

Optimal Estimation of Jacobian and Hessian Matrices That Arise in Finite Difference Calculations

By D. Goldfarb* and Ph. L. Toint

Abstract. In this paper, the problem of estimating Jacobian and Hessian matrices arising in the finite difference approximation of partial differential equations is considered. Using the notion of computational molecule or stencil, schemes are developed that require the minimum number of differences to estimate these matrices. A procedure applicable to more complicated structures is also given.

1. Introduction. In the past few years, there has been a growing interest in studying efficient ways to obtain good estimates of sparse Jacobian or Hessian matrices. Two basic approaches have been followed. In the first approach, it is assumed that one has an estimate of the Jacobian (Hessian) which one wishes to revise given "arbitrary" changes in the variables and the corresponding changes in the functions (gradients). This has led to the development of several sparse quasi-Newton methods (see [11], [14], [12], for example). In the second approach, it is assumed that one can specify the changes in the variables. One then forms an estimate of the Jacobian (Hessian) matrix from the observed changes in the functions (gradients). When the dimension of the matrix is large and the evaluation of each function (gradient) vector is expensive, it is necessary to take the sparsity structure of the matrix into account, in order to make this approach efficient. Several methods for doing this for arbitrary sparsity structures have already appeared in the literature [2]–[5], [7]–[10], [13].

In this paper, we consider the second approach for cases where the sparsity structure arises from finite difference approximations to partial differential equations. Both symmetric and unsymmetric matrices are considered. Although the algorithms presented do not generalize to every pattern generated by finite differences, they cover the most useful structures, and, in some cases, extend existing methods to more complex situations.

In the next section, we describe the method of Curtis, Powell and Reid [5], henceforth referred to as the CPR method, for estimating sparse Jacobian matrices. We then show how the so-called "computational molecule" or "stencil" [1] of the finite difference operator associated with the Jacobian matrix under study can be used to determine the groupings (i.e. function differencing directions) used by the

Received September 2, 1981; revised August 3, 1982 and March 19, 1983.

1980 *Mathematics Subject Classification.* Primary 65M05, 65N05.

Key words and phrases. Finite differences, sparsity, coverings.

* The research of this author was supported in part by the National Science Foundation under Grant No. MCS-8006065, and by the Army Research Office under Contract No. DAAG-29-82-K-0163.

©1984 American Mathematical Society
0025-5718/84 \$1.00 + \$.25 per page

CPR method. This allows sparse Jacobians of the kind considered in this paper to be estimated with a minimum number of function vector differences. In Section 3, we describe the Lower Triangular Substitution (LTS) method of Powell and Toint [10] for estimating sparse Hessian matrices. Analogously to what is done in Section 2, the computational molecule is used to develop optimal estimation schemes. Section 4 presents a tearing procedure which is applicable when the LTS method is used on more complicated structures. Finally, some concluding remarks are offered in Section 5.

2. Computational Molecules and Estimation of Sparse Jacobian Matrices. We now consider the problem of estimating the Jacobian matrix $J(\cdot)$ of some differentiable function $g(\cdot)$ from R^n into R^n at some point x of interest. We also assume that we know the sparsity pattern of $J(\cdot)$ a priori. In order to obtain an estimate of $J(x)$; we compute the value of $g(\cdot)$ at x and at some other points $x + s_k$. To achieve this using as few s_k as possible, we must make use of our knowledge of the sparsity pattern of $J(x)$. We finally assume that we can only compute full function vectors and not any part of them separately. Each element of the Jacobian matrix $J(x)$ is, in general, a nonlinear function of x . When the function $g(\cdot)$ is linear, $J(x)$ is a constant matrix J . Since $J(x)$ has the same sparsity pattern whether $g(x)$ is linear or not, we shall henceforth refer to $J(x)$ simply as J .

Curtis, Powell and Reid [5] were the first to point out that the number of evaluations of $g(\cdot)$ needed to estimate J could be reduced by taking advantage of the sparsity pattern of J . Their idea was to form groups of columns of J that could be estimated together, using only one function difference. This requires that the columns belonging to the same group have all their nonzero elements on different rows. Any group of columns of J will be called CPR valid if and only if this criterion is met. The difference related to that group is then taken along the direction formed by adding together suitable multiples of the canonical basis vectors that correspond to the columns in the group.

For example, let us consider the 4×4 pattern

$$(1) \quad \begin{array}{cccc} * & * & & \\ * & * & * & \\ & * & * & * \\ & & * & * \end{array}$$

The first and fourth columns form a CPR valid group, since they have their nonzero elements in different rows (1, 2 and 3, 4); hence they can be both estimated by using a difference along the direction

$$(2) \quad s_1 = h_1 e_1 + h_4 e_4,$$

where $\{e_i, i = 1, 2, 3, 4\}$ is the canonical basis of R^4 . Columns 2 and 3 each form a separate group since each has at least one nonzero in a row in common with every other column. Estimation of these columns therefore requires two more differences. Thus applying the CPR method to the Jacobian (1) allows it to be estimated in three

differences instead of four. It is also quite easy to verify that the maximum number of nonzero elements on a single row is a lower bound on the number of groups, and hence of differences, that will be needed to estimate a particular sparse Jacobian matrix. Moreover Newsam and Ramsdell [9] have shown that it is always possible (mathematically) to estimate a Jacobian matrix in this number of differences.

In [5], Curtis, Powell and Reid proposed forming groups by considering successively the columns of J in their natural order. Unfortunately, this procedure does not produce a minimal number of groups on most Jacobian structures arising in finite difference calculations. More recently, Coleman and Moré [2] further discussed the ordering in which the columns should be considered, in order to minimize the number of groups. They showed that, for a general sparse pattern, the problem is equivalent to a certain coloring problem on a suitable graph, and proposed the use of graph coloring algorithms to obtain a small number of groups. But these methods still do not yield the minimum number of groups on all sparsity patterns that are encountered in finite difference problems, as observed by the authors themselves.

We now develop another way to gather the columns of the pattern in suitable groups, when we are dealing with a finite difference approximation of a partial differential equation problem. The resulting algorithms will achieve the minimum number of groups needed to estimate the Jacobian in our framework.

The sparsity pattern of J associated with a finite difference approximation to a partial differential equation problem is completely determined by the “computational molecule” or “stencil” of the finite difference operator, the ordering of the components of x (i.e. the numbering of the mesh points) and the given boundary conditions. For the moment we will consider only Dirichlet boundary conditions.

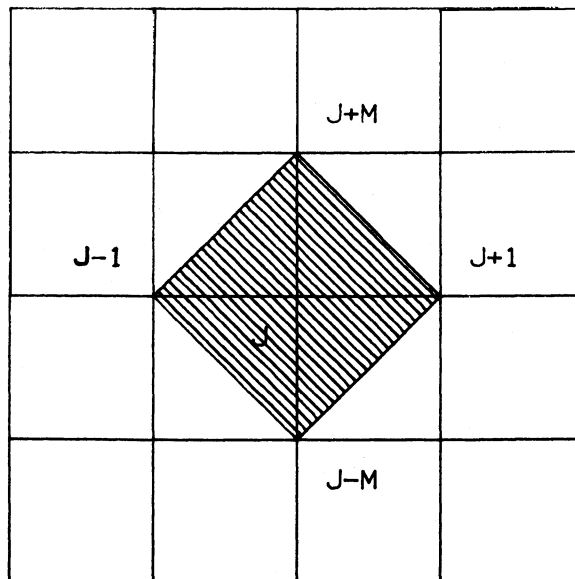


FIGURE 2-1

The computational molecule for the five-point finite difference Laplace operator

Such conditions essentially have no effect on the sparsity pattern of J . In Section 4, we will discuss techniques for handling cases where the boundary conditions affect this pattern. Further, we observe that although the ordering of the components of x affects the sparsity pattern, it does not affect the essential sparsity structure of J ; e.g. permuting the rows and columns with J does not change the intersection graph associated to J aside from the labels assigned to nodes of this graph. This associated graph has vertices $\{1, 2, \dots, n\}$ corresponding to the column indices of J , and edges (i, j) if $i \neq j$ and columns i and j have a nonzero in the same row.

Since the CPR method depends only upon the essential sparsity structure of J , we need only consider the computational molecule of the finite difference operator to form CPR valid groups. To illustrate this, consider the two-dimensional five-point Laplacian operator applied to a rectangular region with $m \times k$ mesh numbered in the standard way (see Figure 2-1). The points that are covered by the molecule centered at the j th mesh point are those with indices $j - m, j - 1, j, j + 1, j + m$; consequently, the only nonzero elements in the j th row (column) of J occur in columns (rows) $j - m, j - 1, j, j + 1$ and $j + m$.

Partitioning the columns of J into a minimal number of groups suitable for the CPR algorithm now becomes a simple task, given the molecule. In the particular case at hand, we completely cover all mesh points by disjoint molecules as shown in Figure 2-2 to determine one of the groups. This group consists of those columns corresponding to the centers of the molecules of this cover. This is clearly a CPR valid group, since all the molecules used to form the cover are disjoint.

The other groups in the partition of the columns of J are obtained by shifting this cover of molecules respectively one line up, one line left, one line right and one line

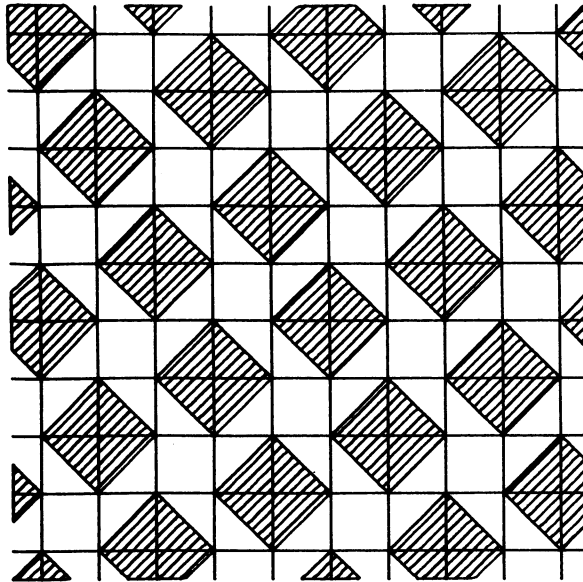


FIGURE 2-2

*Cover of the finite difference mesh
by the computational molecules*

down. One can also view the formation of a new group as the gathering of all columns whose indices are those of variables occupying the same position in every molecule of the cover. In the case of an $m \times n$ mesh numbered in the standard way, the five groups determined by the above approach are

$$G_p = \{ i + m(j - 1) \mid 1 \leq i \leq m, 1 \leq j \leq n, (i + 3j) \bmod 5 = p \}$$

for $p = 0, 1, 2, 3, 4$.

Five groups is clearly the minimum number required by the CPR method in this example, since the maximum number of nonzero elements in a row of the sparsity pattern is five. This provides a new justification of a method described in [8] for this particular case.

It should also be clear that this approach is not restricted to rectangular regions, but can be applied as well to nonrectangular ones, with all sorts of holes and odd shapes.

Minimum cardinality partitions of the columns of J are also given below for some other common two-dimensional computational molecules.

A 3-diagonal Jacobian. The 3 groups corresponding to the cover in Figure 2-3 for an $m \times n$ mesh are

$$G_p = \{ i + m(j - 1) \mid 1 \leq i \leq m, 1 \leq j \leq n, (i + 2j) \bmod 3 = p \}$$

for $p = 0, 1, 2$.

A 7-diagonal Jacobian. The 7 groups corresponding to the cover in Figure 2-4 for an $m \times n$ mesh are

$$G_p = \{ i + m(j - 1) \mid 1 \leq i \leq m, 1 \leq j \leq n, (i + 4j) \bmod 7 = p \}$$

for $p = 0, 1, \dots, 6$.

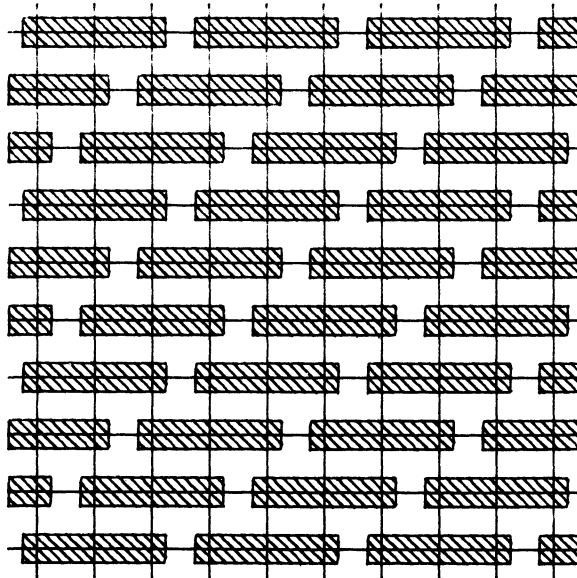


FIGURE 2-3
Stencil and cover for a 3-diagonal Jacobian

A 9-diagonal Jacobian. The 9 groups corresponding to the cover in Figure 2-5 for an $m \times n$ mesh are

$$G_p = \{i + m(j - 1) \mid 1 \leq i \leq m, 1 \leq j \leq n, (i + 6j) \bmod 9 = p\}$$

for $p = 0, 1, \dots, 8$.

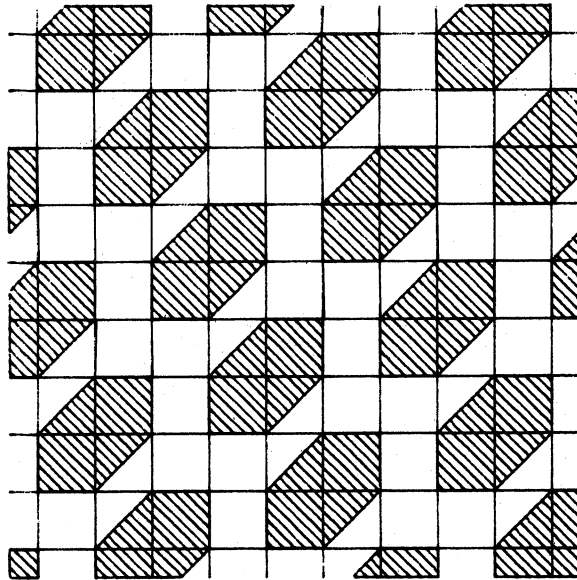


FIGURE 2-4

Stencil and cover for a 7-diagonal Jacobian

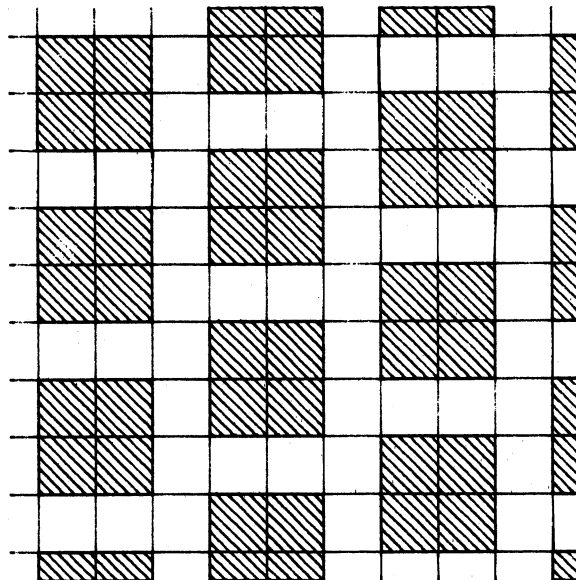


FIGURE 2-5

Stencil and cover for a 9-diagonal Jacobian

An 11-diagonal Jacobian. The 11 groups corresponding to the cover in Figure 2-6 for an $m \times n$ mesh are

$$G_p = \{i + m(j - 1) \mid 1 \leq i \leq m, 1 \leq j \leq n, (i + 7j) \bmod 11 = p\}$$

for $p = 0, 1, \dots, 10$.

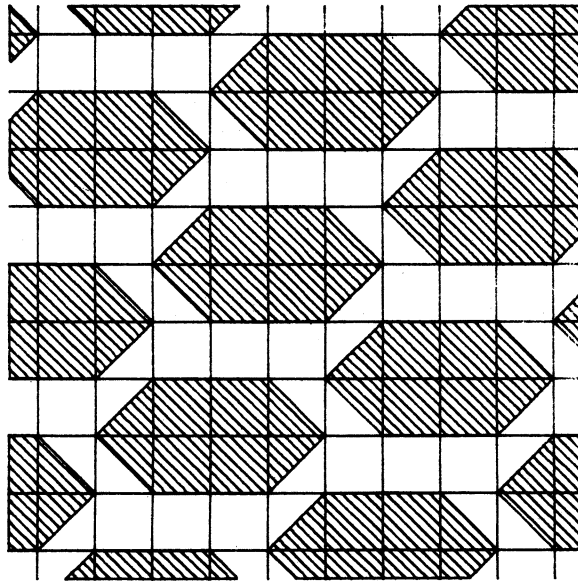


FIGURE 2-6
Stencil and cover for an 11-diagonal Jacobian

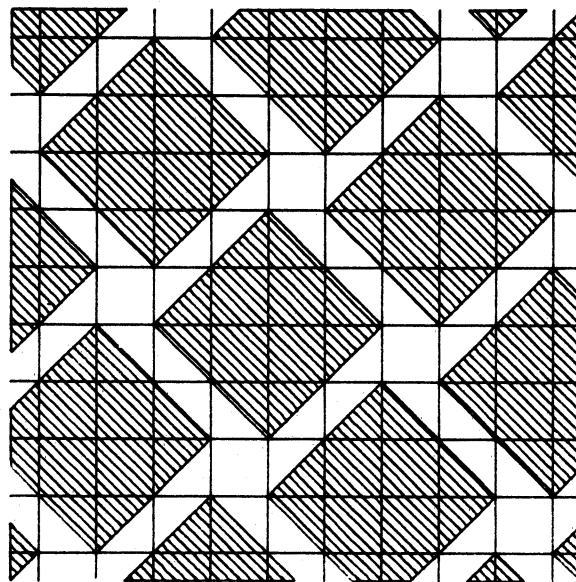


FIGURE 2-7
Stencil and cover for a 13-diagonal Jacobian

A 13-diagonal Jacobian. The 13 groups corresponding to the cover in Figure 2-7 for an $m \times n$ mesh are

$$G_p = \{ i + m(j -) \mid 1 \leq i \leq m, 1 \leq j \leq n, (i + 5j) \bmod 13 = p \}$$

for $p = 0, 1, \dots, 12$.

Each of the above partitions is a minimum cardinality partition** of the columns of J in that the number of groups equals the number of mesh points covered by the computational molecule, i.e. the number of nonzero diagonals in the sparsity pattern of J . Observe, however, that in all of the above cases, there are other minimum cardinality partitions besides those that are given.

Although the above list covers most of the useful ones, the method we propose does not extend to all possible stencils in the discretized plane. Indeed, to be able to apply the procedure, we must build a cover of the mesh using the disjoint molecules, and this is not possible for all molecules. For example, the molecule corresponding to the 9-point Laplacian operator, covering mesh points $j - 2m, j - m, j - 2, j - 1, j, j + 1, j + 2, j + m, j + 2m$, cannot be used for such a cover.

It is easy to prove that the placement of disjoint molecules shown in Figure 2-8 is the closest that one can get to a covering of the mesh points. This figure also shows that ten groups suffice for this Jacobian, since the nine-point stencil centered at mesh point j , augmented by the tenth-point $m + m + 2$, forms a cover. These additional points are indicated by circles in the figure.

The ten groups formed in this manner for an $m \times n$ mesh are

$$G_p = \{ i + m(j - 1) \mid 1 \leq i \leq m, 1 \leq j \leq n, (i + 3j) \bmod 10 = p \}$$

for $p = 0, 1, \dots, 9$.

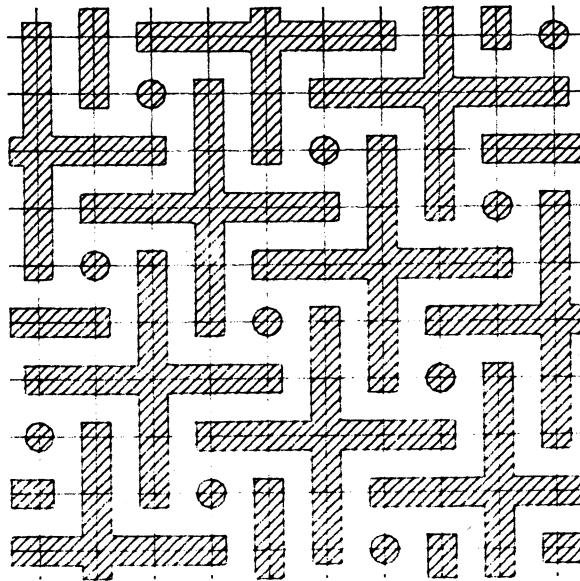


FIGURE 2-8
Stencil for the 9-point Laplacian operator

** Minimum cardinality partitions have also been given for the five-diagonal and the nine-diagonal Jacobian by Newsam and Ramsdell [9], who observed independently that such partitions could be easily found by considering the stencil associated with the Jacobian.

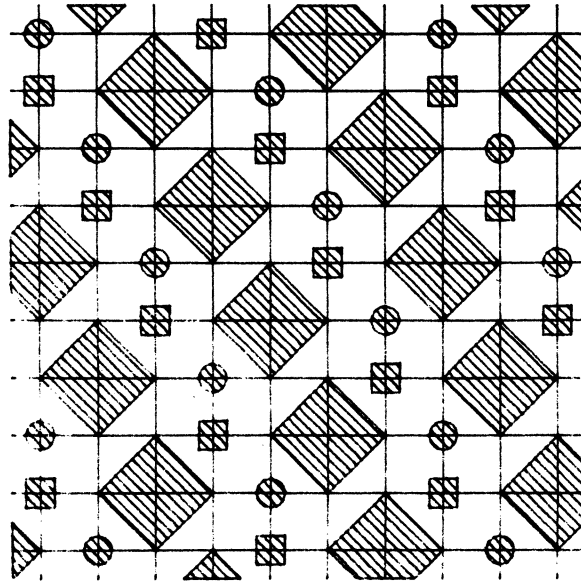


FIGURE 2-9
 One horizontal layer of the covering
 for the 3-dimensional 7-point Laplacian

Our approach can also be extended to 3-dimensional meshes, if we can find a suitable covering of this space with the 3-dimensional stencils defined by the operator. This problem is more difficult than the 2-dimensional case, but not hopeless, as shown by the following example. We consider again the Laplacian operator on an $m \times n \times q$ mesh, and the computational molecule covering mesh points $j - nm, j - m, j - 1, j, j + 1, j + m$ and $j + nm$. This is the natural extension to 3 dimensions of the 5-point operator analyzed above. This molecule can be viewed as a cube standing on one of its corners. It is not difficult to see that the cover of Figure 2-9 is adequate, where the slice of the mesh shown is horizontal, and where a small circle represents the bottom corner of the stencil, a large diamond its section at mid-height and a small square its top corner. When moving one layer up, this cover is simply shifted two positions right and one down.

The seven groups associated with this covering for an $m \times n \times q$ mesh are

$$G_p = \{ i + m(j - 1) + mn(k - 1) \mid 1 \leq i \leq m, 1 \leq j \leq n, 1 \leq k \leq q, \\ (i + 5j + 3k) \bmod 7 = p \}$$

for $p = 0, 1, \dots, 6$.

A similar extension to 3 dimensions is also clearly possible for the 2-dimensional 9-point operator, whose extended computational molecule is now a cube standing on one of its faces.

3. Estimation of Sparse Hessian Matrices. We now turn to the case where our function $g(\cdot)$ is in fact a gradient, and can be expressed as

$$(3) \quad g(x) = H(x)x + b(x),$$

where the elements of $H(x)$ and $b(x)$ are nonlinear functions of x . As above, the structure of $H(x)$ is identical to that of the constant H that would be obtained in the linear case, and this structure is induced by the computational molecule of the given underlying operator. The problem we face is nevertheless different, because we want to use the inherent symmetry of H to reduce the number of groups (and the number of needed differences) even further. This can be accomplished by using an extension of the CPR method to the symmetric case: the LTS method proposed by Powell and Toint in [10].

Let us briefly recall how the LTS (Lower Triangular Substitution) procedure works. Instead of considering the whole sparsity pattern to form CPR valid groups of columns, this method only considers its lower-triangular part. In [10], Powell and Toint propose a general ordering procedure for the columns of this reduced pattern, and they show that it is possible using one direction for each group to obtain an estimate of the whole Hessian matrix by taking differences in $g(\cdot)$ along directions corresponding to these groups. This procedure involves an algebraic substitution, hence the name.

Consider again, for illustration purposes, the 4×4 pattern of (1), and assume now symmetry of the corresponding matrix. Applying the CPR method to the lower-triangular part of this pattern yields two groups, namely

$$(4) \quad G_1 = \{1, 3\} \quad \text{and} \quad G_2 = \{2, 4\}.$$

Using the directions

$$(5) \quad s_1 = h_1 e_1 + h_3 e_3 \quad \text{and} \quad s_2 = h_2 e_2 + h_4 e_4$$

corresponding to these groups yields the following equations for determining H :

$$(6) \quad \begin{array}{ll} H_{11}h_1 = y_1, & H_{12}h_2 = z_1 \\ H_{21}h_1 + H_{23}h_3 = y_2, & H_{22}h_2 = z_2, \\ H_{33}h_3 = y_3, & H_{32}h_2 + H_{34}h_4 = z_3, \\ H_{43}h_3 = y_4, & H_{44}h_4 = z_4, \end{array}$$

where

$$(7) \quad y = g(x + s_1) - g(x) \quad \text{and} \quad z = g(x + s_2) - g(x).$$

It is now obvious that all entries of H can be determined, starting from the last row and working upwards by taking symmetry of H into account. More details and some error bounds can be found in [10].

Returning to our computational molecules, we now see that only the part of the molecule that determines the pattern in the lower-triangular part of H is relevant to the LTS method. This is the part of the molecule covering mesh points with indices larger than that of its center. We refer to these partial molecules as “forward computational molecules” for this reason. If we consider once more the 5-point 2-dimensional Laplacian operator applied on a rectangular $m \times n$ mesh numbered in the standard way, its forward molecule is now a triangle covering mesh points j , $j + 1$ and $j + m$, as shown in Figure 3-1.

A suitable covering of the mesh can now be constructed as shown in Figure 3-2, and the three resulting groups will be

$$G_p = \{ i + m(j - 1) \mid 1 \leq i \leq m, 1 \leq j \leq n, (i + 2j) \bmod 3 = p \}$$

for $p = 0, 1, 2$.

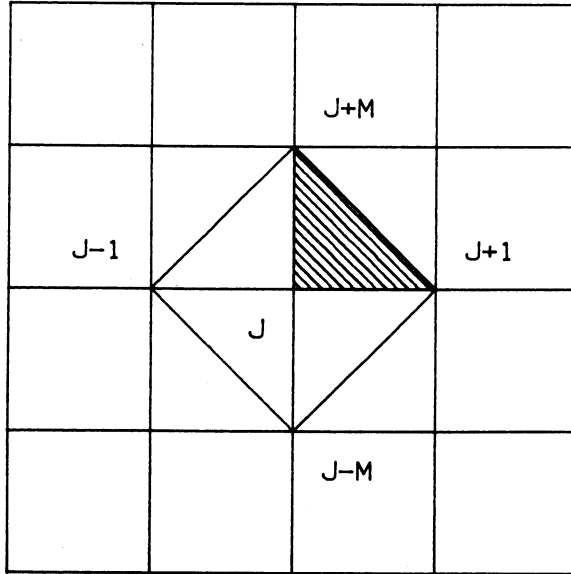


FIGURE 3-1

The forward molecule for the five-point Laplacian operator in two dimensions

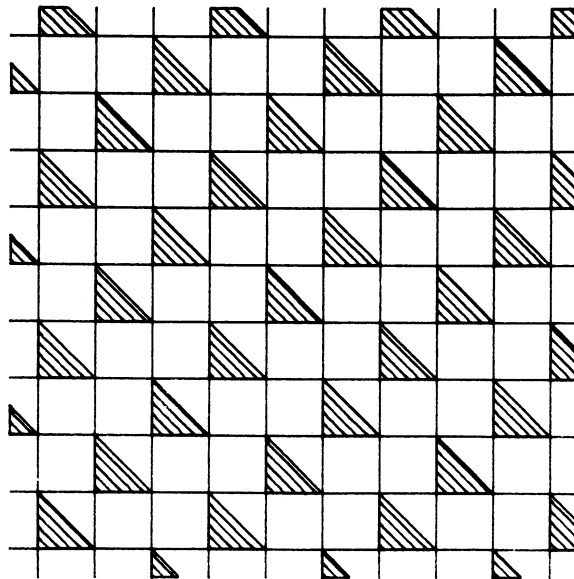


FIGURE 3-2

Cover for a 5-diagonal Hessian

This simple example provides a generalization of the algorithm of [8] to the symmetric case, and the partition clearly has minimum cardinality. Again, this technique can be very easily applied on nonrectangular regions with or without holes.

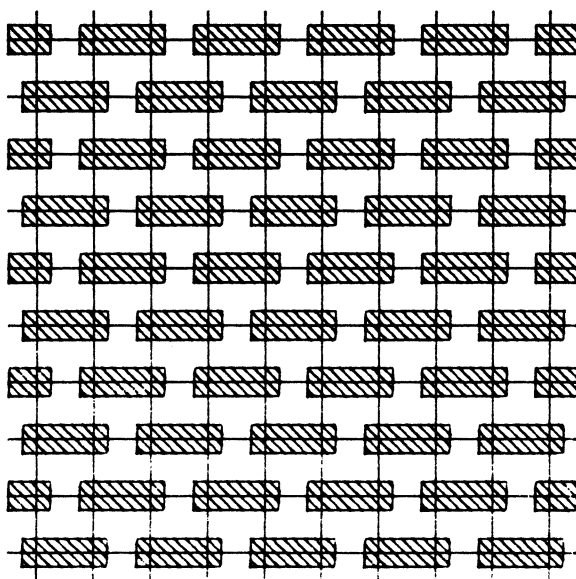


FIGURE 3-3

Forward molecule and cover for a 3-diagonal Hessian

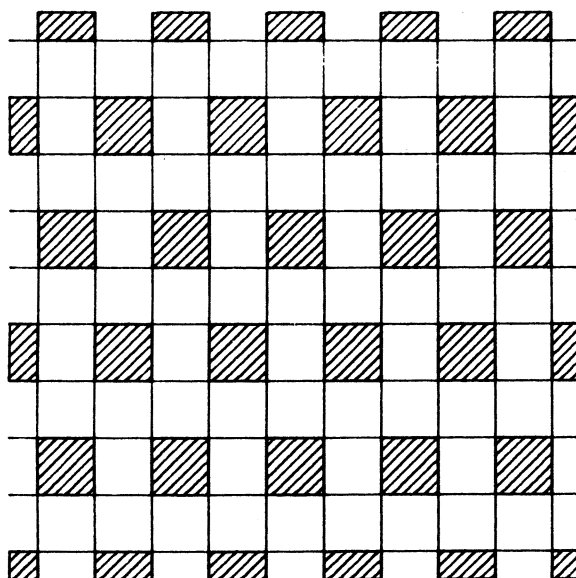


FIGURE 3-4

Forward molecule and cover for a 7-diagonal Hessian

As in the previous section, we give the forward molecules and covers for some other common 2-dimensional operators in Figures 3-3-3-7.

Each of these coverings results in a number of groups equal to the number of diagonals in the lower-triangular part of the Hessian matrix (2, 4, 5, 6 and 7 respectively), and hence, is minimum.

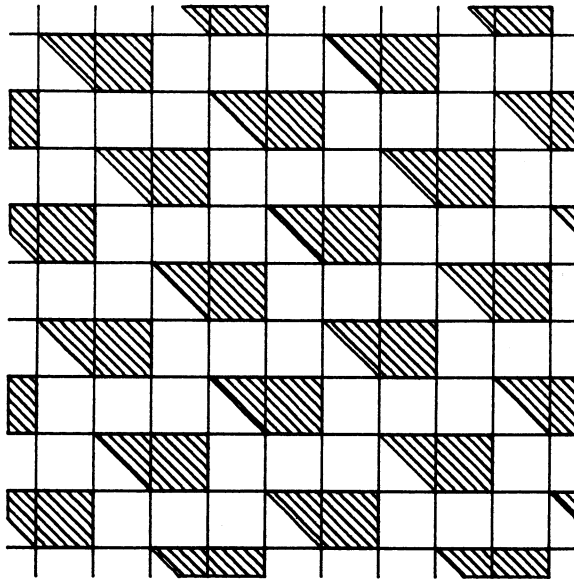


FIGURE 3-5

Forward molecule and cover for a 9-diagonal Hessian

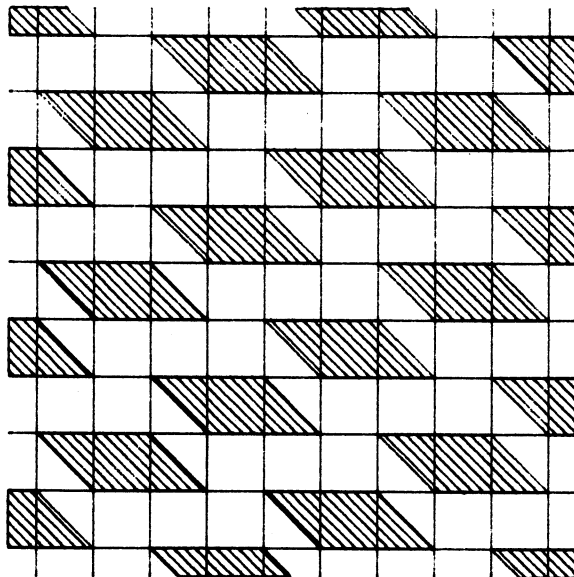


FIGURE 3-6

Forward molecule and cover for an 11-diagonal Hessian

Observe that, in some cases, the fact that the forward molecule differs in shape from the molecule itself may be important. For example, it is now possible to use the forward molecule associated with the 9-point Laplacian operator (a kind of L) to cover the 2-dimensional region, and hence to estimate the corresponding Hessian

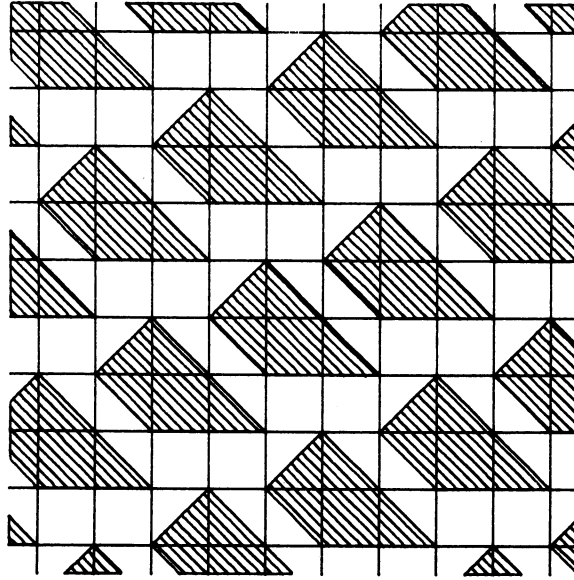


FIGURE 3-7

Forward molecule and cover for a 13-diagonal Hessian

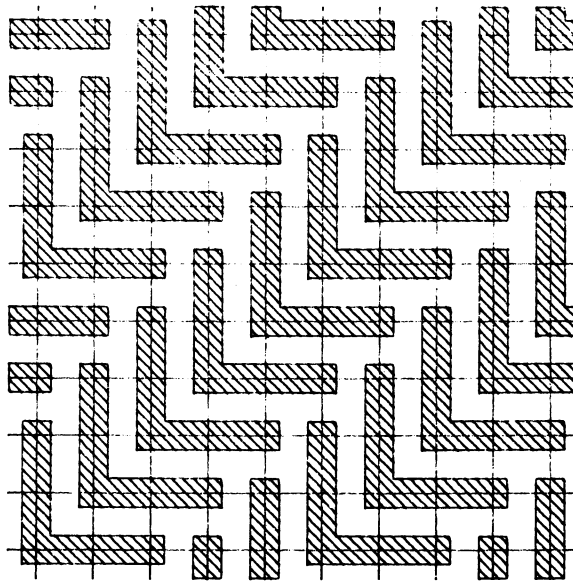


FIGURE 3-8

Forward molecule and cover for the 9-point Laplacian operator

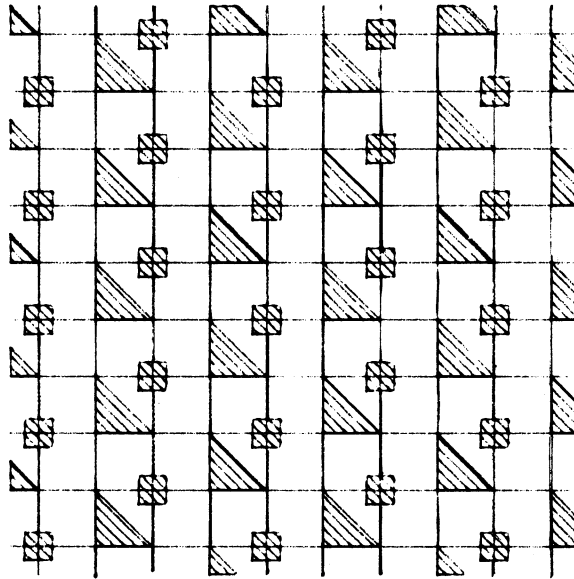


FIGURE 3-9
*One horizontal layer of the forward covering
 for the 3-dimensional 7-point Laplacian*

matrix with 5 differences (see Figure 3-8). As shown above, it was impossible to use that operator in the manner proposed in the unsymmetric case.

As in the unsymmetric case, extension to 3-dimensional problems is sometimes possible, provided the new 3-dimensional forward molecule can be used to cover the 3-dimensional mesh. As an example, we consider again the extension of the 5-point Laplacian operator to 3 dimensions: the corresponding Hessian matrix now has 7 diagonals. The forward molecule associated with this operator is a tetrahedron which can be used to cover the mesh, as shown in Figure 3-9.

In this figure, the triangles represent the bases of the forward molecules, while the small squares stand for the top corners. When moving one layer up, the same pattern is reproduced shifted one position down and one position left. This means that it is possible to estimate the Hessian matrix corresponding to the 3-dimensional 7-point Laplacian operator with the minimal number of 4 differences in $g(\cdot)$, using an LTS procedure.

4. A Tearing Procedure for More Complicated Structures. In this section, we investigate a procedure that is useful when dealing with more complicated Hessian structures. This procedure applies to several different cases, but will be motivated and analyzed on the following example. Consider the problem where the Laplacian operator is applied on a 2-dimensional rectangular region with periodic boundary conditions on the vertical sides of the region. Then a block-tridiagonal matrix results in which the diagonal blocks are tridiagonal except for nonzero elements in the

first estimate all other elements of the row, then use the supplementary direction and these values to substitute for one of the conflicting elements, and finally substitute for the other, using the direction associated with its LTS group.

The method used in this simple example can be formulated as a general algorithm as follows. Assume that, for some reason, we wish to use certain predefined column groups in our estimation procedure. In general, we cannot assume that these groups will be CPR valid, and hence some rows will contain different nonzero elements belonging to columns in the same group. Let $S(i, k)$ be defined as the set of elements with row index i in a column j in group k , where $j \leq i$; i.e.

$$(9) \quad S(i, k) = \{(i, j) \mid j \leq i, H_{ij} \neq 0 \text{ and } k(j) = k\},$$

where $k(j)$ is the group number of column j in our a priori grouping. If for some i , the set $S(i, k)$ contains more than one element, then group k is not CPR valid and we need more information to estimate the elements in $S(i, k)$.

Consider now the set S of elements contained in the sets $S(i, k)$ with cardinality greater than one, and the set C of columns corresponding to the elements in S ; i.e.

$$S = \{(i, j) \mid (i, j) \in S(i, k) \text{ and } \text{card}[S(i, k)] > 1 \text{ for some } k\}$$

and

$$C = \{j \mid (i, j) \in S\}.$$

The idea is then to construct a new CPR grouping for these columns, considered in their natural order. In this grouping, we shall take into account the fact that some information is already available through the a priori groups and differences.

The grouping algorithm we propose can now be described.

1. Form the set C . Let $M(S)$ be the matrix whose nonzeros are indexed by S .
2. If C is empty, stop. Otherwise, form a CPR valid group by considering the nonzero columns of $M(S)$.
3. Delete from C the indices of all columns belonging to the group just formed, and all corresponding elements from S and their respective sets $S(i, k)$.
4. If any set $S(i, k)$ reduces to a single element (i, j) , delete that element from S . Further, if column j of $M(S)$ becomes all zeros, delete j from C .
5. Go to step 2.

Step 4 of the algorithm clearly uses the fact that the a priori group k provides enough information to estimate one of the elements of $S(i, k)$ once the others are known.

The usual LTS substitution procedure for a particular row is now modified as follows.

1. Substitute for all elements that are the only members of their group in that row.
2. If all the elements of the row have been computed, consider the next row.
3. Else,
 - a. choose k such that $S(i, k)$ contains more than one element, where i is the current row number.
 - b. Find a supplementary group that contains, in the i th row, only one of these elements, and use that group to substitute for that particular element.

- c. If, for some k , $S(i, k)$ contains more than one element, go back to a.
- d. Substitute for the remaining unknown elements of the row, by using the a priori groups.

In the example above, one can easily see that S contains only elements (12,6), (12,7), (18, 12), and (18,13), and

$$C = \{6, 7, 12, 13\}.$$

The only supplementary group will then gather columns 6 and 12, and the resulting direction will be given by (8).

This general method can be easily applied to more complicated structures. For example, we could consider our periodic problem where the Laplacian is replaced by the biharmonic operator, and the a priori grouping given by the covering of Figure 3-7. Another interesting case is the estimation of an outlying diagonal around a central band pattern: the algorithm proposed in [15] is the particularization of the method just proposed to that type of structure.

Clearly, once the lower-triangular pattern is fixed, the ordering in the new procedure must be the natural one. However, the user is free to consider any permutation of the full columns of the Hessian matrix, so that the resulting structure $M(S)$ can be covered with as few CPR groups as possible. This obviously depends on the a priori grouping itself, and is problem dependent. One may attempt to choose the a priori groups in a way to optimize, in a certain sense, the resulting $S(i, k)$ in the lower-triangular pattern $M(S)$.

Observe also that the proposed procedure is not, strictly speaking, an LTS method. Indeed, the groups that are formed on the lower-triangular part of the considered pattern are not CPR valid.

5. Concluding Remarks. Our approach to determining minimum cardinality partitions of the columns of Jacobian and Hessian matrices that are, respectively, CPR and LTS valid was based upon covering the finite difference mesh with disjoint (full and forward) computational molecules. It should be evident that in the two-dimensional case, this is equivalent to the problem of “tiling” the plane with these molecules, allowing translations only. We could have characterized the molecules as “polyominoes” [6], which are simply connected squares joined along their edges. For example, the computational molecule for the five-point Laplacian operator can be represented by the “pentomino” shown in Figure 5-1.

Thus our approach in the two-dimensional cases corresponded to finding a tiling of the plane by translations of a given polyomino.

It is worth mentioning that the partitions obtained in this paper may be of some use when parallelizing such algorithms as the Gauss-Seidel and SOR algorithms for solving systems of linear equations that arise from finite difference approximations to partial differential equations.

It should also be pointed out that the tearing approach of Section 4 can be used for unsymmetric as well as symmetric matrices. However, if this is done, we obtain a “substitution” method as opposed to a “direct” CPR like method.

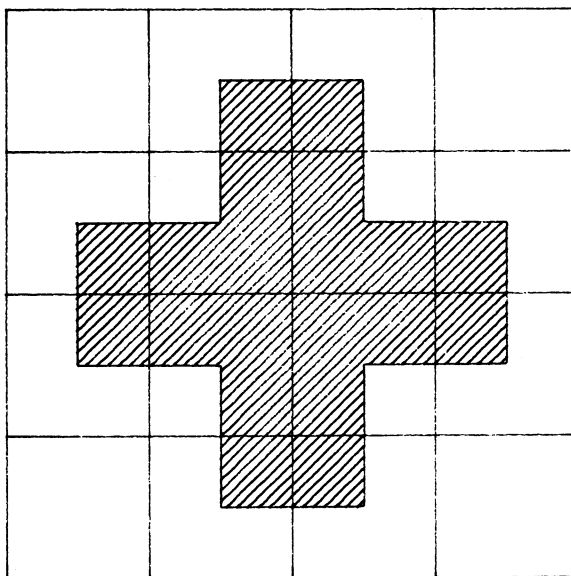


FIGURE 5-1
Pentomino for the 5-point Laplacian operator

6. Acknowledgements. It is a pleasure to acknowledge Tom Coleman's interest in the early developments of the ideas presented here. A discussion of the problem with M. J. D. Powell was also, as usual, very helpful.

Department of Industrial Engineering and Operations Research
 Columbia University
 New York, New York 10027

Department of Mathematics
 Facultés Universitaires de Namur
 Namur, Belgium

1. W. G. BICKLEY, "Finite difference formulae for the square lattice," *Quart. J. Mech. Appl. Math.*, v. 1, 1948, pp. 35–42.
2. T. F. COLEMAN & J. J. MORÉ, "Estimation of sparse Jacobian matrices and graph coloring problems," *SIAM J. Numer. Anal.*, v. 20, 1983, pp. 187–209.
3. T. F. COLEMAN & J. J. MORÉ, *Software for Estimating Sparse Jacobian Matrices*, Technical Report ANL-82-37, Argonne National Laboratory, Argonne, Illinois, 1982.
4. T. F. COLEMAN & J. J. MORÉ, *Estimation of Sparse Hessian Matrices and Graph Colouring Problems*, Technical Report ANL-81-535, Argonne National Laboratory, Argonne, Illinois, 1982.
5. A. CURTIS, M. J. D. POWELL & J. REID, "On the estimation of sparse Jacobian matrices," *J. Inst. Math. Appl.*, v. 13, 1974, pp. 117–119.
6. S. W. GOLOMB, *Polyominoes*, Chas. Scribner's Sons, New York, 1965.
7. S. T. MCCORMICK, "Optimal approximation of sparse Hessians and its equivalence to a graph coloring problem," *Math. Programming*, v. 26, 1983, pp. 153–171.
8. D. K. MELGAARD & R. F. SINCOVEC, "General software for two-dimensional nonlinear partial differential equations," *ACM Trans. Math. Software*, v. 7, 1981, pp. 106–125.
9. G. N. NEWSAM & J. D. RAMSDELL, "Estimation of sparse Jacobian matrices," *SIAM J. Algebraic Discrete Methods*, v. 4, 1983, pp. 404–418.
10. M. J. D. POWELL & PH. L. TOINT, "On the estimation of sparse Hessian matrices," *SIAM J. Numer. Anal.*, v. 16, 1979, pp. 1060–1074.

11. L. K. SCHUBERT, "Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian," *Math. Comp.*, v. 24, 1970, pp. 27–30.
12. D. F. SHANNO, "On variable metric methods for sparse Hessians," *Math. Comp.*, v. 34, 1980, pp. 499–514.
13. M. N. THAPA, *Optimization of Unconstrained Problems With Sparse Hessian Matrices—Newton-Type Methods*, Technical Report SOL82-8, Stanford University, Stanford, Calif., 1982.
14. PH. L. TOINT, "On sparse and symmetric matrix updating subject to a linear equation," *Math. Comp.*, v. 31, 1977, pp. 954–961.
15. PH. L. TOINT, *Estimation of Sparse Hessian Matrices: A Block Substitution Procedure and its Application to the Multi-Diagonal Case*, Technical Report 81/10, Dept. of Math., FUN Namur, Belgium, 1981.