BIT 31 (1991), 447–468.

ITERATIVE REFINEMENT ENHANCES THE STABILITY OF *QR* FACTORIZATION METHODS FOR SOLVING LINEAR EQUATIONS

NICHOLAS J. HIGHAM

Department of Mathematics, University of Manchester, Manchester M13 9PL, England

Abstract.

Iterative refinement is a well-known technique for improving the quality of an approximate solution to a linear system. In the traditional usage residuals are computed in extended precision, but more recent work has shown that fixed precision is sufficient to yield benefits for stability. We extend existing results to show that fixed precision iterative refinement renders an *arbitrary* linear equations solver backward stable in a strong, componentwise sense, under suitable assumptions. Two particular applications involving the *QR* factorization are discussed in detail: solution of square linear systems and solution of least squares problems. In the former case we show that one step of iterative refinement suffices to produce a small componentwise relative backward error. Our results are weaker for the least squares problem, but again we find that iterative refinement improves a componentwise measure of backward stability. In particular, iterative refinement mitigates the effect of poor row scaling of the coefficient matrix, and so provides an alternative to the use of row interchanges in the Householder *QR* factorization. A further application of the results is described to fast methods for solving Vandermonde-like systems.

AMS (MOS) subject classifications: Primary: 65F05, 65G05.

Key-words: Iterative refinement, linear system, least squares problem, *QR* factorization, Gaussian elimination, partial pivoting, rounding error analysis, backward error, componentwise error bounds, Householder transformations, Givens transformations, confluent Vandermonde-like matrices.

1. Introduction.

Iterative refinement is an established technique for improving a computed solution $\hat{x}$ to a linear system $Ax = b$. The process consists of three steps:

1. Compute $r = b - A\hat{x}$.
2. Solve $Ad = r$.
3. Update $y = \hat{x} + d$.

(Repeat from step 1 if necessary, with $\hat{x}$ replaced by y).

Received July 1990. Revised March 1991.

Traditionally, the method is used with Gaussian elimination, and r is computed in extended precision before being rounded to working precision. Iterative refinement for Gaussian elimination was used in the 1940s on desk calculators, but the first thorough analysis of the method was given by Wilkinson in 1963 [35]. The behaviour of mixed precision iterative refinement is now well understood (see [15, 25, 32] for example): if double precision is used in the computation of r, and A is not too ill-conditioned, then the iteration produces a solution correct to working precision, and the rate of convergence depends on the condition number of A.

In the last ten years or so an alternative usage of iterative refinement has gained popularity, in which the residual is computed in the working precision. This interest in fixed precision iterative refinement was prompted by two papers that appeared in the late 1970s. Jankowski and Woźniakowski [23] proved that an arbitrary linear equation solver is made backward stable by the use of fixed precision iterative refinement, as long as the solver is not too unstable to begin with and A is not too ill-conditioned. Skeel [30] proved that Gaussian elimination with partial pivoting becomes stable in a much stronger sense than usual after just one step of fixed precision iterative refinement, again under suitable assumptions.

The purpose of this work is to show that an *arbitrary* linear equation solver can be made stable in the strong, componentwise sense considered by Skeel with the use of fixed precision iterative refinement, as long as certain mild conditions are satisfied. In a sense this result combines the best features of the results of Jankowski and Woźniakowski and of Skeel. The result has three particularly interesting implications:

1. QR factorization with iterative refinement for solving $Ax = b$ matches the stability of Gaussian elimination with partial pivoting and iterative refinement.
2. QR factorization with iterative refinement for solving least squares problems yields a small componentwise backward error, asymptotically, and consequently the overall method is insensitive to poor row scaling of the coefficient matrix.
3. The fast methods for solving Vandermonde-like systems of [11, 13, 17, 18] are numerically stable when coupled with iterative refinement. This had previously been observed empirically, but theoretical explanations were lacking.

The outline of this paper is as follows. In section 2 we develop results for fixed precision iterative refinement with an arbitrary linear equation solver. For the particular case of Gaussian elimination we compare our results with those of Skeel. We also discuss the application of our results to fast algorithms for solving Vandermonde-like systems. In section 3 the results of section 2 are applied to the method of QR factorization for solving linear systems. Numerical experiments are reported to illustrate the theory.

The least squares problem is considered in section 4. We analyze iterative refinement applied to the augmented equations in conjunction with a QR factoriz-

ation. We obtain an asymptotic result which shows that iterative refinement leads to a small componentwise backward error. Again, numerical experiments are included.

The work in sections 3 and 4 makes use of a componentwise error analysis for the Householder and Givens QR factorization algorithms. Since traditional error analyses for these algorithms involve normwise bounds we had to develop a new componentwise analysis. The results of the analysis are stated in an appendix, and the proofs may be found in [20].

For an excellent, up to date survey of both fixed and mixed precision iterative refinement and their applications see [9].

We stress that in this work we concentrate exclusively on *fixed precision* iterative refinement, and we will often refer to it simply as "iterative refinement".

2. Main result.

To assess the stability of linear equation solvers with iterative refinement we will use the notion of *componentwise relative backward error*. The componentwise relative backward error for an approximate solution y to $Ax = b$, where $A \in \mathbb{R}^{n\times n}$, is the quantity

$$\omega(y) = \min\{\varepsilon : (A + \Delta A)y = b + \Delta b, \quad |\Delta A| \le \varepsilon|A|, \quad |\Delta b| \le \varepsilon|b|\}, \tag{2.1}$$

where matrix absolute values and inequalities are interpreted componentwise. Thus ω is the size of the smallest perturbation we have to make to A and b for y to be an exact solution of the perturbed system, where each individual perturbation is measured relative to the element that it perturbs. Note that ω can be much larger than the normwise relative backward error

$$\eta(y) = \min\{\varepsilon : (A + \Delta A)y = b + \Delta b, \quad \|\Delta A\| \le \varepsilon\|A\|, \quad \|\Delta b\| \le \varepsilon\|b\|\}. \tag{2.2}$$

In fact, ω may be infinite, in which case no perturbations of the specified structure exist.

When working in floating point arithmetic with unit roundoff u the best that we can hope for is $\omega \approx u$. The question we are interested in is whether iterative refinement helps to achieve this goal. A result of Oettli and Prager [26] provides the convenient expression

$$\omega(y) = \max_i \frac{|b - Ay|_i}{(|A||y| + |b|)_i}, \tag{2.3}$$

where $\zeta/0$ is interpreted as zero if $\zeta = 0$ and infinity otherwise. Thus the approach we take in our analysis is to attempt to bound $|b - Ay|$ by a scalar multiple of $|A||y| + |b|$.

We will use the following model of floating point arithmetic, which allows for possible lack of a guard digit in addition and subtraction:

$$\begin{aligned} fl(x \text{ op } y) &= (x \text{op } y)(1+\delta), \quad |\delta| \le u, \ \text{op} = *, /, \\ fl(x \pm y) &= x(1+\alpha) \pm y(1+\beta), \quad |\alpha|, |\beta| \le u, \\ fl(\sqrt{x}) &= \sqrt{x}(1+\delta), \quad |\delta| \le u. \end{aligned} \tag{2.4}$$

Computed quantities are denoted with a hat.

To make our analysis as widely applicable as possible we make only very general assumptions about the linear equation solver and the method for computing the residual. We assume that the computed solution $\hat{x}$ to $Ax = b$ satisfies

$$|b - A\hat{x}| \le u(g(A,b)|\hat{x}| + h(A,b)) \tag{2.5}$$

where g: $\mathbb{R}^{n \times (n+1)} \to \mathbb{R}^{n \times n}$ and h: $\mathbb{R}^{n \times (n+1)} \to \mathbb{R}^n$ have nonnegative entries. The functions g and h may depend on n and u as well as on the data A and b. We also assume that the residual $r = b - A\hat{x}$ is computed in such a way that

$$|\hat{r} - r| \le ut(A,b,\hat{x}), \tag{2.6}$$

where t: $\mathbb{R}^{n \times (n+2)} \to \mathbb{R}^n$ is nonnegative. It is straightforward to show that if r is computed in the conventional way, in the working precision via inner products or saxpy operations, then we can take

$$t(A,b,\hat{x}) = \frac{\gamma_{n+1}}{u}(|A||\hat{x}| + |b|), \tag{2.7}$$

where $\gamma_k \equiv ku/(1-ku)$.

THEOREM 2.1. *Let $A \in \mathbb{R}^{n \times n}$ be nonsingular. Suppose the linear system $Ax = b$ is solved in floating point arithmetic using a solver S together with one step of iterative refinement. Assume that the computed solution $\hat{x}$ produced by S satisfies* (2.5) *and that the computed residual $\hat{r}$ satisfies* (2.6). *Then the corrected solution $\hat{y}$ satisfies*

$$|b - A\hat{y}| \le u(h(A,\hat{r}) + t(A,b,\hat{y}) + |A||\hat{y}|) + uq, \tag{2.8}$$

where $q = O(u)$ if $t(A,b,\hat{x}) - t(A,b,\hat{y}) = O(\|\hat{x} - \hat{y}\|_2)$.

PROOF. The residual $r = b - A\hat{x}$ of the original computed solution $\hat{x}$ satisfies

$$|r| \le u(g(A,b)|\hat{x}| + h(A,b)). \tag{2.9}$$

The computed residual is $\hat{r} = r + \Delta r$, where $|\Delta r| \le ut(A,b,\hat{x})$. The computed correction $\hat{d}$ satisfies

$$A\hat{d} = \hat{r} + f_1, \quad |f_1| \le u(g(A,\hat{r})|\hat{d}| + h(A,\hat{r})). \tag{2.10}$$

Finally, for the corrected solution we have

$$\hat{y} = fl(\hat{x} + \hat{d}) = \hat{x} + \hat{d} + f_2, \quad |f_2| \le u(|\hat{x}| + |\hat{d}|). \tag{2.11}$$

Collecting together the above results we obtain

$$\begin{aligned} b - A\hat{y} &= b - A\hat{x} - A\hat{d} - Af_2 \\ &= \hat{r} - \Delta r - A\hat{d} - Af_2 \\ &= -f_1 - \Delta r - Af_2. \end{aligned}$$

Hence

$$\begin{aligned} |b - A\hat{y}| &\le u(g(A,\hat{r})|\hat{d}| + h(A,\hat{r})) + ut(A,b,\hat{x}) + u|A|(|\hat{x}| + |\hat{d}|) \\ &= u(h(A,\hat{r}) + t(A,b,\hat{y}) + |A||\hat{y}|) + uq, \end{aligned} \tag{2.12}$$

where

$$q = t(A,b,\hat{x}) - t(A,b,\hat{y}) + g(A,\hat{r})|d| + |A|(|\hat{x}| - |\hat{y}| + |\hat{d}|).$$

The claim about the order of q follows since $\hat{x} - \hat{y}$, $|\hat{x}| - |\hat{y}|$ and $\hat{d}$ are all of order u. ∎

Theorem 2.1 shows that, to first order, the componentwise relative backward error ω will be small after one step of iterative refinement as long as $h(A,\hat{r})$ and $t(A,b,\hat{y})$ are bounded by a modest scalar multiple of $|A||\hat{y}| + |b|$. This is true for t if the residual is computed in the conventional way (see (2.7)), and in some cases we may take $h \equiv 0$, as shown below. Note that the function g of (2.5) does not appear in the first order term of (2.8). This is the essential reason why iterative refinement works: potential instability manifested in g is suppressed by the refinement stage.

A weakness of Theorem 2.1 is that the bound (2.8) is asymptotic. Since a strict bound for q is not given it is difficult to draw firm conclusions about the size of ω. The next result overcomes this drawback, at the cost of some specialization (and a rather long proof).

We introduce the condition number of Skeel [29]

$$\operatorname{cond}(B) = \| |B^{-1}|\ |B| \|_\infty, \tag{2.13}$$

and the measure of ill-scaling of the vector $|B|\ |x|$

$$\sigma(B,x) = \frac{\max_i (|B|\ |x|)_i}{\min_i (|B|\ |x|)_i}.$$

THEOREM 2.2. *Under the conditions of Theorem* 2.1, *suppose that* $g(A,b) = G|A|$ *and* $h(A,b) = H|b|$, *where* $G, H \in \mathbb{R}^{n\times n}$ *have nonnegative entries, and that the residual is computed in the conventional manner. Then there is a function*

$$f(t_1,t_2) \approx (t_2(t_1 + n + 1)/\operatorname{cond}(A^{-1}) + 2(t_1 + n + 2)^2(1 + ut_2)^2)/(n + 1)$$

such that if

$$\operatorname{cond}(A^{-1})\sigma(A,\hat{y}) \le (f(\|G\|_\infty, \|H\|_\infty)u)^{-1}$$

then

$$|b - A\hat{y}| \le 2\gamma_{n+1}|A|\ |\hat{y}|.$$

PROOF. From (2.12) in the proof of Theorem 2.1, using the formula (2.7) for t, we have

$$|b - A\hat{y}| \le u|\hat{r}| + \gamma_{n+1}|b| + (\gamma_{n+1} + u)|A|\ |\hat{x}| + u(I + G)|A|\ |\hat{d}|. \tag{2.14}$$

The inequality (2.9) implies

$$|b| - |A|\ |\hat{x}|\ |b - A\hat{x}| \le u(G|A|\ |\hat{x}| + H|b|),$$

or $(I - uH)|b| \le (I + uG)|A|\ |\hat{x}|$. If $u\|H\|_\infty < \frac{1}{2}$ (say) then $I - uH$ is nonsingular with a nonnegative inverse satisfying $\|(I - uH)^{-1}\|_\infty \le 2$ and we can solve for $|b|$ to obtain $|b| \le (I - uH)^{-1}(I + uG)|A|\ |\hat{x}|$. It follows from this relation and consideration of the rest of the proof that the simplifying step of replacing b by 0 in the analysis has little effect on the bounds – it merely produces unimportant perturbations in f in the statement of the theorem. Making this replacement in (2.14), and approximating $\gamma_{n+1} + u \simeq \gamma_{n+1}$ we have

$$|b - A\hat{y}| \le u|\hat{r}| + \gamma_{n+1}|A|\ |\hat{x}| + u(I + G)|A|\ |\hat{d}|. \tag{2.15}$$

Our task is now to bound $|A|\ |\hat{x}|$, $|\hat{r}|$ and $|A|\ |\hat{d}|$ in terms of $|\hat{y}|$. By manipulating (2.11) we obtain the inequality

$$|\hat{x}| \le (1 - u)^{-1}(|\hat{y}| + (1 + u)|\hat{d}|) \simeq |\hat{y}| + |\hat{d}|. \tag{2.16}$$

Also, we can bound $|\hat{r}|$ by

$$|\hat{r}| \le |r| + |\Delta r| \le u(G|A|\ |\hat{x}| + H|b|) + \gamma_{n+1}(|A|\ |\hat{x} + |b|),$$

and dropping the $|b|$ terms and using (2.16) gives

$$|\hat{r}| \le (uG + \gamma_{n+1}I)|A|\ |\hat{x}| \le (uG + \gamma_{n+1}I)|A|(|\hat{y}| + |\hat{d}|). \tag{2.17}$$

Substituting from (2.16) and (2.17) into (2.15) we find

$$\begin{aligned} |b - A\hat{y}| &\le uH(uG + \gamma_{n+1}I)|A|(|\hat{y}| + |\hat{d}|) + \gamma_{n+1}|A|(|\hat{y}| + |\hat{d}|) + u(I + G)|A|\ |\hat{d}| \\ &= (\gamma_{n+1}I + uH(uG + \gamma_{n+1}I))|A|\ |\hat{y}| \\ &\quad + (\gamma_{n+1}I + u(I + G) + uH(uG + \gamma_{n+1}I))|A|\ |\hat{d}| \\ &\equiv (\gamma_{n+1}I + M_1)|A|\ |\hat{y}| + M_2|A|\ |\hat{d}|, \end{aligned} \tag{2.18}$$

where

$$\|M_1\|_\infty \le u\|H\|_\infty(u\|G\|_\infty + \gamma_{n+1}),$$

$$\|M_2\|_\infty \le \gamma_{n+2} + u\|G\|_\infty + u\|H\|_\infty(u\|G\|_\infty + \gamma_{n+1}).$$

Now from (2.10), making use of (2.17),

$$\begin{aligned} |\hat{d}| &= |A^{-1}(\hat{r} + f_1)| \\ &\le |A^{-1}|(|\hat{r}| + uG|A|\ |\hat{d}| + uH|\hat{r}|) \\ &\le |A^{-1}|((I + uH)(uG + \gamma_{n+1}I)|A|(|\hat{y}| + |\hat{d}|) + uG|A|\ |\hat{d}|). \end{aligned}$$

After pre-multiplying by $|A|$ this may be re-arranged as

$$(I - uM_3)|A|\ |\hat{d}| \le u|A|\ |A^{-1}|M_4|A|\ |\hat{y}|, \tag{2.19}$$

where

$$M_3 = |A|\ |A^{-1}|((I + uH)(G + (\gamma_{n+1}/u)I) + G),$$

$$M_4 = (I + uH)(G + (\gamma_{n+1}/u)I).$$

Using $\gamma_{n+1}/u \le (n+1)/(1 - (n+1)u) \simeq n + 1$, we have the bounds

$$\|M_3\|_\infty \le \operatorname{cond}(A^{-1})(\|G\|_\infty + n + 1)(2 + u\|H\|_\infty),$$

$$\|M_4\|_\infty \le (\|G\|_\infty + n + 1)(1 + u\|H\|_\infty).$$

If $u\|M_3\|_\infty < 1/2$ (say) then $(I - uM_3)^{-1} \ge 0$ with $\|(I - uM_3)^{-1}\|_\infty \le 2$ and we can rewrite (2.19) as

$$|A|\ |\hat{d}| \le u(I - uM_3)^{-1}|A|\ |A^{-1}|M_4|A|\ |\hat{y}|. \tag{2.20}$$

Substituting this bound into (2.18) we otain

$$\begin{aligned} |b - A\hat{y}| &\le (\gamma_{n+1}I + M_1 + uM_2(I - uM_3)^{-1}|A|\ |A^{-1}|M_4)|A|\ |\hat{y}| \\ &\equiv (\gamma_{n+1}I + M_5)|A|\ |\hat{y}| \\ &\le \omega|A|\ |\hat{y}|, \end{aligned} \tag{2.21}$$

where

$$\omega = \gamma_{n+1} + \|M_5\|_\infty \sigma(A, \hat{y}).$$

Finally, we bound $\|M_5\|_\infty$. Writing $g = \|G\|_\infty, h = \|H\|_\infty$, we have

$$\begin{aligned} \|M_5\|_\infty \le u^2gh + uh\gamma_{n+1} + 2u(\gamma_{n+2} + ug + u^2gh + uh\gamma_{n+1})\cdot \\ \cdot \operatorname{cond}(A^{-1})(g + n + 1)(1 + uh) \end{aligned}$$

and this expression is approximately bounded by

$$u^2(h(g + n + 1) + 2(g + n + 2)^2\,(1 + uh)^2 \operatorname{cond}(A^{-1})).$$

Requiring $\|M_5\|_\infty \sigma(A, \hat{y})$ not to exceed γ_{n+1} leads to the result. ■

Theorem 2.2 says that as long as A is not too ill-conditioned and $|A|\ |\hat{y}|$ is not too badly scaled ($\operatorname{cond}(A^{-1})\sigma(A, \hat{y})$ is not too large) and the solver S is not too unstable ($f(\|G\|_\infty, \|H\|_\infty)$ is not too large) then $\omega \le 2\gamma_{n+1}$ after one step of iterative refinement. Note that the term $\gamma_{n+1}|A|\ |\hat{y}|$ in (2.21) comes from the error bound for

evaluation of the residual, so this bound for ω is about the smallest we could expect to prove.

It is instructive to apply Theorem 2.2 to Gaussian elimination and to make a comparison with results of Skeel. Suppose, then, that the solver S is Gaussian elimination with or without pivoting, and in the latter case assume (without loss of generality) that no interchanges are required. Standard error analysis results (see, e.g., [22]) show that in (2.5) we can take

$$g(A,b) = c_n|\hat{L}|\ |\hat{U}|, \qquad h(A,b) = 0,$$

where $\hat{L}$, $\hat{U}$ are the computed LU factors of A and $c_n \simeq n$. To apply Theorem 2.2 we use $A \simeq \hat{L}\hat{U}$ and write

$$g(A,b) \simeq c_n|\hat{L}|\ |\hat{L}^{-1}A| \leq c_n|\hat{L}|\ |\hat{L}^{-1}|\ |A|,$$

which shows that we can take $G = c_n|\hat{L}|\ |\hat{L}^{-1}|$ and $f(\|G\|_\infty,\ \|H\|_\infty) \simeq 2nu\||\hat{L}|\ |\hat{L}^{-1}|\|_\infty^2$. Without pivoting the growth factor-type term $\||\hat{L}|\ |\hat{L}^{-1}|\|_\infty$ is unbounded, but with partial pivoting it cannot exceed 2^n and is typically $O(n)$ [33].

We can conclude that for Gaussian elimination with partial pivoting one step of iterative refinement will usually be enough to yield a small componentwise relative backward error as long as A is not too ill-conditioned and $|A|\ |\hat{y}|$ is not too badly scaled. Without pivoting the same holds true with the added proviso that the computation of the original $\hat{x}$ must not be "too unstable". Some numerical experiments are reported in the next section.

Note that for some special classes of matrix the componentwise relative backward error is guaranteed to be small for the original $\hat{x}$ produced by Gaussian elimination without pivoting; see [22] for details and references. In such cases there is no benefit in doing iterative refinement in fixed precision.

These results for Gaussian elimination are very similar to those of Skeel [30]. The main differences are that Skeel's analysis covers an arbitrary number of refinement steps with residuals computed in single or double precision, his analysis is specific to Gaussian elimination, and his results involve $\sigma(A,x)$ rather than $\sigma(A,\hat{y})$. Our statements and proofs of Theorems 2.1 and 2.2 were strongly influenced by the work in [30].

A second application of our results is to methods given in [13, 11, 17, 18] for solving $n \times n$ Vandermonde systems in $O(n^2)$ operations. It is known that some of these methods can be unstable, but practical experience indicates that iterative refinement usually cures the instability [17, 18]. An error analysis covering all the algorithms is given in [18, Theorem 3.2] and it shows that a result of the form (2.5) holds (with $h = 0$). Hence Theorem 2.1 is applicable. (Theorem 2.2 is not directly applicable because $g(A,b)$ is not of the form $G|A|$.)

When solving Vandermonde systems the coefficient matrix is usually not available, so residuals are computed using some form of nested multiplication. In the case of (confluent) Vandermonde matrices based on the monomials, the residuals are formed using Horner's rule, and it is straightforward to show that (2.7) holds (error

analysis of Horner's rule is given in [35, pp. 36–37], for example). Hence for standard Vandermonde matrices Theorem 2.1 leads to an asymptotic componentwise stability result. For (confluent) Vandermonde-like matrices based on orthogonal polynomials the residuals are computed using an extension of the Clenshaw recurrence [18, 31]. A complete error analysis of this recurrence is not available but it is easy to see that (2.7) will not always hold. Nevertheless it is clear that a *normwise* bound can be obtained (see [27] for the special case of the Chebyshev polynomials) and hence an asymptotic normwise stability result can be deduced from Theorem 2.1. Thus our results provide theoretical backing for the use of iterative refinement with fast solvers for Vandermonde systems.

3. *QR* factorization for linear systems.

In this section we consider the use of fixed precision iterative refinement with QR factorization methods for solving $Ax = b$, where $A \in \mathbb{R}^{n \times n}$. Specifically, we suppose that a QR factorization $A = QR$ is computed using Householder or Givens transformations and x is obtained by solving $Rx = Q^T b$.

Since we are interested in the componentwise relative backward error we need a componentwise backward error result for QR factorization solution of $Ax = b$. The standard result is expressed in terms of norms: from Wilkinson's analysis of Householder or Givens QR factorization and back substitution [36, pp. 236, 240, 247] it follows that the computed $\hat{x}$ satisfies

$$(3.1) \quad (A + \Delta A)\hat{x} = b + \Delta b, \qquad \|\Delta A\|_2 \le p(n)u\|A\|_2, \qquad \|\Delta b\|_2 \le p(n)u\|b\|_2,$$

where $p(n)$ is a linear polynomial. This result shows that the normwise relative backward error $\eta(\hat{x})$ is small.

We have carried out a detailed componentwise error analysis, the result of which is presented in the appendix. Lemma A.1 shows that $\hat{x}$ satisfies

$$(3.2) \qquad |b - A\hat{x}| \le u(G|A|\ |\hat{x}| + H|b|),$$

where $\|G\|_2$ and $\|H\|_2$ are both bounded by a low degree polynomial in n. The matrices G and H have no special structure, and so (3.2) suggests that the componentwise relative backward error need not be small when x is computed via a QR factorization. In fact, we know of no class of A for which Householder or Givens QR factorization is guaranteed to yield a small componentwise relative backward error.

Suppose, then, that we carry out a step of iterative refinement, to obtain $\hat{y}$. By the form of the bound (3.2) we can invoke Theorem 2.2. We conclude that the componentwise relative backward error $\omega(\hat{y})$ will be small as long as A is not too ill-conditioned and $|A|\ |\hat{y}|$ is not too badly scaled. This is, of course, precisely the same conclusion as for Gaussian elimination with partial pivoting (GEPP).

This conclusion is interesting because it sheds further light on the comparison between the competing methods of QR factorization and GEPP for solving linear

systems. The accepted reasoning is that GEPP is faster but QR factorization has guaranteed stability (in the sense of normwise backward error). We have shown that QR factorization matches GEPP in the ability to produce a small componentwise relative backward error when combined with iterative refinement. Concerning speed, Golub and Van Loan [15, p. 257] comment

> The flop counts tend to exaggerate the Gaussian elimination advantage. When memory traffic and vectorization overheads are considered, the QR approach is comparable in efficiency.

The two methods for solving $Ax = b$ therefore seem to be quite closely matched – for general, dense matrices, at least.

We have carried out numerical experiments in MATLAB to investigate the practical performance of iterative refinement. We used Householder QR factorization and Gaussian elimination both with and without partial pivoting. For each method we computed for each iterate z the componentwise relative backward error $\omega(z)$ using (2.3) and took "$\omega(z) \leq u$" as the termination criterion. In our MATLAB computing environment $u = 2^{-52} \simeq 2.22 \times 10^{-16}$. Some selected results are presented in Tables 3.1–3.7. In the columns of the tables are reported the ω values for each method. "Fail" in the GE column denotes that Gaussian elimination broke down with a zero pivot. Also reported are the standard condition number $\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty$, the condition number appropriate to componentwise perturbations in the data, $\operatorname{cond}(A) = \||A^{-1}|\,|A|\|_\infty$ of (2.13) (which is no larger than $\kappa_\infty(A)$), and $\theta(A, x) = \operatorname{cond}(A^{-1})\sigma(A, x)$, which is the quantity that certainly must not exceed u^{-1} if we are to be able to conclude from Theorem 2.2 that one step of iterative refinement suffices for these methods.

The matrices referred to in the tables are from the test collection [19]. Briefly, element(n) is tridiagonal with zero diagonal entries, invhilb(n) is the inverse of the Hilbert matrix, pascal(n) is a positive definite matrix made up from the entries of Pascal's triangle, compan(n) is a companion matrix, orthog(n) is a symmetric and orthogonal matrix, and gfpp(n) is a matrix for which the growth factor for GEPP is maximal. In each case the right-hand side b was chosen as a random vector with elements between 0 and 1.

The results show several noteworthy features.

GEPP performs as predicted by our analysis and by Skeel's analysis. In fact, iterative refinement converges in one step even when $\theta(A, x) > u^{-1}$ in the examples reported and in most others we have tried. GE also achieves a small componentwise relative backward error, but can require more than one refinement step, even when $\theta(A, x)$ is small.

QR factorization with iterative refinement performs as predicted by our analysis. In most of the examples we tried where $\theta(A, x) > u^{-1}$ the refinement still converged but took two iterations.

Over the whole body of tests it was noticeable that the componentwise relative backward error for the original QR solution was usually larger than that for the original GEPP solution, by a factor typically between 2 and 10 (for $n \leq 50$). A consequence is that the refinement step was needed more often for the QR method than for GEPP.

It is worth stressing that the QR factorization yielded a small *normwise* relative backward error in every case ($\eta(\hat{x}) < u$, in fact), as it must, in view of (3.1). For GEPP, $\eta(\hat{x}) \approx 3 \times 10^{-4}$ for $A = \text{gfpp}\ (50)$, but $\eta(\hat{x}) < u$ in the other cases reported.

We note that Arioli, Demmel and Duff [2] have shown how to sidestep difficulties with iterative refinement of Gaussian elimination caused by a large value of $\sigma(A, x)$ (as is likely to arise when A and x are sparse). Their approach is to relax the definition of componentwise relative backward error as follows: if

$$(|A|\ |y| + |b|)_i \leq 1000nu(\|A(i,:)\|_\infty \|y\|_\infty + |b_i|),$$

then the inequality $|\Delta b|_i \leq \varepsilon |b_i|$ in (2.1) is relaxed to $|\Delta b|_i \leq \varepsilon \|A(i,:)\|_1 \|y\|_\infty$, which amounts to replacing $|b_i|$ in the denominator of (2.3) by $\|A(i,:)\|_1 \|y\|_\infty$. (Here, $A(i,:)$ denotes the ith row of A.) See [1] for more details. Although this strategy was developed with Gaussian elimination in mind it applies equally well to the QR factorization.

Finally, we mention row equilibration. Here, we solve the *scaled* system $(DA)x = Db$ by GEPP or QR factorization, where $B = DA$ has rows of unit 1-norm. This approach avoids the effects of poor row scaling, and we have $\kappa_\infty(B) = \text{cond}\,(A)$. However, as explained in detail in [14], there is no guarantee that row equilibration will lead to a small componentwise relative backward error, and so row equilibration is a less powerful tool than iterative refinement.

Table 3.1. ω values for $A = \text{clement}\ (10)$

$\theta(A,x) = 3.85\text{e}6$ $\text{cond}\,(A) = 9.80\text{e}0$, GEPP	GE	$\kappa_\infty(A) = 4.18\text{e}1$ QR
1.91e-13 2.52e-17	Fail	2.37e-12 9.98e-17

Table 3.2. ω values for $A = \text{invhilb}\ (10)$

$\theta(A,x) = 3.99\text{e}18$ $\text{cond}\,(A) = 5.92\text{e}12$, GEPP	GE	$\kappa_\infty(A) = 3.54\text{e}13$ QR
1.02e-16	1.25e-17	1.98e-11 1.73e-15 4.96e-17

Table 3.3. ω values for $A = \text{pascal}\ (10)$

$\theta(A,x) = 2.74\text{e}12$ $\text{cond}\,(A) = 5.02\text{e}8$, GEPP	GE	$\kappa_\infty(A) = 8.13\text{e}9$ QR
2.70e-15 3.88e-17	6.77e-18	8.73e-14 6.92e-18

Tabel 3.4. ω values for $A = \text{compan}\ (25)$

$\theta(A,x) = 1.36\text{e}6$ $\text{cond}\,(A) = 2.95\text{e}1$, GEPP	GE	$\kappa_\infty(A) = 4.37\text{e}3$ QR
2.08e-14 1.98e-17	2.08e-14 1.98e-17	6.43e-15 1.98e-17

Table 3.5. ω values for A = orthog (25)

$\theta(A,x) = 3.02e1$ $\mathrm{cond}(A) = 2.09e1$, GEPP	GE	$\kappa_\infty(A) = 2.10e1$ QR
2.53e-16	4.61e-07	4.54e-16
4.59e-17	1.56e-13	5.31e-17
	4.34e-17	

Tabel 3.6. ω values for A = clement (50)

$\theta(A,x) = 2.40e18$ $\mathrm{cond}(A) = 1.44e6$, GEPP	GE	$\kappa_\infty(A) = 3.50e7$ QR
3.88e-15	Fail	1.43e-07
7.74e-17		1.04e-15
		6.71e-17

Table 3.7. ω values for A = gfpp (50)

$\theta(A,x) = 4.51e2$ $\mathrm{cond}(A) = 50$, GEPP	GE	$\kappa_\infty(A) = 50$ QR
8.03e-04	8.03e-04	3.22e-16
8.06e-17	8.06e-17	3.82e-17

4. Least squares by *QR* factorization.

Let $A \in \mathbb{R}^{m \times n}$ be of full rank $n \leq m$ and let $b \in \mathbb{R}^m$. If x solves the least squares (LS) problem $\min_x \|Ax - b\|_2$ then

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \tag{4.1}$$

since this augmented system of dimension $m + n$ is a representation of the normal equations $A^TAx = A^Tb$. It is well-known that an approximate LS solution $\hat{x}$ can be improved by applying iterative refinement to the augmented system [5, 6, 12], but it is usually assumed that double precision is used when calculating the residuals.

In some recent work the use of single precision iterative refinement on the augmented system has been considered. Björck [8] states without proof that iterative refinement yields benefits for a certain componentwise measure of stability (namely β in (4.6) below) when the solution method is based on a QR factorization of A. Arioli, Duff and de Rijk [2] make a thorough study of iterative refinement for the case where A is sparse and the solution method is Gaussian elimination with symmetric pivoting applied to the whole augmented system. They draw on Skeel's analysis of iterative refinement for Gaussian elimination. We also mention in passing that Björck [7] analyses fixed precision iterative refinement applied to the so-called "semi-normal equations" for the LS problem.

In this section we analyze fixed precision iterative refinement for the augmented system with Householder or Givens QR factorization as the method of solution. The outline is as follows. First we provide some theoretical support for this application of iterative refinement, by using componentwise error analysis together with Theorem 2.1 to obtain a bound for the residual of the augmented system after one step of refinement. Based on this bound we identify an appropriate definition of backward error, and show this backward error to be small, asymptotically. We observe that in small residual problems it can be difficult to achieve a small backward error, and we

suggest practical ways to overcome this difficulty. Finally, we describe some numerical experiments.

To begin, we recall the well-known result that the Householder and Givens QR factorization algorithms provide a stable way to solve the LS problem in the sense of normwise backward error (see [24]). There is no reason to expect any componentwise measure of backward error to be small, but Theorem 2.1 suggests that iterative refinement applied to the augmented system may help to achieve this goal.

The analysis of iterative refinement is more difficult for the LS problem than it is for a general square linear system. This is largely due to the following three reasons, to which we will return later in the section.

1. No explicit formula is known for any backward error of a general approximate LS solution y (see [21] for a detailed discussion).

2. A perturbation in A results in a special, symmetric perturbation of the coefficient matrix in (4.1).

3. If we are interested only in x, then r in (4.1) can be regarded as an arbitrary vector parameter that can be chosen to minimize the backward error.

In the appendix we give the result of a detailed componentwise error analysis for the solution of the system

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{4.2}$$

via a Householder or Givens QR factorization of A. We allow $g \neq 0$ in (4.2) (cf. 4.1)) so that the analysis is applicable to the refinement step as well as to the initial solution phase. Lemma A.2 implies that the computed solution $(\hat{r}^T, \hat{x}^T)^T$ to (4.2) satisfies

$$\left| \begin{bmatrix} f \\ g \end{bmatrix} - \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \hat{r} \\ \hat{x} \end{bmatrix} \right| \leq \begin{bmatrix} \mu_{m,n} G|A|\ |\hat{x}| + \mu_{m,1}(H_1|f| + H_2|\hat{r}|) \\ \mu_{m,n}|A^T|G^T|\hat{r}| + \mu_{m,1}|A^T|H_3|\hat{r}| \end{bmatrix}$$

$$\leq \mu_{m,n} \left(\begin{bmatrix} H_2 & G|A| \\ |A^T|\bar{G} & 0 \end{bmatrix} \begin{bmatrix} |\hat{r}| \\ |\hat{x}| \end{bmatrix} + \begin{bmatrix} H_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} |f| \\ |g| \end{bmatrix} \right),$$

where $\bar{G} = G^T + H_3$, $\mu_{m,n}$ is approximately the product of the unit roundoff u with a linear polynomial in m and n, and all these G and H matrices are bounded in norm by low degree polynomials in m and n. If we express this bound in the form of (2.5) then

$$\text{“}g(A,b)\text{”} = \frac{\mu_{m,n}}{u} \begin{bmatrix} H_2 & G|A| \\ |A^T|\bar{G} & 0 \end{bmatrix},$$

$$\text{“}h(A,b)\text{”} = \frac{\mu_{m,n}}{u} \begin{bmatrix} H_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} |f| \\ |g| \end{bmatrix},$$

where the A and b inside the quotes denote the matrix and right-hand side of the linear system (4.2) and not the data of the LS problem. To apply Theorem 2.2 we would need to express "$g(A,b) = \tilde{G}|A|$", that is,

$$\frac{\mu_{m,n}}{u}\begin{bmatrix} H_2 & G|A| \\ |A^T|\tilde{G} & 0 \end{bmatrix} \equiv \tilde{G}\begin{bmatrix} I & |A| \\ |A^T| & 0 \end{bmatrix}.$$

Unfortunately, this cannot be done in any useful way.

Since Theorem 2.2 is not applicable, we turn to Theorem 2.1, which tells us that the corrected solution $(\hat{s}^T, \hat{y}^T)^T$ obtained after one step of iterative refinement satisfies (with terms re-ordered from (2.8))

$$\left|\begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix}\begin{bmatrix} \hat{s} \\ \hat{y} \end{bmatrix}\right| \le \mu_{m,n}\begin{bmatrix} H_1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} |\hat{r}_1| \\ |\hat{r}_2| \end{bmatrix} + u\begin{bmatrix} I & |A| \\ |A^T| & 0 \end{bmatrix}\begin{bmatrix} |\hat{s}| \\ |\hat{y}| \end{bmatrix} + \gamma_{m+n+1}\left(\begin{bmatrix} I & |A| \\ |A^T| & 0 \end{bmatrix}\begin{bmatrix} |\hat{s}| \\ |\hat{y}| \end{bmatrix} + \begin{bmatrix} |b| \\ 0 \end{bmatrix}\right) + O(u^2). \tag{4.3}$$

Here, $(\hat{r}_1^T, \hat{r}_2^T)^T$ denotes the residual of the augmented system corresponding to the original computed solution, and we are assuming this residual is computed in the usual way, so that (2.7) holds. We will make two simplifications to the bound (4.3). First, since $(\hat{r}_1^T, \hat{r}_2^T)^T = O(u)$ the first term in the bound may be included in the $O(u^2)$ term. Second, (4.3) yields $|b - \hat{s} - A\hat{y}| = O(u)$ and so $|\hat{s}| \le |A|\ |\hat{y}| + |b| + O(u)$. With these two simplifications, together with $\gamma_{m+n+1} + u \le \gamma_{m+n+2}$, (4.3) may be written

$$\left|\begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix}\begin{bmatrix} \hat{s} \\ \hat{y} \end{bmatrix}\right| \le 2\gamma_{m+n+2}\left(\begin{bmatrix} 0 & |A| \\ |A^T| & 0 \end{bmatrix}\begin{bmatrix} |\hat{s}| \\ |\hat{y}| \end{bmatrix} + \begin{bmatrix} |b| \\ 0 \end{bmatrix}\right) + O(u^2). \tag{4.4}$$

To interpret this inequality in backward error terms we consider a perturbed augmented system

$$\begin{bmatrix} I & A + \Delta A_1 \\ (A + \Delta A_2)^T & 0 \end{bmatrix}\begin{bmatrix} \bar{r} \\ \bar{x} \end{bmatrix} = \begin{bmatrix} b + \Delta b \\ 0 \end{bmatrix}, \tag{4.5}$$

and we define

$$\beta(\bar{r}, \bar{x}) = \min\{\varepsilon : (4.5) \text{ holds with } |\Delta A_i| \le \varepsilon|A|,\ i = 1{:}2,\ |\Delta b| \le \varepsilon|b|\}. \tag{4.6}$$

By appealing to the most general form of the Oettli-Prager result (in which $|A|$ and $|b|$ in (2.1) and (2.3) are replaced by arbitrary $E \ge 0$ and $f \ge 0$) we obtain the expression

$$\begin{aligned} \beta(\bar{r}, \bar{x}) &= \max_i \frac{\left|\begin{bmatrix} b \\ 0 \end{bmatrix} - \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix}\begin{bmatrix} \bar{r} \\ \bar{x} \end{bmatrix}\right|_i}{\left(\begin{bmatrix} 0 & |A| \\ |A|^T & 0 \end{bmatrix}\begin{bmatrix} |\bar{r}| \\ |\bar{x}| \end{bmatrix} + \begin{bmatrix} |b| \\ 0 \end{bmatrix}\right)_i} \\ &= \max\left\{\max_i \frac{|b - (\bar{r} + A\bar{x})|_i}{(|A|\ |\bar{x}| + |b|)_i}, \max_i \frac{|A^T\bar{r}|_i}{(|A^T|\ |\bar{r}|)_i}\right\}. \end{aligned} \tag{4.7}$$

From (4.4) it follows that *if the* $O(u^2)$ *terms can be neglected* then $\beta(\hat{s}, \hat{y}) \leq 2\gamma_{m+n+2}$.

The backward error $\beta(\bar{r}, \bar{x})$ was introduced by Björck [8]. Strictly, it should be called a *pseudo* componentwise backward error because it allows the two occurrences of A in the augmented system to undergo different perturbations and this does not correspond in any simple way to perturbing the original data of the LS problem. (It is shown in [16] how to compute the genuine componentwise backward error that results when $\Delta A_1 = \Delta A_2$ is forced in (4.6).)

Two important properties of β are as follows (see [21], [10] or [2] for further details):

1. If the rows or columns of A are scaled then the class of perturbations ΔA_i, Δb in the definition of β scales in the same way. Hence β is invariant under row and column scalings of A.
2. A bound for the forward error $\|x - \hat{y}\|_2$ can be obtained in terms of β. This bound is potentially much smaller than the standard forward error bound for the LS problem involving $\kappa_2(A)$ (partly because of its better scaling properties). Moreover, it is identical to the bound that would be obtained if the perturbations ΔA_1 and ΔA_2 were equal – in other words, allowing $\Delta A_1 \neq \Delta A_2$ in the definition of β does not weaken the corresponding forward error bounds.

To summarize, we have shown that, asymptotically, the backward error $\beta(\hat{s}, \hat{y})$ is small after one step of iterative refinement. It is not clear how to obtain a more precise result, analogous to Theorem 2.2, say (not even if we switch to using norms in the definition of β). However, we can obtain further understanding from some simple observations and numerical tests.

An easily identifiable case where our asymptotic result is dubious is when the residual for the LS problem is zero or relatively small, i.e., when $\|r\|_2/(\|A\|_2\|x\|_2) = O(u)$. In such cases the computation of r will be subject to severe numerical cancellation and the computed $\hat{s}$ is likely to have few correct significant digits. As a result, the inequality $|A^T\hat{s}| \leq \varepsilon|A^T|\ |\hat{s}|$ cannot be guaranteed to hold with $\varepsilon = O(u)$, and so in view of (4.7) we cannot always expect $\beta(\hat{s}, \hat{y}) = O(u)$. One way to accommodate this difficulty is to adopt the technique of Arioli, Demmel and Duff, which we mentioned at the end of section 3, and thus to allow a wider class of perturbations in the right-hand side in the definition of β. This approach was used in the LS context by Arioli, Duff and de Rijk [2]. Further details are given below.

Another useful observation is that since $\bar{r}$ may be regarded as an arbitrary parameter it may be beneficial to replace it by zero if the true residual is small. If we set $\bar{r} = 0$ in (4.7) the troublesome $A^T\bar{r}$ term disappears and we are left with

$$\beta(0, \bar{x}) = \max_i \frac{|b - A\bar{x}|_i}{(|A|\ |\bar{x}| + |b|)_i}$$

(which, of course, is essentially (2.3)). After iterative refinement has converged, or has been terminated, we can check whether $\beta(0, \hat{y}) < \beta(\hat{s}, \hat{y})$, and, if so, regard $\beta(0, \hat{y})$ as

the backward error for the augmented system. It may be worthwhile to take $\beta(0, \hat{y})$ as the backward error even when $\beta(0, \hat{y}) > \beta(\hat{s}, \hat{y})$, because the corresponding forward error bound is minimized as a function of the residual $\bar{r}$ when $\bar{r} = 0$ [2, 10, 21].

We have experimented with the iterative refinement procedure discussed above using MATLAB. We report results for three problems.

(1) **Problem PR.** The matrix

$$A = \begin{bmatrix} 0 & 2 & 1 \\ 10^6 & 10^6 & 0 \\ 10^6 & 0 & 10^6 \\ 0 & 1 & 1 \end{bmatrix}$$

is from [28] and we took $b = (1, 1, 1, 1)^T$.

The following two classes of test problems were suggested by Björck [7].

(2) **Problem V.** The matrix $A \in \mathbb{R}^{21 \times 6}$ has the form $A = VD$ where $v_{ij} = (i-1)^{j-1}$ and the diagonal matrix D is chosen so that the columns of A have unit 2-norm. The solution vector is taken to be $x = D^{-1}(10^5, 10^4, \ldots, 1)$ and the right-hand sides are defined by $b = Ax + \theta h$, where $A^T h = 0$ with h normalized so that $\kappa_2(A)\|h\|_2 = 1.5\|A\|_2\|x\|_2$. The scalar θ controls the size of the residual. Problems where the coefficient matrix has widely differing row norms are obtained by choosing $w \gg 1$ and setting $D_w = \operatorname{diag}(d_i)$, where $d_i = w$ for $i = 1, 11, 21$ and $d_i = 1$ otherwise, and defining $A_w = D_w A$, $b_w = D_w b$ and $h_w = D_w^{-1} h$.

(3) **Problem H.** Here, $A \in \mathbb{R}^{6 \times 5}$ comprises the first five columns of the inverse of the Hilbert matrix of order 6. The solution vector is given by $x_i = 1/i$, and $b = Ax + \theta h$ where $A^T h = 0$ and $\kappa_2(A)\|h\|_2 = 3.72 \times 10^3 \|A\|_2 \|x\|_2$.

The details of our implementation of iterative refinement are as follows. We used Householder QR factorization and solved the augmented system (4.2) as prescribed in the appendix (see the explanation following (A.1)). For the convergence test we took $\beta(\hat{s}, \hat{y}) \leq u$, where, based on (4.7), $\beta(\bar{r}, \bar{x}) = \max\{\beta_1, \beta_2\}$ with

$$\beta_1 = \max_i \frac{|b - (\bar{r} + A\bar{x})|_i}{(|A|\ |\bar{x}| + |b|)_i},$$

$$\beta_2 = \max_i \frac{|A^T \bar{r}|_i}{(|A^T|\ |\bar{r}|)_i + \mu_i},$$

and where, with

$$\mathscr{L} \equiv \{i : (|A^T||\bar{r}|)_i \leq \tau_i \equiv 1000(m+n)u\|A(:,i)\|_\infty \|(\bar{r}^T, \bar{x}^T)\|_\infty\},$$

$$\mu_i = \begin{cases} \|A(:,i)\|_1 \|(\bar{r}^T, \bar{x}^T)\|_\infty, & \text{if } i \in \mathscr{L}, \\ 0, & \text{otherwise.} \end{cases}$$

The values of μ_i and τ_i are those recommended in [1] and used in [2] (see our comments at the end of section 3). Upon convergence we can assert that the

computed $\hat{s}$ and $\hat{y}$ satisfy

$$\begin{bmatrix} I & A + \Delta A_1 \\ (A + \Delta A_2)^T & 0 \end{bmatrix} \begin{bmatrix} \hat{s} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} b + \Delta b \\ \Delta c \end{bmatrix}, \tag{4.8}$$

where $|\Delta A_i| \leq u|A|$, $i = 1, 2$, $|\Delta b| \leq u|b|$ and $|\Delta c_i| \leq u\mu_i$ if $i \in \mathscr{L}$ or $\Delta c_i = 0$ otherwise. (We could of course extend the use of the tolerances τ_i and μ_i to the first m of the augmented equations, as in [2], but this was not necessary in our tests with dense matrices.)

Selected results are reported in Tables 4.1–4.5. The notation is as follows. The coefficient matrix of the augmented system is denoted by C; $\operatorname{cond}(A) = \| |A^+| \, |A| \|_\infty$ is a generalization of the condition number (2.13) and appears in forward error bounds of [2, 21] (A^+ is the pseudo-inverse of A); and $\rho(x)$ is a computed approximation to the relative residual $\|b - Ax\|_2/(\|A\|_2\|x\|_2)$ of the LS solution. As well as showing β_1, β_2 and $\mathscr{L}$ for each iteration, we tabulate $\beta(0, \hat{y})$, where $\hat{y}$ is the final computed solution. Here are our comments on the results.

1. Iterative refinement converged in these and most other examples, although not always after one iteration. An example where it failed to converge is Problem V with $w = 10^{14}$ and $\theta = 0$ (after ten iterations $\beta(\hat{s}, \hat{y}) > 10^{-2}$); however, if in this problem we interchange the large rows of A to the top (cf. point 4 below) then $\beta(\hat{r}, \hat{x}) = 1.04 \times 10^{-15}$, and after one step of iterative refinement $\beta(\hat{s}, \hat{y}) = 5.98 \times 10^{-17}$.

2. As we would expect, the modified definition of β_2 was invoked for most of the small residual problems, as indicated by the set $\mathscr{L}$ being nonempty.

3. In several instances where $\rho(x) \simeq u$, $\beta(0, \hat{y}) \simeq \beta(\hat{s}, \hat{y})$. Also, $\beta(0, \hat{y}) < \beta(\hat{s}, \hat{y})$ in Table 4.2 for $\theta = 0$. Since $\beta(0, \hat{y})$ is trivial to compute it seems advisable to evaluate it whenever $\rho(x) = O(u)$ and to consider using it in forward error bounds.

4. The coefficient matrix A can have widely differing row norms in weighted LS problems and in LS problems produced by the method of weighting for linearly constrained LS problems [15, secs. 5.6.2, 12.1.5], [34]. It is well-known that in such circumstances the solution computed by Householder QR factorization can be unstable in the sense of having a large componentwise backward error [28], [24, Ch. 17]. This is confirmed in Tables 4.1, 4.3 and 4.4, where the β_1 values are large for the first iteration. The usual advice is to avoid this instability by incorporating "partial pivoting style" row interchanges, as suggested in [28]. Our results suggest that an alternative that may be worth considering is to use iterative refinement to stabilize the solution (or indeed one could use both techniques together). In sparse problems a possible advantage of iterative refinement is that it allows the row ordering to be chosen to minimize intermediate

fill-in rather than to preserve stability (the sparsity of the R factor depends on the column ordering but not the row ordering). We note, however, that to implement iterative refinement it is necessary to store and re-use the orthogonal factor Q, which may be undesirable. The method of semi-normal equations [7] avoids the use of Q but has different stability properties. In the tables below P_6 stands for $\{1,2,3,4,5,6\}$.

Table 4.1. Problem PR.

cond $(A) = 3.46e0$, $\kappa_2(A) = 8.32e5$ cond $(C) = 9.23e5$, $\kappa_2(C) = 1.73e6$				
$\rho(x)$	$\beta(0,\hat{y})$	β_1	β_2	I
2.40e-7	1.30e-1	3.55e-11	2.06e-10	$\emptyset$
		5.74e-17	3.70e-18	$\emptyset$

Table 4.2. Problem V. $w = 1$.

cond $(A) = 2.56e3$, $\kappa_2(A) = 2.22e3$ cond $(C) = 3.36e6$, $\kappa_2(C) = 2.66e6$					
θ	$\rho(x)$	$\beta(0,\hat{y})$	β_1	β_2	$\mathscr{L}$
0	1.53e-16	7.39e-17	4.97e-15	1.16e-32	P_6
			8.00e-17	7.88e-33	P_6
10^{-9}	6.76e-13	2.41e-11	5.18e-15	5.23e-29	P_6
			7.87e-17	4.03e-30	P_6
1	6.76e-4	2.36e-2	4.70e-15	6.01e-16	$\emptyset$
			8.47e-17	1.21e-17	$\emptyset$
10^3	6.76e-1	1.0e0	6.10e-16	5.84e-16	$\emptyset$
			7.62e-17	1.74e-17	$\emptyset$

Table 4.3. Problem V. $w = 10^5$.

cond $(A) = 2.69e3$, $\kappa_2(A) = 9.95e7$ cond $(C) = 5.03e10$, $\kappa_2(C) = 7.99e10$					
θ	$\rho(x)$	$\beta(0,\hat{y})$	β_1	β_2	$\mathscr{L}$
0	2.03e-16	2.62e-16	9.24e-11	4.22e-28	P_6
			1.10e-16	4.22e-28	P_6
10^{-9}	2.12e-16	2.41e-11	9.24e-11	4.60e-28	P_6
			9.29e-17	4.22e-28	P_6
1	1.17e-8	2.36e-2	9.14e-11	1.60e-11	$\emptyset$
			1.11e-16	2.12e-17	$\emptyset$
10^3	1.17e-5	1.00e-0	7.81e-12	1.60e-11	$\emptyset$
			8.14e-17	9.94e-18	$\emptyset$

Table 4.4. Problem V. $w = 10^{10}$.

cond $(A) = 2.69e3$, $\kappa_2(A) = 9.95e12$ cond $(C) = 5.03e15$, $\kappa_2(C) = 7.99e15$					
θ	$\rho(x)$	$\beta(0,\hat{y})$	β_1	β_2	$\mathscr{L}$
0	3.09e-17	2.62e-16	1.74e-5	9.86e-23	P_6
			9.76e-12	9.86e-23	P_6
			7.20e-15	3.10e-28	P_6
			7.44e-17	1.80e-32	P_6
10^{-9}	3.43e-17	2.41e-11	1.74e-5	9.86e-23	P_6
			9.76e-12	9.86e-23	P_6
			7.25e-15	3.10e-28	P_6
			1.04e-16	1.99e-32	P_6
1	1.17e-13	2.36e-2	1.70e-5	1.49e-19	P_6
			2.84e-11	9.86e-23	P_6
			7.75e-15	2.60e-28	P_6
			9.84e-17	4.36e-30	P_6
10^3	1.17e10	1.00e0	7.45e-7	2.87e-6	$\emptyset$
			1.60e-9	1.55e-11	$\emptyset$
			7.21e-14	1.02e-15	$\emptyset$
			7.65e-17	1.06e-17	$\emptyset$

Table 4.5. Problem H.

cond $(A) = 4.16e6$, $\kappa_2(A) = 4.70e6$ cond $(C) = 5.77e6$, $\kappa_2(C) = 8.89e6$					
θ	$\rho(x)$	$\beta(0,\hat{y})$	β_1	β_2	$\mathscr{L}$
0	2.02e-17	1.78e-15	3.13e-14	2.26e-16	$\emptyset$
			4.22e-17	3.22e-15	$\emptyset$
			6.14e-17	2.23e-16	$\emptyset$
			2.60e-17	3.36e-16	$\emptyset$
			1.88e-17	5.66e-17	$\emptyset$
10^{-9}	7.92e-13	8.66e-10	3.12e-14	2.35e-16	$\emptyset$
			4.01e-17	2.45e-17	$\emptyset$
1	7.92e-4	4.64e-1	1.66e-14	2.03e-16	$\emptyset$
			1.25e-17	1.46e-17	$\emptyset$
10^3	7.92e-1	9.99e-1	2.28e-16	2.56e-16	$\emptyset$
			0.00e0	4.05e-17	$\emptyset$

Our overall conclusion is that iterative refinement provides an effective way to enhance the stability of Householder and Givens QR factorization algorithms for solving the LS problem. Our analysis and experiments put the refinement process on a sound footing, although we have not derived conditions under which it is guaranteed to succeed. LAPACK includes iterative refinement in its routines for Gaussian elimination [3,4]. We suggest it is worth considering supporting iterative refinement for LS problems as well.

Acknowledgements

Des Higham carefully read the manuscript and suggested numerous improvements. I also thank the referee and editor for their helpful comments.

Appendix.

A Error analysis results.

In this appendix we state componentwise error analysis results for the solution of linear systems and least squares problems via the QR factorization. The proofs of these results are given in [20].

Recall that our model for floating point arithmetic is (2.4). The constants used in the results are defined as follows: $\gamma_n = nu/(1 - nu)$; $\mu_{m,n} = \gamma_{am+bn+c}$ for some small integer constants a, b, c; and

$$\theta_{m,n} = \frac{2\mu_{m,n}\sqrt{m(n-1)}}{1 - 2\mu_{m,n}\sqrt{m(n-1)}}.$$

The results are stated for Householder QR factorization, but they remain valid (with slightly different constant terms) when the QR factorization is computed using Givens rotations.

LEMMA A.1 *Let $A \in \mathbb{R}^{n\times n}$ be nonsingular. Suppose we solve the system $Ax = b$ with the aid of a QR factorization computed by the Householder algorithm. The computed $\hat{x}$ satisfies*

$$(A + F)\hat{x} = b + e,$$

where

$$|F| \le \mu_{n,n} G|A|, \quad |e| \le \mu_{n,1} H|b|,$$

with

$$\|G\|_2 \le 3n^2(1 + \theta_{n,n}), \quad \|H\|_2 \le 2n(1 + \theta_{n,1}).$$

The next result concerns the solution of the augmented system for a least squares problem, which we take in the form

$$\begin{aligned} r + Ax &= f, \\ A^T r &= g. \end{aligned} \tag{A.1}$$

If A has the QR factorization

$$A = Q\begin{bmatrix} R_1 \\ 0 \end{bmatrix},$$

where $R_1 \in \mathbb{R}^{n\times n}$, then (A.1) transforms to

$$Q^T r + \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x = Q^T f,$$

$$[R_1^T \; 0]\, Q^T r = g.$$

This system can be solved as follows:

$$h = R_1^{-T} g,$$

$$d = Q^T f = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix},$$

$$r = Q\begin{bmatrix} h \\ d_2 \end{bmatrix},$$

$$x = R_1^{-1}(d_1 - h).$$

LEMMA A.2 *Let $A \in \mathbb{R}^{m\times n}$ be of full rank $n \le m$ and suppose the augmented system (A.1) is solved using a QR factorization of A as described above. The computed $\hat{x}$ and $\hat{r}$ satisfy*

$$\begin{bmatrix} I & A + E_1 \\ (A + E_2)^T & 0 \end{bmatrix} \begin{bmatrix} \hat{r} \\ \hat{x} \end{bmatrix} = \begin{bmatrix} f + e_1 \\ g + e_2 \end{bmatrix},$$

where

$$|E_i| \le \mu_{m,n}\, G|A|, \quad i = 1:2,$$

$$|e_1| \le \mu_{m,1}(H_1|f| + H_2|\hat{r}|),$$

$$|e_2| \le \mu_{m,1}|A^T|\, H_3|\hat{r}|,$$

with

$$\|G\|_2 \le 3mn(1 + \theta_{m,n}),$$

$$\|H_1\|_2 \le 3m^{3/2}(1 + \theta_{m,1}),$$

$$\|H_2\|_2 \le 5m^{3/2}(1 + \theta_{m,1}),$$

$$\|H_3\|_2 \le 7m^{3/2}\, \mu_{m,1}(1 + \theta_{m,1}).$$

REFERENCES

[1] M. Arioli, J. W. Demmel and I. S. Duff, *Solving sparse linear systems with sparse backward error*, SIAM J. Matrix Anal. Appl., 10(1989), pp. 165–190.

[2] M. Arioli, I. S. Duff and P. P. M. de Rijk, *On the augmented system approach to sparse least-squares problems*, Numer. Math., 55 (1989), pp. 667–684.

[3] C. H. Bischof, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling and D. C. Sorensen, *Provisional contents*, LAPACK Working Note #5, Report ANL-88-38, Mathematics and Computer Science Division, Argonne National Laboratory, Illinois, 1988.

[4] C. H. Bischof and J. J. Dongarra, *A project for developing a linear algebra library for high-performance computers*, Preprint MCS-P105-0989, Mathematics and Computer Science Division, Argonne National Laboratory, 1989.

[5] Å. Björck, *Iterative refinement of linear least squares solutions* I, BIT, 7 (1967), pp. 257–278.

[6] Å. Björck, *Comment on the iterative refinement of least-squares solutions*, J. Amer. Stat. Assoc., 73 (1978), pp. 161–166.

[7] Å. Björck, *Stability analysis of the method of seminormal equations for linear least squares problems*, Linear Algebra and Appl., 88/89 (1987), pp. 31–48.

[8] Å. Björck, *Componentwise backward errors and condition estimates for linear least squares problems*, Manuscript, Department of Mathematics, Linköping University, Sweden, March 1988.

[9] Å. Björck, *Iterative refinement and reliable computing*, in *Reliable Numerical Computation*, M. G. Cox and S. J. Hammarling, eds., Oxford University Press, 1990, pp. 249–266.

[10] Å. Björck, *Component-wise perturbation analysis and error bounds for linear least squares solutions*; BIT 31:2 (1991), pp. 238–244.

[11] Å. Björck and T. Elfving, *Algorithms for confluent Vandermonde systems*, Numer. Math., 21 (1973), pp. 130–137.

[12] Å. Björck and G. H. Golub, *Iterative refinement of linear least squares solutions by Householder transformation*, BIT, 7 (1967), pp. 322–337.

[13] Å. Björck and V. Pereyra, *Solution of Vandermonde systems of equations*, Math. Comp., 24 (1970), pp. 893–903.

[14] J. W. Demmel and N. J. Higham, *Improved error bounds for underdetermined system solvers*, Numerical Analysis Report No. 189, University of Manchester, England (and LAPACK Working Note #23), 1990.

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

[16] D. J. Higham and N. J. Higham, *Backward error and condition of structured linear systems*, Numerical Analysis Report No. 192, University of Manchester, England, 1990; to appear in SIAM J. Matrix Anal. Appl.

[17] N. J. Higham, *Fast solution of Vandermonde-like systems involving orthogonal polynomials*, IMA Journal of Numerical Analysis, 8 (1988), pp. 473–486.

[18] N. J. Higham, *Stability analysis of algorithms for solving confluent Vandermonde-like systems*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 23–41.

[19] N. J. Higham, *A collection of test matrices in MATLAB*, Technical Report 89-1025, Department of Computer Science, Cornell University, 1989; to appear in ACM Trans. Math. Soft.

[20] N. J. Higham, *Iterative refinement enhances the stability of QR factorization methods for solving linear equations*, Numerical Analysis Report No. 182, University of Manchester, 1990.

[21] N. J. Higham, *Computing error bounds for regression problem*, in *Statistical Analysis of Measurement Error Models and Applications*, P. J. Brown and W. A. Fuller, eds., Contemporary Mathematics 112, Amer. Math. Soc., 1990, pp. 195–208.

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

[23] M. Jankowski and H. Woźniakowski, *Iterative refinement implies numerical stability*, BIT, 17 (1977), pp. 303–311.

[24] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

[25] C. B. Moler, *Iterative refinement in floating point*, J. Assoc. Comput. Mach., 14 (1967), pp. 316–321.

[26] 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.
[27] J. Oliver, *An error analysis of the modified Clenshaw method for evaluating Chebyshev and Fourier series*, J. Inst. Maths. Applics., 20 (1977), pp. 379–391.
[28] M. J. D. Powell and J. K. Reid, *On applying Householder transformations to linear least squares problems*, Proc. IFIP Congress 1968, North-Holland, 1969, pp. 122–126.
[29] R. D. Skeel, *Scaling for numerical stability in Gaussian elimination*, J. Assoc. Comput. Mach., 26 (1979), pp. 494–526.
[30] R. D. Skeel, *Iterative refinement implies numerical stability for Gaussian elimination*, Math. Comp., 35 (1980), pp. 817–832.
[31] F. J. Smith, *An algorithm for summing orthogonal polynomial series and their derivatives with applications to curve-fitting and interpolation*, Math. Comp., 19 (1965), pp. 33–36.
[32] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.
[33] L. N. Trefethen and R. S. Schreiber, *Average-case stability of Gaussian elimination*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 335–360.
[34] C. F. Van Loan, *On the method of weighting for equality-constrained least-squares problems*, SIAM J. Numer. Anal., 22 (1985), pp. 851–864.
[35] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Notes on Applied Science No. 32, Her Majesty's Stationery Office, London, 1963.
[36] J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, 1965.