Journal of Engineering Mathematics 21: 47–58 (1987)

Diakoptics as a general approach in engineering

P.W. Aitchison
Department of Applied Mathematics, University of Manitoba, Winnipeg, Manitoba, Canada, R3T 2N2

Received 2 September 1986; accepted 6 October 1986

Abstract

A method is described for reducing the solution of a matrix equation of linear equations, $Mz = d$, to that of a number of simpler equations. The method is one mathematical version of diakoptics, and can be used to solve engineering problems involving linear equations without specialization to each individual application. The method has the usual advantages of diakoptics: different parts of the problem can be solved on different computers (distributed computation), decreased computer storage requirements in fast storage, applicability to some types of parallel processors and decreased computation time in some circumstances. Consideration is given to reducing M to specific forms including block triangular, banded and the usual block diagonal. Consideration is also given to the case where M is not square and a new application is given to solving least-squares problems.

1. Introduction and notation

A system of equations, $Mz = d$, is sometimes in a form which could be solved by a fast solution algorithm, if there were not some 'disturbing' non-zero entries in M. If these 'disturbing' entries are placed in a matrix B, and $M = A + B$, then using the technique of tearing, the solution of $Mz = d$ can be related to the solution of the simpler system $Az = d$ by equations whose complexity depends on the rank of B.

The technique was originated in electrical engineering by Kron [31], and usually goes under the name diakoptics. Applications of diakoptics in engineering were quite slow to develop, but now there are numerous descriptions and announcements at conferences of applications, and of extensions of various forms of the method. Most of these are connected to specific applications, [5,7,11–17,20,23,24,27,28,31–35,37,39,41,42]. Diakoptics has been considered for use in parallel computation or in distributed computer systems [10,30]. A few authors have considered the basic mathematical techniques unconnected with specific applications [1,2,4,6,8,9,21,22,29,40], but only for the case where M is non-singular. These mathematical results are mostly related, but apparently not [1], to the formulae for the inverse of a sum of matrices, as described in [25].

Section 2 describes a simple mathematical form of the result, and summarizes some of the established results. Section 3 considers specific forms of the reduced matrix A, with a simple numerical example. Section 4 considers the cases where M is not square or not non-singular and it includes a new diakoptic least-squares approach.

The letters A, B, M, X, Y, etc. denote matrices, while letters z, d, e, etc. denote column matrices or column vectors, and 1 is used to denote an identity or unit matrix of appropriate size. If B has order $m \times n$ and it has r non-zero columns, then B^r denotes the order $m \times r$ matrix containing all the non-zero columns of B. z_r, $(A^{-1})_r$, etc. denote matrices of r selected rows from z and A^{-1}; these r rows are usually the rows in the same row position as are the r non-zero columns in B. Similarly, $(A^{-1})_{n-r}$ and z_{n-r} denote the $(n-r) \times n$ and $(n-r) \times 1$ matrices of the rows of A^{-1} and z which are not contained in $(A^{-1})_r$ and z_r. Partitioned matrices are denoted as in $[A \mid X]$ or $\left[\frac{A}{F}\right]$. The blanks in the matrices used for numerical illustrations indicate zeros. The abbreviation n.s. is used for 'non-singular', A^* represents the conjugate transpose of A, and A^{-1} represents the inverse of A and z^{T} is the transpose of z.

2. Basic results on tearing

Suppose the $n \times n$ matrix M can be decomposed as $M = A + B$, where A is non-singular and B has exactly r non-zero columns which are the columns of the $n \times r$ matrix B^r. The following results (1) and (2) are proved in the Appendix.

$$M \text{ is n.s. if and only if } \left[1 + (A^{-1})_r B^r\right] \text{ is n.s.,} \tag{1}$$

where $(A^{-1})_r$ is the $r \times n$ matrix of those rows of A^{-1} with the same row number as the columns of B^r in B.

The system $Mz = d$ is equivalent to (both have the same solution or both have no solutions)

$$\begin{aligned} &\left[1 + (A^{-1})_r B^r\right] z_r = (A^{-1})_r d, \\ &z_{n-r} = (A^{-1})_{n-r}(d - B^r z_r), \end{aligned} \tag{2}$$

where z_r is the vector of variables corresponding to the non-zero columns of B (the connecting variables) and z_{n-r} is the vector of remaining variables.

The efficient solution of the system (2) would be carried out by successively proceeding along the following steps (A)–(E) (using any appropriate linear equation solution algorithm).

(A) Solve for the vector x: $Ax = d$, so that $x = A^{-1}d$.
(Note that solving equations is much more efficient than calculating A^{-1}).

(B) Solve for $n \times r$ matrix X: $AX = B^r$, so that $X = A^{-1}B^r$.
(This means solve the equation for each right-hand side of B^r and put together all the solutions as the columns of X).

(C) Solve for z_r: $(1 + X_r)z_r = x_r$, where X_r and x_r consist of rows of X and x corresponding to the variables in z_r.
Since $(1 + X_r) = 1 + (A^{-1})_r B^r$ and $x_r = (A^{-1})_r d$, then
$z_r = (1 + (A^{-1})_r B^r)^{-1}(A^{-1})_r d$.

(D) Calculate $z_{n-r} = x_{n-r} - X_{n-r} z_r$, so that $z_{n-r} = (A^{-1})_{n-r}(d - B^r z_r)$ as in (2).

(E) Construct z from z_r and z_{n-r}.
Here x_r, X_r, x_{n-r} and X_{n-r} are the rows of x and X corresponding to the rows of z contained in z_r and z_{n-r}.

(3)

The advantages of this method arise because the solutions of the matrix equations in (A) and (B) will be very fast if A has a special structure such as block diagonal or banded. In particular, if A is block diagonal with $A = \mathrm{diag}[A_1, A_2, \ldots, A_p]$, then the solution of the equation $Ax = d$, for example, can be obtained by solving the much smaller systems $A_i x_i = d_i$, for $i = 1, \ldots, p$, where d_i is the appropriate part of d. The complete solution vector then is the composite

$$x = \begin{bmatrix} x_1 \\ \hline x_2 \\ \hline \vdots \\ \hline x_p \end{bmatrix}.$$

The third equation, $(1 + X_r)z_r = x_r$, has an $r \times r$ coefficient matrix which will be small provided r is small.

If M is n.s., then the inverse of M may be calculated in terms of A^{-1} by the formula which is proved in the Appendix:

$$M^{-1} = A^{-1}\{1 - B^r[1 + (A^{-1})_r B^r]^{-1}(A^{-1})_r\}. \tag{4}$$

This inverse may be efficiently calculated successively as follows:

$$\begin{array}{ll} \text{solve for } n \times n \text{ matrix } Y: & AY = 1, \\ \text{solve for } r \times n \text{ matrix } Z: & (1 + Y_r B^r)Z = Y_r, \\ \text{calculate} & M^{-1} = Y(1 - B^r Z). \end{array} \tag{5}$$

Here Y_r consists of the rows of Y corresponding to the rows of z in z_r.

The result (1) and result (2), when M is n.s., are special cases of more general results contained in Bunch and Rose [9, §4], Steward [40], and Bückner [8, pp 448–449] and is related to Kron's method [31] and to the Sherman–Morrison-Woodbury formula for the inverse of the sum of two matrices [25]. The result (4) is a variation on the Sherman-Morrison-Woodberry formula and is derived in the Appendix from equation (24) of [25] where the original references are cited.

Notice, however, that (2) leads to a different formula for M^{-1} which can be obtained from the Sherman-Morrison-Woodbury formula with some difficulty, namely (using partitioned matrix notation):

$$M^{-1} = \left(\frac{(A^{-1})_{n-r}}{0}\right) + \left(\frac{-(A^{-1})_{n-r}B^r}{1}\right)[1 + (A^{-1})_r B^r]^{-1}(A^{-1})_r. \tag{6}$$

This formula assumes that the columns of M are ordered so that the r columns of B^r occur last and the remaining columns occur first in the column ordering from left to right, and similarly for the row ordering. There will be computational advantages in using (6) rather than (4) in some circumstances.

Generalizations of results (1) to (5) are given by (1′) to (5′) below.

Let $M = A + B$, where A is n.s., and suppose B has rank r (but not necessarily only r non-zero columns). B has full-rank factorisation $B = FDG$, where F has order $n \times r$, G has order $r \times n$ and both have rank r, and D is a full-rank diagonal matrix (see, for example, Ben-Israel and Greville [3, p. 22]); D can be the identity matrix in order to simplify this.

$$M \text{ is n.s. if and only if } D^{-1} + GA^{-1}F \text{ is n.s.} \tag{1′}$$

The system $Mz = d$ is equivalent to the system

$$\begin{array}{l} (D^{-1} + GA^{-1}F)y_r = GA^{-1}d, \\ z = A^{-1}d - A^{-1}Fy_r, \end{array} \tag{2′}$$

where y_r is an $r \times 1$ vector of auxiliary variables related to z by $y_r = Gz$. An efficient solution is given by successively solving,

$$\begin{array}{ll} \text{solve for } x: & Ax = d, \\ \text{solve for } X: & AX = F, \\ \text{solve for } y_r: & (D^{-1} + GX)y_r = Gx, \\ \text{calculate } z: & z = x - Xy_r. \end{array} \tag{3′}$$

Similarly,

$$M^{-1} = A^{-1}\left[1 - F(D^{-1} + GA^{-1}F)^{-1}GA^{-1}\right]. \tag{4'}$$

An efficient algorithm for calculating M^{-1} is to successively solve:

$$\begin{array}{ll} \text{solve for the } n \times n \text{ matrix } Y: & AY = 1, \\ \text{solve for the } r \times n \text{ matrix } Z: & (1 + GYF)Z = GY, \\ \text{calculate} & M^{-1} = Y(1 - FZ). \end{array} \tag{5'}$$

The proofs of (1′) and (2′), being very similar to the proofs of (1) and (2), are omitted, and the other results may be related to previously cited references.

If M is singular and possibly not square, then formulae (4) and (4′) can be adapted to give the generalised inverse matrix of M, which here means any matrix, denoted M^-, satisfying $MM^-M = M$ (see [3] for more details). Equation (4′) becomes

$$M^- = A^-\left[1 - FD(D + DGA^-FD)^- DGA^-\right] \tag{7}$$

where A^- and $(D + DGA^-FD)^-$ are generalised inverses of the possibly singular matrices A and $(D + DGA^-FD)$. This holds provided the row and column space of $B = FDG$ are subsets of the row and column spaces of A. The proofs of these formulae are without major difficulty and are omitted.

The role of generalized inverses in equations is that $Mz = d$ has a solution $z = M^-d$, provided a solution exists, and has general multiple solutions given by $z = M^-d + (1 - M^-M)p$, for arbitrary vector p. (see [3]).

3. Exploiting a 'near'-banded or 'near'-block triangular structure

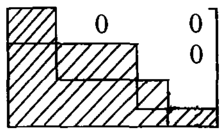
According to Bunch and Rose [9, §4,5] single-element tearing for symmetric matrices, M, is not advantageous over other methods (and may be worse) unless additional structural information is available for the matrix A, where $M = A + B$, such as A being banded, triangular, block triangular or of other special form. These special forms of A are considered in more detail in this section.

The opportunities of tearing in these cases include the following: (a) if A is block diagonal or block triangular then it enables very large problems where M has appropriate structure to be broken up into a number of simpler problems; (b) it enables a problem where M has only a small number of non-zero entries outside of the special structure, such as banded, to be converted into a problem with matrix having the special structure or conversion to a better form of the special structure, such as reduced band width; (c) when A is block diagonal, it enables different parts of the problem to be solved on entirely separate and independent computers (steps (A) and (B) of (3)), with only the final parts requiring assembly of these separate parts on a central computer (solving the $r \times r$ system in step (C) of (3) and other minor calculations in (D) and (E)). In other words, there is substantial parallel computation possible.

An example of type (b) is given, with operational counts, in example 6, Section (4) of Bunch and Rose [9]. In order for tearing to be advantageous, even in these cases where M has special structure, the number, r, of non-zero columns in B, where $M = A + B$, must be relatively small compared with the size of M. That is, the solution of the $r \times r$ auxiliary systems, $(1 + X_r)z_r = x_r$ (see equation (3)), must not approach the complexity of the direct solution of the original problem $Mz = d$ (Note that $1 + X_r$ may not be as sparse as was M).

The problem of solving the system $Az = d$ or of finding A^{-1} when A is a non-singular banded matrix (for some non-negative integer b, $-b \leqslant i-j \leqslant b$ for all non-zero entries a_{ij} of A) has been investigated in great detail, especially for tri-diagonal systems. The solution is very efficient because there is no fill-in of L during an LU factorization and only $5n$ multiplications and divisions are required where A is $n \times n$ (see for example [26, §7.1]). For one method of putting a matrix into a banded form, by rearranging rows and columns, see [18] or [19]. Further information is in [38]. For clarification purposes only, Examples 1 and 2 below provide simple examples of the band-width reduction process and of the reduction of the system to two smaller systems.

Another special case of $Az = d$, with a solution composed of a series of smaller problems, occurs when $Az = d$ has block triangular structure ($a_{jj} = 0$ for all $j > i$, *or* for all $i > j$, except for those entries lying in some square contiguous non-overlapping diagonal block submatrices):

$A =$ 

The blocks may be single diagonal entries, and triangular matrices are special cases of this form. The solution algorithm involves successively solving a system, whose coefficient matrix is the diagonal block, for the variables associated with that block. The order of solving the blocks is from upper left to lower right if A is lower block triangular and the reverse order if A is upper block triangular. Duff and Reid [15] give an implementation of the fast algorithm of Tarjan for converting a matrix into block triangular form and Steward [40] gives a not-so-fast algorithm for choosing which elements to tear in order to reduce large diagonal blocks to a number of smaller blocks in a block triangular matrix. The algorithms are illustrated in Steward's paper, and the tearing process is a straightforward application of equation (2).

Example 1. Decreasing the bandwidth in $Mz = d$, when M is a banded matrix. Let M be given, and A, B^r be chosen as follows:

$$M = \begin{bmatrix} 2 & 1 & & & & & \\ 3 & 1 & 1 & & & & \\ & -1 & 1 & 1 & & & \\ & & & 2 & 1 & & \\ & & & 1 & 2 & & \\ & & & & 1 & 1 & 1 \\ & & & 1 & 2 & -1 & 1 \end{bmatrix}, \quad d = \begin{bmatrix} 2 \\ 1 \\ -2 \\ 1 \\ 2 \\ 0 \\ 4 \end{bmatrix},$$

and choose

$$A = \begin{bmatrix} 2 & 1 & & & & & \\ 3 & 1 & 1 & & & & \\ & -1 & 1 & 1 & & & \\ & & & 2 & 1 & & \\ & & & 1 & 2 & & \\ & & & & 1 & 1 & 1 \\ & & & & & -1 & 1 \end{bmatrix}, \quad B^r = \begin{bmatrix} & \\ & \\ & \\ & \\ & \\ & \\ 1 & 2 \end{bmatrix},$$

so

$$z_r = [z_4, z_5]^{\mathrm{T}}, \quad z_{n-4} = [z_1\ z_2\ z_3\ z_6\ z_7]^{\mathrm{T}}.$$

Using (2) and (3), and using LU-factorization or any other algorithm, $Ax = D$ and $AX = B^r$ have solutions

$$x = [1 \quad 0 \quad -2 \quad 0 \quad 1 \quad -\tfrac{5}{2} \quad \tfrac{3}{2}]^{\mathrm{T}}, \quad X = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -\tfrac{1}{2} & \tfrac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}^{\mathrm{T}}.$$

Hence $(1 + X_r)z_r = x_r$ (see equation (3)) becomes $z_r = x_r = [0 \ 1]^{\mathrm{T}}$.

$$z_{n-r} = x_{n-r} - X_{n-r}z_r \text{ becomes } z_{n-r} = [1 \quad 0 \quad -2 \quad -\tfrac{5}{2} \quad \tfrac{3}{2}]^{\mathrm{T}} - [0 \quad 0 \quad 0 \quad -1 \quad 1]^{\mathrm{T}}.$$

So

$$z_{n-r} = [1 \quad 0 \quad -2 \quad -\tfrac{3}{2} \quad \tfrac{1}{2}]^{\mathrm{T}},$$

and the complete solution is

$$z = [1 \quad 0 \quad -2 \quad 0 \quad 1 \quad -\tfrac{3}{2} \quad \tfrac{1}{2}]^{\mathrm{T}}.$$

Example 2. Separating $Mz = b$, where M is a banded matrix, into two similar problems and an auxiliary problem.

Take M and d as above and choose

$$A = \begin{bmatrix} 2 & 1 & & & & & \\ 3 & 1 & 1 & & & & \\ & -1 & 1 & 1 & & & \\ & & & 2 & & & \\ & & & & 2 & & \\ & & & & 1 & 1 & 1 \\ & & & & 2 & -1 & 1 \end{bmatrix}, \quad B^r = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad z_r = \begin{bmatrix} z_4 \\ z_5 \end{bmatrix}, \quad z_{n-r} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_6 \\ z_7 \end{bmatrix}.$$

$Ax = d$ and $AX = B^r$ each separate into two parts:

$$A_1x_1 = d_1, \quad A_2x_2 = d_2, \quad A_1X_1 = B_1^r, \quad A_2X_2 = B_2^r,$$

where

$$A_1 = \begin{bmatrix} 2 & 1 & & \\ 3 & 1 & 1 & \\ & -1 & 1 & 1 \\ & & & 2 \end{bmatrix}, \quad A_2 \begin{bmatrix} 2 & & \\ 1 & 1 & 1 \\ 2 & -1 & 1 \end{bmatrix}, \quad d_1 = \begin{bmatrix} 2 \\ 1 \\ -2 \\ 1 \end{bmatrix}, \quad d_2 = \begin{bmatrix} 2 \\ 0 \\ 4 \end{bmatrix}, \quad B_1^r = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix},$$

$$B_2^r = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix},$$

Solutions by any convenient algorithm are

$$x_1 = [\tfrac{1}{2} \ 1 \ -\tfrac{3}{2} \ \tfrac{1}{2}]^{\mathrm{T}}, \quad x_2 = [1 \ -\tfrac{3}{2} \ \tfrac{1}{2}]^{\mathrm{T}}, \quad X_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\tfrac{1}{2} & 1 & \tfrac{1}{2} & \tfrac{1}{2} \end{bmatrix}^{\mathrm{T}},$$

$$X_2 = \begin{bmatrix} \tfrac{1}{2} & -\tfrac{1}{4} & -\tfrac{1}{4} \\ 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}.$$

$$(1 + X_r)z_r = x_r \text{ becomes } \begin{bmatrix} 1 & \tfrac{1}{2} \\ \tfrac{1}{2} & 1 \end{bmatrix} z_r = \begin{bmatrix} \tfrac{1}{2} \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} z_4 \\ z_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

and

$$z_{n-r} = x_{n-r} - X_{n-r}z_r \text{ becomes } z_{n-r} = \begin{bmatrix}\tfrac{1}{2} & 1 & -\tfrac{3}{2} & -\tfrac{3}{2} & \tfrac{1}{2}\end{bmatrix}^{\mathrm{T}} - \begin{bmatrix}-\tfrac{1}{2} & 1 & \tfrac{1}{2} & 0 & 0\end{bmatrix}^{\mathrm{T}},$$

or $z_{n-r} = [1 \quad 0 \quad -2 \quad -\tfrac{3}{2} \quad \tfrac{1}{2}]^{\mathrm{T}}$. The complete solution is $z = [1 \quad 0 \quad -2 \quad 0 \quad 1 \quad -\tfrac{3}{2} \quad \tfrac{1}{2}]^{\mathrm{T}}$.

4. Tearing when M is not square, and least squares

Extensions of the tearing process can be used in the cases where M is not square. Consider first the system $Mz = d$ where M has order $m \times n$ with $m \leqslant n$. Assume that, after rearranging columns if necessary, the first m columns of M are decomposed as $A + B$, so that in partitioned form $M = [A + B \mid E]$, and correspondingly $z = [z_m \mid z_{n-m}]^{\mathrm{T}}$. If B has r non-zero columns where r is small compared with m, while A is n.s. and has a special structure which allows quick solutions of equations of which it is coefficient matrix, then a useful reduction formula similar to equation (2) is as follows:

$$\begin{aligned} &Mz = d, \text{ or } [A + B \mid E] = d \text{ is equivalent to } \left[1 + (A^{-1})_r B^r\right] z_r = (A^{-1})_r (d - Ez_{n-m}),\\ &z_{m-r} = (A^{-1})_{m-r}(d - B^r z_r - Ez_{n-m}), \end{aligned} \tag{8}$$

where B^r consists of the r non-zero columns of B, $(A^{-1})_r$ and z_r are the r rows of A^{-1} and z_m corresponding to the non-zero columns of B, while $(A^{-1})_{m-r}$ and z_{m-r} are the remaining rows of A^{-1} and z_m. A generalization of result (8) in the form of equation (2′) is also possible but is not given here.

Notice that $A + B$ is n.s. if and only if $[1 + (A^{-1})_r B^r]$ is n.s., by equation (1), in which case z_{n-m} becomes simply a vector of arbitrary parameters in (8). If $(1 + (A^{-1})_r B^r)$ is singular, or it is not known that $1 + (A^{-1})_r B^r$ is n.s., then the solution becomes more complicated because the vector z_{n-m} must now be chosen so that any linear dependence of the rows of $1 + (A^{-1})_r B^r$ is also a linear dependence of $(A^{-1})_r(d - Ez_{n-m})$.

An efficient solution algorithm of (8) for the case where $A + B$, and so $1 + (A^{-1})_r B^r$, are n.s. is a simple adaption of (3). A proof of (8) is given in the Appendix.

This completes the consideration of the case of fewer equations than variables. In a similar way formulae can be given for the solution where there are more equations than variables. However, in such large equation systems derived from a physical model, using data from the actual physical situation, the system would almost always be inconsistent. Such inconsistencies can result from inaccuracies in the data and because of the imperfect fit of the mathematical model with the physical situation being modelled. This limits the usefulness of this method. The least-squares approximate solution of $Mz = d$ is usually used in such cases and can be found using well known theory. That is, if M is $m \times n$ with $m \geqslant n$ then the least-squares solution of $Mz = d$ is given by a solution of the system with symmetric matrix M^*M:

$$(M^*M)z = M^*d, \tag{9}$$

where M^* is the conjugate transpose of M. If M has rank n, then M^*M is non-singular, so that the least-squares solution is unique, and the solution of (9) by LU-decomposition or other usual method is particularly simple since the method is stable under any feasible pivoting order, and requires fewer operations than the non-symmetric case. In many cases, a better solution method for least-squares problems is by singular-value decomposition as discussed in [36].

Unfortunately the coefficient matrix M^*M in (9) usually has a lot of fill-in and is not suitable for diakoptic solution methods. However, a diakoptical solution can be carried out in two stages as follows. Assume M is $m \times n$ with $m \geqslant n$ and $M = A + B$, where A has a structure as shown with nonzero blocks, D_i, which have at least as many rows as columns:

$$\begin{bmatrix} D_1 & & & \\ & D_2 & & 0 \\ & & \ddots & \\ & 0 & & D_p \end{bmatrix} \tag{10}$$

Assume B has only r non-zero columns which together form the $m \times r$ matrix B^r. Hence (9) can be written

$$M^*Az + M^*Bz = M^*d, \quad M^*A \text{ is now square.}$$

The diakoptical solution using formulae (2) is given by

$$\left(1 + \left[(M^*A)^{-1}\right]_r M^*B^r\right)z_r = \left[(M^*A)^{-1}\right]_r M^*d, \tag{11}$$

and

$$z_{n-r} = \left[(M^*A)^{-1}\right]_{n-r}(M^*d - M^*B^r z_r). \tag{12}$$

The only additional difficulty in solving (11) and (12) is in dealing with $(M^*A)^{-1}$, and this can be dealt with by working with a transposed form of $(M^*A)^{-1}$, as follows:

$$(M^*A)^{-1} = \left([(M^*A)^*]^{-1}\right)^* \quad \left(\text{since for any matrix } C,\ (C^*)^{-1} = (C^{-1})^*\right)$$

$$= \left[(A^*M)^{-1}\right]^*.$$

Consider, temporarily, the unconjugated matrix $(A^*M)^{-1}$,

$$(A^*M)^{-1} = [A^*(A+B)]^{-1} = (A^*A + A^*B)^{-1}.$$

Note that A^*B will have the same zero columns as B, and $(A^*B)^r = A^*B^r$. Using formula (4) (or (6) can be used in a similar way)

$$(A^*M)^{-1} = (A^*A)^{-1}\left\{1 - A^*B^r\left(1 + \left[(A^*A)^{-1}\right]_r A^*B^r\right)^{-1}\left[(A^*A)^{-1}\right]_r\right\}, \tag{13}$$

and so, taking the conjugate, using the fact that $([(A^*A)^{-1}]_r)^* = [(A^*A)^{-1}]^r$,

$$(M^*A)^{-1} = \left[(A^*M)^{-1}\right]^*$$

$$= \left\{1 - \left[(A^*A)^{-1}\right]^r\left(1 + [B^r]^*A\left[(A^*A)^{-1}\right]^r\right)^{-1}[B^r]^*A\right\}(A^*A)^{-1}. \tag{14}$$

Substitution of (14) into (11) and (12) gives the complete result. Notice that, in (14), $(A^*A)^{-1}$ can be calculated in terms of the separate blocks, D_i, of A, since by (10), $A^*A = \operatorname{diag}[D_1^*D_1, D_2^*D_2, \ldots, D_p^*D_p]$ (i.e. the block diagonal matrix with blocks $D_1^*D_1, \ldots, D_p^*D_p$) and $(A^*A)^{-1} = \operatorname{diag}[(D_1^*D_1)^{-1}, \ldots, (D_p^*D_p)^{-1}]$.

The rather complicated formulae (11), (12), (13) and (14) have a relatively simple solution algorithm as follows, where M has size $m \times n$ and $A_1 = D_1^* D_1$, $A_2 = D_2^* D_2, \ldots, A_p = D_p^* D_p$:
(a) calculate $A_i = D_i^* D_i$, $B_i = A_i^{-1}$ and $C_i = D_i^*[B^r]_i$ for $i = 1, \ldots, p$, where $[B^r]_i$ is the block of rows of B^r corresponding in position to the rows of D_i in A;
(b) calculate the matrices E_i of size $n_i \times r$, where n_i is the number of columns of D_i,

$$E_i = B_i C_i, \text{ for } i = 1, \ldots, p, \left(\text{so that } E_i = (D_i^* D_i)^{-1} D_i^* [B^r]_i\right);$$

Form the composite matrix, $E = (A^*A)^{-1}A^*B^r$ of size $n \times r$

$$E = \left[\begin{array}{c} E_1 \\ \hline E_2 \\ \hline \vdots \\ \hline E_p \end{array}\right];$$

(c) solve the system of equations for the $r \times r$ matrix

$$[1 + (E_r)^*] F = M^* B^r,$$

$$[1 + (E_r)^*] f = M^* d$$

(so $F = (1 + (B^r)^* A[(A^*A)^{-1}]^r)^{-1} M^* B^r$, and r-vector $f = (1 + (B^r)^* A[(A^*A)^{-1}]^r)^{-1} M^* d$);
(d) solve the $r \times r$ system for z_r to give the formula of equation (11),

$$(1 + F) z_r = f;$$

(e) calculate, using equation (12), the remaining z-variables

$$z_{n-r} = (E_{n-r})^* (M^*d - M^*B^r z_r);$$

(f) form the solution vector z from z_r and z_{n-r}.

The solution time of a state-estimation problem using formulae (11) to (14) using steps (a) to (f), is most dependent on the calculation of the inverses of the blocks A_i in step (a). Each such block of size $d \times d$ requires approximately d^3 multiplications so that p equal blocks would require about $p(n/p)^3 = n^3/p^2$ such operations. In comparison direct solution by LU-decomposition of the full $n \times n$ system of equations (9) requires about $(1/3)n^3$ operations, plus the multiplication required to form the products M^*M and M^*d of about n^3 operations. However, sparse-matrix methods or other more efficient least-squares solutions show that computation time-savings by the diakoptics method may be small or non-existent in a least-squares solution on a single computer.

However, the method can save computer storage and can be very effective on a distributed computer system where the individual calculations for $i = 1, 2, \ldots, p$ in steps (a) and (b) are carried out on different computers, perhaps in different areas of an electrical system. The information from the separate computers would only have to be brought together to complete the solution in steps (c), (d), (e), and (f). However, the last four steps involve a relatively small part of the solution computation provided r is not too large. The most computation occurs in step (c) when solving $r + 1$ systems of linear equations, each with the same $r \times r$ coefficient matrix. If LU-decomposition were used then this would involve roughly $(1/3)r^3$ multiplications for the LU-decomposition and roughly r^3 for the forward and backwards substitutions giving a total of $(4/3)r^3$. There is also another $r \times r$ system of equations in step (4) using about another $r^3/3$ operations.

Appendix

Proof of (1)

Assuming A is n.s., $M = A + B$ is n.s. if and only if $1 + A^{-1}B$ is n.s., if and only if $1_r + (A^{-1})_r B$ has linearly independent rows, (where 1_r and $(A^{-1})_r$ are the rows of 1 and A^{-1} corresponding to the nonzero columns of B) since $1 + A^{-1}B$ differs from $1_r + (A^{-1})_r B$ by having extra rows each with a single non-zero entry in a zero column of $1_r + (A^{-1})_r B$. Finally, $1_r + (A^{-1})_r B$ has linearly independent rows if and only if $1 + (A^{-1})_r B^r$ is n.s., since the two differ only by zero columns, thus proving (1).

Proof of (2)

When M is singular or n.s. $Mz = d$ or $(A + B)z = d$ is equivalent to $z + A^{-1}Bz = A^{-1}d$, and using $Bz = B^r z_r$, and separating into subsets of r rows and of $(n - r)$ rows, is equivalent to

$$z_r + (A^{-1})_r B^r z_r = (A^{-1})_r d,$$
$$z_{n-r} + (A^{-1})_{n-r} B^r z_r = (A^{-1})_{n-r} d,$$

from which the result (2) follows.

Proof of (4)

$$(A + B)^{-1} = (A + BA^{-1}A)^{-1} = (A^{-1})(1 + BA^{-1})^{-1} = A^{-1}\left[1 + B^r (A^{-1})_r\right]^{-1},$$

and by formula (24) of [25], with $A = 1$, $U = 1$, $B = B^r$, $V = (A^{-1})_r$,

$$(A + B)^{-1} = A^{-1}\left\{1 - B^r\left[1 + (A^{-1})_r B_r\right]^{-1}(A^{-1})_r\right\}.$$

Proof of (8)

The system $Mz = [A + B \mid E]z = d$ is equivalent to

$$A^{-1}[A + B \mid E]\left[\frac{z_m}{z_{n-m}}\right] = A^{-1}d,$$

and to $(1 + A^{-1}B)z_m + A^{-1}Ez_{n-m} = A^{-1}d$, and to $z_m + A^{-1}B^r z_r + A^{-1}Ez_{n-m} = A^{-1}d$, since $Bz_m = B^r z_r$. This is equivalent to the r rows given by

$$z_r + (A^{-1})_r B^r z_r + (A^{-1})_r E z_{n-m} = (A^{-1})_r d,$$

and the remaining $m - r$ rows

$$z_{m-r} + (A^{-1})_{m-r} B' z_r + (A^{-1})_{m-r} E z_{n-m} = (A^{-1})_{m-r} d.$$

This completes the proof of (8).

References

1. P.W. Aitchison: Diakoptics or tearing – a mathematical approach, *Quarterly of Applied Mathematics* 41 (1983) 265–272.
2. H. Asai and M. Tanaka: Diakoptics: multi-level tearing method and estimation of the computation time, *Proceedings of International Symposium on Circuits and Systems*, 1984, Vol. 1, IEEE, New York (1984) 364–367.
3. Adi Ben-Israel and T.N.E. Greville: *Generalized Inverses: Theory and Applications*, Wiley, New York (1974), reprinted by Krieger (1980).
4. M.V. Bhat and H.K. Kesaven: Diakoptic equations and sparsity, *IEEE PES Summer Meeting*, Anaheim (1974); IEEE (1974) paper C74.384-4.
5. M.D. Bradshaw: An electromagnetic analog to the method of diakoptics (Kron's 'tearing technique') of circuit analysis, *23rd. Midwest Symposium on Circuits and Systems*, Toledo (1980) 562–566.
6. A. Brameller, M.N. John and M.R. Scott: *Practical Diakoptics for Electrical Networks*, Chapman and Hall, London (1969).
7. A. Brameller and K.L. Lo: The application of diakoptics and the escalator method to the solution of very large eigenvalue problems, *International journal for Numerical Methods in Engineering* 2 (1970) 535–549.
8. H.F. Bückner: Numerical methods for integral equations; in: *Survey of Numerical Analysis*, McGraw-Hill, New York (1962).
9. J.R. Bunch and D.J. Rose: Partitioning, tearing and modification of sparse linear systems, *Journal of Mathematical Analysis and Applications* 48 (1974) 574–593.
10. G. Cafaro, P. Pugliese and F. Vacca: Parallel solution of turn networks, *International Journal of Electrical Power and Energy Systems* 6 (1984) 131–138.
11. C.F. Chen and K.U. Wang: Diakoptics for eigenvalues of large scale networks, *23rd Midwest Symposium on Circuits and Systems*, Toledo (1980); North Holland, California (1980) 572–574.
12. L.O. Chua and L. Chen: Diakoptic and generalized hybrid analysis, *IEEE Transactions on Circuits and Systems* 23 (1976) 694–705.
13. J. Constantinescu: Study of the transient processes in large-scale power systems, *Revue Roumaine des Sciences Techniques* (*Serie d'Electrotechnique et Energetique*) 27 (1982) 211–227.
14. P. Cristea and M. Iordache: A diakoptic analysis method, *IEEE International Symposium on Circuits and Systems* (1983), IEEE, New York (1983) 633–636.
15. I.S. Duff and J.K. Reid: An implementation of Tarjan's algorithm for the block triangularization of a matrix, *ACM Transactions on Mathematical Software* 4 (1978) 137–147 and 189–192.
16. M. El-Marsafawy, R.W. Menzies and R.M. Mathur: New diakoptic technique for load-flow solution of very large power-systems using the bus admittance matrix, *Proc. Institution of Electrical Engineers* 126 (1979) 1301–1302.
17. T. Fukao and E. Teratsuji: An extension of diakoptics and its application to singular decomposition problem of networks, *Electrical Engineering in Japan* 102 (1) (1982) 141–148.
18. A. George and J.W. Liu: *Computer Solution of Large Sparse Positive Definite Systems*, Prentice Hall, New Jersey (1981) Ch. 4.
19. N.E. Gibbs, W.G. Poole and P.K. Stockmeyer: An algorithm for reducing the band width and profile of a sparse matrix, *SIAM Journal of Numerical Analysis* 13 (1976) 236–250.
20. G. Goubau, N.N. Puri and F.K. Schwering: Diakoptic theory for multi-element antennas, *IEEE Transactions on Antennas and Propagation* 30 (1982) 15–26.
21. I.N. Hajj: Diakoptics and block elimination, *IEEE 1978 Summer Power Meeting at Los Angeles*, paper A78, 564-7.
22. I.N. Hajj: Sparsity considerations in network solution by tearing, *IEEE Transactions on Circuits and Systems* 27 (1980) 357–366.
23. H.H. Happ: The application of diakoptics to the solution of power system problems, in: *Electrical Power Problems, The Mathematical Challenge*, SIAM (1980) 69–103.
24. H.H. Happ: *Piecewise Methods and Applications to Power Systems*, John Wiley, Chichester (1980).
25. H.V. Henderson and S.R. Searle: On deriving the inverse of a sum of matrices, *SIAM Review* 23 (1981) 53–60.
26. P. Henrici: *Discrete Variable Methods in Ordinary Differential Equations*, John Wiley, New York (1962).
27. P.B. Johns and K. Akhtarzad: Time domain approximations in the solution of fields by time domain diakoptics, *International Journal for Numerical Methods in Engineering* 18 (1982) 1361–1373.

28. H.K. Kesaven and J. Dueckman: Multi-terminal representations and diakoptics, *Journal of the Franklin Institute* 313 (1982) 337–352.
29. A. Klos: What is diakoptics? *Electrical Power and Energy Systems* 4 (1982) 192–195.
30. I.N. Kotarova and O.Yu. Shamaeva: A parallel diakoptics method of solving complex problems on distributed computational systems, *Cybernetics* 15 (1979) 131–141.
31. G. Kron: Diakoptics – piecewise solution of large scale systems, *The Electrical Journal*, in 20 parts from 158 (1957) 1673–1677 to 162 (1959) 431–436.
32. B. Majundar and T.N. Saha: Diakoptic approach to symmetrical short circuit analysis, *Journal of the Institution of Engineers (India), part EL, Electrical Engineering Division* 61 (1980) 1–4.
33. S.Y. Mansour, D.H. Kelly and D.O. Koval: Diakoptical active-reactive dispatch of generation, *Proceedings of the 1984 IEEE International Symposium on Circuits and Systems*, Vol. 1, IEEE, New York (1984) 136–139.
34. O.W. Marcus, K. Reiss, B.X. Weis and Z.P. Tomaszewski: Basic considerations on an adaptive iteration algorithm for large scale network analysis, *Proceedings of the 1983 IEEE Symposium on Circuits and Systems*, Vol. 1, IEEE, New York (1983) 230–233.
35. J. Myslik: Diakoptical analysis of the networks considering the changes of branch impedances, *Electrotech. Obz. (Czechoslovakia)* 73 (1984) 196–200.
36. W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling: *Numerical Recipes: the Art of Scientific Computing*, Cambridge University Press, Cambridge (1986).
37. N.B. Rabatt and H.Y. Hsieh: A latent macromodular approach to large-scale sparse networks, *IEEE Transactions on Circuits and Systems* 23 (1976) 745–752.
38. J.K. Reid: A survey of sparse matrix computation, in: *Electrical Power Problems: the Mathematical Challenge*, SIAM (1980) 41–68.
39. A. Sangiovanni-Vincentelli, L. Chen and L.O. Chua: A new tearing approach-node tearing nodal analysis, *Proceedings of 1977 IEEE International Symposium on Circuits and Systems*, IEEE, New York (1977) 143–147.
40. D.V. Steward: Partitioning and tearing of systems of equations, *SIAM J. Numerical Analysis*, Series B, 2 (1965) 345–365.
41. P.K.U. Wang and C.F. Chen: Sensitivity calculation and network optimization through decomposition, *Proceedings of 1983 IEEE Symposium on Circuits and Systems*, Vol. 3, IEEE, New York, (1983) 1034–1037.
42. F.F. Wu: Solution of large-scale networks by tearing, *IEEE Transactions on Circuits and Systems* 23 (1976) 706–713.