}|}(\mathbf{A},\mathbf{b}) \le 2 \frac{\| |\mathbf{A}^{-1}| \, |\mathbf{A}| \, |\mathbf{x}| \|_\infty}{\| \mathbf{x} \|_\infty} \, , \tag{7}$$

and $\| |\mathbf{A}^{-1}| \, |\mathbf{A}| \, |\mathbf{x}| \|_\infty / \| \mathbf{x} \|_\infty \le \| |\mathbf{A}^{-1}| \, |\mathbf{A}| \|_\infty$, we get the simpler condition number

$$\kappa_{|\mathbf{A}|}(\mathbf{A}) \equiv \| |\mathbf{A}^{-1}| \, |\mathbf{A}| \|_\infty \ge 0.5 \, \kappa_{|\mathbf{A}|,|\mathbf{b}|}(\mathbf{A},\mathbf{b}) \, . \tag{8}$$

The purpose of the condition number is, of course, to provide error bounds: if $\mathbf{A}$ is perturbed by $|\Delta \mathbf{A}| \le \omega |\mathbf{A}|$ and $\mathbf{b}$ by $|\Delta \mathbf{b}| \le \omega |\mathbf{b}|$, and if ω is small enough, then $\mathbf{x}$ will be perturbed by no more than about $\omega \, \kappa_{|\mathbf{A}|,|\mathbf{b}|}(\mathbf{A},\mathbf{b})$. More rigorously, Skeel (1979) shows that, for ω defined as in (3),

$$\frac{\| \delta \mathbf{x} \|_\infty}{\| \mathbf{x} \|_\infty} \le \frac{\omega \, \kappa_{|\mathbf{A}|,|\mathbf{b}|}(\mathbf{A},\mathbf{b})}{1 - \omega \, \kappa_{|\mathbf{A}|}(\mathbf{A})} \, . \tag{9}$$

Similarly, if we define

$$\kappa_{\mathbf{E}}(\mathbf{A}) \equiv \| |\mathbf{A}^{-1}| \, \mathbf{E} \|_\infty \, , \tag{10}$$

we have, for ω defined as in (2),

$$\frac{\|\delta x\|_\infty}{\|x\|_\infty} \le \frac{\omega \kappa_{E,f}(A,b)}{1-\omega \kappa_E(A)} . \tag{11}$$

It is easy to see that the problem is no more badly conditioned with respect to the componentwise relative backward error measure than with respect to the usual normed backward error measure. This is because

$$\kappa(A) \equiv \|A^{-1}\|_\infty \|A\|_\infty \ge \||A^{-1}||A|\|_\infty = \kappa_{|A|}(A) . \tag{12}$$

It is possible for $\kappa_{|A|}(A)$ to be much smaller than $\kappa(A)$. For example, we can make $\kappa(A)$ arbitrarily large by multiplying one of the rows of A by a large enough constant. However, $\kappa_{|A|}(A)$ is independent of the row scaling of A.

2.2 Backward error

As stated in the introduction, it is in practice necessary to modify the choice $f=|b|$ of the componentwise relative backward error. This need arises because of the factor $\sigma(A,x)$ in Theorem 2; when $\sigma(A,x)$ is large, convergence of the backward error ω in equation (3) to the roundoff level is not guaranteed. Take, for example, A sparse and irreducible, and x sparse such that some $b_i = \sum_j a_{ij} x_j$ are zero because each $a_{ij} x_j = 0$. Since A^{-1} is structurally full (Duff, Erisman, Gear, and Reid 1985), x will be structurally full as well, so that a computed component $\hat{x}_k$ can be zero only through exact cancellation. In practice, this means that all components of the computed solution $\hat{x}$ will be nonzero, with the entries which should be zero containing roundoff error of unpredictable sign. Therefore both $r_i = (A\hat{x}-b)_i$ and $(|A||\hat{x}|+|b|)_i$ may be small but of similar orders of magnitude, so that ω stays large even after some steps of iterative refinement.

Ideally, we would like to choose f to satisfy the following four criteria:

(i) the backward error ω (in (2)) usually converges to machine precision after one step of iterative refinement,

(ii) ωf is "small" compared to b,

(iii) the resulting error bound in (11) is as small as possible, and

(iv) ω is row-scaling independent.

We have experimented with two choices for f which come close to meeting these four criteria; this will be borne out by the numerical experiments in Section 5. It turns out we must sacrifice the sparsity structure of b in order to guarantee a small backward error bound ω (criterion (i)). A trivial way to do this is to set $E=0$ and $f=|r|/\varepsilon=|A\hat{x}-b|/\varepsilon$, whence $\delta A=0$, $\delta b=r$ and $\omega=\varepsilon$. Of course this is unsatisfactory because $\delta b=r$ may be much larger in norm than b if the system is ill-conditioned, violating criterion (ii). Our approach is to keep $E=|A|$ and choose f_i larger than $|b_i|$ only if it is necessary to keep ω small.

We will choose f in an a posteriori way, letting it depend on the computation as follows: Let

$\mathbf{w}=|\mathbf{A}|\,|\hat{\mathbf{x}}|+|\mathbf{b}|$ be the vector of denominators in equation (3). We then choose a threshold τ_i for each w_i, so that when $w_i>\tau_i$ we can use the usual scaling factor $f_i=|b_i|$. Otherwise, when $w_i\le\tau_i$, we choose a larger f_i. Correspondingly, we divide the equations of $\mathbf{A}\mathbf{x}=\mathbf{b}$ into two categories, those where $w_i>\tau_i$, and those where $w_i\le\tau_i$. We may assume without loss of generality that the leading m equations of $\mathbf{A}\mathbf{x}=\mathbf{b}$, which we denote by $\mathbf{A}^{(1)}\mathbf{x}^{(1)}=\mathbf{b}^{(1)}$, belong to the first category, and the remaining $n-m$ equations, $\mathbf{A}^{(2)}\mathbf{x}^{(2)}=\mathbf{b}^{(2)}$, belong to the second. As stated above, we will let $\mathbf{f}^{(1)}=|\mathbf{b}^{(1)}|$ in the first category. There are several possibilities for τ_i, but in practice the following one has worked well: $\tau_i=1000\,n\,\varepsilon(\|\mathbf{A}_{i.}\|_\infty\,\|\hat{\mathbf{x}}\|_\infty+|b_i|)$, where $\mathbf{A}_{i.}$ is the ith row of A. Note that τ_i is about 1000 times larger than the maximum possible roundoff error committed in computing w_i, and w_i can only be less than τ_i if each product $a_{ij}\hat{x}_j$ is tiny. We performed other runs to check the sensitivity of this choice and found that a change of say a factor of ten (to 100) could occasionally change the number of iterations and the error estimate but usually not by much. We note, however, that this can be viewed as a local choice and could be varied while performing iterative refinement, possibly increasing it in order to decrease ω.

Given the vector τ of the thresholds τ_i, we can choose $\mathbf{f}^{(2)}$ in at least two ways. The first way that we consider is as follows. We let $\mathbf{f}^{(2)}=|\mathbf{A}^{(2)}|\,\mathbf{e}\,\|\hat{\mathbf{x}}\|_\infty$, where $\mathbf{e}$ is the column vector of all ones. This corresponds to the usual normwise backward error, and so the components r_i of the residual are almost guaranteed to be small compared to these $f_i^{(2)}$, insofar as Gaussian elimination alone guarantees a small residual in the norm sense. Since we have not modified the definition of E, we are further guaranteed a solution $\hat{\mathbf{x}}$ which preserves the sparsity structure of A.

There is a difficulty with this choice of $\mathbf{f}$, however: we are no longer guaranteed that $\|\delta\mathbf{b}\|_\infty$ is small compared to $\|\mathbf{b}\|_\infty$. This can only happen when A is very ill-conditioned, since $\|\mathbf{A}^{(2)}\|_\infty\,\|\hat{\mathbf{x}}\|_\infty/\|\mathbf{b}\|_\infty$ is a lower bound on the condition number $\|\mathbf{A}^{-1}\|_\infty\,\|\mathbf{A}\|_\infty$ of A. We have constructed artificial examples where this happens, but not observed it in practice. There is also the possibility that large components in $\mathbf{f}$ will make the condition number $\kappa_{|\mathbf{A}|,\mathbf{f}}(\mathbf{A},\mathbf{b})$ too large and so make the error estimate $\omega\,\kappa_{|\mathbf{A}|,\mathbf{f}}(\mathbf{A},\mathbf{b})$ too pessimistic, but note that this condition number is still bounded by $2\,\kappa_{|\mathbf{A}|}(\mathbf{A})$. We may avoid this possibility as follows. Given the two backward errors

$$\omega_i\equiv\max_j\frac{|\mathbf{A}^{(i)}\hat{\mathbf{x}}-\mathbf{b}^{(i)}|_j}{(|\mathbf{A}^{(i)}|\,|\hat{\mathbf{x}}|+|\mathbf{f}^{(i)}|)_j}\quad,i=1,2, \tag{13}$$

the residual satisfies

$$|\mathbf{r}|=\begin{pmatrix}|\mathbf{A}^{(1)}\hat{\mathbf{x}}-\mathbf{b}^{(1)}|\\ |\mathbf{A}^{(2)}\hat{\mathbf{x}}-\mathbf{b}^{(2)}|\end{pmatrix}\le\begin{pmatrix}\omega_1\,(|\mathbf{A}^{(1)}|\,|\hat{\mathbf{x}}|+|\mathbf{b}^{(1)}|)\\ \omega_2\,(|\mathbf{A}^{(2)}|\,|\hat{\mathbf{x}}|+|\mathbf{A}^{(2)}|\,\mathbf{e}\,\|\hat{\mathbf{x}}\|_\infty)\end{pmatrix} \tag{14}$$

and, to first order, the error is bounded by

$$\frac{\|\delta x\|_\infty}{\|x\|_\infty} = \frac{\|A^{-1}r\|_\infty}{\|x\|_\infty} \le \frac{\||A^{-1}||r|\|_\infty}{\|x\|_\infty}$$

$$\le \omega_1 \frac{\left\| |A^{-1}| \begin{pmatrix} |A^{(1)}||\hat{x}|+|b^{(1)}| \\ 0 \end{pmatrix} \right\|_\infty}{\|\hat{x}\|_\infty} + \omega_2 \frac{\left\| |A^{-1}| \begin{pmatrix} 0 \\ |A^{(2)}||\hat{x}|+f^{(2)} \end{pmatrix} \right\|_\infty}{\|\hat{x}\|_\infty} \qquad (15)$$

$$\equiv \omega_1 \kappa_{\omega_1} + \omega_2 \kappa_{\omega_2} .$$

The advantage of this formulation is that components of $f^{(2)}$ may be very large compared to the components of $b^{(2)}$, causing ω_2 to be very small and κ_{ω_2} to be correspondingly large but without affecting ω_1 or κ_{ω_1}. This formulation is tested in the numerical experiments in Section 5.

A second possible choice for $f^{(2)}$ is to use $f^{(2)} = \|b\|_\infty e$. This choice of $f^{(2)}$ assures us that a small backward error indeed means $\|\delta b\|_\infty / \|b\|_\infty$ will be small, but gives us less assurance that the backward error will converge to machine precision. We have not seen it fail in practice. As with the other choice of f, we can bound the error using two backward errors defined as in (13) and the sum of their products with two condition numbers as in (15). Section 5 also reports on numerical experience with this backward error measure.

Both the previous choices for $f^{(2)}$ can violate one of the criteria (ii) or (iv). The choice $f^{(2)} = |A^{(2)}| e \|\hat{x}\|_\infty$ guarantees that ω_i, $i=1,2$, are row-scaling independent (criterion (iv)), while it can violate criterion (ii). The choice $f^{(2)} = \|b\|_\infty e$ satisfies criterion (ii), but the corresponding ω_2 is row-scaling dependent. Both, as we shall see, satisfy criteria (i) and (iii).

We also see that the bound depends on the accuracy with which we can compute the residual r and the backwards error ω in (2). How much can roundoff contaminate the computed ω, especially when $r = A\hat{x} - b$ is computed by an arithmetic with machine precision ε? A standard error analysis shows that the error in the computed r, δr, is bounded by $(\gamma+1)\varepsilon(|A||\hat{x}|+|b|)$, where γ is the maximum number of nonzero entries in a row of A. When $E=|A|$ and $f=|b|$, this means that the computed ω cannot differ from the true ω by more than about $\pm(\gamma+1)\varepsilon$ which will be within the tolerance of our sparse modification of Skeel's stopping criterion in Theorem 2. Since the computed ω is almost certainly at least about $\gamma\varepsilon$, the final error bound $\omega\,\kappa_{|A|,|b|}(A,b)$, can be low by no more than a factor of 2. The same is true for ω_i, $i=1,2$.

At this point, one might ask what choice of E and f minimizes the resulting error estimate (11). It is easy to see that any choice of E and f such that $E|x|+f$ is a multiple of $|r|$, say $E=0$ and $f=|r|$, yields the minimum product $\omega\,\kappa_{E,f}(A,b) = \||A^{-1}||r|\|_\infty / \|x\|_\infty$. Since the true error is $\|\delta x\|_\infty / \|x\|_\infty = \|A^{-1}r\|_\infty / \|x\|_\infty$, we see that the bound is as tight as ignoring signs in r allows. For this special choice of E and f, we should also add $(\gamma+1)\varepsilon(|A||\hat{x}|+|b|)$ to $|r|$ since roundoff may lower the computed value of $|r|$ by the same amount. The choice $E=0$ and $f=|r|+(\gamma+1)\varepsilon(|A||\hat{x}|+|b|)$ yields a new error bound of

$$\frac{\|\delta x\|_\infty}{\|x\|_\infty} \leq \frac{\||A^{-1}||r|\|_\infty}{\|x\|_\infty} + (\gamma+1)\epsilon\kappa_{|A|,|b|}(A,b). \tag{16}$$

Thus we see that the condition number $\kappa_{|A|,|b|}(A,b)$ plays a central role independent of the notion of backward error, just because it reflects the possible roundoff errors in the computed residual. Furthermore, after only a few steps of iterative refinement Theorem 2 guarantees that, to first order, the bound (9) will be about the same as the bound (16). In our experiments we have seen that, usually, the estimates of the real error given by (9) and (15) have the same order of accuracy as the estimates obtained by the bound (16).

Note that if we set $e_{ij} = \|A\|_\infty$ and $f_i = \|b\|_\infty$, the backward error of $\hat{x}$ with respect to E and f is given by $\|A\hat{x}-b\|_\infty / (\|A\|_\infty \|\hat{x}\|_1 + \|b\|_\infty)$. It is also easy to see that

$$\kappa_{E,f}(A,b) = \frac{\|A^{-1}\|_\infty \|A\|_\infty \|x\|_1 + \|A^{-1}\|_\infty \|b\|_\infty}{\|x\|_\infty} \tag{17}$$

which is within a factor of $2n$ of $\|A^{-1}\|_\infty \|A\|_\infty$. Thus, this choice of E and f, which permits equally large perturbations in all entries of A and b, gives essentially the same backward error and condition number as the usual normed backward error.

We note, in conclusion, that Skeel's original motivation (Skeel 1979) was to analyze the effects of row and column scaling of A on the accuracy and the stability of the LU factorization. He concluded that the optimal way to scale depended on the solution: the columns should be scaled (thus scaling the solution components) so that the components of the scaled solution are all equal in magnitude, and the rows should be scaled so each component of |A||x| (x is the solution) is equal in magnitude. This is unfortunately hard to use in practice since it requires much information about the solution. Fortunately, one step of iterative refinement tends to overcome the effects of bad row scaling, as we have seen.

3 Different models of floating-point arithmetic

Theorem 2 assumed that arithmetic was implemented rather cleanly, i.e. that the floating-point result $fl(a \Diamond b)$ of the operation $a \Diamond b, (\Diamond \in \{+,-,\times,/\})$ satisfies

$$fl(a \Diamond b) = (a \Diamond b)(1+e) \tag{18}$$

with $|e| \leq \varepsilon$, where ε is called the machine precision. This model eliminates both the possibility of underflow as well as machines like the CRAYs, where for lack of a guard digit in the hardware we can only assert that

$$fl(a \pm b) = a(1+e_1) \pm b(1+e_2) \tag{19}$$

where $|e_i| \leq \varepsilon$. Thus, when a and b are very close and we are subtracting, this model permits a large relative error in the computed difference. For example, on any CRAY or many CDC machines, the computed difference of 2^t and the next smaller floating-point number is wrong by a factor of 2 (see, Kahan 1981).

Despite this difficulty, it is possible to carry through the proof of Theorem 2 using the weaker model (19) instead of (18) and arrive at essentially the same conclusion: one step of iterative refinement, even without computing the residual using arithmetic of machine precision ε^2, is enough to guarantee a small componentwise relative backward error as long as the matrix is not too ill-conditioned and $\sigma(A,x)$ is not too large. One might expect problems in bounding the error in the computed residual $fl(A\hat{x}-b)$, since the result might be off by a factor of 2, but in the analysis this potential error is dominated by the error in computing $A\hat{x}$, so the proof goes through. Similarly, the error in updating $\hat{x}-d$ is swamped by larger errors.

The other exception to the model in (18) is underflow. The extension of error analysis to include underflow is discussed in some detail by Demmel (1984), and we just summarize the results here. In place of (18) we use the model

$$fl(a \Diamond b) = (a \Diamond b)(1+e)+v \tag{20}$$

where $|e| \leq \varepsilon$ as before, and v represents the underflow error. Let λ be the underflow threshold, that is the smallest positive, normalized floating-point number. Then, on machines where computed quantities which would be smaller than λ are replaced by zero, $|v|$ is bounded by λ. On machines with IEEE standard floating-point arithmetic (see IEEE 1985, IEEE 1987), gradual underflow lowers the bound on $|v|$ to $\varepsilon\lambda$.

The statement of Theorem 2 must be modified as follows to account for underflow. For gradual underflow, we can say the following: if the inputs A and b and the output $\hat{x}$ are normalized (that is, exceed λ in magnitude), and if the residuals are computed by an arithmetic of machine precision either ε or ε^2, then gradual underflow can only degrade performance to the level of the residual computation using the arithmetic of machine precision ε. For conventional underflow, the norms of A, b and $\hat{x}$ must exceed λ/ε for this statement to be true.

The use of extended range and precision in intermediate computations does not change these conclusions. Assuming r and d are stored in the same format as A, b and $\hat{x}$, underflows in r and d have the same potential effects on performance as they did when they were not computed in extended format.

We have not yet considered the effect of underflow on the rate of convergence of the iteration. There are matrices for which the iteration converges only if underflows do not occur, but the matrices are so ill-conditioned as to make the computed solution untrustworthy anyway. As long as some entry of A is large enough (λ for gradual underflow and λ/ε for conventional underflow) then underflows will have an effect on the convergence rate comparable to roundoff.

4 An estimator for $\kappa_{|A|,|b|}(A,b)$

In order to estimate the accuracy of a computed solution of $Ax=b$, two ingredients are needed: a bound on the backward error (however it is measured) and a condition number with respect to the choice of backward error. As discussed in Section 2.2, the product of the two previous quantities provides an approximate upper bound on the relative error in the computed solution.

In the case of the conventional normwise backward error, the condition number is essentially given by $\kappa(A)=\|A^{-1}\|_\infty \|A\|_\infty$. There has been much work on such estimators for $\kappa(A)$ in recent years (for example Cline *et al.* 1979, see Higham 1987a for a complete list of references), and cheap, reliable estimators are available in standard software packages such as LINPACK (Dongarra *et al.* 1979). It is natural to seek an analogous estimator for $\kappa_{|A|,|b|}(A,b)$.

From (5) we see that the quantity we need to estimate is

$$\| |A^{-1}| E|x| + |A^{-1}| f \|_\infty = \| |A^{-1}| (E|x| + f) \|_\infty. \tag{21}$$

In place of the true solution x, we may use its computed approximation $\hat{x}$. In the case of componentwise relative backward error, we may also just use the simpler condition number $\kappa_{|A|}(A)$ which requires us to estimate

$$\| |A^{-1}| |A| \|_\infty = \| |A^{-1}| |A| e \|_\infty \tag{22}$$

where e is the vector of all ones. Either way, we need to be able to estimate

$$\| |A^{-1}| g \|_\infty \tag{23}$$

where g is a nonnegative vector which is easy to compute (in the above examples it costs just one matrix-vector multiply).

Let $G=\mathrm{diag}(g_1,\ldots,g_n)$. Then $g=Ge$ and

$$\| |A^{-1}| g \|_\infty = \| |A^{-1}| G e \|_\infty = \| |A^{-1} G| e \|_\infty = \| |A^{-1} G| \|_\infty = \| A^{-1} G \|_\infty. \tag{24}$$

$\| A^{-1} G \|_\infty$ can be estimated by the algorithm of Hager (1984) and Higham (1987), which estimates the 1-norm (or infinity-norm) of a $n \times n$ matrix given the ability to multiply a vector by both the matrix and its transpose. We can multiply any vector z by the operator $A^{-1}G$ by multiplying z by the diagonal matrix G, and then solving $Ay=Gz$ using the LU factorization of A. Multiplying by $(A^{-1}G)^T$ is equally easy.

Our estimate of condition numbers $\kappa_{|A|,|b|}(A,b)$ includes a dependence on the calculated solution. We also performed runs for different solutions (for example, $x_i=i^2$, $i=1,\ldots,n$) and found little sensitivity. Note that the experiments in Set 1 in Section 5 give us results close to the upper bound of twice $\kappa_{|A|}$.

5 Numerical experiments

We tested the stopping criteria, the backward errors (13) and the error bound (15) by modifying the sparse linear system solver MA28 in the Harwell Subroutine Library (Duff 1977). As we mentioned in Section 1, MA28 can drop entries of **L** and **U** that fall below a tolerance (called *drop tol* in our tables) in order to further decrease fill-in (*drop tol*=0 corresponds to standard Gaussian elimination). The resulting **L** and **U** factors are then used to solve $\mathbf{Ax}=\mathbf{b}$ for **x** by forward and back substitution in the usual way, followed by some steps of iterative refinement.

All tests were done on an IBM 3084. In single precision, the machine precision, ε, is $16^{-5} \approx 10^{-6}$. In double precision, it is $16^{-13} \approx 2\times10^{-16}$.

All our runs are on a common set of test matrices from the Harwell-Boeing test set (Duff, Grimes, and Lewis 1987). Their names, number of nonzero entries and condition numbers $\kappa(A)$ and $\kappa_{|A|}(A)$ are given in Table 1. The name of each matrix includes its dimensions, for example GRE115 is 115 by 115. Two matrices are identified as GRE216. Both of these have the same structure, but they have quite different numerical values. We also ran our tests on some other matrices from the set and obtained results broadly comparable with these displayed.

For each run, we chose the value of the solution **x** and then we computed the right-hand side **b** by multiplying the solution by the test matrix. All matrices have also been scaled before computing the right-hand side, thus obtaining two test problems for each matrix. The scaling is computed using the Harwell routine MC19, which makes the nonzeros of the scaled matrix near to unity by minimizing the sum of the squares of logarithms of the moduli of the nonzeros (Curtis and Reid 1972). This scaling does not guarantee that $\kappa(A)$ and $\kappa_{|A|}(A)$ must decrease (see Table 1) although on many matrices the effect is very beneficial, particularly for the classical condition number. This is particularly so for the second GRE216 example, where, before the scaling, the matrix was essentially singular. Note in general that many of the matrices are poorly conditioned, particularly before scaling.

In all the runs, the standard normwise backward error

$$\eta \equiv \frac{\|\mathbf{r}\|_\infty}{\|\mathbf{A}\|_\infty \|\hat{\mathbf{x}}\|_\infty + \|\mathbf{b}\|_\infty}, \tag{25}$$

the condition number $\kappa(A)$ and the error bound $\eta\,\kappa(A)$ were computed and compared to the other backward errors, condition numbers and error bounds.

We ran our tests with different choices for the vectors τ and **f** defined in Section 2.2 and different right-hand sides **b**. According to these different choices, we group the experiments into 3 sets. We also include some runs using drop tolerances (set 4).

The main data for our numerical experiments are presented in Tables A1-A15 in the Appendix. In this section, we display summaries of these results.

	Nonzeros	Before scaling		After scaling	
		$\kappa(A)$	$\kappa_{\lvert A\rvert}(A)$	$\kappa(A)$	$\kappa_{\lvert A\rvert}(A)$
GRE115	421	0.93D+02	0.86D+02	0.69D+04	0.13D+03
GRE185	975	0.38D+06	0.15D+06	0.39D+06	0.14D+06
GRE216	812	0.28D+03	0.22D+03	0.20D+03	0.18D+03
GRE216	812	0.83D+15	0.29D+14	0.56D+08	0.85D+07
GRE343	1310	0.47D+03	0.37D+03	0.30D+03	0.26D+03
GRE512	1976	0.46D+03	0.37D+03	0.40D+03	0.36D+03
GRE1107	5664	0.18D+09	0.98D+08	0.77D+10	0.24D+09
WEST67	294	0.91D+03	0.31D+03	0.30D+03	0.13D+03
WEST132	413	0.11D+13	0.80D+07	0.94D+04	0.21D+04
WEST156	362	0.12D+32	0.38D+09	0.91D+12	0.15D+09
WEST167	506	0.69D+11	0.52D+06	0.46D+04	0.12D+04
WEST381	2134	0.53D+07	0.38D+05	0.38D+06	0.53D+04
WEST479	1888	0.49D+12	0.37D+07	0.27D+06	0.20D+05
WEST497	1721	0.38D+12	0.13D+07	0.42D+07	0.63D+04
WEST655	2808	0.49D+12	0.37D+07	0.42D+06	0.36D+05
WEST989	3518	0.13D+13	0.10D+08	0.58D+06	0.52D+05
WEST1505	5414	0.14D+13	0.10D+08	0.67D+08	0.21D+07
WEST2021	7310	0.28D+13	0.21D+08	0.86D+06	0.10D+06

Table 1. Condition numbers before and after scaling.

In all cases, the stopping criterion was

Stop if $\omega \leq \varepsilon$ *or* ω *does not decrease by at least a factor of* 2 .

All the runs used IBM double precision, except for the experiments in single and mixed precision in set 1. This stopping criterion differs from that used in Theorem 2 ($\omega \leq (n+1)\varepsilon$). The value in Theorem 2 can be too large, especially for very large and sparse matrices, and the iterative refinement could stop too early. Generally, our stopping criterion terminates the iterative refinement with a value of ω less than ε. If the convergence is slow (for example, using double precision, the second GRE216 matrix in Table A7 stops after 4 iterations with $\omega = 0.4 \times 10^{-15} \approx 2\ \varepsilon$), our stopping criterion recognizes this early. However, the final value of ω is still of order ε. Somewhat surprisingly we find there is no advantage in including a factor $(\gamma+1)$ in our stopping criterion. Indeed, its inclusion would often result in no iterations, and there are only few occasions in sets 1 to 3 where the $\omega \leq \varepsilon$ criterion is not met. Note that, in the runs in sets 2 to 4, ω is replaced by $\omega_1 + \omega_2$ (as in equations (13)–(15)). If we used a similar condition on η, in most of the examples we did not perform any steps of iterative refinement because the first solution satisfied the stopping criterion, but, before scaling, the estimation of the error $\| \delta x \|_\infty / \| x \|_\infty$ given by $\eta\, \kappa(A)$ was very poor because of the very large value of $\kappa(A)$.

We discuss the experiments for each of our four sets of values in turn. In all the following tables, the row corresponding to "Num. iter." gives the number of steps performed by the iterative refinement algorithm and the row corresponding to "Num. cases" gives the number of examples for which the iterative refinement performed that number of iterations. By "Error" we denote the

max-norm of the difference between the computed solution and the actual solution used to generate the right-hand side, divided by the max-norm of the actual solution.

In the following, we denote by $\omega_i^{(j)}$ and by $\kappa_{\omega_i}^{(j)}$, $i=1,2$, $j=1,2,3,4$, the componentwise backward errors defined by (13) and the corresponding condition numbers defined by (15). The superscript identifies the set of tests.

Set 1:

For these tests we chose $\tau_i=0$, so that all equations belonged to category 1. Thus the backwards error was given by $\omega_1^{(1)}$ as defined in (13), the condition number $\kappa_{\omega_1}^{(1)}$ and the error bound by $\omega_1^{(1)}\,\kappa_{\omega_1}^{(1)}$ as defined in (15). Because all the equations belong to category 1, $\kappa_{\omega_1}^{(1)}=\kappa_{|A|,|b|}(A,b)$, and $\omega_2^{(1)}=0$. The right-hand sides b were chosen so that the true solution x had all components equal to 1. The drop tolerance was zero. These test were run in single precision, double precision, and mixed precision (all single precision, except for double precision computation of residuals). The Tables A1-A5 in the Appendix are relative to Set 1.

	min	avr	max
$Log_{10}\frac{\kappa(A)\,(Before\ scaling)}{\kappa(A)\,(After\ scaling)}$	-1.9	4.1	19
$Log_{10}\frac{\kappa_{\omega_1}^{(1)}\,(Before\ scaling)}{\kappa_{\omega_1}^{(1)}\,(After\ scaling)}$	-0.38	1.4	6.5

	Before scaling			After scaling		
	min	avr	max	min	avr	max
$Log_{10}(\kappa(A)/\kappa_{\omega_1}^{(1)})$	-0.26	3.6	22	-0.26	0.91	3.5

Table 2. Summary of results for the condition numbers of set 1.

In Table 2, summarizing the results in Table A1, we observe that the condition number $\kappa_{\omega_1}^{(1)}$ is always less, for both scaled and unscaled matrices, than twice the classical condition number $\kappa(A)$, as must be the case from the theory. In some examples, $\kappa_{\omega_1}^{(1)}$ is much better than $\kappa(A)$ (for example, in the WEST156 example before scaling $\kappa_{\omega_1}^{(1)} < 3.2\times 10^{-23}\ \kappa(A)$). Moreover, Table 2 shows that the classical condition number $\kappa(A)$, without any form of scaling, is rather unreliable as a measure of the ill-conditioning of the system. Table 3 (summarizing the results in Tables A2 and A3) reflects the previous considerations, so that the estimation $\omega_1^{(1)}\,\kappa_{\omega_1}^{(1)}$ of the error is generally quite tight, while $\eta\,\kappa(A)$ can be too pessimistic before scaling. Note that it is possible for our bound to be less tight than that from the classical theory but, when this happens in the experiments, our bound is only 3 times greater than the classical one in the worst case.

Throughout, our estimate of condition numbers $\kappa_{|A|,|b|}(A,b)$ includes a dependence on the calculated solution. We also performed runs for different solutions (for example, $x_i = i^2$, $i=1,\ldots,n$)

	Before scaling			After scaling		
Num. iter.	0	1	≥ 2	0	1	≥ 2
Num. cases	0	17	1	1	16	1
	min	avr	max	min	avr	max
$Log_{10}(\eta)$	–18	–16	–16	–17	–16	–16
$Log_{10}(\omega_1^{(1)})$	–16	–16	–16	–16	–16	–16
$Log_{10}\frac{\eta\,\kappa(A)}{Error}$	0.78	4.7	22	0.93	2.0	4.1
$Log_{10}\frac{\omega_1^{(1)}\,\kappa_{\omega_1}^{(1)}}{Error}$	0.48	1.5	2.5	0.43	1.4	3.3
$Log_{10}\frac{\eta\,\kappa(A)}{\omega_1^{(1)}\,\kappa_{\omega_1}^{(1)}}$	–0.32	3.2	20	–0.41	0.53	3.0

Table 3. Summary of results for set 1.

and found little sensitivity. Note that our choice of x in Set 1 gives us results close to the upper bound of twice $\kappa_{|A|}$. In Tables A4 and A5, we report the results of the algorithm using single and mixed precision. Unfortunately, the test matrices are in many cases so ill-conditioned that the iterative refinement diverged, that is $\omega_1^{(1)}$ increased after some steps as in, for example, GRE1107 and the second GRE216 example in Table A4. In practice, IBM single precision is too poor to produce good results, and the use of mixed precision does not help. Note, however, that our algorithm still terminates after only a few steps. In every case, we tried running the iterative refinement for twenty steps and in no cases did we get much improvement over the results shown. Our algorithm for computing the condition numbers encounters numerical difficulties partly because of the ill-conditioning of these matrices and partly because we use threshold pivoting in the LU factorization. We could have used iterative refinement in this computation, but this would be at variance with our desire for a cheap estimator. Our feeling is that single precision calculations are inappropriate here.

Set 2:

For these tests we chose $\tau_i = 1000\, n\, \varepsilon(\| A_{i.} \|_\infty \| \hat{x} \|_\infty + |b_i|)$ and $f^{(2)} = |A^{(2)}|\, e\, \| \hat{x} \|_\infty$, where e is the column vector of all ones. This leads to the backward errors $\omega_1^{(2)}$ and $\omega_2^{(2)}$ defined in (13) and the condition numbers $\kappa_{\omega_1}^{(2)}$ and $\kappa_{\omega_2}^{(2)}$ and error bound $\omega_1^{(2)}\,\kappa_{\omega_1}^{(2)} + \omega_2^{(2)}\,\kappa_{\omega_2}^{(2)}$ defined in (15). The right-hand sides were chosen so that the true solution x had every fifth entry equal to 1 ($x_1 = x_6 = x_{11} = \ldots = 1$) and the rest zero. The drop tolerance was zero. These tests were done in double precision only. Tables A6 to A8 show the results of runs on set 2. We present a summary of these results in Tables 4 and 5.

We also ran all the test examples of set 2 replacing zero with 10^{-16} in x and obtained similar results. It is necessary to emphasize that, in most of the examples of set 2, the standard ω

	min	avr	max
$Log_{10}\frac{\kappa(A)\,(Before\ scaling)}{\kappa(A)\,(After\ scaling)}$	−1.9	4.1	19
$Log_{10}\frac{\kappa^{(2)}_{\omega_1}\,(Before\ scaling)}{\kappa^{(2)}_{\omega_1}\,(After\ scaling)}$	−0.37	1.3	7.0
$Log_{10}\frac{\kappa^{(2)}_{\omega_2}\,(Before\ scaling)}{\kappa^{(2)}_{\omega_2}\,(After\ scaling)}$	−0.43	1.6	6.1

	Before scaling			After scaling		
	min	avr	max	min	avr	max
$Log_{10}(\kappa(A)/\kappa^{(2)}_{\omega_1})$	0.45	4.3	23	0.26	1.5	3.8
$Log_{10}(\kappa(A)/\kappa^{(2)}_{\omega_2})$	0.52	4.3	23	0.30	1.8	5.2

Table 4. Summary of results for the condition numbers for set 2.

computed by (3) was very large (sometimes of order 1), so that we would get no useful information if we use a very large value for τ_i. Notice that, in all our runs, $\omega_2^{(2)}$ is very small compared with $\omega_1^{(2)}$, in agreement with our comments after equation (15).

It may appear that our error estimate is sometimes poor, but the relatively good solution obtained is really fortuitous as can be seen by the results in the Appendix using the same matrix but with a different right-hand side (the examples shown by the GRE1107 results in Tables A3 and A8 and by the second GRE216 results in Tables A2 and A7).

	Before scaling			After scaling		
Num. iter.	0	1	≥ 2	0	1	≥ 2
Num. cases	1	13	4	2	12	4
	min	avr	max	min	avr	max
$Log_{10}(\eta)$	−23	−17	−16	−17	−17	−16
$Log_{10}(\omega_1^{(2)})$	−16	−16	−15	−16	−16	−15
$Log_{10}(\omega_2^{(2)})$	−32	−27	−19	−31	−28	−19
$Log_{10}\frac{\eta\,\kappa(A)}{Error}$	0.65	4.5	19	0.97	2.2	4.0
$Log_{10}\frac{\omega_1^{(2)}\,\kappa^{(2)}_{\omega_1}+\omega_2^{(2)}\,\kappa^{(2)}_{\omega_2}}{Error}$	0.58	1.7	4.3	0.50	1.6	2.7
$Log_{10}\frac{\eta\,\kappa(A)}{\omega_1^{(2)}\,\kappa^{(2)}_{\omega_1}+\omega_2^{(2)}\,\kappa^{(2)}_{\omega_2}}$	−0.17	2.8	16	−0.23	0.63	2.4

Table 5. Summary of results for set 2.

Set 3:

For these tests we chose $\tau_i = 1000\, n\, \varepsilon(\|\mathbf{A}_{i.}\|_\infty \|\hat{\mathbf{x}}\|_\infty + |b_i|)$ just as in Set 2, and $\mathbf{f}^{(2)} = \|\mathbf{b}\|_\infty \mathbf{e}$, where $\mathbf{e}$ is the column vector of all ones. This leads to backward errors $\omega_1^{(3)}$ and $\omega_2^{(3)}$ defined in (13) and the condition numbers $\kappa_{\omega_1}^{(3)}$ and $\kappa_{\omega_2}^{(3)}$ and error bound $\omega_1^{(3)}\, \kappa_{\omega_1}^{(3)} + \omega_2^{(3)}\, \kappa_{\omega_2}^{(3)}$ defined in (15). The right-hand sides were chosen so that the true solution $\mathbf{x}$ had every fifth entry equal to 1 and the rest zero. The drop tolerance was zero. These tests were done in double precision only. The Tables A9-A11 are relative to Set 3 of parameters, and we summarize these in Tables 6 and 7.

	min	avr	max
$Log_{10} \frac{\kappa(A)\,(Before\ scaling)}{\kappa(A)\,(After\ scaling)}$	-1.9	4.1	19
$Log_{10} \frac{\kappa_{\omega_1}^{(3)}\,(Before\ scaling)}{\kappa_{\omega_1}^{(3)}\,(After\ scaling)}$	-0.37	1.3	7.0
$Log_{10} \frac{\kappa_{\omega_2}^{(3)}\,(Before\ scaling)}{\kappa_{\omega_2}^{(3)}\,(After\ scaling)}$	-1.9	4.0	14

	Before scaling			After scaling		
	min	avr	max	min	avr	max
$Log_{10}(\kappa(A) / \kappa_{\omega_1}^{(3)})$	0.45	4.3	23	0.26	1.5	3.8
$Log_{10}(\kappa(A) / \kappa_{\omega_2}^{(3)})$	0.10	0.97	6.4	0.38	0.86	2.6

Table 6. Summary of results for the condition numbers for set 3.

	Before scaling			After scaling		
Num. iter.	0	1	≥ 2	0	1	≥ 2
Num. cases	1	13	4	2	12	4
	min	avr	max	min	avr	max
$Log_{10}(\eta)$	-23	-17	-16	-17	-17	-16
$Log_{10}(\omega_1^{(3)})$	-16	-16	-15	-16	-16	-15
$Log_{10}(\omega_2^{(3)})$	-30	-27	-17	-31	-28	-19
$Log_{10} \frac{\eta\, \kappa(A)}{Error}$	0.65	4.5	19	0.97	2.2	4.0
$Log_{10} \frac{\omega_1^{(3)}\, \kappa_{\omega_1}^{(3)} + \omega_2^{(3)}\, \kappa_{\omega_2}^{(3)}}{Error}$	0.58	2.2	7.9	0.50	1.6	2.7
$Log_{10} \frac{\eta\, \kappa(A)}{\omega_1^{(3)}\, \kappa_{\omega_1}^{(3)} + \omega_2^{(3)}\, \kappa_{\omega_2}^{(3)}}$	-0.17	2.3	11	-0.23	0.63	2.4

Table 7. Summary of results for the set 3.

Comparing Tables 4 and 6 we observe that, while $\kappa_{\omega_1}^{(2)}$ and $\kappa_{\omega_2}^{(2)}$ are usually quite close, $\kappa_{\omega_2}^{(3)}$ can be

much larger than $\kappa^{(3)}_{\omega_1}$, (for example, see the WEST156 example before scaling, where $\kappa^{(3)}_{\omega_2}$ is 10^{16} times larger than $\kappa^{(3)}_{\omega_1}$) and the error estimation can be pessimistic. Also note that, comparing line 7 of Tables 5 and 7, this choice of f does not give as good a bound as our choice for f in set 2, although the difference is minimal after scaling.

Set 4:

For these tests we used nonzero drop tolerances ($drop\ tol = 10^{-5}$, $drop\ tol = 10^{-3}$). We changed τ_i from its earlier value to $\tau_i = 1000\,n\,(\varepsilon + drop\ tol)(\|A_{i.}\|_\infty \|\hat{x}\|_\infty + |b_i|)$ and used $f^{(2)} = |A^{(2)}|\,e\,\|\hat{x}\|_\infty$, where e is the column vector of all ones. The entries of b and x were chosen as in Set 3. Double precision was used. Tables A12-A15 are the results of runs using this set of parameters, and the results are summarized in Table 8.

	drop tol. = 10^{-5}			*drop tol.* = 10^{-3}		
Num. iter.	0	1	≥ 2	0	1	≥ 2
Num. cases	2	6	10	2	1	15
	min	avr	max	min	avr	max
$Log_{10}(\eta)$	−18	−16	−16	−18	−15	−4.6
$Log_{10}(\omega_1^{(4)})$	−∞	−∞	−17	−∞	−∞	−∞
$Log_{10}(\omega_2^{(4)})$	−16	−16	−15	−16	−15	−2.8
$Log_{10}\frac{\eta\,\kappa(A)}{Error}$	0.66	2.3	3.7	0.90	2.1	4.6
$Log_{10}\frac{\omega_1^{(4)}\,\kappa^{(4)}_{\omega_1} + \omega_2^{(4)}\,\kappa^{(4)}_{\omega_2}}{Error}$	0.66	1.6	2.8	0.64	1.6	3.8
$Log_{10}\frac{\eta\,\kappa(A)}{\omega_1^{(4)}\,\kappa^{(4)}_{\omega_1} + \omega_2^{(4)}\,\kappa^{(4)}_{\omega_2}}$	−0.14	0.64	2.5	−0.95	0.45	2.9

Table 8. Summary of results for set 4. The −∞ entries correspond to values of $\omega_1^{(4)} = 0$.

Note that, in this set, we nearly always have $\omega_1^{(4)} = 0$. This corresponds to putting all of the error into b, that is $\delta A = 0$ and $\delta b = A\hat{x} - b$, obtaining the situation which was discussed at the beginning of Section 2.2. In this case, f does not depend on b explicitly, but our bounds are still good. Note again that our stopping criterion terminates after only a few iterations if the iteration diverges. We checked this divergence by forcing more iterations and observed either oscillation or divergence.

We observed, contrary to Zlatev (1986), that little gain in sparsity was obtained (see for example Table A15), while even moderate values of drop tolerance caused divergence of the iterative refinement. A drop tolerance strategy appears to work well only on very structured sparse matrices such as those resulting from discretizations of partial differential equations. We confirmed this with a few test runs. See, for example, the results in Table 9.

drop tol	0	10^{-2}	10^{-1}
Fill-in	23619	16085	4697
Num. iter.	2	14	16
Error	0.32D-14	0.25D-14	0.29D-01

Table 9. Fill-in, numbers of iterations and error for the five point operator on a 30×30 grid, using $x_i = 1, i = 1,...,n$ and different values of *drop tol.*

Finally, Duff, Erisman, and Reid (1986, page 276) described an example of Gear (1975) where the error matrix for minimizing the Frobenius norm of the error becomes arbitrarily large if the perturbations are constrained to the original pattern. On this example, after one step of iterative refinement, using as a starting point the solution

$$\hat{\mathbf{x}} = \begin{pmatrix} (\delta - \sigma)/\delta \\ 1/\delta \\ 1/\delta \\ (\delta - \sigma)/\delta \end{pmatrix}, \quad \sigma = 10^{-15},$$

we can guarantee that the error matrix E has the same pattern as the original matrix. That is

$$\mathbf{E} \leq \omega \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & |\delta| & 0 & 0 \\ 0 & 0 & |\delta| & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} = \omega |\mathbf{A}|,$$

with $\omega \leq 10^{-16}$, $\delta = 10^{-8}$. It is interesting to notice that $\kappa(\mathbf{A}) = 1 + 1/\delta$ and $\kappa_{|\mathbf{A}|}(\mathbf{A}) = 4$.

6 Conclusions

We have shown that, when the iterative refinement is converging, it is possible and inexpensive to guarantee solutions of sparse linear systems which are exact solutions of a nearby system whose matrix has the same sparsity structure. Thus we have answered the open problem posed by Duff, Erisman and Reid (1986, page 276) concerning obtaining bounded perturbations while maintaining sparsity. If the equations arise from the discretization of a partial differential equation, then a componentwise tiny error should indicate that the solution obtained is that of a neighbouring partial differential equation, a conclusion that would not be available if classical error bounds were being used.

We have extended work of Skeel (1980) and Demmel (1984) to include the possibility of having sparse right-hand sides and solutions vectors and have shown that, although we can not always guarantee the solution to a nearby problem whose right-hand side sparsity is the same, we can develop suitable bounds for perturbations in the right-hand side.

We discuss methods of inexpensively and accurately calculating a condition number appropriate to this tighter backward error. This condition number is not bigger than that of Wilkinson and can

indeed be much smaller, particularly if the matrix is badly row-scaled. For example, in set 1, the average of the logarithms of the ratio of the classical condition number before and after scaling is 4.1, while for the Skeel condition number the corresponding value is 1.4.

We have incorporated our backward error estimator in the iterative refinement step of a direct sparse matrix solver and find that we often require zero or one step of iterative refinement to guarantee that the computed solution is the solution of a nearby system with the same sparsity structure as the original matrix. We also observe that we do not require any extra precision in calculating residuals, thus confirming remarks made by Skeel (1980). Additionally, when combined with our condition number estimator, a good estimate of the actual error is obtained. Furthermore, when iterative refinement diverges, our stopping criterion recognizes this early.

We observed, contrary to Zlatev (1986), that little gain in sparsity was obtained while even moderate values of drop tolerance caused divergence of the iterative refinement. A drop tolerance strategy appears to work well only on very structured sparse matrices such as those resulting from discretizations of partial differential equations.

In this paper, we have been using iterative refinement to improve the solution obtained using an LU factorization. We have also considered the case when our LU factorization can be quite inaccurate (set 4). In this case, one could use other techniques including SOR and CG and it is a open question as to how far our analysis could be continued to cover these cases.

Acknowledgement

We would like thank John Reid for reading a draft of this paper and making some helpful remarks.

References

Cline, A. K., Moler, C. B., Stewart, G. W. and Wilkinson J. H. (1979). An estimate for the condition number of a matrix. *SIAM J. Numer. Anal.* **16** , 368-375.

Curtis A. R. and Reid, J. K. (1972). On the automatic scaling of matrices for Gaussian elimination. *J. Inst. Maths. Applics.* **10**, 118-124.

Demmel, J. W. (1984). Underflow and the Reliability of Numerical Software, *SIAM J. Sci. Stat. Comput.* **5**, 887-919.

Dongarra, J. J., Bunch, J. R., Moler, C. B. and Stewart, G..W. (1979). LINPACK User's Guide. SIAM, Philadelphia, 1979.

Duff, I. S. (1977). MA28 – a set of Fortran subroutines for sparse unsymmetric linear equations. Report AERE R8730, HMSO, London.

Duff, I. S., Erisman, A. M., Gear, C. W., and Reid, J. K. (1985). Some remarks on inverses of sparse matrices. Report CSS 171, CSS Division, Harwell Laboratory, England.

Duff, I. S., Erisman, A. M., and Reid, J. K. (1986). Direct methods for sparse matrices. Oxford University Press, London.

Duff, I. S., Grimes, R. G., Lewis, J. G. (1987). Sparse matrix test problems. Report CSS 191, CSS Division, Harwell Laboratory, England.

Gear, C. W. (1975). Numerical errors in sparse linear equations. Report UIUCDCS-F-75-885, Department of Computer Science, University of Illinois at Urbana-Champaign, Illinois.

Hager, W. W. (1984). Condition estimators, *SIAM J. Sci. Stat. Comput.* **5**, 311-316.

Higham, N. J. (1987a). A survey of condition number estimation for triangular matrices. *SIAM Review* **29**, 575-596.

Higham, N. J. (1987b). Fortran codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation. Numerical Analysis Report No. 135, University of Manchester, England.

IEEE (1985). Standard for Binary Floating Point Arithmetic. ANSI/IEEE Std 754-1985, IEEE, New York.

IEEE (1987). Radix and Format Independent Standard for Floating Point Arithmetic. ANSI/IEEE Std 854-1987, IEEE, New York.

Kahan, W. (1981). Why Do We Need a Floating Point Arithmetic Standard? IEEE Floating Point Subcommittee Working Document P754/81-2.8.

Markowitz, H. M. (1957). The elimination form of the inverse and its application to linear programming. *Management Sci.* **3**, 255-269.

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

Skeel, R. D. (1979). Scaling for numerical stability in Gaussian elimination. *J. ACM* **26**, 494-526.

Skeel, R. D. (1980). Iterative refinement implies numerical stability for Gaussian elimination. *Math. Comp.* **35**, 817-832.

Wilkinson, J. H. (1961). Error analysis of direct methods of matrix inversion. *J. ACM* **8**, 281-330.

Zlatev, Z., Wasniewski, J. and Schaumburg, K. (1986). Condition number estimators in a sparse matrix software. *SIAM J. Sci. Stat. Comput.* **7**, 1175-1189.

APPENDIX Tables of results of numerical experiments

In all the following tables, the column corresponding to "Num. iter." gives the number of steps performed by the iterative refinement algorithm. By "Error" we denote the max-norm of the difference between the computed solution and the actual solution used to generate the right-hand side, divided by the max-norm of the actual solution.

	Before scaling		After scaling	
	$\kappa(A)$	$\kappa_{\omega_1}^{(1)}$	$\kappa(A)$	$\kappa_{\omega_1}^{(1)}$
GRE115	0.93D+02	0.17D+03	0.69D+04	0.26D+03
GRE185	0.38D+06	0.30D+06	0.39D+06	0.29D+06
GRE216	0.28D+03	0.44D+03	0.20D+03	0.35D+03
GRE216	0.83D+15	0.58D+14	0.56D+08	0.17D+08
GRE343	0.47D+03	0.74D+03	0.30D+03	0.51D+03
GRE512	0.46D+03	0.73D+03	0.40D+03	0.72D+03
GRE1107	0.18D+09	0.20D+09	0.77D+10	0.48D+09
WEST67	0.91D+03	0.15D+03	0.30D+03	0.16D+03
WEST132	0.11D+13	0.12D+08	0.94D+04	0.33D+04
WEST156	0.12D+32	0.38D+09	0.91D+12	0.30D+09
WEST167	0.69D+11	0.80D+06	0.46D+04	0.18D+04
WEST381	0.53D+07	0.75D+05	0.38D+06	0.85D+04
WEST479	0.49D+12	0.57D+07	0.27D+06	0.25D+05
WEST497	0.38D+12	0.20D+07	0.42D+07	0.12D+05
WEST655	0.49D+12	0.57D+07	0.42D+06	0.41D+05
WEST989	0.13D+13	0.16D+08	0.58D+06	0.70D+05
WEST1505	0.14D+13	0.16D+08	0.67D+08	0.35D+07
WEST2021	0.28D+13	0.32D+08	0.86D+06	0.12D+06

Table A1. Set 1. Condition numbers before and after scaling.

	Num. iter.	η	$\eta\,\kappa(A)$	$\omega_1^{(1)}$	$\omega_1^{(1)}\,\kappa_{\omega_1}^{(1)}$	Error
GRE115	1	0.52D-16	0.48D-14	0.59D-16	0.10D-13	0.79D-15
GRE185	1	0.12D-15	0.47D-10	0.16D-15	0.48D-10	0.16D-12
GRE216	1	0.67D-16	0.19D-13	0.67D-16	0.29D-13	0.26D-15
GRE216	1	0.73D-16	0.61D-01	0.11D-15	0.64D-02	0.21D-02
GRE343	1	0.10D-15	0.47D-13	0.10D-15	0.74D-13	0.50D-15
GRE512	1	0.83D-16	0.38D-13	0.83D-16	0.61D-13	0.26D-15
GRE1107	1	0.93D-16	0.17D-07	0.11D-15	0.22D-07	0.74D-10
WEST67	1	0.49D-16	0.45D-13	0.89D-16	0.13D-13	0.24D-14
WEST132	1	0.93D-17	0.98D-05	0.15D-15	0.18D-08	0.18D-09
WEST156	1	0.77D-18	0.90D+13	0.11D-15	0.42D-07	0.38D-09
WEST167	1	0.80D-16	0.55D-05	0.12D-15	0.95D-10	0.48D-11
WEST381	2	0.45D-16	0.24D-09	0.16D-15	0.12D-10	0.23D-11
WEST479	1	0.19D-16	0.94D-05	0.17D-15	0.96D-09	0.42D-10
WEST497	1	0.77D-16	0.29D-04	0.11D-15	0.22D-09	0.23D-10
WEST655	1	0.19D-16	0.94D-05	0.21D-15	0.12D-08	0.54D-10
WEST989	1	0.95D-16	0.13D-03	0.13D-15	0.21D-08	0.17D-09
WEST1505	1	0.93D-16	0.13D-03	0.16D-15	0.26D-08	0.17D-09
WEST2021	1	0.98D-16	0.27D-03	0.16D-15	0.52D-08	0.88D-10

Table A2. Set 1. $x_i = 1$, i=1,...,n, double precision before scaling.

	Num. iter.	η	$\eta\,\kappa(A)$	$\omega_1^{(1)}$	$\omega_1^{(1)}\,\kappa_{\omega_1}^{(1)}$	Error
GRE115	1	0.64E-16	0.44E-12	0.83E-16	0.22E-13	0.42E-14
GRE185	1	0.62E-16	0.24E-10	0.64E-16	0.18E-10	0.54E-13
GRE216	1	0.54E-16	0.11E-13	0.79E-16	0.28E-13	0.13E-14
GRE216	1	0.89E-16	0.50E-08	0.93E-16	0.16E-08	0.17E-09
GRE343	1	0.76E-16	0.23E-13	0.88E-16	0.45E-13	0.10E-14
GRE512	1	0.76E-16	0.31E-13	0.93E-16	0.66E-13	0.27E-14
GRE1107	1	0.39E-16	0.30E-06	0.10E-15	0.48E-07	0.25E-10
WEST67	1	0.35E-16	0.11E-13	0.14E-15	0.21E-13	0.89E-15
WEST132	1	0.28E-16	0.26E-12	0.98E-16	0.33E-12	0.73E-14
WEST156	0	0.57E-16	0.52E-04	0.16E-15	0.48E-07	0.98E-08
WEST167	1	0.29E-16	0.13E-12	0.11E-15	0.20E-12	0.44E-14
WEST381	1	0.15E-15	0.58E-10	0.17E-15	0.15E-11	0.56E-12
WEST479	1	0.35E-16	0.94E-11	0.22E-15	0.56E-11	0.12E-12
WEST497	1	0.17E-16	0.70E-10	0.11E-15	0.13E-11	0.26E-12
WEST655	1	0.52E-16	0.22E-10	0.19E-15	0.80E-11	0.19E-12
WEST989	1	0.25E-16	0.15E-10	0.12E-15	0.80E-11	0.33E-12
WEST1505	1	0.50E-16	0.34E-08	0.17E-15	0.60E-09	0.82E-10
WEST2021	1	0.50E-16	0.43E-10	0.18E-15	0.22E-10	0.19E-12

Table A3. Set 1. $x_i = 1$, i=1,..,n, double precision after scaling.

	Num. iter.	η	$\eta\,\kappa(A)$	$\omega_1^{(1)}$	$\omega_1^{(1)}\,\kappa_{\omega_1}^{(1)}$	Error
GRE115	1	0.15E-06	0.10E-02	0.29E-06	0.77E-04	0.18E-04
GRE185	2	0.33E-06	0.13E+00	0.33E-06	0.95E-01	0.40E-02
GRE216	1	0.36E-06	0.73E-04	0.39E-06	0.14E-03	0.43E-05
GRE216	2	0.59E-06	0.33E+02	0.83E-06	0.11E+02	0.43E-01
GRE343	1	0.39E-06	0.11E-03	0.42E-06	0.17E-03	0.29E-05
GRE512	1	0.74E-06	0.30E-03	0.74E-06	0.42E-03	0.15E-04
GRE1107	4	0.18E-05	0.13E+04	0.11E-03	0.13E+03	0.86E+00
WEST67	1	0.15E-06	0.45E-04	0.46E-06	0.19E-05	0.97E-05
WEST132	1	0.18E-06	0.17E-02	0.47E-06	0.41E-04	0.82E-04
WEST156	0	0.22E-07	0.20E+05	0.54E-06	0.42E+01	0.95E+00
WEST167	1	0.84E-07	0.38E-03	0.41E-06	0.19E-04	0.40E-04
WEST381	1	0.48E-07	0.19E-01	0.51E-06	0.11E-03	0.23E-02
WEST479	1	0.22E-06	0.61E-01	0.95E-06	0.62E-03	0.83E-03
WEST497	1	0.12E-06	0.49E+00	0.50E-06	0.15E-03	0.17E-02
WEST655	1	0.74E-07	0.31E-01	0.73E-06	0.78E-03	0.77E-03
WEST989	1	0.11E-06	0.63E-01	0.49E-06	0.89E-03	0.72E-03
WEST1505	1	0.11E-06	0.73E+01	0.70E-06	0.63E-01	0.10E+00
WEST2021	1	0.11E-06	0.93E-01	0.72E-06	0.22E-02	0.56E-03

Table A4. Set 1. $x_i = 1$, i=1,...,n, single precision after scaling.

	Num. iter.	η	$\eta\,\kappa(A)$	$\omega_1^{(1)}$	$\omega_1^{(1)}\,\kappa_{\omega_1}^{(1)}$	Error
GRE115	1	0.20E-06	0.14E-02	0.40E-06	0.10E-03	0.57E-05
GRE185	2	0.26E-06	0.10E+00	0.58E-06	0.17E+00	0.16E-02
GRE216	1	0.33E-06	0.66E-04	0.72E-06	0.25E-03	0.31E-05
GRE216	4	0.16E-06	0.89E+01	0.11E-05	0.14E+02	0.62E-01
GRE343	1	0.33E-06	0.97E-04	0.72E-06	0.27E-03	0.26E-05
GRE512	2	0.25E-06	0.10E-03	0.60E-06	0.31E-03	0.72E-05
GRE1107	4	0.17E-05	0.12E+04	0.20E-03	0.24E+03	0.84E+00
WEST67	1	0.20E-06	0.60E-04	0.51E-06	0.21E-05	0.86E-05
WEST132	1	0.15E-06	0.14E-02	0.75E-06	0.66E-04	0.13E-03
WEST156	1	0.11E-07	0.98E+04	0.59E-06	0.46E+01	0.18E+01
WEST167	1	0.12E-06	0.53E-03	0.58E-06	0.28E-04	0.16E-04
WEST381	1	0.17E-06	0.67E-01	0.73E-06	0.16E-03	0.31E-03
WEST479	1	0.77E-07	0.21E-01	0.63E-06	0.41E-03	0.24E-03
WEST497	1	0.12E-06	0.51E+00	0.67E-06	0.20E-03	0.21E-03
WEST655	1	0.74E-07	0.31E-01	0.82E-06	0.89E-03	0.69E-03
WEST989	1	0.94E-07	0.55E-01	0.88E-06	0.16E-02	0.65E-03
WEST1505	1	0.12E-06	0.80E+01	0.79E-06	0.71E-01	0.12E+00
WEST2021	1	0.99E-07	0.86E-01	0.80E-06	0.25E-02	0.15E-03

Table A5. Set 1. $x_i = 1$, i=1,...,n, mixed precision after scaling.

	Before scaling			After scaling		
	$\kappa(A)$	$\kappa_{\omega_1}^{(2)}$	$\kappa_{\omega_2}^{(2)}$	$\kappa(A)$	$\kappa_{\omega_1}^{(2)}$	$\kappa_{\omega_2}^{(2)}$
GRE115	0.93D+02	0.33D+02	0.23D+02	0.69D+04	0.58D+02	0.56D+02
GRE185	0.38D+06	0.50D+05	0.54D+05	0.39D+06	0.46D+05	0.52D+05
GRE216	0.28D+03	0.90D+02	0.82D+02	0.20D+03	0.11D+03	0.10D+03
GRE216	0.83D+15	0.37D+14	0.48D+13	0.56D+08	0.35D+07	0.37D+07
GRE343	0.47D+03	0.16D+03	0.13D+03	0.30D+03	0.10D+03	0.11D+03
GRE512	0.46D+03	0.14D+03	0.14D+03	0.40D+03	0.14D+03	0.14D+03
GRE1107	0.18D+09	0.40D+08	0.31D+08	0.77D+10	0.91D+08	0.83D+08
WEST67	0.91D+03	0.54D+02	0.78D+02	0.30D+03	0.51D+02	0.41D+02
WEST132	0.11D+13	0.26D+07	0.25D+07	0.94D+04	0.61D+03	0.83D+03
WEST156	0.12D+32	0.12D+09	0.13D+09	0.91D+12	0.28D+09	0.54D+07
WEST167	0.69D+11	0.45D+05	0.35D+06	0.46D+04	0.86D+03	0.40D+03
WEST381	0.53D+07	0.16D+05	0.63D+04	0.38D+06	0.23D+04	0.13D+04
WEST479	0.49D+12	0.12D+06	0.22D+07	0.27D+06	0.57D+04	0.34D+04
WEST497	0.38D+12	0.75D+06	0.33D+06	0.42D+07	0.73D+03	0.54D+04
WEST655	0.49D+12	0.66D+06	0.14D+07	0.42D+06	0.12D+05	0.32D+04
WEST989	0.13D+13	0.45D+07	0.47D+07	0.58D+06	0.21D+05	0.11D+05
WEST1505	0.14D+13	0.49D+07	0.53D+07	0.67D+08	0.27D+07	0.17D+05
WEST2021	0.28D+13	0.50D+07	0.89D+07	0.86D+06	0.42D+05	0.11D+05

Table A6. Set 2. Condition numbers before and after scaling.

	Num. iter.	η	$\eta\,\kappa(A)$	$\omega_1^{(2)}$	$\omega_2^{(2)}$	$\omega_1^{(2)}\,\kappa_{\omega_1}^{(2)}+\omega_2^{(2)}\,\kappa_{\omega_2}^{(2)}$	Error
GRE115	1	0.35D-16	0.32D-14	0.84D-16	0.89D-28	0.27D-14	0.71D-15
GRE185	1	0.94D-16	0.35D-10	0.19D-15	0.24D-25	0.94D-11	0.14D-12
GRE216	1	0.12D-16	0.34D-14	0.56D-16	0.64D-27	0.50D-14	0.13D-15
GRE216	4	0.51D-16	0.42D-01	0.41D-15	0.25D-26	0.15D-01	0.76D-06
GRE343	1	0.14D-16	0.65D-14	0.56D-16	0.74D-26	0.90D-14	0.11D-15
GRE512	1	0.25D-16	0.11D-13	0.83D-16	0.27D-25	0.12D-13	0.19D-15
GRE1107	2	0.42D-16	0.78D-08	0.20D-15	0.58D-24	0.82D-08	0.83D-10
WEST67	1	0.42D-16	0.38D-13	0.16D-15	0.27D-30	0.88D-14	0.12D-14
WEST132	1	0.24D-16	0.25D-04	0.13D-15	0.80D-28	0.34D-09	0.16D-10
WEST156	1	0.12D-22	0.14D+09	0.86D-16	0.15D-31	0.10D-07	0.10D-10
WEST167	0	0.28D-17	0.19D-06	0.20D-15	0.25D-18	0.92D-11	0.37D-12
WEST381	1	0.78D-17	0.41D-10	0.15D-15	0.40D-29	0.24D-11	0.29D-12
WEST479	3	0.33D-19	0.16D-07	0.33D-15	0.14D-28	0.39D-10	0.91D-12
WEST497	1	0.12D-17	0.44D-06	0.16D-15	0.28D-28	0.12D-09	0.30D-11
WEST655	3	0.88D-19	0.43D-07	0.26D-15	0.15D-25	0.17D-09	0.29D-11
WEST989	1	0.14D-16	0.19D-04	0.14D-15	0.29D-27	0.61D-09	0.26D-10
WEST1505	1	0.23D-16	0.31D-04	0.20D-15	0.67D-27	0.99D-09	0.46D-10
WEST2021	1	0.19D-16	0.52D-04	0.22D-15	0.32D-27	0.11D-08	0.24D-10

Table A7. Set 2. $x_i = 1$, i=1,6,.., *else* $x_i = 0$, before scaling.

	Num. iter.	η	$\eta\,\kappa(A)$	$\omega_1^{(2)}$	$\omega_2^{(2)}$	$\omega_1^{(2)}\,\kappa_{\omega_1}^{(2)}+\omega_2^{(2)}\,\kappa_{\omega_2}^{(2)}$	Error
GRE115	1	0.32E-17	0.22E-13	0.96E-16	0.36E-27	0.56E-14	0.29E-15
GRE185	1	0.64E-16	0.25E-10	0.11E-15	0.41E-24	0.52E-11	0.57E-13
GRE216	2	0.60E-16	0.12E-13	0.15E-15	0.10E-28	0.16E-13	0.81E-15
GRE216	1	0.12E-15	0.68E-08	0.14E-15	0.94E-25	0.50E-09	0.77E-10
GRE343	1	0.60E-16	0.18E-13	0.22E-15	0.48E-26	0.23E-13	0.67E-15
GRE512	1	0.86E-16	0.35E-13	0.22E-15	0.25E-25	0.31E-13	0.67E-15
GRE1107	3	0.77E-16	0.59E-06	0.20E-14	0.18E-22	0.18E-06	0.10E-08
WEST67	1	0.40E-16	0.12E-13	0.16E-15	0.28E-30	0.79E-14	0.13E-14
WEST132	1	0.17E-16	0.16E-12	0.17E-15	0.78E-31	0.11E-12	0.54E-14
WEST156	0	0.61E-17	0.56E-05	0.10E-15	0.14E-29	0.30E-07	0.32E-08
WEST167	0	0.21E-16	0.94E-13	0.18E-15	0.50E-19	0.16E-12	0.24E-14
WEST381	1	0.35E-16	0.13E-10	0.12E-15	0.57E-29	0.27E-12	0.86E-13
WEST479	2	0.37E-17	0.10E-11	0.16E-15	0.33E-30	0.90E-12	0.28E-13
WEST497	1	0.52E-17	0.22E-10	0.11E-15	0.13E-30	0.81E-13	0.22E-14
WEST655	2	0.13E-16	0.55E-11	0.19E-15	0.60E-29	0.22E-11	0.61E-14
WEST989	1	0.32E-16	0.19E-10	0.20E-15	0.63E-29	0.43E-11	0.48E-13
WEST1505	1	0.32E-16	0.21E-08	0.20E-15	0.36E-28	0.54E-09	0.97E-11
WEST2021	1	0.32E-16	0.27E-10	0.20E-15	0.95E-29	0.85E-11	0.18E-13

Table A8. Set 2. $x_i = 1$, i=1,6,.., *else* $x_i = 0$, after scaling.

	Before scaling			After scaling		
	$\kappa(A)$	$\kappa_{\omega_1}^{(3)}$	$\kappa_{\omega_2}^{(3)}$	$\kappa(A)$	$\kappa_{\omega_1}^{(3)}$	$\kappa_{\omega_2}^{(3)}$
GRE115	0.93D+02	0.33D+02	0.38D+02	0.69D+04	0.58D+02	0.29D+04
GRE185	0.38D+06	0.50D+05	0.93D+05	0.39D+06	0.46D+05	0.92D+05
GRE216	0.28D+03	0.90D+02	0.84D+02	0.20D+03	0.11D+03	0.82D+02
GRE216	0.83D+15	0.37D+14	0.18D+15	0.56D+08	0.35D+07	0.19D+08
GRE343	0.47D+03	0.16D+03	0.10D+03	0.30D+03	0.10D+03	0.85D+02
GRE512	0.46D+03	0.14D+03	0.14D+03	0.40D+03	0.14D+03	0.12D+03
GRE1107	0.18D+09	0.40D+08	0.42D+08	0.77D+10	0.91D+08	0.21D+10
WEST67	0.91D+03	0.54D+02	0.45D+02	0.30D+03	0.51D+02	0.24D+02
WEST132	0.11D+13	0.26D+07	0.39D+11	0.94D+04	0.61D+03	0.27D+04
WEST156	0.12D+32	0.12D+09	0.44D+25	0.91D+12	0.28D+09	0.23D+11
WEST167	0.69D+11	0.45D+05	0.68D+09	0.46D+04	0.86D+03	0.15D+04
WEST381	0.53D+07	0.16D+05	0.29D+07	0.38D+06	0.23D+04	0.30D+05
WEST479	0.49D+12	0.12D+06	0.28D+12	0.27D+06	0.57D+04	0.28D+05
WEST497	0.38D+12	0.75D+06	0.10D+12	0.42D+07	0.73D+03	0.85D+06
WEST655	0.49D+12	0.66D+06	0.18D+12	0.42D+06	0.12D+05	0.20D+05
WEST989	0.13D+13	0.45D+07	0.73D+12	0.58D+06	0.21D+05	0.11D+06
WEST1505	0.14D+13	0.49D+07	0.11D+13	0.67D+08	0.27D+07	0.17D+06
WEST2021	0.28D+13	0.50D+07	0.14D+13	0.86D+06	0.42D+05	0.12D+06

Table A9. Set 3. Condition numbers before and after scaling.

	Num. iter.	η	$\eta\,\kappa(A)$	$\omega_1^{(3)}$	$\omega_2^{(3)}$	$\omega_1^{(3)}\,\kappa_{\omega_1}^{(3)} + \omega_2^{(3)}\,\kappa_{\omega_2}^{(3)}$	Error
GRE115	1	0.35D-16	0.32D-14	0.84D-16	0.89D-28	0.27D-14	0.71D-15
GRE185	1	0.94D-16	0.35D-10	0.19D-15	0.24D-25	0.94D-11	0.14D-12
GRE216	1	0.12D-16	0.34D-14	0.56D-16	0.64D-27	0.50D-14	0.13D-15
GRE216	4	0.51D-16	0.42D-01	0.41D-15	0.25D-26	0.15D-01	0.76D-06
GRE343	1	0.14D-16	0.65D-14	0.56D-16	0.12D-25	0.90D-14	0.11D-15
GRE512	1	0.25D-16	0.11D-13	0.83D-16	0.34D-25	0.12D-13	0.19D-15
GRE1107	2	0.42D-16	0.78D-08	0.20D-15	0.58D-24	0.82D-08	0.83D-10
WEST67	1	0.42D-16	0.38D-13	0.16D-15	0.50D-30	0.88D-14	0.12D-14
WEST132	1	0.24D-16	0.25D-04	0.13D-15	0.80D-28	0.34D-09	0.16D-10
WEST156	1	0.12D-22	0.14D+09	0.86D-16	0.17D-27	0.75D-03	0.10D-10
WEST167	0	0.28D-17	0.19D-06	0.20D-15	0.18D-16	0.12D-07	0.37D-12
WEST381	1	0.78D-17	0.41D-10	0.15D-15	0.40D-29	0.24D-11	0.29D-12
WEST479	3	0.33D-19	0.16D-07	0.33D-15	0.14D-28	0.39D-10	0.91D-12
WEST497	1	0.12D-17	0.44D-06	0.16D-15	0.28D-28	0.12D-09	0.30D-11
WEST655	3	0.88D-19	0.43D-07	0.26D-15	0.15D-25	0.17D-09	0.29D-11
WEST989	1	0.14D-16	0.19D-04	0.14D-15	0.29D-27	0.61D-09	0.26D-10
WEST1505	1	0.23D-16	0.31D-04	0.20D-15	0.67D-27	0.99D-09	0.46D-10
WEST2021	1	0.19D-16	0.52D-04	0.22D-15	0.32D-27	0.11D-08	0.24D-10

Table A10. Set 3. $x_i = 1$, i=1,6,.., $else\, x_i = 0$, before scaling.

	Num. iter.	η	$\eta\,\kappa$(A)	$\omega_1^{(3)}$	$\omega_2^{(3)}$	$\omega_1^{(3)}\,\kappa_{\omega_1}^{(3)}+\omega_2^{(3)}\,\kappa_{\omega_2}^{(3)}$	Error
GRE115	1	0.32E-17	0.22E-13	0.96E-16	0.36E-27	0.56E-14	0.29E-15
GRE185	1	0.64E-16	0.25E-10	0.11E-15	0.41E-24	0.52E-11	0.57E-13
GRE216	2	0.60E-16	0.12E-13	0.15E-15	0.12E-28	0.16E-13	0.81E-15
GRE216	1	0.12E-15	0.68E-08	0.14E-15	0.94E-25	0.50E-09	0.77E-10
GRE343	1	0.60E-16	0.18E-13	0.22E-15	0.71E-26	0.23E-13	0.67E-15
GRE512	1	0.86E-16	0.35E-13	0.22E-15	0.31E-25	0.31E-13	0.67E-15
GRE1107	3	0.77E-16	0.59E-06	0.20E-14	0.18E-22	0.18E-06	0.10E-08
WEST67	1	0.40E-16	0.12E-13	0.16E-15	0.57E-30	0.79E-14	0.13E-14
WEST132	1	0.17E-16	0.16E-12	0.17E-15	0.78E-31	0.11E-12	0.54E-14
WEST156	0	0.61E-17	0.56E-05	0.10E-15	0.14E-29	0.30E-07	0.32E-08
WEST167	0	0.21E-16	0.94E-13	0.18E-15	0.50E-19	0.16E-12	0.24E-14
WEST381	1	0.35E-16	0.13E-10	0.12E-15	0.57E-29	0.27E-12	0.86E-13
WEST479	2	0.37E-17	0.10E-11	0.16E-15	0.33E-30	0.90E-12	0.28E-13
WEST497	1	0.52E-17	0.22E-10	0.11E-15	0.13E-30	0.81E-13	0.22E-14
WEST655	2	0.13E-16	0.55E-11	0.19E-15	0.60E-29	0.22E-11	0.61E-14
WEST989	1	0.32E-16	0.19E-10	0.20E-15	0.63E-29	0.43E-11	0.48E-13
WEST1505	1	0.32E-16	0.21E-08	0.20E-15	0.36E-28	0.54E-09	0.97E-11
WEST2021	1	0.32E-16	0.27E-10	0.20E-15	0.95E-29	0.85E-11	0.18E-13

Table A11. Set 3. $x_i = 1$, i=1,6,.., $else\, x_i = 0$, after scaling.

		$drop\ tol = 10^{-5}$		$drop\ tol = 10^{-3}$	
	κ(A)	$\kappa_{\omega_1}^{(4)}$	$\kappa_{\omega_2}^{(4)}$	$\kappa_{\omega_1}^{(4)}$	$\kappa_{\omega_2}^{(4)}$
GRE115	0.69E+04	0.00E+00	0.12E+03	0.00E+00	0.12E+03
GRE185	0.39E+06	0.00E+00	0.17E+06	0.00E+00	0.14E+06
GRE216	0.20E+03	0.00E+00	0.21E+03	0.00E+00	0.21E+03
GRE216	0.84E+08	0.00E+00	0.15E+08	0.00E+00	0.10E+07
GRE343	0.30E+03	0.00E+00	0.31E+03	0.00E+00	0.26E+03
GRE512	0.40E+03	0.00E+00	0.43E+03	0.00E+00	0.37E+03
GRE1107	0.63E+10	0.00E+00	0.23E+09	0.00E+00	0.55E+07
WEST67	0.30E+03	0.29E+01	0.16E+03	0.00E+00	0.14E+03
WEST132	0.94E+04	0.00E+00	0.24E+04	0.00E+00	0.22E+04
WEST156	0.91E+12	0.00E+00	0.29E+09	0.00E+00	0.16E+06
WEST167	0.46E+04	0.00E+00	0.16E+04	0.00E+00	0.13E+04
WEST381	0.38E+06	0.00E+00	0.65E+04	0.00E+00	0.54E+04
WEST479	0.27E+06	0.00E+00	0.23E+05	0.00E+00	0.20E+05
WEST497	0.42E+07	0.00E+00	0.65E+04	0.00E+00	0.63E+04
WEST655	0.42E+06	0.00E+00	0.43E+05	0.00E+00	0.37E+05
WEST989	0.58E+06	0.00E+00	0.63E+05	0.00E+00	0.53E+05
WEST1505	0.67E+08	0.00E+00	0.35E+07	0.00E+00	0.21E+07
WEST2021	0.86E+06	0.00E+00	0.12E+06	0.00E+00	0.10E+06

Table A12. Set 4. Condition numbers after scaling for $drop\ tol. = 10^{-5}$ and $drop\ tol. = 10^{-3}$.

	Num. iter.	η	$\eta\,\kappa(A)$	$\omega_1^{(4)}$	$\omega_2^{(4)}$	$\omega_1^{(4)}\,\kappa_{\omega_1}^{(4)}+\omega_2^{(4)}\,\kappa_{\omega_2}^{(4)}$	Error
GRE115	2	0.99E-18	0.68E-14	0.00E+00	0.69E-16	0.85E-14	0.15E-14
GRE185	3	0.55E-16	0.22E-10	0.00E+00	0.50E-16	0.83E-11	0.80E-13
GRE216	1	0.90E-16	0.18E-13	0.00E+00	0.83E-16	0.18E-13	0.88E-15
GRE216	29	0.10E-15	0.84E-08	0.00E+00	0.50E-15	0.77E-08	0.63E-10
GRE343	1	0.90E-16	0.27E-13	0.00E+00	0.83E-16	0.26E-13	0.81E-15
GRE512	1	0.86E-16	0.35E-13	0.00E+00	0.11E-15	0.48E-13	0.68E-15
GRE1107	15	0.62E-16	0.39E-06	0.00E+00	0.19E-15	0.44E-07	0.27E-09
WEST67	1	0.50E-16	0.15E-13	0.13E-16	0.61E-16	0.10E-13	0.85E-15
WEST132	2	0.36E-16	0.33E-12	0.00E+00	0.67E-16	0.16E-12	0.46E-14
WEST156	0	0.61E-17	0.56E-05	0.00E+00	0.54E-16	0.16E-07	0.32E-08
WEST167	0	0.21E-16	0.94E-13	0.00E+00	0.67E-16	0.11E-12	0.24E-14
WEST381	2	0.23E-16	0.89E-11	0.00E+00	0.54E-16	0.36E-12	0.78E-13
WEST479	3	0.26E-16	0.71E-11	0.00E+00	0.57E-16	0.13E-11	0.55E-13
WEST497	1	0.58E-17	0.25E-10	0.00E+00	0.55E-16	0.36E-12	0.47E-14
WEST655	2	0.55E-16	0.23E-10	0.00E+00	0.91E-16	0.39E-11	0.22E-13
WEST989	1	0.13E-15	0.75E-10	0.00E+00	0.19E-15	0.12E-10	0.18E-13
WEST1505	2	0.64E-16	0.43E-08	0.00E+00	0.10E-15	0.35E-09	0.10E-10
WEST2021	2	0.95E-16	0.82E-10	0.00E+00	0.13E-15	0.16E-10	0.59E-13

Table A13. Set 4. $x_i = 1$, i=1,6,.., *else* $x_i = 0$, after scaling and *drop tol.* $= 10^{-5}$.

	Num. iter.	η	$\eta\,\kappa(A)$	$\omega_1^{(4)}$	$\omega_2^{(4)}$	$\omega_1^{(4)}\,\kappa_{\omega_1}^{(4)}+\omega_2^{(4)}\,\kappa_{\omega_2}^{(4)}$	Error
GRE115	4	0.35E-17	0.24E-13	0.00E+00	0.48E-16	0.59E-14	0.80E-15
GRE185	15	0.46E-16	0.15E-10	0.00E+00	0.61E-16	0.87E-11	0.14E-12
GRE216	1	0.65E-16	0.13E-13	0.00E+00	0.74E-16	0.16E-13	0.11E-14
GRE216	3	0.26E-04	0.15E+03	0.00E+00	0.11E-02	0.12E+04	0.22E+01
GRE343	3	0.66E-16	0.20E-13	0.00E+00	0.87E-16	0.23E-13	0.72E-15
GRE512	4	0.63E-16	0.26E-13	0.00E+00	0.89E-16	0.32E-13	0.79E-15
GRE1107	3	0.64E-05	0.10E+04	0.00E+00	0.16E-02	0.90E+04	0.13E+01
WEST67	2	0.37E-16	0.11E-13	0.00E+00	0.45E-16	0.61E-14	0.14E-14
WEST132	3	0.25E-16	0.23E-12	0.00E+00	0.52E-16	0.11E-12	0.21E-14
WEST156	0	0.59E-18	0.73E-08	0.00E+00	0.54E-16	0.87E-11	0.18E-12
WEST167	0	0.21E-16	0.94E-13	0.00E+00	0.67E-16	0.84E-13	0.24E-14
WEST381	4	0.17E-16	0.67E-11	0.00E+00	0.53E-16	0.29E-12	0.33E-13
WEST479	7	0.34E-17	0.91E-12	0.00E+00	0.55E-16	0.11E-11	0.51E-13
WEST497	4	0.30E-17	0.13E-10	0.00E+00	0.58E-16	0.36E-12	0.36E-14
WEST655	5	0.28E-16	0.12E-10	0.00E+00	0.65E-16	0.24E-11	0.55E-13
WEST989	5	0.32E-16	0.19E-10	0.00E+00	0.64E-16	0.34E-11	0.11E-12
WEST1505	10	0.32E-16	0.20E-08	0.00E+00	0.90E-16	0.19E-09	0.23E-10
WEST2021	5	0.32E-16	0.27E-10	0.00E+00	0.94E-16	0.98E-11	0.72E-13

Table A14. Set 4. $x_i = 1$, i=1,6,.., *else* $x_i = 0$, after scaling and *drop tol.* $= 10^{-3}$.

	Nonzeros	Fill-in		
		$drop\ tol = 0.0$	$drop\ tol = 10^{-5}$	$drop\ tol = 10^{-3}$
GRE115	421	647	651	605
GRE185	975	3173	3028	2929
GRE216	812	2544	2263	2262
GRE216	812	2767	2580	2180
GRE343	1310	5334	4891	4890
GRE512	1976	11535	11020	11007
GRE1107	5664	47603	45255	41181
WEST67	294	267	202	204
WEST132	413	89	87	83
WEST156	362	27	20	15
WEST167	506	96	96	92
WEST381	2134	2057	1867	1711
WEST479	1888	1121	982	790
WEST497	1721	279	263	252
WEST655	2808	2092	1791	1709
WEST989	3518	1156	1139	1135
WEST1505	5414	2032	1934	1821
WEST2021	7310	2539	2466	2410

Table A15. Set 4. Number of nonzero entries in the original matrices and fill-in for $drop\ tol. = 0.0$, $drop\ tol. = 10^{-5}$ and $drop\ tol. = 10^{-3}$ after scaling.