

IMPROVING THE SOLVABILITY OF ILL-CONDITIONED SYSTEMS OF LINEAR EQUATIONS BY REDUCING THE CONDITION NUMBER OF THEIR MATRICES

MUHAMMAD FAROOQ AND ABDELLAH SALHI

ABSTRACT. This paper is concerned with the solution of ill-conditioned Systems of Linear Equations (SLE's) via the solution of equivalent SLE's which are well-conditioned. A matrix is first constructed from that of the given ill-conditioned system. Then, an adequate right-hand side is computed to make up the instance of an equivalent system. Formulae and algorithms for computing an instance of this equivalent SLE and solving it will be given and illustrated.

1. Introduction

Ill-conditioned systems of linear equations (SLE's) are notoriously difficult to solve to any useful accuracy, [5]. Their matrices are characterized by large condition numbers. The difficulty may be negotiated by solving different but equivalent systems which are well-conditioned; well-conditioned SLE's have matrices with small condition numbers. The approach we put forward here constructs a new matrix and a new right-hand side that constitute an instance of an equivalent SLE to the one given which is ill-conditioned. Moreover, this new matrix has a small condition number compared to that of the matrix of the initial SLE. This means that solving this equivalent system must be better than solving the original one by virtue of the difference in the magnitude of the condition numbers of their matrices.

Before embarking in the derivation of the necessary formulae and algorithms to construct such equivalent SLE's, let us first motivate this approach.

Consider the following system of linear equations (SLE) in two variables.

$$\begin{aligned}x + y &= 2, \\x + 1.0001y &= 2.\end{aligned}$$

Received February 3, 2010; Revised October 8, 2010.

2010 *Mathematics Subject Classification.* 65N12.

Key words and phrases. condition number, systems of linear equations, ill-conditioning, equivalent systems, Schur complement.

It has solution $x = 2, y = 0$. If we change its RHS to $(2.0001, 2)$ the solution becomes $(3.0001, -1)$. These results, undoubtedly, bear the hallmark of an ill-conditioned SLE. Let us see why this is so. Draw the lines corresponding to each equation with a blunt pencil, (Figure 1). The intersection of the two lines is a set of points. Because the lines are nearly parallel, the candidate for solution, as can be seen in Figure 1, may vary wildly from $(2, 0)$. Near-parallelism is what exacerbates inaccuracy by creating a potentially very large set of solutions from which to pick. If we sharpen our pencil well and draw again the two lines, the “intersection set” will be reduced dramatically, but there will still be a lot of points to pick from, (Figure 2). The sharp pencil here may be interpreted as calculations with high accuracy, while a blunt one represents low accuracy calculations.

Now, if we cannot do much to the accuracy we have at our disposition, i.e., we have exhausted the capabilities of our computer, for instance, and as ill-conditioning is independent of accuracy, then what else can we do? Since near-parallelism is really the issue, an obvious way is to replace one of the near-parallel lines by a perpendicular line to either of them, but passing through the “point of intersection” of the near-parallel lines. If such a line could be found, then the drawing will have a better defined intersection set particularly if the pencil is really sharp! (See Figure 3). Note that, although there is only one perpendicular line to the one we decide to keep passing through the solution, at that point there is an infinite near parallel lines, each of which being a perfectly valid replacement to one of the problematic lines.

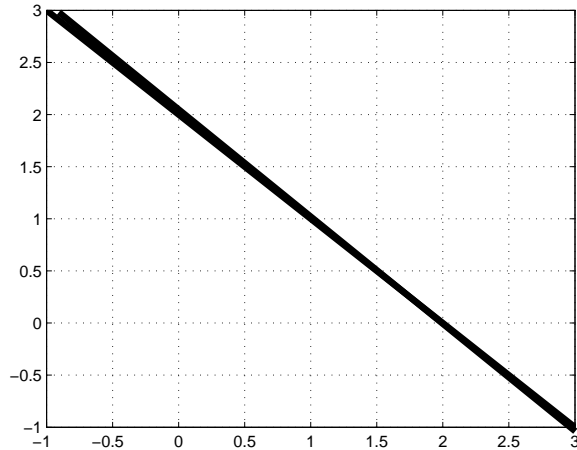


FIGURE 1. Thick line representation of ill-conditioning.

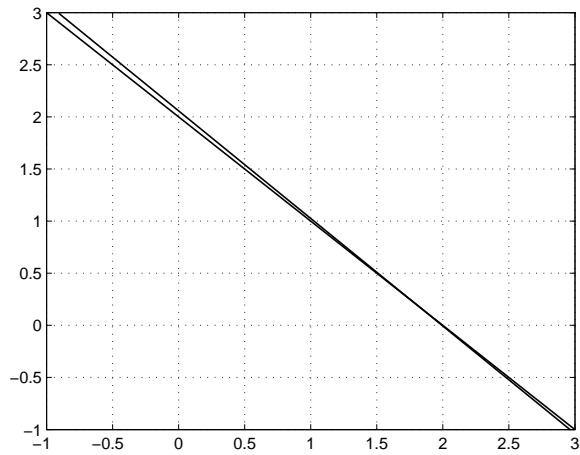


FIGURE 2. Thin line representation of ill-conditioning.

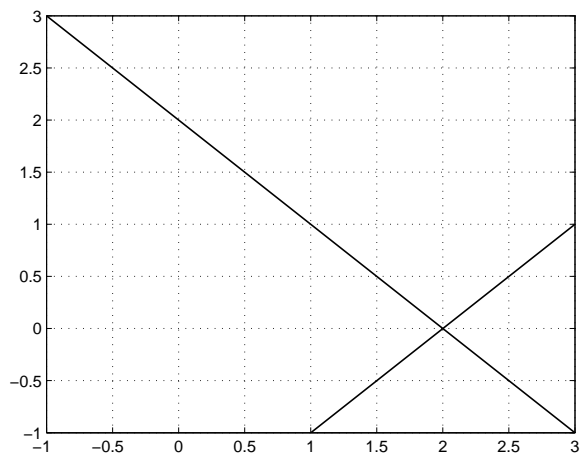


FIGURE 3. Resolution of ill-conditioning.

Let us try this idea on our system. Note that the condition number of the initial system is ≈ 4002 . If we replace the 2^{nd} equation of this system with one the line of which is perpendicular to that of this 2^{nd} equation, such as

$$x - y = b'_2$$

the condition number of the matrix of this new system is 1. However, we still do not know equation 2 explicitly; its RHS, \mathbf{b}' , must be calculated first, and in such a way that this new system be equivalent to the initial one. Write the new system as

$$x + y = 2,$$

$$x - y = b'_2$$

and denote it as

$$(1) \quad A' \mathbf{x} = \mathbf{b}'.$$

This gives

$$\mathbf{x} = A'^{-1} \mathbf{b}' = A^{-1} \mathbf{b}$$

with

$$(2) \quad \mathbf{b}' = A' A^{-1} \mathbf{b}.$$

Putting the values of A' , A^{-1} , and \mathbf{b} in (2), we find

$$\mathbf{b}' = \frac{\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1.0001 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \end{pmatrix}}{0.0001},$$

$$\begin{pmatrix} 2 \\ b'_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 2001 & -2000 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \end{pmatrix},$$

which gives us $b'_2 = 2$. With this RHS, the new system has the same solution as the initial one, and, importantly, it is well-conditioned. If we change the RHS to $(2, 2.0001)$ and solve, the solution is $(2, -0.0001)$ which is very close to that of the system before perturbation of the RHS. This simple motivation for our approach to improving solvability of SLE's through reducing the condition number of their matrices can be extended to any ill-conditioned SLE as follows. Consider, without loss of generality, the extreme case where the matrix of the given SLE is singular. This can be explained by the presence of at least two parallel equations leading to the loss of the matrix full-rank. Removing this parallelism by replacing one of the parallel lines with another which is not parallel to either of them, or indeed to any other equation in the system, will restore the matrix of the SLE to full-rank, making it non-singular. This means that it has a condition number less than infinity. It is therefore clear that removing the parallelism in a matrix reduces its condition number and hence ill-conditioning.

In the following, we will present a general approach to generating well-conditioned SLE's equivalent to square ill-conditioned ones of any dimension.

2. General approach

Consider the linear algebraic system $A\mathbf{x}=\mathbf{b}$ or $\sum a_{ij}x_j = b_i, i = 1, \dots, n$, and $j = 1, \dots, n$. Assume that the last equation is nearly similar (parallel) to any one of the other equations. We want to replace it with another one (perpendicular to it), resulting in an equivalent system, $A'\mathbf{x}=\mathbf{b}'$, of the form:

$$\sum_{j=1}^n a_{ij}x_j = b_i, i = 1, \dots, n - 1,$$

$$\sum_{j=1}^n a'_{nj}x_j = b'_n.$$

We, therefore, need to find the left-hand side row vector $a'_{n1}, a'_{n2}, \dots, a'_{nn}$, and the right-hand side b'_n .

2.1. Finding the left-hand side vector: Algorithm

Finding a suitable row vector $(a'_{n1}, a'_{n2}, \dots, a'_{nn})$, i.e., a vector that is orthogonal to $(a_{n1}, a_{n2}, \dots, a_{nn})^T$, is straightforward and can be done as follows.

- Algorithm.* 1. Generate random numbers $a'_{n1}, a'_{n2}, \dots, a'_{n(n-1)}$.
 2. Compute $a'_{nn} = -\frac{\sum_{j=1}^{n-1} a_{nj} \times a'_{nj}}{a_{nn}}$.

Note that the random numbers $a'_{n1}, a'_{n2}, \dots, a'_{n(n-1)}$, should be of the same magnitude as $a_{n1}, a_{n2}, \dots, a_{nn}$.

2.2. Computing the right-hand side last entry b'_n

Since $A\mathbf{x}=\mathbf{b}$ and $A'\mathbf{x}=\mathbf{b}'$ are equivalent, we have $\mathbf{x} = A^{-1}\mathbf{b} = A'^{-1}\mathbf{b}'$, where

$$A^{-1} = \frac{adj(A)}{|A|} \text{ and } A'^{-1} = \frac{adj(A')}{|A'|}$$

with

$$adj(A) = \begin{pmatrix} A_{11} & A_{21} & \dots & A_{n1} \\ \cdot & \cdot & \cdot & \cdot \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{pmatrix} \text{ and } adj(A') = \begin{pmatrix} A'_{11} & A'_{21} & \dots & A'_{n1} \\ \cdot & \cdot & \cdot & \cdot \\ A'_{1n} & A'_{2n} & \dots & A'_{nn} \end{pmatrix}.$$

A_{ij} and A'_{ij} represent the cofactors of corresponding elements of A and A' respectively, for all i and j . Since only the n^{th} row of the original system has been changed, it is clear that the cofactors of $a_{n1}, a'_{n1}, a_{n2}, a'_{n2}, \dots, a_{nn}, a'_{nn}$ are the same. Therefore $A_{n1} = A'_{n1}, A_{n2} = A'_{n2}, \dots, A_{nn} = A'_{nn}$. As the two systems are equivalent, b' can be calculated as in (2). More explicitly, we can write

$$\frac{\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \cdot & \cdot & \cdot & \cdot \\ a'_{n1} & a'_{n2} & \dots & a'_{nn} \end{pmatrix} \begin{pmatrix} A_{11} & A_{21} & \dots & A_{n1} \\ \cdot & \cdot & \cdot & \cdot \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{pmatrix} \begin{pmatrix} b_1 \\ \cdot \\ b_n \end{pmatrix}}{\Delta} = \begin{pmatrix} b_1 \\ \cdot \\ b'_n \end{pmatrix}$$

or

$$\frac{\begin{pmatrix} \sum_{j=1}^n a_{1j}A_{1j} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \sum_{j=1}^n a_{2j}A_{2j} & 0 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \sum_{j=1}^n a'_{nj}A_{1j} & \sum_{j=1}^n a'_{nj}A_{2j} & \cdot & \cdot & \cdot & \sum_{j=1}^n a'_{nj}A_{nj} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_n \end{pmatrix}}{\Delta} = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b'_n \end{pmatrix}$$

or

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \frac{\sum_{j=1}^n a'_{nj}A_{1j}b_1 + \sum_{j=1}^n a'_{nj}A_{2j}b_2 + \dots + \sum_{j=1}^n a'_{nj}A_{nj}b_n}{\Delta} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b'_n \end{pmatrix}.$$

Now, by identification we have

$$b'_n = \frac{\sum_{j=1}^n a'_{nj}A_{1j}b_1 + \sum_{j=1}^n a'_{nj}A_{2j}b_2 + \dots + \sum_{j=1}^n a'_{nj}A_{nj}b_n}{\Delta},$$

or

$$b'_n = \frac{\sum_{j=1}^n a'_{nj}(A_{1j}b_1 + A_{2j}b_2 + \dots + A_{nj}b_n)}{\Delta},$$

i.e.,

$$(3) \quad b'_n = \frac{\sum_{j=1}^n a'_{nj}(\sum_{i=1}^n A_{ij}b_i)}{\Delta}.$$

The problem, of course, with calculating b'_n in this way is the need to compute Δ or A^{-1} first, which we know are not easy to calculate because of the ill-conditioning of A . If we can avoid these calculations when computing \mathbf{b}' , then that would improve this formula a great deal. Note that we do not need to compute A^{-1} explicitly to find b'_n .

3. A better approach to computing \mathbf{b}'

The above method may not be as workable as one would wish it to be for the simple reason that it relies, although only implicitly, on A^{-1} , the inverse of the matrix of the original system to find the RHS of the equivalent system with a better behaved matrix. Here we suggest an alternative approach which relies on the A'^{-1} to do the same. This is much better since, numerically, A' is more stable. We follow the same methodology.

Consider (1) again. Write (2) as:

$$(4) \quad AA'^{-1}\mathbf{b}' = \mathbf{b}.$$

More explicitly, we can write

$$\frac{\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} A'_{11} & A'_{21} & \dots & A'_{n1} \\ \vdots & \vdots & \ddots & \vdots \\ A'_{1n} & A'_{2n} & \dots & A'_{nn} \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b'_n \end{pmatrix}}{\Delta'} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

or

$$\frac{\begin{pmatrix} \sum_{j=1}^n a_{1j}A'_{1j} & 0 & \dots & 0 \\ 0 & \sum_{j=1}^n a_{2j}A'_{2j} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{j=1}^n a_{nj}A'_{1j} & \sum_{j=1}^n a_{nj}A'_{2j} & \dots & \sum_{j=1}^n a_{nj}A'_{nj} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b'_n \end{pmatrix}}{\Delta'} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

i.e.,

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ \frac{\sum_{j=1}^n a_{nj}A'_{1j}b_1 + \sum_{j=1}^n a_{nj}A'_{2j}b_2 + \dots + \sum_{j=1}^n a_{nj}A'_{nj}b'_n}{\Delta'} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

By comparison we get

$$b_n = \frac{\sum_{j=1}^n a_{nj}A'_{1j}b_1 + \sum_{j=1}^n a_{nj}A'_{2j}b_2 + \dots + \sum_{j=1}^n a_{nj}A'_{nj}b'_n}{\Delta'}$$

or

$$b_n = \frac{\sum_{j=1}^n a_{nj}(A'_{1j}b_1 + A'_{2j}b_2 + \dots + A'_{(n-1)j}b_{n-1} + A'_{nj}b'_n)}{\Delta'}$$

Multiplying both sides by Δ' , we get

$$\Delta' b_n = \sum_{j=1}^n a_{nj}(A'_{1j}b_1 + A'_{2j}b_2 + \dots + A'_{(n-1)j}b_{n-1}) + \sum_{j=1}^n a_{nj}A'_{nj}b'_n$$

and

$$\sum_{j=1}^n a_{nj}A'_{nj}b'_n = \Delta' b_n - \sum_{j=1}^n a_{nj} \left(\sum_{i=1}^{n-1} A'_{ij} b_i \right).$$

Therefore

$$(5) \quad b'_n = \frac{\Delta' b_n - \sum_{j=1}^n a_{nj} \left(\sum_{i=1}^{n-1} A'_{ij} b_i \right)}{\sum_{j=1}^n a_{nj}A'_{nj}}.$$

Theorem 3.1. Expressions (3) and (5) of \mathbf{b}' are equivalent.

Proof. The proof is straightforward since both (3) and (5) are derived from the equivalence between $A\mathbf{x}=\mathbf{b}$ and $A'\mathbf{x}=\mathbf{b}'$ and the equation $A^{-1}\mathbf{b} = A'^{-1}\mathbf{b}'$. \square

4. The positive definite case

We have considered the case of a general system. It is interesting, however, to know if gains can be made if the system has a Symmetric Positive Definite matrix. This case is treated differently in [4]. Here, we can deal with it as follows. Write the system $A\mathbf{x}=\mathbf{b}$ in factored form as

$$(6) \quad LL^T\mathbf{x}=\mathbf{b}.$$

Then, extract the equivalent system from (6) as

$$(7) \quad L^T\mathbf{x}=\mathbf{b}_f,$$

where

$$\mathbf{b}_f = L^{-1}\mathbf{b}.$$

The system $A\mathbf{x}=\mathbf{b}$ and (7) are now equivalent and the condition number of L or L^T is the square root of the condition number of A . If the condition number of L or L^T is still considered to be large, then we can improve it by using the general approach of Section 2, and better still, that of Section 3.

5. A new approach

The above two approaches to compute the RHS of the equivalent but better behaved system to the one given initially, may not be as practical as one may wish because of their dependence on the inverse of the coefficient matrices. Here we suggest a new approach which does not depend on the inverse of either matrix. This approach relies on the bordering method of Faddeeva, [3]. Since the ill-conditioning is caused by the last row of the coefficient matrix, so the given matrix A can be regarded as a result of bordering the $(n-1) \times (n-1)$ submatrix of A with the offending row and the last column. We partition our system as follows assuming that the $(n-1) \times (n-1)$ submatrix of A is well behaved.

$$(8) \quad \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdot & a_{1,n-1} & a_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n-1,1} & a_{n-2,2} & \cdot & a_{n-1,n-1} & a_{n-1,n} \\ \hline a_{n1} & a_{n2} & \cdot & a_{n,n-1} & a_{nn} \end{array} \right) \begin{pmatrix} x_1 \\ \cdot \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \cdot \\ b_{n-1} \\ b_n \end{pmatrix}.$$

The above system can be written in a more compact form as follows

$$\begin{pmatrix} A_{n-1} & \mathbf{u} \\ \mathbf{v} & a_{nn} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ x_n \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{n-1} \\ b_n \end{pmatrix},$$

where A_{n-1} , \mathbf{u} , \mathbf{v} , \mathbf{b}_{n-1} are as per the partition in (8) above and

$$\mathbf{y} = \begin{pmatrix} x_1 \\ \cdot \\ x_{n-1} \end{pmatrix}.$$

By the bordering method of Faddeeva [3] we have

$$(9) \quad \mathbf{x} = \begin{pmatrix} \mathbf{x}_{n-1} \\ 0 \end{pmatrix} + \frac{b_n - \mathbf{v}\mathbf{x}_{n-1}}{\alpha_n} \begin{pmatrix} \mathbf{q}_{n-1} \\ 1 \end{pmatrix},$$

where $\mathbf{x}_{n-1} = A_{n-1}^{-1}\mathbf{b}_{n-1}$, $\mathbf{q}_{n-1} = -A_{n-1}^{-1}\mathbf{u}_{n-1}$ and $\alpha_n = a_{nn} - \mathbf{v}A_{n-1}^{-1}\mathbf{u} = a_{nn} + \mathbf{v}\mathbf{q}_{n-1}$.

Note that \mathbf{x}_{n-1} and \mathbf{y} should not be confused with each other. They are different as the former is the solution of the incomplete problem while the latter is the $n - 1$ entries of the solution to the original problem.

Now replacing the last row of (8), the corresponding equivalent system is as follows

$$(10) \quad \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdot & a_{1(n-1)} & a_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n-1,1} & a_{n-1,2} & \cdot & a_{n-1,n-1} & a_{n-1,n} \\ \hline a'_{n1} & a'_{n2} & \cdot & a'_{n,n-1} & a'_{nn} \end{array} \right) \begin{pmatrix} x_1 \\ \cdot \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \cdot \\ b_{n-1} \\ b'_n \end{pmatrix},$$

which can also be written as

$$\begin{pmatrix} A_{n-1} & \mathbf{u} \\ \mathbf{v}' & a'_{nn} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ x_n \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{n-1} \\ b'_n \end{pmatrix}.$$

Again using the bordering approach of Faddeeva, [3], the solution to the equivalent system is

$$(11) \quad \mathbf{x} = \begin{pmatrix} \mathbf{x}_{n-1} \\ 0 \end{pmatrix} + \frac{b'_n - \mathbf{v}'\mathbf{x}_{n-1}}{\beta_n} \begin{pmatrix} \mathbf{q}_{n-1} \\ 1 \end{pmatrix},$$

where

$$\beta_n = a'_{nn} - \mathbf{v}'A_{n-1}^{-1}\mathbf{u} = a'_{nn} + \mathbf{v}'\mathbf{q}_{n-1}.$$

But, since (8) and (10) are equivalent, from (9) and (11) we can write

$$(12) \quad b'_n = \mathbf{v}'\mathbf{x}_{n-1} + \frac{\beta_n}{\alpha_n}(b_n - \mathbf{v}\mathbf{x}_{n-1}).$$

Theorem 5.1. Expressions (9) and (11) are equivalent.

Proof. Putting the value of b'_n in (11) and simplifying we get the result in (9). To prove the other part, from (12)

$$b_n = \mathbf{v}\mathbf{x}_{n-1} + \frac{\alpha_n}{\beta_n}(b'_n - \mathbf{v}'\mathbf{x}_{n-1}).$$

Putting this value of b_n in (9) we get (11). □

5.1. Illustration 1

To illustrate, consider the partitioned system

$$(13) \quad \left(\begin{array}{ccc|c} 2 & 5 & 7 & 9 \\ 4 & 8 & 2 & 3 \\ 3 & 1 & 5 & -10 \\ \hline 2.0001 & 5.0001 & 7.0001 & 9.0001 \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 7 \\ 9 \\ 1 \end{pmatrix},$$

which in matrix form is

$$\begin{pmatrix} A_{n-1} & \mathbf{u} \\ \mathbf{v} & a_{nn} \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ x_n \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{n-1} \\ b_n \end{pmatrix},$$

where A_{n-1} , u , v , b_{n-1} have obvious values and $\kappa(A) = 6.295709e^{+005}$. By (9) we obtain

$$\mathbf{x} = 1.0e^{+003} \begin{pmatrix} 9.789437 \\ -5.220816 \\ -1.321646 \\ 1.753026 \end{pmatrix}.$$

Now replacing the last row of (13) by the one which is perpendicular to the first, the new system will be

$$(14) \quad \left(\begin{array}{ccc|c} 2 & 5 & 7 & 9 \\ 4 & 8 & 2 & 3 \\ 3 & 1 & 5 & -10 \\ 1 & 3 & 4 & -5 \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 7 \\ 9 \\ b'_n \end{pmatrix},$$

where b'_n is to be computed. Using equation (12), we find

$$b'_n = -1.9925e^{+004}.$$

Now solving (14) with this new RHS and using result (11)

$$\mathbf{x} = 1.0e^{+003} \begin{pmatrix} 9.789437 \\ -5.220816 \\ -1.321646 \\ 1.754026 \end{pmatrix},$$

with $\kappa(A') = 10.259081$. We can see that for the solution of the equivalent new system we only need a few vector operations because we already have all the relevant information to solve this new system. It is more stable compared to the original system, at least from the point of view of condition numbers.

5.2. Computational complexity

Looking at the bordering approach of Faddeeva, [3], again we can conclude that, provided the offending row is detected beforehand, the solution to an ill-conditioned $n \times n$ linear system of equations, the ill-conditioning of which is caused by the said offending row or equation, is equivalent to the solution of the $(n-1) \times (n-1)$ subsystem of linear equations with 2 right hand sides.

Therefore, computationally, the solution of the system is equivalent to solving the incomplete (well-behaved) subsystem of dimension $n-1$ with two right-hand sides. More explicitly, since the complexity of an n dimensional SLE is $O(n^3)$, and given that the product of an $n \times n$ matrix with an n -vector is $O(n^2)$, the time complexity of this approach is $O(n^3)$.

6. Extending the method to handle multiple near-parallel rows of matrix A

In Section 5 we proposed a method which deals with SLE with two offending equations (near-parallel). However, if we replace more than one row of A , i.e., $n-k$ rows, then we need to calculate the corresponding $n-k$ RHS components of the equivalent system. To do this we use the idea of block matrices and partition matrix A such that each block of A is a matrix of compatible dimensions. Suppose we have an SLE which is ill-conditioned and this ill-conditioning is caused by the last $n-k$ rows of the system. Then it can be written as

$$(15) \quad \left(\begin{array}{ccc|ccc} a_{11} & \cdot & a_{1k} & a_{1,k+1} & \cdot & a_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{k1} & \cdot & a_{kk} & a_{k,k+1} & \cdot & a_{kn} \\ \hline a_{k+1,1} & \cdot & a_{k+1,k} & a_{k+1,k+1} & \cdot & a_{k+1,n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & \cdot & a_{nk} & a_{n,k+1} & \cdot & a_{nn} \end{array} \right) \begin{pmatrix} x_1 \\ \cdot \\ x_k \\ x_{k+1} \\ \cdot \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \cdot \\ b_k \\ b_{k+1} \\ \cdot \\ b_n \end{pmatrix},$$

or, in a more compact form as

$$(16) \quad \begin{pmatrix} A_k & U \\ V & M \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{f} \end{pmatrix},$$

where A_k is a square matrix of dimension k , U a matrix of dimension $k \times (n-k)$, V a matrix of dimension $(n-k) \times k$ and M a matrix of dimension $(n-k) \times (n-k)$ and both A_k and M are regular. If we set

$$B = M - VA_k^{-1}U,$$

which is the Schur complement [1, 6] of A_k in A and is of the same dimension as M , then using the block bordering method of Brezinski [2] which is a generalization of the bordering method of Faddeeva, [3], the solution of system (15) is

$$(17) \quad \mathbf{x} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{Q} \\ \mathbf{I} \end{pmatrix} B^{-1}(\mathbf{f} - V\mathbf{y}),$$

where $\mathbf{0}$ is the zero vector. Now if we replace the last $n-k$ rows of system (15) which cause ill-conditioning, the new equivalent system will be

$$(18) \quad \left(\begin{array}{ccc|ccc} a_{11} & \cdot & a_{1k} & a_{1,k+1} & \cdot & a_{1n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{k1} & \cdot & a_{kk} & a_{k,k+1} & \cdot & a_{kn} \\ \hline a'_{k+1,1} & \cdot & a'_{k+1,k} & a'_{k+1,k+1} & \cdot & a'_{k+1,n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a'_{n1} & \cdot & a'_{nk} & a'_{n,k+1} & \cdot & a'_{nn} \end{array} \right) \begin{pmatrix} x_1 \\ \cdot \\ x_k \\ x_{k+1} \\ \cdot \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \cdot \\ b_k \\ b'_{k+1} \\ \cdot \\ b'_n \end{pmatrix}.$$

System (18) can be written in a compact form as

$$(19) \quad \begin{pmatrix} A_k & U \\ V' & M' \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{f}' \end{pmatrix}.$$

Set $B' = M' - V'A_k^{-1}U$, which is the Schur complement [1, 6] of A_k in A' and is of the same dimension as M' . However, \mathbf{f}' needs to be determined. Again using the block bordering method of [2], the solution of (19) will be

$$(20) \quad \mathbf{x} = \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{Q} \\ \mathbf{I} \end{pmatrix} B'^{-1}(\mathbf{f}' - V'\mathbf{y}).$$

Since (16) and (19) are equivalent, so from (17) and (20), after simplification, we get

$$(21) \quad \mathbf{f}' = V'\mathbf{y} + B'B^{-1}(\mathbf{f} - V\mathbf{y}).$$

Theorem 6.1. *Expressions (17) and (20) are equivalent.*

Proof. Putting the entries of \mathbf{f}' in (20) and simplifying we get the result in (17). To prove the other part, from (21) we have

$$\mathbf{f} = V\mathbf{y} + BB'^{-1}(\mathbf{f}' - V'\mathbf{y}).$$

Putting this value of \mathbf{f} in (17) gives result (20). The theorem follows. □

6.1. Illustration 2

Consider

$$A\mathbf{x} = \left(\begin{array}{ccc|cc} 2 & 5 & 7 & 9 & 1 \\ 4 & 8 & 2 & 3 & -1 \\ -1 & 0 & -4 & 1 & 2 \\ \hline 2.0001 & 5.0001 & 7.0001 & 9.0001 & 1.0001 \\ 4.0001 & 8.0001 & 2.0001 & 3.0001 & -1.0001 \end{array} \right) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 7 \\ 5 \end{pmatrix} = \mathbf{b},$$

with $\kappa(A) = 5.094029e^{+005}$. In a more compact form

$$\begin{pmatrix} A_k & U \\ V & M \end{pmatrix} \begin{pmatrix} \mathbf{y} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{f} \end{pmatrix},$$

where A_k, U, V, M, \mathbf{d} and \mathbf{f} have obvious values and the solution of the system is

$$\mathbf{x} = 1.0e^{+005} \begin{pmatrix} 1.2088 \\ -0.6485 \\ -0.2338 \\ 0.2735 \\ -0.0000 \end{pmatrix}.$$

The Schur complement of A_k in A is

$$B = 1.0e^{-003} \begin{pmatrix} 0.2194 & 0.2226 \\ 0.2194 & 0.0226 \end{pmatrix}.$$

To improve $\kappa(A)$, replace the last two rows of A to get

$$A' = \left(\begin{array}{ccc|cc} 2 & 5 & 7 & 9 & 1 \\ 4 & 8 & 2 & 3 & -1 \\ -1 & 0 & -4 & 1 & 2 \\ \hline 0 & 1 & 2 & 3 & -46 \\ 1 & -1 & 1 & 2 & 4 \end{array} \right),$$

which has condition number $\kappa(A') = 32.6028$, which is much smaller than $\kappa(A)$. We have

$$A' = \begin{pmatrix} A_k & U \\ V' & M' \end{pmatrix},$$

where A_k and U remain unchanged. We need \mathbf{b}' and, hence \mathbf{f}' in

$$\mathbf{b}' = \begin{pmatrix} \mathbf{d} \\ \mathbf{f}' \end{pmatrix},$$

which is to be determined from (21). Having the Schur complement of A_k in A' as

$$B' = \begin{pmatrix} -1.0806 & -47.1774 \\ 7.9355 & 7.2581 \end{pmatrix},$$

and putting values in (21) we get

$$\mathbf{f}' = 1.0e^{+005} \begin{pmatrix} -0.2956 \\ 2.1706 \end{pmatrix}.$$

Putting \mathbf{f}' in (20) we get

$$\mathbf{x} = 1.0e^{+005} \begin{pmatrix} 1.2088 \\ -0.6485 \\ -0.2338 \\ 0.2735 \\ -0.0000 \end{pmatrix}.$$

7. Conclusion

The near-parallelism of the hyperplanes corresponding to two or more equations of a SLE causes the matrix of this system to have a large condition number and to be generally ill-conditioned. By removing this parallelism, an equivalent SLE can be constructed which is well-conditioned. We have derived appropriate formulae and algorithms for this purpose and illustrated them. Although the computational complexity of our approach, as provided in Section 5.2 shows that it is only $O(n^3)$, it remains, however, to check how practical this approach is on large scale SLE's. This work is in progress.

References

- [1] C. Brezinski, *Other manifestations of the schur complement*, Linear Algebra Appl. **111** (1988), 231–247.
- [2] C. Brezinski, M. Redivo Zaglia, and H. Sadok, *A Breakdown-free Lanczos type algorithm for solving linear systems*, Numer. Math. **63** (1992), no. 1, 29–38.
- [3] V. N. Faddeeva, *Computational Methods of Linear Algebra*, Dover Publications, Inc., New York, USA, 1959.
- [4] H. J. Kim, Kyung Choi, H. B. Lee, H. K. Jung, and S. Y. Hahn, *A new algorithm for solving ill-conditioned linear system*, IEEE Transactions on Magnetics **32** (1996), no. 3, 1373–1376.
- [5] J. R. Rice, *Matrix Computations and Mathematical Software*, McGraw-Hill, New York, 1981.
- [6] F. Zhang and Roger A. Horn, *Basic Properties of the Schur Complement*, In F. Zhang, editor, *The Schur Complement and its applications*, pages 17–46. Springer, New York, 2005.

MUHAMMAD FAROOQ
DEPARTMENT OF MATHEMATICAL SCIENCES
THE UNIVERSITY OF ESSEX, WIVENHOE PARK
COLCHESTER CO4 3SQ, UK
E-mail address: mfarooj@essex.ac.uk

ABDELLAH SALHI
DEPARTMENT OF MATHEMATICAL SCIENCES
THE UNIVERSITY OF ESSEX, WIVENHOE PARK
COLCHESTER CO4 3SQ, UK
E-mail address: as@essex.ac.uk