RAL-TR-2001-005

Incremental norm estimation for dense and sparse matrices[1]

Iain S. Duff[2] and Christof Vömel[3]

ABSTRACT

We discuss the benefits of an incremental approach to norm estimation for triangular matrices. Our investigation covers both dense and sparse matrices. If we use our incremental norm estimation on explicitly generated entries of the inverse of the triangular matrix, we can relate our results to incremental condition estimation (ICE). We show that our algorithm extends more naturally to the sparse case than ICE. Our scheme can be used to develop a robust pivot selection criterion for QR factorization or as the basis of a rank-revealing factorization.

Keywords: matrix norm, condition number, incremental estimators, approximate singular vectors, sparse triangular matrices, QR factorization, rank-revealing.

AMS(MOS) subject classifications: 65F05, 65F35, 65F50.

---

[1]Current reports available by anonymous ftp to ftp.numerical.rl.ac.uk in directory pub/reports. This report is in file dvRAL01005.ps.gz. Report also available through URL http://www.numerical.rl.ac.uk/reports/reports.html. Also published as Technical Report TR/PA/00/83 from CERFACS, 42 Ave G. Coriolis, 31057 Toulouse Cedex, France.

[2] i.s.duff@rl.ac.uk, The work of this author was supported in part by the EPSRC Grant GR/M78502.

[3] voemel@cerfacs.fr, CERFACS, 42 Ave G. Coriolis, 31057 Toulouse Cedex, France.

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxon OX11 0QX

December 20, 2000

Contents

1 Introduction

There are many cases when it is interesting and important to detect the ill-conditioning of a square matrix A from the triangular factors arising in its LU or QR factorization. Applications include the calculation of forward error bounds based on the condition number of A and robust pivot selection criteria.

Another particularly interesting field of applications is provided by rank-revealing factorizations. During the process of determining a rank-revealing permutation, several (and, in the extreme case, an exponential number of) leading or trailing submatrices have to be investigated for their conditioning, see for example the survey by Chandrasekaran and Ipsen (1994). A condition estimator is used to determine the conditioning of these matrices. Conceptually, there are two major classes of these estimators. The first class are *static* in the sense that they estimate the condition number of a fixed triangular matrix. These methods are surveyed in Higham (1987). The second class can be used for *dynamic* estimation when a triangular matrix is calculated one column or row at a time. These incremental schemes (often called incremental condition estimation or ICE) were originally presented in Bischof (1990) and are particularly attractive for monitoring a factorization as it proceeds. This was exploited in Pierce and Lewis (1997) where a generalization of the original scheme to sparse matrices (Bischof, Pierce and Lewis 1990) was incorporated in a multifrontal QR algorithm to generate a good initial permutation for rank-revealing 'on-the-fly'.

A completely different rank-revealing strategy is proposed in Meyer and Pierce (1995). Instead of using condition estimation together with the triangular factors from a LU factorization (as for example in Chan (1984)), a method based on an *implicit* LU factorization is employed. This so-called Direct Projection Method (Benzi and Meyer 1995) calculates an upper triangular matrix Z such that $AZ = L$ is a lower triangular matrix, with $Z = U^{-1}$ where U is the triangular factor of Crout's LU factorization. To our knowledge, this the first time information on the *inverse* of a triangular factor was used to detect ill-conditioning. All the previous approaches only used the triangular factors themselves so that the condition estimators had to estimate the reciprocal of the smallest singular value. On the contrary, working with the matrix inverse implies the estimation of the largest singular value, that is the Euclidean matrix norm. This motivated us to think about the design of an efficient norm estimator which can be applied in that framework.

When we were reformulating the ICE scheme from Bischof (1990) to the task of norm estimation, we discovered that this scheme has a major shortcoming. Namely that the scheme allows the use of approximate vectors for only one side; that is, approximate right singular vectors for lower triangular matrices and approximate left singular vectors for upper triangular matrices. While this might at first glance not seem very critical, it has a severe implication on the use of ICE on *sparse* matrices. As we will see, ICE fails on sparse matrices because it uses approximate singular vectors from the *wrong* side. Therefore, sophisticated modifications have to be introduced to adapt the scheme to the sparse case (Bischof et al. 1990).

The purpose of this presentation is to show how to address this problem with a genuine matrix norm estimator. By generalizing the underlying ideas of ICE to the norm case, we develop an incremental scheme that can be based on approximate singular vectors from both sides. This scheme is as reliable as ICE for dense matrix but is *directly* applicable to sparse matrices. It is particularly interesting in calculations that involve matrix inverses, for example the Direct Projection Method (Benzi and Meyer 1995) and the rank-revealing approach of Meyer and Pierce (1995). In addition, the incremental estimation of the matrix norm, that is the *largest* singular value, can be used to complement ICE which gives an estimate of the smallest singular value. In this way we can obtain an incremental estimation of the *true* condition number.

In Section 2, we first briefly discuss the original condition estimation scheme of Bischof (1990) and then describe how we calculate the matrix norm of a triangular matrix in incremental fashion. As we will see, the norm estimation can be based both on approximate left *and* right singular vectors, in contrast to ICE. This allows a *direct* application of our schemes to sparse matrices and the modifications necessary for ICE, as developed by Bischof et al. (1990), can be avoided.

Of course, our norm estimation is of particular interest when the inverse of the triangular factor is available. In Section 3, we develop, as an example, a QR factorization with inverted triangular factor. This algorithm will later be used for testing our norm estimator.

The inversion of sparse matrices is additionally associated with the problem of fill-in. However, in the case of triangular matrices fill-in can be avoided by storing the inverse in factored form as proposed by Alvarado and Schreiber (1993). We describe the details of this approach in Section 4.1 and illustrate problems that can occur when we try to detect ill-conditioning from the factored form.

We show the reliability of our incremental norm estimator in Section 5, by presenting results obtained from a variety of dense and sparse test cases from standard matrix collections (Duff, Grimes and Lewis 1989, Higham 1995).

Finally, we give our conclusions in Section 6.

2 Incremental estimators

In this section, we present the details of our incremental norm estimator. The principal conceptual difference between our scheme and the original incremental condition estimator (ICE) (Bischof 1990) is that ours uses matrix-vector multiplications whereas ICE is based on the solution of triangular systems. A more detailed comparison between the schemes is given in Section 2.5.

2.1 The original incremental condition estimator (ICE)

In order to appreciate the general difficulties of determining the conditioning by examining the triangular factors, we first present two classical test matrices from Kahan (1966):

EXAMPLE 1 *Consider* $T_n \in \mathbf{R}^{n\times n}$ *where*

$$T_n = \begin{bmatrix} 1 & -\gamma & \dots & -\gamma \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -\gamma \\ 0 & \dots & 0 & 1 \end{bmatrix}$$

with $\gamma > 0$. *The components of the inverse* $T_n^{-1} = (\alpha_{ij})$ *satisfy the recursion* $(\alpha_{i-1j}) = (1+\gamma)(\alpha_{ij}), i = j-2, \dots, 1$, *hence it is given by*

$$\alpha_{ij} = \begin{cases} 1, & i = j \\ \gamma(1+\gamma)^{j-i-1}, & i < j. \end{cases}$$

EXAMPLE 2 *Consider* $K_n(c) \in \mathbf{R}^{n\times n}$ *with*

$$K_n(c) = \mathrm{diag}(1, s, s^2, \dots, s^{n-1}) \begin{bmatrix} 1 & -c & \dots & -c \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -c \\ 0 & \dots & 0 & 1 \end{bmatrix}$$

where $c, s \in (0,1)$ *with* $c^2 + s^2 = 1$. *Its inverse is given by* $K_n^{-1}(c) = (\alpha_{ij})$ *with*

$$\alpha_{ij} = \begin{cases} s^{1-i}, & i = j \\ s^{1-i}c(1+c)^{j-i-1}, & i < j. \end{cases}$$

Both matrices are very ill-conditioned which is easy to see from the entries of the inverses but is not evident from the entries in the matrices themselves. A Householder QR factorization with column pivoting (Golub and van Loan 1996) will reveal the ill-conditioning of Example 1 but will not work on Example 2. We will use both these matrices in our experiments in Section 5.

In Bischof (1990), an elegant approach to condition estimation is presented which updates an estimate of the smallest singular value of a triangular matrix when it is augmented by adding another column. We now describe this approach applied to upper triangular matrices.

Given an upper triangular matrix T, we can calculate its smallest singular value by finding a vector d of unit length so that the solution x of $x^H T = d^H$ has maximum norm. That is, we find

$$d = \arg \max_{\|d\|_2 = 1} \|d^H T^{-1}\|_2.$$

Once we have solved this problem (at least approximately), it is shown in Bischof (1990) how to compute a cheap estimate of the smallest singular value for the augmented matrix

$$\hat{T} = \begin{bmatrix} T & v \\ & \gamma \end{bmatrix}.$$

The right-hand side $\hat{d}$ for the augmented system $\hat{x}^H \hat{T} = \hat{d}^H$ can be chosen as

$$\hat{d} = \hat{d}(s, c) = \begin{pmatrix} sd \\ c \end{pmatrix}, \tag{2.1}$$

where $s^2 + c^2 = 1$, and the solution to this augmented system has the form

$$\hat{x} = \begin{pmatrix} sx \\ \frac{c - s\alpha}{\gamma} \end{pmatrix} \tag{2.2}$$

with $\alpha = x^H v$. In other words, $\hat{x}^H \hat{T} = \hat{d}$ can be solved for $\hat{x}$ *without any back-substitution involving* T.

The parameters (s, c) are chosen to maximize the norm of $\hat{x}$. This maximization problem can be treated analytically, and we refer the reader to the very elegant demonstration in Bischof (1990).

The low cost of this approach together with the quality of the estimates obtained have made it an attractive safeguard for the computation of the QR factorization, as was already suggested in Bischof (1990) and later on was successfully employed in the sparse multifrontal rank revealing QR factorization (Pierce and Lewis 1997).

2.2 Incremental norm estimation by approximate left singular vectors

Analogously to Section 2.1, we seek a cheap incremental *norm* estimator when augmenting an upper triangular matrix. We can design an efficient scheme by proceeding in a very similar way to the ICE construction.

Computing the matrix norm using a left singular vector means we wish to find a vector y of unit length such that

$$y = \arg \max_{\|y\|_2 = 1} \|y^H T\|_2.$$

An incremental norm estimator has then to specify a cheap heuristic for the computation of $\hat{y}$ corresponding to the augmented matrix

$$\hat{T} = \begin{bmatrix} T & v \\ & \gamma \end{bmatrix}. \tag{2.3}$$

We will see that we can avoid a matrix-vector product in this computation if we restrict the search to vectors $\hat{y}$ of the form

$$\hat{y} = \hat{y}(s,c) = \begin{pmatrix} sy \\ c \end{pmatrix}, \tag{2.4}$$

where $s^2 + c^2 = 1$.

Since

$$\begin{aligned}
\|\hat{y}^H \hat{T}\|_2^2 &= \hat{y}^H \hat{T}\hat{T}^H \hat{y} \\
&= \left(sy^H, c\right) \begin{bmatrix} T & v \\ & \gamma \end{bmatrix} \begin{bmatrix} T^H & \\ v^H & \gamma \end{bmatrix} \begin{pmatrix} sy \\ c \end{pmatrix} \\
&= (s,c) \begin{bmatrix} y^H T T^H y + (y^H v)^2 & \gamma(y^H v) \\ \gamma(y^H v) & \gamma^2 \end{bmatrix} \begin{pmatrix} s \\ c \end{pmatrix} \\
&= (s,c)\, B \begin{pmatrix} s \\ c \end{pmatrix},
\end{aligned}$$

we can rewrite the objective function as a quadratic form, where $B \in \mathbf{R}^{2\times 2}$.

Theorem 3 *The matrix B is s.p.d. if $\hat{T}$ is nonsingular. Hence the maximization problem*

$$\max_{\|(s,c)\|_2 = 1} \|\hat{y}(c,s)^H \hat{T}\|_2^2 \tag{2.5}$$

has as solution the eigenvector (s^*, c^*) of unit length belonging to the largest eigenvalue of B.

The calculation of $\|\hat{y}^H \hat{T}\|_2$ by a matrix-vector product at every step can be avoided by using the updating formula

$$\|\hat{y}^H \hat{T}\|_2 = \sqrt{s^2 \|y^H T\|_2^2 + \left(s(y^H v) + c\gamma\right)^2} \tag{2.6}$$

which is a consequence of

$$\hat{y}^H \hat{T} = \begin{pmatrix} s y^H T \\ s y^H v + c\gamma \end{pmatrix}. \tag{2.7}$$

If we introduce the quantities

$$\alpha = y^H v, \quad \delta = \|y^H T\|_2$$

and

$$\eta^2 = \alpha^2 + \delta^2, \quad \mu = \delta\gamma, \quad \nu = \alpha\gamma,$$

we find as the solution of (2.5):

for the case $\alpha \neq 0$

$$\begin{pmatrix} s \\ c \end{pmatrix} = \frac{u}{\|u\|_2}, \; u = \begin{pmatrix} \eta^2 - \gamma^2 + \sqrt{\eta^4 + 2\nu^2 - 2\mu^2 + \gamma^4} \\ 2\nu \end{pmatrix}$$

for the case $\alpha = 0$

$$\begin{pmatrix} s \\ c \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 \\ 0 \end{pmatrix}, & \text{if } \delta > |\gamma|, \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \text{otherwise.} \end{cases}$$

Using (2.6), we can completely omit the calculation of $\hat{y}^H \hat{T}$ and compute $\hat{\delta}$ directly from δ.

2.3 Incremental norm estimation by approximate right singular vectors

In the previous section, we showed how to construct incrementally an approximation of the left singular vector corresponding to the largest singular value. We will now develop the scheme for the corresponding right singular vector. This might seem very natural, however we emphasize that it is not possible to extend the original ICE scheme described in Section 2.1 to right singular vectors. In Section 2.5, we look more closely at this problem of extending ICE.

For an upper triangular matrix T, the issue is now to find a vector z of unit length so that

$$z = \arg \max_{\|z\|_2 = 1} \|Tz\|_2.$$

With the augmented matrix $\hat{T}$ defined as in (2.3), our approximate right singular vector is assumed to be of the form

$$\hat{z} = \hat{z}(s, c) = \begin{pmatrix} sz \\ c \end{pmatrix}, \tag{2.8}$$

where $s^2 + c^2 = 1$, exactly as in (2.4).

We state again the objective function as a quadratic form viz.

$$\begin{aligned} \|\hat{T}\hat{z}\|_2^2 &= (s, c) \begin{bmatrix} z^H T^H T z & z^H T^H v \\ v^H T z & v^H v + \gamma^2 \end{bmatrix} \begin{pmatrix} s \\ c \end{pmatrix} \\ &= (s, c)\, C \begin{pmatrix} s \\ c \end{pmatrix}, \end{aligned}$$

and see, by the same arguments as in Section 2.2, that the solution (s^*, c^*) can be calculated analytically.

By exploiting the recurrence

$$\hat{T}\hat{z} = \begin{pmatrix} sTz + cv \\ c\gamma \end{pmatrix}, \tag{2.9}$$

we see that, as in Section 2.2, we can avoid a matrix-vector product at each stage.

If we define

$$\beta = v^H Tz, \quad \epsilon = \|Tz\|_2 \quad \kappa^2 = v^H v + \gamma^2,$$

we have

for the case $\beta \neq 0$

$$\begin{pmatrix} s \\ c \end{pmatrix} = \frac{u}{\|u\|_2}, \; u = \begin{pmatrix} \epsilon^2 - \kappa^2 + \sqrt{\epsilon^4 + 4\beta^2 - 2\epsilon^2\kappa^2 + \kappa^4} \\ 2\beta \end{pmatrix}$$

and for the case $\beta = 0$

$$\begin{pmatrix} s \\ c \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 \\ 0 \end{pmatrix}, & \text{if } \epsilon > |\gamma|, \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \text{otherwise.} \end{cases}$$

We point out that our algorithm works only with the vector Tz, the approximate right singular vector z is neither needed nor computed. However, it might become useful to know z. In this case, we suggest multiplying Tz by T^H and normalizing the result. This strategy of performing one step of power iteration with an appropriately chosen vector z was originally used by the LINPACK condition estimator (Cline, Moler, Stewart and Wilkinson 1979), but we choose z differently.

2.4 Incremental norm estimation for sparse matrices

The incremental condition estimator described in Section 2.1 is intended to be used with dense matrices. In its original form, ICE cannot be applied to sparse matrices as we illustrate through Example 4.

EXAMPLE 4 *Consider the triangular matrix* $A \in \mathbf{R}^{(n_B+n_C+n_D)\times(n_B+n_C+n_D)}$ *with*

$$A = \begin{bmatrix} B & 0 & E_B \\ 0 & C & E_C \\ 0 & 0 & D \end{bmatrix}.$$

After step n_B*, we obtain from ICE an approximate left singular vector* x_B*. In the next step, the first column of the second block column is appended, where* $\gamma = a(n_B+1, n_B+1)$ *is the only nonzero entry. ICE will now take either* $\hat{x} = (x_B, 0)^H$

or $\hat{x} = (0, 1)^H$ as an approximate left singular vector for the augmented triangular matrix. But once a component in the approximate left singular vector is set to zero, this choice cannot be undone later in the calculation, independent of the entries in the right border. Thus, the sparsity of the matrix A can cause the quality of the estimate to be very poor.

The modifications proposed to ICE in Bischof et al. (1990) to overcome this problem are as follows: for each block on the diagonal generate a separate approximate left singular vector, and then merge these subvectors together where the weights of each subvector are computed using a block version of ICE. This requires again the computation of the eigenvector belonging to the largest eigenvalue of a s.p.d. matrix, but this matrix will be of order k where k is the number of diagonal blocks rather than of order 2. As this eigenvector (for $k > 4$) can no longer be computed analytically, the solution of a secular equation using rational approximations is used.

The reason for the failure of ICE on sparse matrices is that, while the upper triangular matrix is augmented *column by column*, the incremental condition estimator uses *left* approximate singular vectors and thus calculates a weighted linear combination of the *rows*. This problem will not occur if it was possible to base ICE on approximate right singular vectors.

What does this imply for the incremental norm estimation? As in the case of ICE, we expect to encounter similar problems to the incremental norm estimation for sparse matrices if we use approximate singular vectors from the *wrong* side. Fortunately, we can use approximate right singular vectors in the case of columnwise augmentation of the matrix as we have shown in Section 2.3. This allows us to use the same simple and elegant scheme for dense matrices in the sparse case also.

2.5 The relationship between incremental norm and condition estimation

We now present a more detailed investigation of the relationship between the incremental norm and incremental condition estimators described in the previous sections. In particular, we show why incremental condition estimation is less flexible with respect to the use of approximate singular vectors from both sides.
For the following discussion, we use the nomenclature

$$\hat{T} = \begin{bmatrix} T & v \\ & \gamma \end{bmatrix}, \quad \hat{T}^{-1} = \begin{bmatrix} T^{-1} & u \\ & \gamma^{-1} \end{bmatrix}.$$

Let us first look at the incremental condition estimator ICE. The scheme constructs from a vector d of unit norm the next vector $\hat{d}$ incrementally as

$$\hat{d} = \hat{d}(s, c) = \begin{pmatrix} sd \\ c \end{pmatrix}.$$

The elegance of the scheme lies in the fact that it is *not* necessary to solve the equation $\hat{x}^H \hat{T} = \hat{d}^H$ if the solution of the previous equation $x^H T = d$ is known. Instead, $\hat{x}$ can be computed directly from x through the update formula given in equation (2.2).

The problem is that the analogous update formula for ICE based on approximate right singular vectors is not practical. An investigation of the equation $\hat{T}\hat{z} = \hat{d}$ reveals the update formula

$$\hat{z} = \begin{pmatrix} sz + cu \\ c/\gamma \end{pmatrix}. \tag{2.10}$$

Note that this formula involves the vector u which is part of the *inverse* matrix $\hat{T}$. As was shown in Sections 2.2 and 2.3, the incremental norm estimator does not have the same problem. Estimators based on both left as well as right singular vectors have update formulae that involve only terms of the original matrix T (see equations (2.7) and (2.9)).

The incremental approach to norm estimation is a direct generalization of the concepts used in ICE. Indeed it is the case that the application of the incremental norm estimator, using approximate left singular vectors, to the matrix T^{-1} is mathematically *equivalent* to applying the incremental condition estimator ICE to the matrix T. This follows by substituting the matrix T by its inverse T^{-1} in the derivation of Section 2.2.

3 Triangular factorizations with inverse factors

In this section, we describe briefly the incorporation of the inversion of a triangular factor into a QR factorization. This algorithm will be the basis for our numerical tests which are reported in Section 5. We remark that the explicit computation of matrix inverses arises for example in signal processing applications (Cybenko 1987, Pan and Plemmons 1989).

3.1 The QR factorization with simultaneous inversion of R

There are several ways to combine a standard QR factorization with a simultaneous inversion of R. It is important to consider both the performance and the stability of the inversion algorithm. Both aspects were investigated in Du Croz and Higham (1992). Of all the methods discussed in that paper, we decided to implement a method rich in Level 2 BLAS matrix-vector multiplies. Lemma 5 describes the basis of our inversion algorithm.

LEMMA 5 *Assume that $R \in \mathbf{R}^{i \times i}$ and that the first $i-1$ columns of $Y = R^{-1}$ have already been computed. Then, the i^{th} column of Y can be computed from*

$$\begin{aligned} Y(i,i) * R(i,i) &= 1, \\ Y(1:i-1,i) * R(i,i) &= -Y(1:i-1,1:i-1) * R(1:i-1,i). \end{aligned}$$

This is a consequence of a columnwise evaluation of $YR = I$.

If we combine a QR factorization based on the modified Gram-Schmidt algorithm (Golub and van Loan 1996) with the simultaneous inversion described by Lemma 5, we get Algorithm 1.

Algorithm 1 *QR factorization with simultaneous inversion.*

```
[n, n] = size(A);
Q = zeros(m, n);
R = zeros(n);
Y = zeros(n);
for i = 1:n do
  R(i, i) = norm(A(:, i), 2);
  Q(:, i) = A(:, i)/R(i, i);
  for j = i+1:n do
    R(i, j) = (Q(:, i))' * A(:, j);
    A(:, j) = A(:, j) - R(i, j) * Q(:, i);
  end for
  Y(i, i) = 1/R(i, i);
  if i > 1 then
    Y(1 : i - 1, i) = Y(1 : i - 1, 1 : i - 1) * R(1 : i - 1, i);
    Y(1 : i - 1, i) = -Y(i, i) * Y(1 : i - 1, i);
  end if
end for
```

3.2 Stability issues of triangular matrix inversion

The numerical stability properties of general triangular matrix inversion were investigated in Du Croz and Higham (1992). The inversion in Algorithm 1 is just Method 2 from Du Croz and Higham (1992) adapted for upper triangular matrices. An error analysis similar to the one performed there establishes the following componentwise residual bound for the computed inverse $\bar{Y}$:

$$|\bar{Y}R - I| \leq c_n u |\bar{Y}||R| + \mathcal{O}(u^2)$$

where c_n denotes a constant of order n and u the unit roundoff. The interpretation of this result is that the residual $\bar{Y}R - I$ can be bounded componentwise by a small multiple of the unit roundoff times the size of the entries in $\bar{Y}$ and R.

This bound illustrates the reliability of our method for matrix inversion, but the remark on page 18 of Du Croz and Higham (1992) should be recalled:

... we wish to stress that all the analysis here pertains to matrix analysis alone. It is usually the case that when a computed inverse is used as part of a larger

computation the stability properties are less favourable, and this is one reason why matrix inversion is generally discouraged.

Indeed, the authors give an example illustrating that the solution of $Rx = b$ by the evaluation $R^{-1}b$ need *not* be backward stable if R^{-1} has first to be computed from R.

4 Inverses in sparse factored form

4.1 Sparse storage of triangular inverses

The inverse of a sparse matrix A is generally less sparse than A itself and indeed, if the matrix is irreducible, its inverse is structurally dense, see for example Duff, Erisman and Reid (1986). As was observed by the pioneers in linear programming some few decades ago (Bartels and Golub 1969, Forrest and Tomlin 1972), the inverse of a sparse triangular matrix can be stored with exactly the same storage as the matrix itself, that is without fill-in. This is described in Lemma 6.

LEMMA 6 *Let $R \in \mathbf{R}^{n\times n}$ be an upper triangular matrix. Denote by R_i an elementary matrix, equal to the identity matrix except for row i where it is identical to the i-th row of R. Then:*

1. $R = R_n R_{n-1} \ldots R_1$
2. *Let $S_i = R_i^{-1}$. Then S_i has exactly the same sparsity structure as R_i and is, apart from row i equal to the identity matrix. Note that S_i does not contain the i-th row of Y.*

Both results can be checked by calculation.

The lemma suggests that we can obtain a no fill-in representation of the inverse by storing the factors $S_i, i = 1, \ldots, n$ instead of R^{-1}.

Although this is very good from the point of view of sparsity it unfortunately causes problems for the detection of ill-conditioning. For example, the factored representation of T_n^{-1}, where T_n is the matrix of Example 1, is given by the tableau

$$\begin{bmatrix} 1 & \gamma & \ldots & \gamma \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \gamma \\ 0 & \ldots & 0 & 1 \end{bmatrix}.$$

Here, row i of Y holds the non-trivial row of the elementary matrix S_i. We see that the exponential growth of the matrix entries does not show up in the factored form, that is the ill-conditioning is *hidden* by this implicit representation of the inverse. From this example, we conclude that we need to calculate the inverse explicitly to avoid hiding the ill-conditioning. For most matrices, it is not possible to do this without fill-in, however, in Alvarado and Schreiber (1993), it is shown how the number of factors in the sparse factored form of the inverse can be reduced while still avoiding fill-in so long as the matrix satisfies a certain condition. The original intention of Alvarado and Schreiber (1993) was to enhance parallelism in the solution of triangular systems, but we use the idea here to help detect ill-conditioning.

In order to explain the method, we introduce the following nomenclature: For an upper triangular matrix $R \in \mathbf{R}^{n\times n}$, its directed acyclic graph $G(R)$ is the pair (V, E) where $V = \{1, \ldots, n\}$ and $E = \{(i,j) | i \neq j \text{ and } R(i,j) \neq 0\}$. For $(i,j) \in E$, i is called a predecessor of j and j a successor of i. The transitive closure of a directed graph $G = (V, E)$ is the graph $G' = (V, E')$ where $E' = \{(i,j) | \exists \text{ path } i \rightarrow j \text{ in } G\}$.

THEOREM 7 *(Gilbert 1994) Let R be a nonsingular upper triangular matrix. Then*

$$G(R^{-1}) = (G(R))'.$$

This theorem allows us to extend Lemma 6 by showing that the restriction to using elementary matrices as factors is not necessary. Instead, we can consider *blocks* of rows of R where the corresponding *generalized* elementary matrix has a transitively closed directed graph. By generalized elementary matrix, we mean the matrix which is equal to the identity except for the rows belonging to the block where it is identical to the corresponding rows of R.

EXAMPLE 8 *Consider the matrix*

$$R = \begin{bmatrix} 11 & 12 & 13 & 14 \\ & 22 & 23 & 0 \\ & & 33 & 34 \\ & & & 44 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 33 & 34 \\ & & & 44 \end{bmatrix} \begin{bmatrix} 11 & 12 & 13 & 14 \\ & 22 & 23 & 0 \\ & & 1 & \\ & & & 1 \end{bmatrix}.$$

From Theorem 7, we see that each of the factors can be inverted without fill-in.

It is desirable to look for a representation of the inverse with the smallest number of factors possible. The inclusion of this row blocking strategy into Algorithm 1 will then result in a hybrid algorithm that uses the sparse representation of the inverse but also reveals possible hidden ill-conditioning of dense submatrices. In particular, this algorithm can handle the pathological matrices in Examples 1 and 2.

To formalize the objectives, the algorithm should find a partition $0 = e_1 < e_2 < \ldots < e_{m+1} = n$ so that

$$R^{-1} = S_1 \ldots S_m \tag{4.1}$$

where S_k is the inverse of the generalized elementary matrix corresponding to the rows $e_k + 1, \ldots, e_{k+1}$ of R and the number of factors m is as small as possible.

The following Lemma is now an immediate consequence of Theorem 7.

LEMMA 9 *Assume that the generalized elementary matrix corresponding to the rows* $e_k+1,\ldots,j-1$ *of* R *is invertible without fill-in. Then the augmented generalized elementary matrix corresponding to the rows* $e_k+1,\ldots,j$ *of* R *is invertible without fill-in if and only if the Condition 10 is satisfied.*

CONDITION 10 *Every successor* s *of* j *is also a successor of all predecessors* $p \geq e_k+1$ *of* j.

The following theorem shows the optimality of the partition.

THEOREM 11 *(Alvarado and Schreiber 1993) A partitioning with maximal row blocking based on Condition 10 leads to a sparse representation of* R^{-1} *with the smallest possible number of factors.*

It is interesting to see how easily the row blocking can be incorporated into our inversion algorithm for triangular matrices. The following analogue to Lemma 5 shows how Algorithm 1 has to be modified.

LEMMA 12 *Assume that Condition 10 holds for* $e_k+1,\ldots,i$ *and that the columns* $e_k+1,\ldots,i-1$ *of* S_k *have already been computed. Then, the* i^{th} *column of* S_k *can be computed from*

$$\begin{aligned} Y(i,i)*R(i,i) &= 1, \\ Y(e_k+1:i-1,i)*R(i,i) &= -Y(e_k+1:i-1,e_k+1:i-1)*R(e_k+1:i-1,i). \end{aligned}$$

We remark that the stability of the partitioned inverse method in the context of solving triangular linear systems has been studied in Higham and Pothen (1994). Generally, the comments given at the end of Section 3.2 also apply here.

4.2 Incremental norm estimation for sparse factored inverses

The application of any incremental scheme to a factored representation is a difficult problem. As can be seen from (2.6) and (2.9), it is always assumed that we have access to the full column being appended. However, in the factored approach a column might not be stored explicitly because of fill-in, see Section 4.1. The column could be generated but the high cost of its computation from the factors might spoil the effectiveness of the scheme.

Although we cannot give a full solution to this problem, we suggest at least a partial remedy as follows:

1. It is possible to use
$$\|R^{-1}\|_2 \approx \prod_{i=1}^{m} \|S_i\|_2. \tag{4.2}$$

to obtain an estimated upper bound for the condition number of $Y = R^{-1}$. In our tests, we found that the product on the right-hand side is often a severe overestimation of $\|R^{-1}\|_2$, even if each factor $\|S_i\|_2$ is an underestimate. Although there are circumstances where an overestimate is useful (for example, if the value is not too large then we are fairly sure the matrix is not ill-conditioned), the use of (4.2) can be very unreliable.

2. The cost for the computation of an off-diagonal block depends on the number of factors in the sparse representation, the graph $G(R^{-1})$, and the position of the block. The example

$$\begin{aligned} Y &= \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ & I & \\ & & I \end{bmatrix} \begin{bmatrix} I & & \\ & S_{22} & S_{23} \\ & & I \end{bmatrix} \begin{bmatrix} I & & \\ & I & \\ & & S_{33} \end{bmatrix} \\ &= \begin{bmatrix} S_{11} & S_{12}S_{22} & (S_{12}S_{23} + S_{13})S_{33} \\ & S_{22} & S_{23}S_{33} \\ & & S_{33} \end{bmatrix} \end{aligned}$$

illustrates this. If R^{-1} is very sparse, the computation of the columns of Y from its factors becomes affordable. We can use a blocked version of our scheme to first calculate approximate singular vectors corresponding to the diagonal blocks and afterwards merge them together to obtain an approximate singular vector for the whole system.

5 Numerical tests

In this section, we present tests of our incremental schemes with dense and sparse matrices. We use the algorithm described in Section 3.1 which allows us to use our norm estimator on both the triangular factor and its inverse.

In Table 5.1, we show the incremental estimates for the QR factorization of sparse matrices from the Harwell-Boeing collection (Duff et al. 1989). Here, the second column displays the exact matrix norm of R as calculated by MATLAB, the third and fourth columns show estimations based on approximate left singular vectors and on approximate right singular vectors, respectively.

In general, both of our estimators give a good approximation to the norm of R. Note that, because we compute the norm using an approximate singular vector, our estimate will always be a lower bound for the norm. However, as indicated in Section 2.4, the incremental approach for upper triangular matrices based on approximate left singular vectors can lead to problems for sparse matrices and we see this in a few of our test cases, most noticeably the case *arc130*, where the incremental approach based on right singular vectors gives a much better estimate than using left singular vectors.

Name	norm(R)	est. (left)	est. (right)
arc130	2.3973e+05	1.9160e+02	2.3712e+05
bfw398a	1.0412e+01	9.4564e+00	8.6924e+00
cavity04	7.1227e+01	3.5360e+01	6.3994e+01
e05r0400	4.5921e+01	1.7958e+01	4.1371e+01
fidap001	1.3019e-01	1.1498e-01	1.1960e-01
fs_183_1	1.1293e+09	8.2283e+08	1.1293e+09
impcol_b	8.6395e+00	3.1833e+00	8.4843e+00
impcol_c	1.2000e+02	9.7782e+00	1.2000e+02
lns__131	9.7721e+09	9.5468e+09	9.1036e+09
nnc261	1.0406e+03	5.7603e+02	1.0175e+03
saylr1	4.8325e+08	4.1494e+08	4.8180e+08
steam1	2.1712e+07	1.8812e+07	2.1712e+07
str____0	1.3938e+01	7.3314e+00	1.3902e+01
west0381	1.7153e+03	1.3968e+03	1.7153e+03

Table 5.1: Results with matrices from the Harwell-Boeing collection.

In Table 5.2, we show the incremental estimates for the norm of R^{-1} from the QR factorization of dense matrices from the Matlab Test Matrix Toolbox (Higham 1995). Specifically, we use the matrices from Example 1, called *condex* in Higham (1995), and Example 2, named *kahan*. Furthermore, we include tests with Hilbert matrices (*hilb*). We apply our incremental norm estimator to the inverse of the matrices, since the ill-conditioning is evident there rather than in matrices themselves. We note that both of our estimates are always very close to the real norm.

Name	Size	norm(inv(R))	est. (left)	est. (right)
condex(n,3)	50	3.7530e+14	3.7220e+14	3.7530e+14
	75	1.2593e+22	1.2489e+22	1.2593e+22
	100	4.2255e+29	4.1906e+29	4.2255e+29
kahan(n)	50	6.4262e+07	6.0921e+07	6.4262e+07
	75	8.4992e+11	8.0573e+11	8.4992e+11
	100	1.1241e+16	1.0657e+16	1.1241e+16
hilb(n)	50	8.9463e+17	3.7405e+17	8.3684e+17
	75	1.7687e+18	5.2885e+17	1.7139e+18
	100	1.8250e+18	7.2637e+17	1.6266e+18

Table 5.2: Results with matrices from the Matlab Test Matrix Toolbox.

In Tables 5.3 and 5.4, we show the incremental estimates for the QR factorization of random matrices with uniform and exponentially distributed singular values, respectively. For each of the different matrix sizes n, we created 50 random matrices $A = U\Sigma V^H$ choosing different random matrices U,V, and singular values either uniformly distributed as

$$\sigma_i = norm(A)/i, \quad 1 \leq i \leq n,$$

or exponentially distributed as

$$\sigma_i = \alpha^i, \quad 1 \leq i \leq n, \quad \alpha^n = norm(A),$$

where the norm of A was chosen in advance. The random orthogonal matrices U and V were generated using a method of Stewart (1980) which is available under the name *qmult* in Higham (1995). The values displayed in the table are the averages from 50 tests each.

Size	norm(R)	est. (left)	est. (right)
50	1.0000e+01	8.8211e+00	9.2427e+00
	1.0000e+06	8.7931e+05	9.2079e+05
	1.0000e+12	8.7692e+11	9.1995e+11
75	1.0000e+01	8.8183e+00	9.2273e+00
	1.0000e+06	8.8188e+05	9.2421e+05
	1.0000e+12	8.8044e+11	9.2064e+11
100	1.0000e+01	8.8147e+00	9.2158e+00
	1.0000e+06	8.7806e+05	9.1797e+05
	1.0000e+12	8.7292e+11	9.1696e+11

Table 5.3: Results (averages) with random matrices, σ_i uniformly distributed

Size	norm(R)	est. (left)	est. (right)
50	1.0000e+01	8.5490e+00	9.0990e+00
	1.0000e+06	8.5224e+05	9.5870e+05
	1.0000e+12	8.8925e+11	9.9151e+11
75	1.0000e+01	8.5177e+00	9.1122e+00
	1.0000e+06	8.1961e+05	9.5153e+05
	1.0000e+12	8.3155e+11	9.8192e+11
100	1.0000e+01	8.5445e+00	9.1128e+00
	1.0000e+06	8.0845e+05	9.3058e+05
	1.0000e+12	8.1969e+11	9.5817e+11

Table 5.4: Results (averages) with random matrices, σ_i exponentially distributed.

These tests show that also in the case of *dense* upper triangular matrices, the norm estimation based on approximate right singular vectors is more reliable than that based on approximate left singular vectors, which corresponds to the approach used in ICE.

6 Conclusions

We have shown how an incremental norm estimator can be developed for a triangular factor. We have pointed out the suitability of the scheme for both dense and sparse matrices due to the fact that it can use approximate singular vectors both from the left and the right side.

In the context of the inversion of triangular matrices, we have related our approach to the incremental condition estimator, ICE. We also discussed the applicability of our scheme when a sparse matrix inverse is stored in factored form.

To demonstrate the efficacy of our approach, we have shown results on some standard test examples including both sparse and dense matrices.

Acknowledgments

Our understanding of this problem and the presentation have been greatly aided by conversations with Per Christian Hansen and Miroslav Tuma. We would like to thank Mario Arioli and Nick Gould for their comments on an earlier version of this paper.

References

Alvarado, F. L. and Schreiber, R. (1993), 'Optimal parallel solution of sparse triangular systems', *SIAM J. Scientific Computing* **14**, 446–460.

Bartels, R. H. and Golub, G. H. (1969), 'The simplex method of linear programming using LU decompositions', *Communications of the Association for Computing Machinery* **12**, 266–268.

Benzi, M. and Meyer, C. D. (1995), 'A direct projection method for sparse linear systems', *SIAM J. Scientific Computing* **16**(5), 1159–1176.

Bischof, C. H. (1990), 'Incremental condition estimation', *SIAM J. Matrix Analysis and Applications* **11**, 312–322.

Bischof, C. H., Pierce, D. J. and Lewis, J. G. (1990), 'Incremental condition estimation for sparse matrices', *SIAM J. Matrix Analysis and Applications* **11**, 644–659.

Chan, T. F. (1984), 'On the existence and computation of LU-factorizations with small pivots', *Math. Comp.* **42**, 535–547.

Chandrasekaran, S. and Ipsen, I. (1994), 'On rank-revealing QR factorizations', *SIAM J. Matrix Analysis and Applications* **15(2)**, 592–622.

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. Numerical Analysis* **16**, 368–375.

Cybenko, G. (1987), 'Fast Toeplitz orthogonalization using inner products', *SIAM J. Scientific and Statistical Computing* **8**, 734–740.

Du Croz, J. J. and Higham, N. J. (1992), 'Stability of methods for matrix inversion', *IMA J. Numerical Analysis* **12**, 1–19.

Duff, I. S., Erisman, A. M. and Reid, J. K. (1986), *Direct Methods for Sparse Matrices*, Oxford University Press, Oxford, England.

Duff, I. S., Grimes, R. G. and Lewis, J. G. (1989), 'Sparse matrix test problems', *ACM Trans. Math. Softw.* **15**(1), 1–14.

Forrest, J. J. H. and Tomlin, J. A. (1972), 'Updating triangular factors of the basis to maintain sparsity in the product form simplex method', *Mathematical Programming* **2(3)**, 263–278.

Gilbert, J. R. (1994), 'Predicting structure in sparse matrix computations', *SIAM J. Matrix Analysis and Applications* **15**, 62–79.

Golub, G. H. and van Loan, C. (1996), *Matrix Computations*, 3. edn, The John Hopkins University Press, Baltimore, Maryland.

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

Higham, N. J. (1995), The Test Matrix Toolbox for MATLAB (version 3.0), Numerical Analysis Report No. 276, Manchester Centre for Computational Mathematics, Manchester, England.

Higham, N. J. and Pothen, A. (1994), 'Stability of the partitioned inverse method for parallel solution of sparse triangular systems', *SIAM J. Scientific Computing* **15**(1), 139–148.

Kahan, W. (1966), 'Numerical linear algebra', *Canadian Mathematical Bulletin* **9**, 757–801.

Meyer, C. D. and Pierce, D. (1995), Steps towards an iterative rank-revealing method, Technical Report ISSTECH-95-013, Boeing Information and Support Services.

Pan, C.-T. and Plemmons, R. J. (1989), 'Least squares modifications with inverse factorizations: parallel implications', *J. Comput. Appl. Math.* **27**, 109–127.

Pierce, D. J. and Lewis, J. G. (1997), 'Sparse multifrontal rank revealing QR factorization', *SIAM J. Matrix Analysis and Applications* **18**(1), 159–180.

Stewart, G. W. (1980), 'The efficient generation of random orthogonal matrices with an application to condition estimators', *SIAM J. Numerical Analysis* **17**, 403–404.