On the Stability of Null-Space Methods for KKT Systems [1]

Roger Fletcher and Tom Johnson

Department of Mathematics and Computer Science, University of Dundee, Dundee DD1 4HN, Scotland, UK.

Numerical Analysis Report NA/167, December 1995

Abstract

This paper considers the numerical stability of null-space methods for KKT systems, particularly in the context of quadratic programming. The methods we consider are based on the direct elimination of variables which is attractive for solving large sparse systems. Ill-conditioning in a certain sub-matrix A in the system is shown to adversely affect the method insofar as it is commonly implemented. In particular it can cause growth in the residual error of the solution, which would not normally occur if Gaussian elimination or related methods were used. The mechanism of this error growth is studied and is not due to growth in the null-space basis matrix Z, as might have been expected, but to the indeterminacy of this matrix. When LU factors of A are available it is shown that an alternative form of the method is available which avoids this residual error growth. These conclusions are supported by error analysis and Matlab experiments on some extremely ill-conditioned test problems. These indicate that the alternative method is very robust in regard to residual error growth, and is unlikely to be significantly inferior to the methods based on an orthogonal basis matrix. The paper concludes with some discussion of what needs to be done when LU factors are not available.

1 Introduction

A Karush-Kuhn-Tucker (KKT) system is a linear system

$$\begin{bmatrix} G & A \\ A^T & 0 \end{bmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{c} \\ \mathbf{b} \end{pmatrix} \tag{1.1}$$

[1] An early version of this paper was presented at the Dundee Biennial Conference in Numerical Analysis, June, 1995 and the Manchester IMA Conference on Linear Algebra, July 1995.

involving a symmetric matrix of the form

$$K = \begin{bmatrix} G & A \\ A^T & 0 \end{bmatrix}. \tag{1.2}$$

Such systems are characteristic of the optimization problem

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\mathbf{x}^T G\mathbf{x} - \mathbf{c}^T\mathbf{x} \\ &\text{subject to} && A^T\mathbf{x} = \mathbf{b} \end{aligned} \tag{1.3}$$

in which there are linear equality constraints, and the objective is a quadratic function. The KKT system (1.1) represents the first order necessary conditions for a locally minimum solution of this problem, and $\mathbf{y}$ is a vector of Lagrange multipliers (see [3] for example). Problems like (1.3) arise in many fields of study, such as in Newton's method for nonlinear programming, and in the solution of partial differential equations involving incompressible fluid flows, incompressible solids, and the analysis of plates and shells. Also problems with inequality constraints are often solved by solving a sequence of equality constrained problems, most particularly in the active set method for quadratic programming.

In (1.2) and (1.3), G is the symmetric $n \times n$ Hessian matrix of the objective function, A is the $n \times m$ Jacobian matrix of the linear constraints, and $m \leq n$. We assume that A has full rank, for otherwise K would be singular. In some applications, A does not immediately have full rank, but can readily be reduced to a full rank matrix by a suitable transformation.

There are various ways of solving KKT systems, most of which can be regarded as symmetry-preserving variants of Gaussian elimination with pivoting (see for example Forsgren and Murray [4]). This approach is suitable for a one-off solution of a large sparse KKT system, by incorporating a suitable data structure which permits fill-in in the resulting factors. Our interest in KKT systems arises in a Quadratic Programming (QP) context, where we are using the so-called *null-space method* to solve the sequence of equality constrained problems that arise. This method is described in Section 2. An important feature of QP is that the successive matrices K differ only in that one column is either added to or removed from A. The null-space method allows this feature to be used advantageously to update factors of the reduced Hessian matrix that arises when solving the KKT system. However in this paper we do not consider the updaing issue, but concentrate on the solution of a single problem like (1.3), but in a null-space context. In fact the null-space method is related to one of the above mentioned variants of Gaussian Elimination, and this point is discussed towards the end of Section 3.

In this paper we study the numerical stability of the null-space method when the matrix K is ill-conditioned. This arises either when the matrix A is close to being rank deficient or when the reduced Hessian matrix is ill-conditioned. It is well known however that Gaussian elimination with pivoting usually enables ill-conditioned systems to be solved with small backward error (that is the computed solution is the exact solution of

a nearby problem). As Wilkinson [6] points out, the size of the backward error depends only on the growth in certain reduced matrices, and the amount of growth is usually negligible for an ill-conditioned matrix. Although it is possible for exponential growth to occur (we give an example for a KKT system), this is most unlikely in practice. A consequence of this is that if the computed solution is substituted into the system of equations, a very accurate residual is obtained. Thus variants of Gaussian elimination with pivoting usually provide a very stable method for solving ill-conditioned systems.

However this argument does not carry over to the null-space method and we indicate at the end of Section 2 that there are serious concerns about numerical stability when A is nearly rank deficient. We describe some Matlab experiments in Section 6 which support these concerns. In particular the residual of the KKT system is seen to be proportional to the condition number of A. We present some error analysis in Section 4 which shows how this arises.

When LU factors of A are available, we show in Section 3 that there is an alternative way of implementing a null-space method, which avoids the numerical instability. This is also supported by Matlab experiments. The reasons for this are described in Section 5, and we present some error analysis which illustrates the difference in the two approaches. In practice, when solving large sparse QP problems, LU factors are not usually available and it is more usual to use some sort of product form method. We conclude with some remarks about what can be done in this situation to avoid numerical instability.

2 Null-Space Methods

A null-space method (see e.g. [3]) is an important technique for solving quadratic programming problems with equality constraints. In this section we show how the method can be derived as a generalised form of constraint elimination. The key issue in this procedure is the formation of a basis for the null space of A. We determine the basis in such a way that we are able to solve large sparse problems efficiently. When A is ill-conditioned we argue that there is serious concern for the numerical stability of the method.

The null space of A may be defined by

$$\mathcal{N}(A) = \left\{ \mathbf{z} \,|\, A^T\mathbf{z} = \mathbf{0} \right\},$$

and has dimension $n - m$ when A has full rank. Any matrix

$$Z = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{n-m}].$$

whose columns are a basis for $\mathcal{N}(A)$ will be referred to as a *null-space matrix* for A. Such a matrix satisfies $A^T Z = 0$ and has linearly independent columns. A general specification for computing a null-space matrix is to choose an $n \times (n - m)$ matrix V such that the matrix

$$\mathbf{A} = \begin{bmatrix} A & V \end{bmatrix}$$

is non-singular. Its inverse is then partitioned in the following way

$$\mathbf{A}^{-1} = [\, A \quad V \,]^{-1} = \begin{bmatrix} Y^T \\ Z^T \end{bmatrix} \begin{matrix} m \\ n-m \end{matrix} . \tag{2.1}$$

It follows from the properties of the inverse that $A^T Z = 0$ and $A^T Y = I_m$. By construction, the columns of Z are linearly independent, and it follows that these columns form a basis for $\mathcal{N}(A)$.

The value of this construction is that it enables us to parametrize the solution set of the (usually) underdetermined system $A^T \mathbf{x} = \mathbf{b}$ in (1.3) by

$$\mathbf{x} = Y\mathbf{b} + Z\mathbf{v}, \qquad \mathbf{v} \in \mathbb{R}^{n-m}. \tag{2.2}$$

Here $Y\mathbf{b}$ is one particular solution of $A^T\mathbf{x} = \mathbf{b}$ and any other solution $\mathbf{x}$ differs from $Y\mathbf{b}$ by a vector, $Z\mathbf{v}$ say, in $\mathcal{N}(\mathcal{A})$. Thus (2.2) provides a general way of eliminating the constraints, by expressing the problem in terms of the *reduced variables* $\mathbf{v}$. Hence if (2.2) is substituted into the objective function of (1.3), we obtain the *reduced problem*

$$\text{minimize} \qquad \tfrac{1}{2}\mathbf{v}^T(Z^T G Z)\mathbf{v} + \mathbf{v}^T Z^T(GY\mathbf{b} - \mathbf{c}). \tag{2.3}$$

We refer to $Z^T G Z$ as the *reduced Hessian matrix* and $Z^T(GY\mathbf{b}-\mathbf{c})$ as the *reduced gradient vector* (at the point $\mathbf{x} = Y\mathbf{b}$). A sufficient condition for (2.3) to have a unique minimizer is that $Z^T G Z$ is positive definite. In this case there exist Choleski factors $Z^T G Z = LL^T$, and (2.3) can be solved by finding a stationary point, that is by solving the linear system

$$LL^T\mathbf{v} = Z^T(\mathbf{c} - GY\mathbf{b}). \tag{2.4}$$

Then substitution of $\mathbf{v}$ into (2.2) determines the solution $\mathbf{x}$ of (1.3). The vector $G\mathbf{x}-\mathbf{c}$ is the gradient of the objective function at the solution, so a vector $\mathbf{y}$ of Lagrange multipliers satisfying $G\mathbf{x} - \mathbf{c} + A\mathbf{y} = \mathbf{0}$ can then be obtained from

$$\mathbf{y} = Y^T(\mathbf{c} - G\mathbf{x}) \tag{2.5}$$

by virtue of the property that $Y^T A = I$. The vectors $\mathbf{x}$ and $\mathbf{y}$ also provide the solution to (1.1) as can readily be verified.

In practice, when A is a large sparse matrix, the matrices Y and Z are usually substantially dense and it is impracticable to store them explicitly. Instead, products with Y and Z or their transposes are obtained by solving linear systems involving $\mathbf{A}$. For example the vector $\mathbf{x} = Y\mathbf{b} + Z\mathbf{v}$ in (2.2) could be computed by solving the linear system

$$\mathbf{A}^T\mathbf{x} = \begin{pmatrix} \mathbf{b} \\ \mathbf{v} \end{pmatrix} \tag{2.6}$$

by virtue of (2.1). Likewise solving the system

$$\mathbf{A}\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \mathbf{t} \tag{2.7}$$

provides the products $\mathbf{u}_1 = Y^T\mathbf{t}$ and $\mathbf{u}_2 = Z^T\mathbf{t}$. These computations require an invertible representation of the matrix $\mathbf{A}$ to be available.

Solving systems involving $\mathbf{A}$ is usually a major cost with the null-space method. To keep this cost as low as possible, it is preferable to choose the matrix V to be sparse. Other choices (for example based on the QR factors of A, see [3]) usually involve significantly more fill-in and computational expense. In particular it is attractive to choose the columns of V to be unit vectors, using some form of pivoting to keep $\mathbf{A}$ as well conditioned as possible. In this case, assuming that the row permutation has been incorporated into A, it is possible to write

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \qquad V = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{2.8}$$

where A_1 is an $m \times m$ nonsingular submatrix. Then (2.1) becomes

$$\begin{bmatrix} Y^T \\ Z^T \end{bmatrix} = \begin{bmatrix} A_1 & \\ A_2 & I \end{bmatrix}^{-1} = \begin{bmatrix} A_1^{-1} & \\ -A_2A_1^{-1} & I \end{bmatrix}$$

and provides an explicit expression for Y and Z. In particular we see that

$$Z^T = [\, -A_2A_1^{-1} \quad I \,]. \tag{2.9}$$

We refer to this choice of V as *direct elimination* as it corresponds to directly using the first m variables to eliminate the constraints (see [3]). We shall adopt this choice of V throughout the rest of the paper.

The reduced Hessian matrix Z^TGZ is also needed for use in (2.3), and can be calculated in a similar way. The method is to compute the vectors $Z^TGZ\mathbf{e}_k$ for $k = 1, 2, \ldots, n-m$, where $\mathbf{e}_k$ denotes column k of the unit matrix I_{n-m}. The computation is carried out from right to left by first computing the vector $\mathbf{z}_k = Z\mathbf{e}_k$ by solving the system

$$\mathbf{A}^T\mathbf{z}_k = \begin{pmatrix} \mathbf{0} \\ \mathbf{e}_k \end{pmatrix}. \tag{2.10}$$

Then the product $G\mathbf{z}_k$ is computed, followed by the solution of

$$\mathbf{A}\mathbf{u} = G\mathbf{z}_k. \tag{2.11}$$

The partition $\mathbf{u}_2$ is then column k of Z^TGZ as required. The lower triangle of Z^TGZ is then used to calculate the Choleski factor L. A similar approach is essentially used in an active set method for QP, in which the Choleski factor of Z^TGZ is built up over a sequence of iterations. (If indefinite QP problems are solved, it may be required to solve KKT systems in which Z^TGZ is indefinite. We note that such systems can also be solved in a numerically stable way which preserves symmetry, see Higham [5] in regard to the method of Bunch and Kaufmann [1]).

An advantage of the null-space approach is that we only need to have available a subroutine for the matrix product $G\mathbf{v}$. Thus we can take full advantage of sparsity or structure in G, without for example having to allow for fill-in as Gaussian elimination would require. The approach is most convenient when Z^TGZ is sufficiently small to allow it to be stored as a dense matrix. In fact there is a close relationship between the null-space method and a variant of Gaussian elimination, as we shall see in the next section, and the matrix Z^TGZ is the same submatrix in both methods. Thus it would be equally easy (or difficult) to represent Z^TGZ in a sparse matrix format with either method.

To summarize the content of this section we can enumerate the steps implied by (2.2) through (2.5)

1. Calculate Z^TGZ as in (2.10) and (2.11).

2. Calculate $\mathbf{s} = Y\mathbf{b}$ by a solve with $\mathbf{A}^T$ as in (2.6) with $\mathbf{v} = \mathbf{0}$.

3. Calculate $\mathbf{t} = \mathbf{c} - G\mathbf{s}$ requiring a product with G.

4. Calculate $\mathbf{u}_2 = Z^T\mathbf{t}$ by a solve with $\mathbf{A}$ as in (2.7).

5. Solve $Z^TGZ\mathbf{v} = \mathbf{u}_2$ to determine $\mathbf{v}$ as in (2.4).

6. Calculate $\mathbf{x} = Y\mathbf{b} + Z\mathbf{v}$ by a solve with $\mathbf{A}^T$ as in (2.6).

7. Calculate $\mathbf{g} = \mathbf{c} - G\mathbf{x}$ requiring a product with G.

8. Calculate $\mathbf{y} = Y^T\mathbf{g}$ by a solve with $\mathbf{A}$, which also provides $\mathbf{z} = Z^T\mathbf{g}$.

When direct elimination based on (2.9) is used, we shall refer to this as *Method 1.* Step 1 requires $2(n-m)$ solves with either $\mathbf{A}$ or $\mathbf{A}^T$ and $n-m$ products with G to set up the reduced Hessian matrix. The remaining steps require 4 solves and 2 products, plus a solve with Z^TGZ. In some circumstances these counts can be reduced. If $\mathbf{b} = \mathbf{0}$ then steps 2 and 3 are not required. If the multiplier part $\mathbf{y}$ of the solution is not of interest then steps 7 and 8 are not needed.

We now turn to the concerns about the numerical stability of the null-space method when A (and hence A_1 and $\mathbf{A}$) is ill-conditioned. In this case A is close to a rank deficient matrix, A' say, which has a null space of higher dimension. When we solve systems like (2.10) and (2.11), the matrix Z that we are implicitly using is badly determined. Therefore, because of round-off error, we effectively get a significantly different Z matrix each time we carry out a solve. Thus the computed reduced Hessian matrix Z^TGZ does not correspond to any one particular Z matrix. As we shall see in the rest of the paper, this can lead to solutions with significant residual error.

3 Using LU factors of A

In this section we consider the possibility that we can readily compute LU factors of A given by

$$\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} U. \tag{3.1}$$

where L_1 is unit lower triangular and U is upper triangular. We can assume that a row permutation has been made which enables us to bound the elements of L_1 and L_2 by $|l_{ij}| \leq 1$. As we shall see, these factors permit us to circumvent the difficulties caused by ill-conditioning to a large extent. (Unfortunately, LU factors are not always available, and some indication is given in Section 7 as to what might be done in this situation.) We also describe how the steps in the null-space method are changed. Finally we explore some connections with Gaussian elimination and other methods, which provide some insight into the likelihood of growth in Z.

A key observation is that if LU factors of A are available, then it is possible to express Z in the alternative form

$$Z^T = \begin{bmatrix} -L_2 L_1^{-1} & I \end{bmatrix} \tag{3.2}$$

in which the UU^{-1} factors arising from (2.9) and (3.1) are cancelled out. A minor disadvantage, compared to (2.9), is that L_2 is needed, which is likely to be less sparse than A_2 and also requires additional storage. However if A is ill-conditioned, this is manifested in U (but not usually L) being ill-conditioned, so that (3.2) enables Z to be defined in a way which is well-conditioned. In calculating the reduced Hessian matrix it is convenient to define

$$\mathbf{L} = \begin{bmatrix} L_1 & \\ L_2 & I \end{bmatrix} \tag{3.3}$$

and replace equations (2.10) and (2.11) by

$$\mathbf{L}^T \mathbf{z}_k = \begin{bmatrix} L_1^T & L_2^T \\ & I \end{bmatrix} \mathbf{z}_k = \begin{pmatrix} \mathbf{0} \\ \mathbf{e}_k \end{pmatrix} \tag{3.4}$$

and

$$\mathbf{L}\mathbf{u} = \begin{bmatrix} L_1 & \\ L_2 & I \end{bmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = G\mathbf{z}_k. \tag{3.5}$$

The steps of the resulting null-space method are as follows (using subscript 1 to denote the first m rows of a vector or matrix, and subscript 2 to denote the last $n - m$ rows).

1. Calculate $Z^T G Z$ as in (3.4) and (3.5).

2. Calculate $\mathbf{s}_1 = L_1^{-T} U^{-T} \mathbf{b}$ and let $\mathbf{s} = \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{0} \end{pmatrix}$.

3. Calculate $\mathbf{t} = \mathbf{c} - G\mathbf{s}$ requiring a product with G.
4. Calculate $\mathbf{u}_2 = Z^T\mathbf{t} = \mathbf{t}_2 - L_2L_1^{-1}\mathbf{t}_1$.
5. Solve $Z^TGZ\mathbf{v} = \mathbf{u}_2$ for $\mathbf{v}$.
6. Calculate $\mathbf{w} = Z\mathbf{v} = \begin{pmatrix} -L_1^{-T}L_2^T\mathbf{v} \\ \mathbf{v} \end{pmatrix}$.
7. Calculate $\mathbf{x} = \mathbf{s} + \mathbf{w}$
8. Calculate $\mathbf{g} = \mathbf{c} - G\mathbf{x}$ requiring a product with G.
9. Calculate $\mathbf{y} = U^{-1}L_1^{-1}\mathbf{g}_1$.
10. Calculate $\mathbf{z} = Z^T\mathbf{g} = \mathbf{g}_2 - L_2L_1^{-1}\mathbf{g}_1$.

In the above, inverse operations involving L_1 and U are done by forward or backward substitution. The method is referred to as *Method 2* in what follows. (For comparability with Method 1, we have also included the calculation of the reduced gradient $\mathbf{z}$, although this would not normally be required.) Note that all solves with the $n \times n$ matrix $\mathbf{A}$ are replaced by solves with smaller $m \times m$ matrices. Also steps 1, 4, 6 and 10 use the alternative definition (3.2) of Z and so avoid a potentially ill-conditioned calculation with $\mathbf{A}$ (or A_1). We consider the numerical stability of both Method 1 and Method 2 in more detail in the next section.

In the rest of this section, we explore some connections between this method and some variants of Gaussian elimination, and we examine the factored forms that are provided by these methods. It is readily observed (but not well known) that there are block factors of K corresponding to any null-space method in this general format. These are the factors

$$K = \begin{bmatrix} A & V & \\ & & I \end{bmatrix} \begin{bmatrix} Y^TGY & Y^TGZ & I \\ Z^TGY & Z^TGZ & \\ I & & \end{bmatrix} \begin{bmatrix} A^T & & \\ V^T & & \\ & I & \end{bmatrix} \tag{3.6}$$

(using blanks to denote a zero matrix). This result is readily verified by using the equation $AY^T + VZ^T = I$ derived from (2.1). This expression makes it clear that inverse representations of the matrices $\mathbf{A}$ and Z^TGZ will be required to solve (1.1). However these factors are not directly useful as a method of solution as they also involve the matrices Y^TGY and Y^TGZ whose computation we wish to avoid in a null-space method. Equation (3.6) also shows that K^{-1} will become large when either $\mathbf{A}$ or Z^TGZ is ill-conditioned, and we would expect the spectral condition number to behave like $\kappa_K \sim \kappa_A^2\kappa_M$ where $M = Z^TGZ$.

When using direct elimination (2.8) we may partition K in the form

$$K = \begin{pmatrix} G_{11} & G_{12} & A_1 \\ G_{21} & G_{22} & A_2 \\ A_1^T & A_2^T & 0 \end{pmatrix}.$$

When A has LU factors (3.1) then it is readily verified that another way of factorizing K is given by

$$\begin{bmatrix} G_{11} & G_{12} & A_1 \\ G_{21} & G_{22} & A_2 \\ A_1^T & A_2^T & 0 \end{bmatrix} = \begin{bmatrix} L_1 & & \\ L_2 & I & \\ & & I \end{bmatrix} \begin{bmatrix} L_1^{-1}G_{11}L_1^{-T} & L_1^{-1}G_1Z & U \\ Z^TG_1^TL_1^{-T} & Z^TGZ & \\ U^T & & \end{bmatrix} \begin{bmatrix} L_1^T & L_2^T & \\ & I & \\ & & I \end{bmatrix} \tag{3.7}$$

where Z is defined by (3.2) and $G_1 = [G_{11}\ G_{12}]$. Note that the matrix U occurs on the reverse diagonal of the middle factor, but that no operations with U^{-1} are required in the calculation of the factors. Thus any ill-conditioning associated with U does not manifest itself until the factors are used in solving the KKT system (1.1). If there is no growth in Z then the backward error in (3.7) will be small, indicating the potential for a small residual solution of the KKT system. We show in Section 5 how this can come about. Another related observation is that if A is rank deficient, then the factors (3.6) do not exist (since the calculation of Y involves A_1^{-1} and hence U^{-1}) whereas (3.7) can be calculated without difficulty.

The factorization (3.7) of K is closely related to some symmetry preserving variants of Gaussian elimination. Let us start by eliminating A_2 and the sub-diagonal elements of A_1 by row operations. (As before we can assume that row pivoting has been used.) The outcome of these row operations is that

$$\begin{bmatrix} G_{11} & G_{12} & A_1 \\ G_{21} & G_{22} & A_2 \\ A_1^T & A_2^T & 0 \end{bmatrix} = \begin{bmatrix} L_1 & & \\ L_2 & I & \\ & & I \end{bmatrix} \begin{bmatrix} L_1^{-1}G_{11} & L_1^{-1}G_{12} & U \\ Z^TG_1^T & Z^TG_2^T & \\ A_1^T & A_2^T & \end{bmatrix} \tag{3.8}$$

where $G_2 = [G_{21}\ G_{22}]$. Note that these row operations are exactly those used by Gaussian elimination to form (3.1). To restore symmetry in the factors, we repeat the above procedure in transposed form, that is we make column operations on A_1^T and A_2^T, which gives rise to (3.7).

We can also interleave these row and column operations without affecting the final result. If we pair up the first row and column operation, then the second row and column operation, and so on, then we get the method of 'ba' pivots described by Forsgren and Murray [4]. Thus these methods essentially share the same matrix factors. The difference is that in the null-space method, Z^TGZ is calculated by matrix solves with $\mathbf{A}$, as described in Section 2, whereas in these other methods it is obtained by row and column operations on the matrix K.

This association with Gaussian elimination enables us to bound the growth in the

factors of K. The bound is attained for the critical case typified by the matrix

$$K = \left[\begin{array}{cccccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 0 & 1 & -1 & -1 & -1 & -1 \\ \hline 1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \end{array}\right]$$

for which $n = 6$ and $m = 4$. Row operations with pivots in the (1,7), (2,8), (3,9) and (4,10) positions leads to the matrix

$$\left[\begin{array}{cccccc|cccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 & 2 & 0 & 1 & 0 & 0 \\ 4 & 4 & 4 & 4 & 4 & 4 & 0 & 0 & 1 & 0 \\ 8 & 8 & 8 & 8 & 8 & 8 & 0 & 0 & 0 & 1 \\ 16 & 16 & 16 & 16 & 16 & 15 & 0 & 0 & 0 & 0 \\ 16 & 16 & 16 & 16 & 15 & 16 & 0 & 0 & 0 & 0 \\ \hline 1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 \end{array}\right] .$$

Then column operations with pivots in the (7,1), (8,2), (9,3) and (10,4) positions gives rise to

$$\left[\begin{array}{cccccc|cccc} 1 & 2 & 4 & 8 & 16 & 16 & 1 & 0 & 0 & 0 \\ 2 & 4 & 8 & 16 & 32 & 32 & 0 & 1 & 0 & 0 \\ 4 & 8 & 16 & 32 & 64 & 64 & 0 & 0 & 1 & 0 \\ 8 & 16 & 32 & 64 & 128 & 128 & 0 & 0 & 0 & 1 \\ 16 & 32 & 64 & 128 & 256 & 255 & 0 & 0 & 0 & 0 \\ 16 & 32 & 64 & 128 & 255 & 256 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

which corresponds to the middle factor in (3.7). In this case $U = I$, $L = A$ and the corresponding matrix Z is given by

$$Z^T = \begin{bmatrix} 8 & 4 & 2 & 1 & 1 & 0 \\ 8 & 4 & 2 & 1 & 0 & 1 \end{bmatrix} .$$

In general it is readily shown that when $m < n$, growth of 2^{2m} in the maximum modulus element of K can occur. For the null-space method based on (3.2), this example also illustrates the maximum possible growth of 2^{m-1} in Z, when $|l_{ij}| \leq 1$. In practice however such growth is most unlikely and it is usual not to get any significant growth in Z.

4 Numerical Stability of Method 1

In this and the next section we consider the effect of ill-conditioning in the matrix K on the solutions obtained by null-space methods based on direct elimination. In particular we are interested to see whether or not we can establish results comparable to those for Gaussian elimination. We shall show that the forward error in $\mathbf{x}$ is not as severe as would be predicted by the condition number of K. We also look at the residual errors in the solution and show that Method 2 is very satisfactory in this respect, whereas Method 1 is not.

In order to prevent the details of the analysis from obscuring the insight that we are trying to provide, we shall adopt the following simple convention. We imagine that we are solving a sequence of problems in which either κ_A or κ_M (the spectral condition numbers of $\mathbf{A}$ and $M = Z^T G Z$) is increasing without bound. We then use the notation $O(h)$ to indicate a quantity that is bounded in norm by $c\|h\|$ on this sequence, where there exists an implied constant c that is independent of κ_A or κ_M, but may contain a modest dependence on n. Also we shall assume that the system is well scaled so that $G = O(1)$ and $A \sim 1$. This enables us for example to deduce that multiplication of an error bound $O(\varepsilon)$ by $\mathbf{A}^{-1}$ causes the bound to be increased to $O(\kappa_A \varepsilon)$. We also choose to assume that the KKT system models a situation in which the exact solution vectors $\mathbf{x}$ and $\mathbf{y}$ exist and are not unreasonably large in norm, that is $\mathbf{x} = O(1)$ and $\mathbf{y} = O(1)$. A similar assumption is needed in order to show that Gaussian elimination provides accurate residuals, so we cannot expect to dispense with this assumption. Sometimes it may be possible to argue that we are solving a physical problem which is known to have a well behaved solution.

Another assumption that we make is that the choice of the matrix V in (2.8) (and hence the partitioning of A) is made using some form of pivoting. Now the exact solution for Z is given by

$$Z^T = [\, -A_2 A_1^{-1} \quad I \,] = [\, -L_2 L_1^{-1} \quad I \,]$$

from (3.3), using the factors of A defined in (3.1). It follows that

$$Z = O(\kappa_L) \tag{4.1}$$

where κ_L is the spectral condition number of $\mathbf{L}$. Assuming that partial pivoting is used, so that $|l_{ij}| \leq 1$, and that negligible growth occurs in L_1^{-1}, it then follows that negligible growth occurs in Z and we can assert that

$$\kappa_L = O(1) \quad \text{and} \quad Z = O(1). \tag{4.2}$$

Another consequence of this assumption is that we are able to neglect terms of $O(\kappa_L \varepsilon)$ relative to terms of $O(\kappa_A \varepsilon)$ when assessing the propagation of errors for Method 2.

We shall now sketch some properties (Wilkinson [6]) of floating point arithmetic of relative precision ε. If a nonsingular system of n linear equations $A\mathbf{x} = \mathbf{b}$ is solved by Gaussian elimination, the computed solution $\hat{\mathbf{x}}$ is the exact solution of a perturbed system $(A + E)\hat{\mathbf{x}} = \mathbf{b}$ where E is referred to as the backward error. E can be bounded by an expression of the form $\rho\phi(n)\varepsilon$ in which ρ measures the growth in A during the elimination and $\phi(n)$ is a modest quadratic in n. For ill-conditioned systems, and assuming that partial pivoting is used, growth is rare and can be ignored. Also this bound usually overstates the dependence on n which is unlikely to be a dominant factor. Hence for the backward error

$$E = O(\varepsilon). \tag{4.3}$$

We can measure the accuracy of the solution either by the forward error $\hat{\mathbf{x}} - \mathbf{x} = -A^{-1}E\hat{\mathbf{x}}$ or by computing the residual $\mathbf{r} = A\hat{\mathbf{x}} - \mathbf{b} = -E\hat{\mathbf{x}}$. Using $A = O(1)$ we have

$$\hat{\mathbf{x}} = \mathbf{x} + O(\kappa_A \varepsilon \hat{\mathbf{x}}).$$

where κ_A is some condition number of A. Since $\mathbf{x} = O(1)$, and assuming that $\kappa_A \varepsilon \ll 1$, it follows that $\hat{\mathbf{x}} = O(1)$ and hence

$$\hat{\mathbf{x}} = \mathbf{x} + O(\kappa_A \varepsilon). \tag{4.4}$$

Likewise we can deduce that

$$\mathbf{r} = -E\hat{\mathbf{x}} = O(\varepsilon). \tag{4.5}$$

These bounds are likely to be realistic and tell us that for Gaussian elimination, ill-conditioning affects the forward error in $\mathbf{x}$ but not the residual $\mathbf{r}$, as long as $\hat{\mathbf{x}}$ is of reasonable magnitude.

Wilkinson also gives expressions for the backward error in a scalar product and hence in the product $\mathbf{s} = \mathbf{b} + A\mathbf{x}$. The computed product $\hat{\mathbf{s}}$ is the exact product of a system in which the relative perturbation in each element of $\mathbf{b}$ and A is no more than $n\varepsilon$ where n is the dimension of $\mathbf{x}$. We can express this as

$$\hat{\mathbf{s}} = \mathbf{s} + O(\varepsilon) \tag{4.6}$$

if we make the assumption that $\mathbf{b}$ and A are $O(1)$.

The first stage in a null-space calculation is the determination of $Z^T G Z$, which we denote by M. In Method 1, this is computed as in (2.10) and (2.11). In (2.10) a column $\mathbf{z}_k$ of the matrix Z is computed which, by applying (4.4), satisfies

$$\hat{\mathbf{z}}_k = \mathbf{z}_k + O(\kappa_A \varepsilon) \tag{4.7}$$

where κ_A is the spectral condition number of $\mathbf{A}$. The product with G introduces negligible error, and the solution of (2.11) together with (4.5) shows that

$$\mathbf{A}\hat{\mathbf{u}} = G\hat{\mathbf{z}}_k + O(\varepsilon).$$

Multiplying by $\mathbf{L}^{-1}$ and extracting the $\hat{\mathbf{u}}_2$ partition gives

$$\begin{aligned} \hat{\mathbf{u}}_2 &= Z^T G\hat{\mathbf{z}}_k + O(\kappa_L\varepsilon) \\ &= Z^T G\mathbf{z}_k + O(\kappa_A\varepsilon) \end{aligned}$$

using (4.7) and then (4.2). Hence we have established that

$$\widehat{M} = M + O(\kappa_A\varepsilon). \tag{4.8}$$

The argument has been given in some detail as it is important to see why the error in M is $O(\kappa_A\varepsilon)$ and not $O(\kappa_A^2\varepsilon)$. We also observe that $M = Z^TGZ = O(1)$ and hence that $\widehat{M} = O(1)$ when $\kappa_A\varepsilon \ll 1$.

We now turn to the solution of the KKT system using Method 1. We shall assume that systems involving $\mathbf{A}$ and M are solved in such a way that (4.5) applies. Using (4.6), and assuming that the computed quantities $\hat{\mathbf{s}}, \hat{\mathbf{t}}, \ldots, \hat{\mathbf{z}}$ are $O(1)$, the residual errors in the sequence of calculations are then

$$\mathbf{A}^T\hat{\mathbf{s}} = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \end{pmatrix} + O(\varepsilon) \tag{4.9}$$

$$\hat{\mathbf{t}} = \mathbf{c} - G\hat{\mathbf{s}} + O(\varepsilon) \tag{4.10}$$

$$\mathbf{A}\hat{\mathbf{u}} = \hat{\mathbf{t}} + O(\varepsilon) \tag{4.11}$$

$$\widehat{M}\hat{\mathbf{v}} = \hat{\mathbf{u}}_2 + O(\varepsilon) \tag{4.12}$$

$$\mathbf{A}^T\hat{\mathbf{x}} = \begin{pmatrix} \mathbf{b} \\ \hat{\mathbf{v}} \end{pmatrix} + O(\varepsilon) \tag{4.13}$$

$$\hat{\mathbf{g}} = \mathbf{c} - G\hat{\mathbf{x}} + O(\varepsilon) \tag{4.14}$$

$$\mathbf{A}\begin{pmatrix} \hat{\mathbf{y}} \\ \hat{\mathbf{z}} \end{pmatrix} = \hat{\mathbf{g}} + O(\varepsilon). \tag{4.15}$$

These results, together with (4.8), may be combined to get the forward errors in the solution vectors $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$. Multiplying through equations (4.9) and (4.13) by $\mathbf{A}^{-T}$ magnifies the errors by a factor κ_A (since we are assuming that $A = O(1)$), giving

$$\hat{\mathbf{s}} = Y\mathbf{b} + O(\kappa_A\varepsilon) \tag{4.16}$$

$$\hat{\mathbf{x}} = Y\mathbf{b} + Z\hat{\mathbf{v}} + O(\kappa_A\varepsilon). \tag{4.17}$$

We can get a rather better bound from (4.11) and (4.15) by first multiplying through by $\mathbf{L}^{-1}$ and using $\kappa_L = O(1)$ to give

$$\hat{\mathbf{u}}_2 = Z^T\hat{\mathbf{t}} + O(\varepsilon) \tag{4.18}$$

$$\hat{\mathbf{z}}_2 = Z^T\hat{\mathbf{g}} + O(\varepsilon) \tag{4.19}$$

from the second partition of the solution. However the first partition of (4.15) gives

$$\hat{\mathbf{y}} = Y^T\hat{\mathbf{g}} + O(\kappa_A\varepsilon). \tag{4.20}$$

Combining (4.8) and (4.12) gives

$$\hat{\mathbf{v}} = M^{-1}\hat{\mathbf{u}}_2 + O(\kappa_A\varepsilon) + O(\kappa_M\varepsilon). \tag{4.21}$$

We can now chain through the forward errors, noting that a product with Z or Z^T does not magnify the error in a previously computed quantity (by virtue of (4.2). However the product $M^{-1}\hat{\mathbf{u}}_2$ in (4.21) magnifies the error in $\hat{\mathbf{u}}_2$ by a factor κ_M and the product $Y^T\hat{\mathbf{g}}$ in (4.20) magnifies the error in $\hat{\mathbf{g}}$ by a factor κ_A. The outcome is that

$$\hat{\mathbf{x}} = \mathbf{x} + O(\kappa_A\kappa_M\varepsilon) \tag{4.22}$$

and

$$\hat{\mathbf{y}} = \mathbf{y} + O(\kappa_A^2\kappa_M\varepsilon). \tag{4.23}$$

As we would expect, the forward errors are affected by the condition numbers of $\mathbf{A}$ and M. However although the condition number of K is expected to be of the order $\kappa_A^2\kappa_M$, we see that this factor only magnifies the error in the $\mathbf{y}$ part of the solution, with the $\mathbf{x}$ part being less badly affected.

When K is ill-conditioned we must necessarily expect that the forward errors are adversely affected. A more important question is to ask whether the solution satisfies the equations of the problem accurately. There are three measures of interest, the residuals $\mathbf{q} = G\mathbf{x} + A\mathbf{y} - \mathbf{c}$ and $\mathbf{r} = A^T\mathbf{x} - \mathbf{b}$ of the KKT system (1.1), and the reduced gradient $\mathbf{z} = Z^T\mathbf{g}$ where $\mathbf{g} = \mathbf{c} - G\mathbf{x}$ is the negative gradient vector at the solution. We note that the vector $\mathbf{z}$ is computed as a by-product of step 8 of Method 1.

If we compute $\mathbf{r}$ we obtain $\hat{\mathbf{r}} = A^T\hat{\mathbf{x}} - \mathbf{b} + O(\varepsilon)$ as in (4.6), and it follows from (4.13) and the definition of $\mathbf{A}$ that $A^T\hat{\mathbf{x}} = \mathbf{b} + O(\varepsilon)$. Thus

$$\hat{\mathbf{r}} = O(\varepsilon). \tag{4.24}$$

When computing $\mathbf{q}$ we obtain

$$\begin{aligned}
\hat{\mathbf{q}} &= G\hat{\mathbf{x}} + A\hat{\mathbf{y}} - \mathbf{c} + O(\varepsilon) && (4.25)\\
&= A\hat{\mathbf{y}} - \hat{\mathbf{g}} + O(\varepsilon) && (4.26)\\
&= -\begin{pmatrix}\mathbf{0}\\ \hat{\mathbf{z}}\end{pmatrix} + O(\varepsilon) && (4.27)
\end{aligned}$$

from (4.14) and (4.15). Thus the accuracy of $\hat{\mathbf{q}}$ depends on that of $\hat{\mathbf{z}}$. From (4.19) and (4.14) it follows that

$$\begin{aligned}
\hat{\mathbf{z}} &= Z^T\mathbf{c} - Z^TG\hat{\mathbf{x}} + O(\varepsilon)\\
&= Z^T\mathbf{c} - Z^TGY\mathbf{b} - Z^TGZ\hat{\mathbf{v}} + O(\kappa_A\varepsilon)
\end{aligned}$$

from (4.17). (Notice that it is important not to use (4.22) here which would give an unnecessary factor of κ_M.) Then (4.8), (4.12), (4.11) and (4.10) can be used, giving

$$\hat{\mathbf{z}} = O(\kappa_A \varepsilon). \tag{4.28}$$

Thus we are able to predict under our assumptions that the reduced gradient $\hat{\mathbf{z}}$ and the residual $\hat{\mathbf{q}}$ are adversely affected by ill-conditioning in A, but not by ill-conditioning in M. However the residual $\hat{\mathbf{r}}$ is unaffected by ill-conditioning either in A or M.

Simulations are described in Section 6 which indicate that these error bounds reliably predict the actual effects of ill-conditioning. Method 1 is seen to be unsatisfactory in that an accurate residual $\mathbf{q}$ cannot be obtained when A is ill-conditioned. We shall show in the next section that Method 2 does not share this disadvantage.

The main results of this section and the next are summarised and discussed in Section 7.

5 Numerical Stability of Method 2

In this section we assess the behaviour of Method 2 in the presence of ill-conditioning in K. Although we cannot expect any improvement for the forward errors, we are able to show that Method 2 is able to give accurate residuals that are not affected by ill-conditioning. The relationship between Method 2 and Gaussian elimination described towards the end of Section 3 gives some hope of proving this result. However this is not immediate because Method 2 does not make direct use of the factors (3.7) in the same way that Gaussian elimination does.

A fundamental difficulty with the analysis of Method 2 is that we can deduce from (4.7) that

$$\hat{Z} = Z + O(\kappa_A \varepsilon) \tag{5.1}$$

and this result cannot be improved if LU factors are available. To see this, we know that the computed factors of any square matrix A satisfy

$$\hat{L}\hat{U} = A + E = A + O(\varepsilon) \tag{5.2}$$

when there is no growth in $\hat{U}$. If $A = LU$ are the exact factors, it follows that

$$L^{-1}\hat{L} = U\hat{U}^{-1} + L^{-1}E\hat{U}^{-1} = U\hat{U}^{-1} + Q + R$$

say, where Q is the strict lower triangular part of $L^{-1}E\hat{U}^{-1}$ and R is the upper triangular part. Because $L^{-1}\hat{L}$ is unit lower triangular and $U\hat{U}^{-1}$ is upper triangular we can deduce that $L^{-1}\hat{L} = I + Q$ and $U\hat{U}^{-1} = I - R$. Since $L^{-1}E\hat{U}^{-1}$ involves an inverse operation with $\hat{U}$ we can expect that $\hat{L}$ and L differ by $O(\kappa_A \varepsilon)$. This result has been confirmed by computing the LU factors of a Hilbert matrix in single and double precision Fortran. On applying the result to our matrix $\mathbf{A}$, it follows that (5.1) holds.

Fortunately all is not lost because we are still able to compute a null-space matrix which accurately satisfies the equation $Z^T A = 0$. Let $\widehat{Z}$ denote the null-space matrix obtained from $\widehat{L}$ in exact arithmetic. It follows that $\widehat{Z}^T \widehat{L} = 0$ and hence from (5.2) that

$$\widehat{Z}^T A = O(\varepsilon). \tag{5.3}$$

We also have $\widehat{Z} = O(1)$ as long as $\kappa_A \varepsilon \ll 1$. Our analysis will express the errors that arise in Method 2 in terms of $\widehat{Z}$ rather than Z and this enables us to avoid the κ_A factor in the residuals.

The first step in Method 2 is to compute $M = Z^T G Z$ as in (3.4) and (3.5). In this section we denote $\widehat{M} = \widehat{Z}^T G \widehat{Z}$ as the value computed from $\widehat{Z}$ in exact arithmetic and use $\widehat{\widehat{M}}$ to denote the computed value of $\widehat{M}$. It readily follows, using results like (4.2), that

$$\widehat{\widehat{M}} = \widehat{M} + O(\varepsilon). \tag{5.4}$$

We now consider the solution of the KKT system using Method 2. As in equations (4.9) through (4.15) we assume that the computed quantities $\hat{\mathbf{s}}$, $\hat{\mathbf{t}}, \ldots,$ $\hat{\mathbf{z}}$ are $O(1)$. Then the residual errors in the sequence of calculations are

$$\begin{aligned}
A_1^T \hat{\mathbf{s}}_1 &= \mathbf{b} + O(\varepsilon) \quad \text{and} \quad \hat{\mathbf{s}}_2 = \mathbf{0} && (5.5)\\
\hat{\mathbf{t}} &= \mathbf{c} - G\hat{\mathbf{s}} + O(\varepsilon) && (5.6)\\
\hat{\mathbf{u}}_2 &= \widehat{Z}^T \hat{\mathbf{t}} + O(\varepsilon) && (5.7)\\
\widehat{\widehat{M}} \hat{\mathbf{v}} &= \hat{\mathbf{u}}_2 + O(\varepsilon) && (5.8)\\
\hat{\mathbf{w}} &= \widehat{Z} \hat{\mathbf{v}} + O(\varepsilon) && (5.9)\\
\hat{\mathbf{x}} &= \hat{\mathbf{s}} + \hat{\mathbf{w}} + O(\varepsilon) && (5.10)\\
\hat{\mathbf{g}} &= \mathbf{c} - G\hat{\mathbf{x}} + O(\varepsilon) && (5.11)\\
A_1 \hat{\mathbf{y}} &= \hat{\mathbf{g}}_1 + O(\varepsilon) && (5.12)\\
\hat{\mathbf{z}} &= \widehat{Z}^T \hat{\mathbf{g}} + O(\varepsilon). && (5.13)
\end{aligned}$$

It is readily seen from these equations that the forward errors will propagate in a similar way to Method 1.

Turning to the residual errors, the computed value of the residual $\mathbf{r}$ is

$$\hat{\mathbf{r}} = A^T \hat{\mathbf{x}} - \mathbf{b} + O(\varepsilon) = A^T \hat{\mathbf{s}} + A^T \widehat{Z} \hat{\mathbf{v}} - \mathbf{b} + O(\varepsilon) = O(\varepsilon) \tag{5.14}$$

from (5.10), (5.9), (5.5) and (5.3). When computing $\mathbf{q}$ we obtain $\hat{\mathbf{q}} = A\hat{\mathbf{y}} - \hat{\mathbf{g}} + O(\varepsilon)$ as for Method 1, and it follows from (5.12) that $\hat{\mathbf{q}}_1 = O(\varepsilon)$. From (5.3) we can deduce that $\widehat{Z}^T \hat{\mathbf{q}} = -\widehat{Z}^T \hat{\mathbf{g}} + O(\varepsilon)$. But $\widehat{Z}^T \hat{\mathbf{q}} = \hat{\mathbf{q}}_2 - \hat{L}_1^{-1} \hat{L}_2 \hat{\mathbf{q}}_1 = \hat{\mathbf{q}}_2 + O(\varepsilon)$ so it follows that

$$\hat{\mathbf{q}}_2 = -\widehat{Z}^T \hat{\mathbf{g}} + O(\varepsilon) = -\hat{\mathbf{z}} + O(\varepsilon). \tag{5.15}$$

Thus the accuracy of the residual $\hat{\mathbf{q}}$ depends on that of $\hat{\mathbf{z}}$, as for Method 1. For $\hat{\mathbf{z}}$ we can use (5.13), (5.11), (5.10) and (5.9) to get

$$\hat{\mathbf{z}} = \hat{Z}^T\mathbf{c} - \hat{Z}^T G\hat{\mathbf{s}} - \hat{Z}^T G\hat{Z}\hat{\mathbf{v}} + O(\varepsilon).$$

Now we can invoke (5.4) and (5.8) giving

$$\hat{\mathbf{z}} = \hat{Z}^T\mathbf{c} - \hat{Z}^T G\hat{\mathbf{s}} - \hat{\mathbf{u}}_2 + O(\varepsilon) = O(\varepsilon) \tag{5.16}$$

from (5.7) and (5.6). Thus we have established under our assumptions that all three measures of accuracy for the KKT system are $O(\varepsilon)$ for Method 2 and are not affected by ill-conditioning in either A or M. These results are again supported by the simulations in the next section.

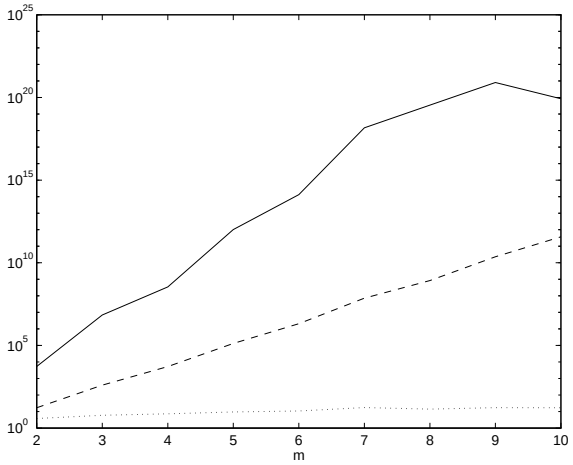

Figure 1. Condition numbers of K, $\mathbf{A}$ and $\mathbf{L}$

6 Numerical Experiments

In order to check the predictions of Sections 4 and 5, some experiments have been carried out on artifically generated KKT systems. These experiments have been carried out in Matlab for which the machine precision is $\varepsilon \simeq 10^{-16}$. They suggest that the upper bounds given by the error analysis accurately reflect the actual behaviour of an ill-conditioned system. Another phenomenon that occurs when the ill-conditioning is very extreme is also explained.

The KKT systems have been constructed in the following way. To make A ill-conditioned we have chosen it as the first m columns of the $n \times n$ Hilbert matrix, where $n = 2m$. Choosing $m = 2, 3, \ldots, 10$ provides a sequence of problems for which the condition number of $\mathbf{A}$ increases exponentially. Factors $PA = LU$ are calculated by the Matlab routine `lu` which uses Gaussian Elimination with partial pivoting, and A is replaced by PA. In the first instance the matrix G is generated by random numbers in the range $[-1, 1]$. However to make $M = Z^T GZ$ positive definite, a multiple of the unit matrix is added to the G_{22} partition of G, chosen so that the smallest eigenvalue of M

is changed to 10^{1-k} for some positive integer k. The assumptions of the analysis require that the KKT system has a solution that is $O(1)$. To achieve this, exact solutions $\mathbf{x}$ and $\mathbf{y}$ are generated by random numbers in $[-1, 1]$, and the right hand sides $\mathbf{c}$ and $\mathbf{b}$ are calculated from (1.1). For each value of m, 10 runs are made with a different random number seed and the statistics are averaged over these 10 runs.

First of all we examine the effect of increasing the condition number of A whilst keeping M well-conditioned. To do this we increase m from 2 up to 10, whilst fixing $k = 1$. The resulting condition numbers of K, $\mathbf{A}$ and $\mathbf{L}$ are plotted in Figure 1. It can be seen that the slope of the unbroken line (κ_K) is about twice that of the dashed line (κ_A). Since $\kappa_M \sim 1$, this is consistent with the estimate $\kappa_K \sim \kappa_A^2 \kappa_M$ that we deduced in Section 3. The condition number of $\mathbf{L}$ (dotted line) shows negligible increase, showing that there is no growth in L_1^{-1}, thus enabling us to assert that $Z = O(1)$. The levelling out of the κ_K graph for $m = 8$, 9 and 10 is due to round-off error corrupting the least eigenvalue of K.

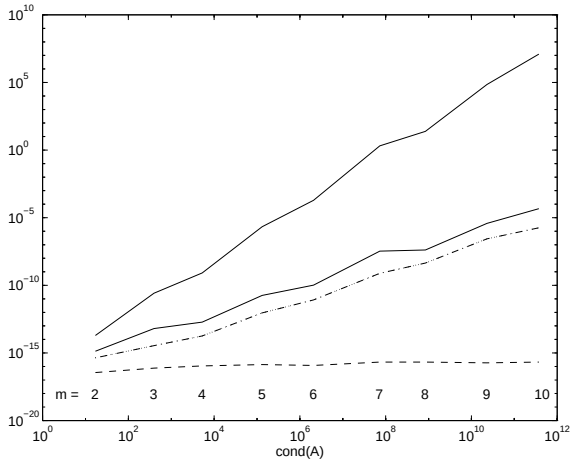

Figure 2. Error growth vs. κ_A for Method 1

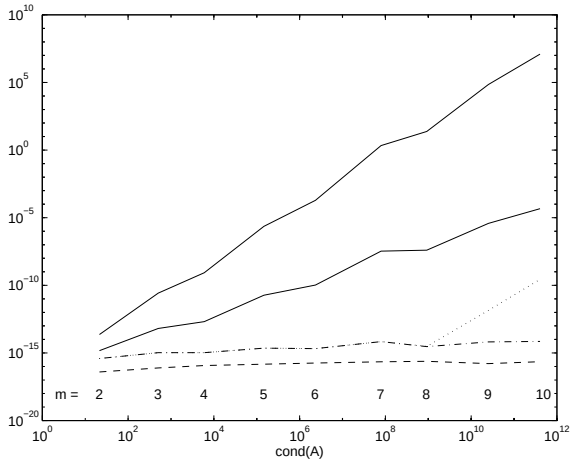

Figure 3. Error growth vs. κ_A for Method 2

The effect of the conditioning of A on the different types of error is illustrated in Figures 2 and 3. The forward error is shown by the two unbroken lines, the upper line being the error in $\mathbf{y}$ and the lower line being the error in $\mathbf{x}$. The upper line has a slope of about 2 on the log-log scale, and the lower line has a slope of about 1, and both have an intercept with the y-axis of about 10^{-16}. This is precisely in accordance with (4.23) and (4.22). It can also be seen that both methods exhibit the same forward error. The computed value of the residual error $\mathbf{r} = A^T\mathbf{x} - \mathbf{b}$ is shown by the dashed line and both methods show the $O(\varepsilon)$ behaviour as predicted by (4.24) and (5.14), with the increasing condition number having no effect.

The difference between Methods 1 and 2 is shown by the computed values of the residual $\mathbf{q} = G\mathbf{x} + A\mathbf{y} - \mathbf{c}$ (dotted line) and the reduced gradient $\mathbf{z} = Z^T\mathbf{g}$ (dash-dot line). As we would expect from (4.27), these graphs are superimposed, and they clearly show the influence of κ_A on the error growth for Method 1, as predicted by (4.28). Negligible error growth is observed for Method 2 as predicted by (5.16), except for an

increase in $\mathbf{q}$ for κ_A greater than about 10^9. This feature is explained later in the section.

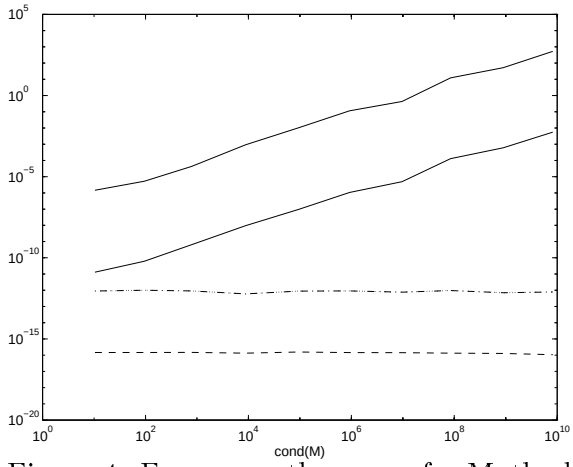

Figure 4. Error growth vs. κ_M for Method 1

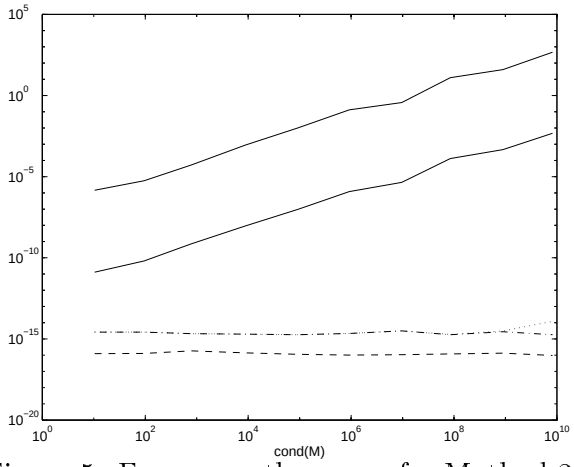

Figure 5. Error growth vs. κ_M for Method 2

We now turn to see the influence of ill-conditioning in M on the errors. To do this we fix $m = 5$, for which $\kappa_A \simeq 10^5$, and carry out a sequence of calculations with $k = 1, 2, \ldots, 10$, which causes κ_M to increase exponentially. Each calculation is the average of ten runs as above. The results are illustrated in Figures 3 and 4, using the same key. The forward errors are again seen to be the same for both methods and they both have a slope of about 1 on the log-log scale, corresponding to the κ_M factor in (4.22) and (4.23). The upper line for the forward error in $\mathbf{y}$ lies about 10^5 units above that for the forward error in $\mathbf{x}$, as the extra factor of κ_A in (4.23) would predict. The residual $\mathbf{r}$ is seen to be unaffected by the conditioning of M as above. The residual $\mathbf{q}$ and the reduced gradient $\mathbf{z}$ are also unaffected by κ_M, but the graphs for Method 1 lie above those for Method 2, due to the κ_A factor in (4.28). All these effects are in accordance with what the error analysis predicts.

To examine the anomalous behaviour of $\mathbf{q}$ in Figure 3 in more detail, we turn to a sequence of more ill-conditioned test problems obtained by using the *last* m columns of the Hilbert matrix to define A. The results for Method 2 are illustrated in Figure 6 and the anomalous behaviour (dotted line) is now very evident. The reason for this becomes apparent when it is noticed that it sets in when the forward error in $\mathbf{y}$, and hence the value of $\widehat{\mathbf{y}}$, becomes greater than unity. This possibility has been excluded in our error analysis by the assumption that $\widehat{\mathbf{y}} = O(1)$. The anomalous behaviour sets in when $\kappa_A^2 \kappa_M \varepsilon \simeq 1$, that is $\kappa_A \simeq (\kappa_M \varepsilon)^{-1/2}$, or in this case $\kappa_A \simeq 10^8$, much as Figures 3 and 6 illustrate. For greater values of κ_A there is a term $O(\widehat{\mathbf{y}}\varepsilon)$ in the expression for $\widehat{\mathbf{q}}$ indicating that the error is of the form $\kappa_A^2 \kappa_M \varepsilon^2$. The fact that this part of the graph of $\widehat{\mathbf{q}}$ is parallel to the graph of the forward error in $\mathbf{y}$ supports this conclusion.

The above calculations have also been carried out using a Vandermonde matrix in place of the Hilbert matrix and very similar results have been obtained.

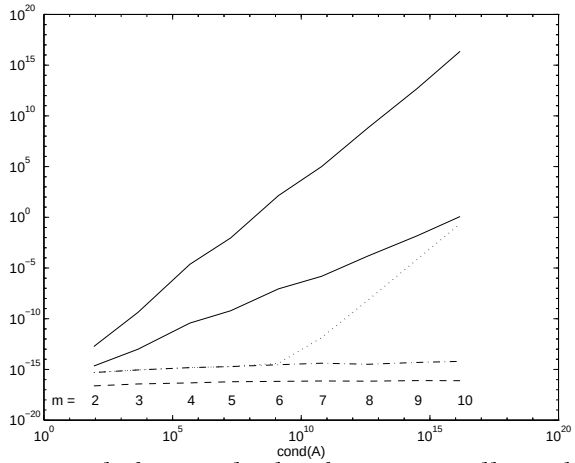

Figure 6. Error growth for Method 2 for a more ill-conditioned matrix

7 Summary and Discussion

In this paper we have examined the effect of ill-conditioning on the solution of a KKT system by null-space methods based on direct elimination. Such methods are important because they are well suited to take advantage of sparsity in large systems. However they have often been criticised for a lack of numerical stability, particularly when compared to methods based on QR factors. We have studied two methods: Method 1 in which an invertible representation of $\mathbf{A}$ in (2.8) is used to solve systems, and Method 2 in which LU factors (3.1) of A are available. We have presented error analysis backed up by numerical simulations which, under certain assumptions on growth, provide the following conclusions

- Both methods have the same forward error bounds, with $\widehat{\mathbf{x}} = \mathbf{x} + O(\kappa_A \kappa_M \varepsilon)$ and $\widehat{\mathbf{y}} = \mathbf{y} + O(\kappa_A^2 \kappa_M \varepsilon)$.
- Both methods give accurate residuals if A is well conditioned, even if M is ill-conditioned.
- Method 2 always gives an accurate residual $\mathbf{q} = G\mathbf{x} + A\mathbf{y} - \mathbf{c}$, whereas $\mathbf{q} = O(\kappa_A \varepsilon)$ for Method 1.
- Both methods give an accurate residual $\mathbf{r} = A^T\mathbf{x} - \mathbf{b}$ if A is ill-conditioned.

These conclusions do indicate that Method 1 is adversely affected by ill-conditioning in A, even though the technique for solving systems involving $\mathbf{A}$ is able to provide accurate residuals. The reasons for this are particularly interesting. For example one might expect that when A is ill-conditioned, then $\mathbf{A}^{-1}$ would be large and we might therefore expect from (2.1) that Z would be large. In fact we have seen that as long as V is chosen suitably, then growth in Z is very unlikely (the argument is similar to that for Gaussian elimination). Of course if V is badly chosen then Z can be large and this will cause

significant error. One might also expect that because the forward error in computing Z is necessarily of order $O(\kappa_A \varepsilon)$, it would follow that no null-space method could provide accurate residuals.

The way forward, which is exploited in the analysis for Method 2, is that Method 2 determines a matrix $\widehat{Z}$ for which $\widehat{Z}^T A = O(\varepsilon)$. Thus, although the null-space is inevitably badly determined when A is ill-conditioned, Method 2 fixes on one particular basis matrix $\widehat{Z}$ that is well behaved. This basis is an exact basis for an $O(\varepsilon)$ perturbation to A. Method 2 is able to solve this perturbed problem accurately. On the other hand Method 1 essentially obtains a different approximation to Z for every solve with $\mathbf{A}$. Thus the computed reduced Hessian matrix $Z^T G Z$ does not correspond accurately to any one particular $\widehat{Z}$ matrix.

In passing, it is interesting to remark that computing the factors

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix},$$

and defining $Z = Q_2$, also provides a stable approach, not so much because it avoids the growth in Z (we have seen that this is rarely a problem), but because it also provides a fixed null-space reference basis, which is an exact basis for an $O(\varepsilon)$ perturbation to A.

In the context of quadratic programming, a common solution method for large sparse systems is to use some sort of product form method (Gauss-Jordan, Bartels-Golub-Reid, Forrest-Tomlin etc.). It is not clear that such methods provide $O(\varepsilon)$ solutions to the systems involving $\mathbf{A}$ that are solved in Method 1 (although B-G-R may be stable in this respect). However the main difficulty comes when the product form becomes too unweildy and is re-inverted. If A is ill-conditioned, the refactorization of $\mathbf{A}$ is likely to determine a basis matrix Z that differs by $O(\kappa_A \varepsilon)$ from that defined by the old product form. Thus the old reduced Hessian matrix $Z^T G Z$ would not correspond accurately to that defined by the new Z matrix after re-inversion. The only recourse would be to re-evaluate $Z^T G Z$ on re-inversion, which might be very expensive. Thus we do not see a product form method on its own as being suitable. Our paper has shown that if a fixed reference basis is generated then accurate residuals are possible. It is hoped to show how this might be done in a subsequent paper by combining a product form method with another method such as LU factorization.

8 References

[1] Bunch J.R. and Parlett B.N. (1971) Direct methods for solving symmetric indefinite systems of linear equations, *Num. Funct. Anal. and Opt.*, **12**, 253-269.

[2] Duff I.S., Erisman A.M. and Reid J.K. (1986) *Direct Methods for Sparse Matrices*, Clarendon Press, Oxford.

[3] Fletcher R. (1987) *Practical Methods of Optimization, 2nd Edn.*, Wiley, Chichester.

[4] Forsgren A. and Murray W. (1993) Newton methods for large-scale linear equality-constrained minimization, *SIAM J. Matrix Anal. Appl.*, **14**, 560-587.

[5] Higham N.J. (1995) Stability of the Diagonal Pivoting Method with Partial Pivoting, MCCM Numerical Analysis Report 265, Manchester University.

[6] Wilkinson J.H. (1965) *The Algebraic Eigenproblem*, Oxford University Press, Oxford.