

## THE EFFICIENT COMPUTATION OF SPARSE JACOBIAN MATRICES USING AUTOMATIC DIFFERENTIATION\*

THOMAS F. COLEMAN<sup>†</sup> AND ARUN VERMA<sup>‡</sup>

**Abstract.** This paper is concerned with the efficient computation of sparse Jacobian matrices of nonlinear vector maps using automatic differentiation (AD). Specifically, we propose the use of a graph coloring technique, bicoloring, to exploit the sparsity of the Jacobian matrix  $J$  and thereby allow for the efficient determination of  $J$  using AD software. We analyze both a direct scheme and a substitution process. We discuss the results of numerical experiments indicating significant practical potential of this approach.

**Key words.** sparse Jacobian matrices, nonlinear systems of equations, nonlinear least squares, graph coloring, bicoloring, automatic differentiation, computational differentiation, sparse finite differencing, partition problem, NP-complete problems, ADOL-C

**AMS subject classifications.** 65K05, 65K10, 65H10, 90C30, 90C05, 68L10

**PII.** S1064827595295349

**1. Introduction.** The efficient numerical solution of nonlinear systems of algebraic equations  $F(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$  usually requires the repeated calculation or estimation of the matrix of first derivatives, the Jacobian matrix  $J(x) \in \mathbb{R}^{m \times n}$ . In large-scale problems, the matrix  $J$  is often sparse and it is important to exploit this fact in order to efficiently determine, or estimate, the matrix  $J$  at a given argument  $x$ . This paper is concerned with the efficient calculation of sparse Jacobian matrices by the judicious application of AD techniques. Specifically, we show how to define “thin” matrices  $V$  and  $W$  such that the nonzero elements of  $J$  can easily be extracted from the calculated pair  $(W^T J, JV)$ . Given an arbitrary  $n$ -by- $t_V$  matrix  $V$ , the product  $JV$  can be directly calculated using AD in the “forward mode”; given an arbitrary  $m$ -by- $t_W$  matrix  $W$ , the product  $W^T J$  can be calculated using AD in the “reverse mode,” e.g., [11, 13].

The forward mode of AD allows for the computation of product  $JV$  in time proportional to  $t_V \cdot \omega(F)$ , where  $\omega(F)$  is the time required to evaluate  $F$ . This fact leads to the following practical question. Given the structure of a sparse Jacobian matrix  $J$ , how can a matrix  $V$  be chosen so that the nonzeros of  $J$  can be easily determined from the product  $JV$ ? A good solution is offered by the sparse finite-differencing literature [3, 4, 5, 7, 6, 8] and adapted to the AD setting [1]. Partition the columns of  $J$  into a set of groups  $G_C$ , where the number of groups in  $G_C$  is denoted by  $|G_C|$  such that the columns in each group  $G \in G_C$  are *structurally orthogonal*.<sup>1</sup> Each group  $G \in G_C$  defines a column  $v$  of  $V$ :  $v_i = 1$  if and only if column  $i$  is in

\*Received by the editors November 27, 1995; accepted for publication (in revised form) September 17, 1996; published electronically April 16, 1998. This research was partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-90ER25013 and in part by the Advanced Computing Research Institute, a unit of the Cornell Theory Center which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute.

<http://www.siam.org/journals/sisc/19-4/29534.html>

<sup>†</sup>Computer Science Department and Center for Applied Mathematics, Cornell University, Ithaca, NY 14850 (coleman@cs.cornell.edu).

<sup>‡</sup>Computer Science Department, Cornell University, Ithaca, NY 14850 (verma@cs.cornell.edu).

<sup>1</sup>Two nonzero  $n$ -vectors  $v, w$  are *structurally orthogonal* if  $v_i * w_i = 0$ ,  $i = 1 : n$ .

group  $G$ ; otherwise,  $v_i = 0$ . It is clear that the nonzeros of  $J$  can be immediately “identified” from the computed product  $JV$ . Graph coloring techniques, applied to the column intersection graph of  $J$ , can be used to try and produce a partition  $G_C$  with low cardinality  $|G_C|$ . This, in turn, induces a thin matrix  $V$ ; i.e., construct  $V \in \mathbb{R}^{n \times t_V}$  where  $t_V = |G_C|$ . However, it is not always possible to ensure that  $|G_C|$  is small: consider a sparse matrix  $J$  with a single dense row.

Alternatively, the reverse mode of AD allows for the computation of the product  $W^T J$  in time proportional to  $t_W \cdot \omega(F)$  where  $\omega(F)$  is the time required to evaluate  $F$ , and  $W \in \mathbb{R}^{m \times t_W}$ . The “transpose” of the argument above can lead to an efficient way to determine  $J$ . That is, apply graph coloring techniques to the row intersection graph of  $J$  to induce a thin matrix  $W$ ; compute  $W^T J$  via the reverse mode of AD (takes time proportional to  $t_W \cdot \omega(F)$ ); trivially extract the nonzeros of  $J$  from the computed matrix  $W^T J$ . Of course it is easy to construct examples where defining a thin matrix  $W$  is not possible—consider the case where  $J$  has a dense column.

Clearly there are problems where a row-oriented approach is preferable, and there are problems where a column-oriented approach is better. Unfortunately, it is easy to devise problems where neither approach is satisfactory: let  $J$  have both a dense row and a dense column. This is exactly when it may pay to use both modes of AD simultaneously: compute a pair  $(W^T J, JV)$ , for suitable choices of  $W$  and  $V$ , and extract the nonzero elements of  $J$  from this computed pair of thin matrices.

Our concern in this paper is with efficiency with respect to number of floating point operations or *flops*. We do not concern ourselves with space requirements in this study. However, it should be noted that the reverse mode of AD often requires significantly more space than the forward mode: if space is tight then our suggested approach, which involves application of both forward and reverse modes, may not be possible. There is current research activity on reducing the space requirements of the reverse mode of AD.

We note that an independent proposal regarding sparse Jacobian calculation is made by Hossain and Steihaug [14]: a graph-theoretic interpretation of the direct determination problem is given and an algorithm based on this interpretation is provided. In this paper we proffer a new direct method and we also propose a substitution method, both based directly on the Jacobian structure. We compare our direct and substitution methods, numerically, and we discuss the round-off properties of the substitution method. In addition, we interface our graph coloring software to the automatic differentiator ADOL-C, [12] and report on a few preliminary computational results.

The remainder of the paper is organized as follows. In section 2, we review the relevant aspects of AD, both forward and reverse modes. In section 3, we formalize the combinatorial problems to be solved both from a matrix point of view and in terms of graph theory. We propose both a direct determination problem and a substitution problem. In section 4, we propose “bicoloring” approaches to both the direct determination and “determination by substitution” problems. The bicoloring technique produces matrices  $V$  and  $W$  where  $JV$  is subsequently determined via the forward mode of AD, and  $W^T J$  is determined via the reverse mode. Typically, the column dimensions of  $V$  and  $W$  will be small: the cost of the application of AD is proportional to the sum of the column dimensions of  $V$  and  $W$  (times the work to evaluate  $F$ ).

In section 5 we present and discuss various numerical experiments. The experiments indicate that our bicoloring approach can significantly reduce the cost of de-

termining  $J$  (over one-sided Jacobian determination).

The substitution method we propose consistently outperforms the direct method. However, the substitution calculation increases the chance of round-off contamination. This effect is discussed in section 6. We end the paper in section 7 with some concluding remarks and observations on possible directions for future research. Specifically, we note that while sparsity is a symptom of underlying structure in a nonlinear problem, it is not a necessary symptom. Moreover, it is often possible to exploit structure in the absence of sparsity and apply AD tools “surgically” to efficiently obtain the Jacobian matrix  $J$ .

**2. Basics of AD.** AD is a chain rule based technique for evaluating the derivatives analytically (and hence without any truncation errors) with respect to input variables of functions defined by a high-level language computer program. In this section we briefly review the basics of AD, borrowing heavily from [11, 13].

A program computing the function  $z = F(x)$ ,  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , can be viewed as a sequence of scalar assignments  $v_i = \psi_i \langle v_j \rangle_{j \rightarrow i}$ , where the relation  $j \rightarrow i$  indicates that  $v_j$  is computed before  $v_i$ . Hence, the vector  $v$  can be thought of as a set of ordered variables such that  $v_j, j \rightarrow i$ , is computed before  $v_i$  using the set of variables  $\{v_k | k \rightarrow j\}$ . Here  $\psi_i$  represent elementary functions, which can be arithmetic operations and/or univariate transcendental functions.

Ordering the variables as above, we can partition variables  $v_j$  into three vectors:

$$\begin{aligned} x &\equiv (v_1, v_2, \dots, v_n) \text{ (independent),} \\ y &\equiv (v_{n+1}, v_{n+2}, \dots, v_{n+p}) \text{ (intermediate),} \\ z &\equiv (v_{n+p+1}, v_{n+p+2}, \dots, v_{n+p+m}) \text{ (dependent).} \end{aligned}$$

In general, the number of intermediate variables is much larger than the dimensions of the problem, i.e.,  $p \gg m, n$ .

Assume that all these elementary functions  $\psi_i$  are well defined and have continuous elementary partials  $c_{ij} = \frac{\partial \psi_i}{\partial v_j}$ ,  $j < i$ . Assuming without loss of generality that the dependent variables  $z$  do not themselves occur as arguments of elementary functions, we can combine the partials  $c_{ij}$  into the  $(p+m)$ -by- $(n+p)$  matrix

$$C = (c_{n+i,j})_{\substack{1 \leq j \leq n+p \\ 1 \leq i \leq p+m}}.$$

Unless elementary functions with more than two arguments are included in the library, each row of  $C$  contains either one or two nonzero entries. We define a number<sup>2</sup>

$$q \equiv nnz(C) \leq 2p.$$

Also, since work involved in an elementary function is proportional to the number of arguments, it follows that

$$(2.1) \quad \omega(F) = \sum_{i=1}^{p+m} \omega(\psi_{n+i}) \sim q,$$

where  $\sim$  indicates proportionality. Because of the ordering relation the square matrix  $C$  is upper trapezoidal with  $n-1$  nontrivial superdiagonals. Thus  $C$  can be partitioned

<sup>2</sup>If  $M$  is a matrix or a vector then “ $nnz(M)$ ” is the number of nonzeros in  $M$ .

as

$$(2.2) \quad C = \left( \begin{array}{c|c} & \\ \hline A & L \\ \hline B & M \end{array} \right),$$

where

$$A \in \mathfrak{R}^{p \times n}, \quad B \in \mathfrak{R}^{m \times n}, \quad L \in \mathfrak{R}^{p \times p}, \quad M \in \mathfrak{R}^{m \times p},$$

and  $L$  is strictly lower triangular. Application of the chain rule yields

$$(2.3) \quad \begin{pmatrix} A & L - I \\ B & M \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} 0 \\ -z \end{pmatrix}.$$

If we eliminate the intermediate vector  $\Delta y$  from (2.3), we get an expression (the Schur complement) for the Jacobian:

$$(2.4) \quad J \equiv B - M(L - I)^{-1}A = B + M(I - L)^{-1}A.$$

Since  $I - L$  is a unit lower triangular matrix, the calculation of the matrix products  $\tilde{A} \equiv (I - L)^{-1}A$  and  $\tilde{M} \equiv M(I - L)^{-1}$  leads to two natural ways to compute  $J$ :

$$(2.5) \quad J = B + M\tilde{A} \quad \text{or} \quad J = B + \tilde{M}A.$$

The alternative expressions for  $J$  given in (2.5) define the two basic *modes* of AD, *forward* and *reverse*.

The *forward mode* corresponds to computing the rows of  $\tilde{A}$ , one by one, as the corresponding rows of  $[A, L - I]$  are obtained from successive evaluation of elementary functions. Since this amounts to solutions of  $n$  linear systems with lower-triangular matrix  $[I - L]$ , followed by multiplication of dense columns of  $\tilde{A}$  by  $M$ , the total computational effort is roughly  $n \cdot q$  or  $n \cdot \omega(F)$ .

The *reverse mode* corresponds to computing  $\tilde{M}$  as a solution to the linear system  $(I - L)^T \tilde{M}^T = M^T$ . This back-substitution process can begin only after all elementary functions and their partial derivatives have been evaluated. Since this amounts to the solution of  $m$  linear systems with lower triangular matrix  $[I - L]$ , followed by multiplication of dense rows of  $\tilde{M}$  by  $A$ , total computational effort is roughly  $m \cdot q$  or  $m \cdot \omega(F)$ .

We are interested in computing products of the form  $JV$  and  $W^T J$ . Product  $JV$  can be computed

$$JV = BV + M[(I - L)^{-1}(AV)],$$

which can clearly be done in time proportional to  $t_V \cdot \omega(F)$  when  $V \in \mathfrak{R}^{n \times t_V}$ . Analogously, product  $W^T J$  can be computed

$$W^T J = W^T B + [(W^T M)(I - L)^{-1}]A,$$

which can be done in time proportional to  $t_W \cdot \omega(F)$  assuming  $W \in \mathfrak{R}^{m \times t_W}$ .

An important subcase worthy of special attention is when  $F$  is a scalar function, i.e.,  $m = 1$ . In this case the Jacobian matrix corresponds to the transpose of the gradient of  $F$  and is a single row vector. Note that the complexity arguments applied to this case imply that the reverse mode of AD yields the gradient in time proportional to  $\omega(F)$ , whereas the forward mode costs  $n \cdot \omega(F)$ . The efficiency of the forward mode evaluation of the gradient can be dramatically increased—i.e., the dependence on  $n$  is removed—if  $F$  has structure that can be exploited [2].

**3. Partition problems and graph theory.** Our basic task is to efficiently determine thin matrices  $V, W$  so that the nonzero elements of  $J$  can be readily extracted from the information  $(W^T J, JV)$ . The pair of matrices  $(W^T J, JV)$  is obtained from the application of both modes of AD: matrix  $W^T J$  is computed by the reverse mode; the forward mode determines  $JV$ . The purpose of this section is to more rigorously formulate the question of determining suitable matrices  $V, W$ , first in the language of “partitions,” and then using graph theoretic concepts.

We begin with an example illustrating the usefulness of simultaneously applying both modes of AD, forward and reverse. Consider the following  $n$ -by- $n$  Jacobian, symmetric in structure but not in value:

$$(3.1) \quad J = \begin{pmatrix} \square & \triangle & \triangle & \triangle & \triangle \\ \square & \diamond & & & \\ \square & & \diamond & & \\ \square & & & \diamond & \\ \square & & & & \diamond \end{pmatrix}.$$

It is clear that a partition of columns consistent with the direct determination of  $J$  requires  $n$  groups. This is because a “consistent column partition” requires that each group contain columns that are structurally orthogonal, and the presence of a dense row implies each group consists of exactly one column. Therefore, if matrix  $V$  corresponds to a “consistent column partition,” then  $V$  has  $n$  columns and the work to evaluate  $JV$  by the forward mode of AD is proportional to  $n \cdot \omega(F)$ . By a similar argument, and the fact that a column of  $J$  is dense, a “consistent row partition” requires  $n$  groups. Therefore, if matrix  $W$  corresponds to a “consistent row partition,” then  $W$  has  $n$  rows and the work to evaluate  $W^T J$  by the reverse mode of AD is proportional to  $n \cdot \omega(F)$ .

**DEFINITION 1.** A bipartition of a matrix  $A$  is a pair  $(G_R, G_C)$  where  $G_R$  is a row partition of a subset of the rows of  $A$  and  $G_C$  is a column partition of a subset of the columns of  $A$ .

In this example the use of a bipartition dramatically decreases the amount of work required to determine  $J$ . Specifically, the total amount of work required is proportional to  $3 \cdot \omega(F)$ . To see this define  $V = (e_1, e_2 + e_3 + e_4 + e_5)$ ;  $W = (e_1)$ , where we follow the usual convention of representing the  $i$ th column of the identity matrix with  $e_i$ . Clearly elements  $\square, \diamond$  are directly determined from the product  $JV$ ; elements  $\triangle$  are directly determined from the product  $W^T J$ .

The basic idea is to partition the rows into a set of groups  $G_R$  and the columns into a set of groups  $G_C$ , with  $|G_R| + |G_C|$  as small as possible, such that every nonzero element of  $J$  can be directly determined from either a group in  $G_R$  or a group in  $G_C$ .

**DEFINITION 2.** A bipartition  $(G_R, G_C)$  of a matrix  $A$  is consistent with direct determination if for every nonzero  $a_{ij}$  of  $A$ , either column  $j$  is in a group of  $G_C$  which has no other column having a nonzero in row  $i$  or row  $i$  is in a group of  $G_R$  which has no other rows having a nonzero in column  $j$ .

Clearly, given a bipartition  $(G_R, G_C)$  consistent with direct determination, we can trivially construct matrices  $W \in \mathbb{R}^{m \times |G_R|}, V \in \mathbb{R}^{n \times |G_C|}$  such that  $A$  can be directly determined from  $(W^T A, AV)$ .

If we relax the restriction that each nonzero element of  $J$  be determined directly, then it is possible that the work required to evaluate the nonzeros of  $J$  can be further reduced. For example we could allow for a "substitution" process when recovering the nonzeros of  $J$  from the pair  $(W^T J, JV)$ . Figures 3.1 and 3.2 illustrate that a substitution method can win over direct determination: Figure 3.1 corresponds to direct determination; Figure 3.2 corresponds to determination using substitution.

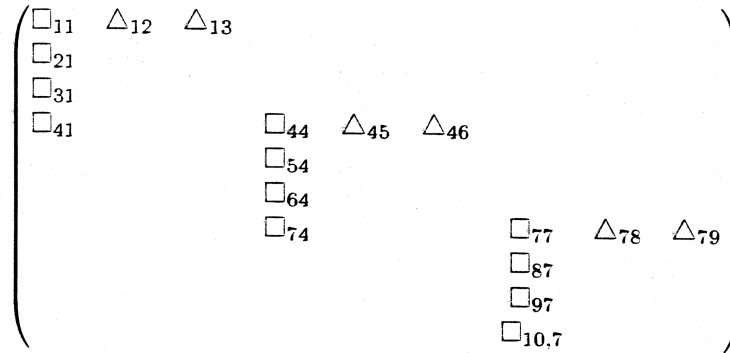


FIG. 3.1. Optimal partition for direct method.

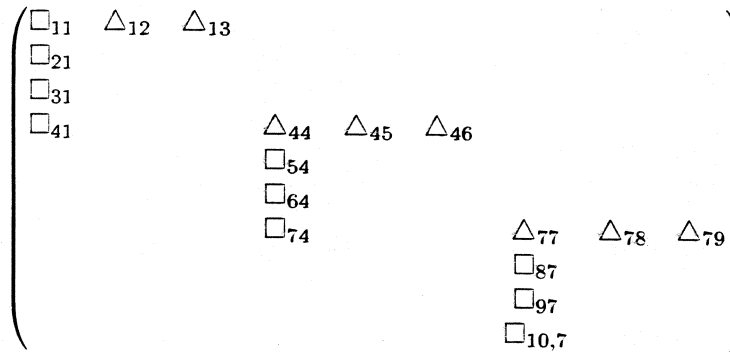


FIG. 3.2. Optimal partition for substitution method.

In both cases elements labelled  $\square$  are computed from the column grouping, i.e., calculated using the product  $JV$ ; elements labelled  $\triangle$  are calculated from the row groupings, i.e., calculated using the product  $W^T J$ . The matrix in Figure 3.1 indicates that we can choose  $G_C$  with  $|G_C| = 2$  and  $G_R$  with  $|G_R| = 1$  and determine all elements directly. That is, choose  $V = (e_1 + e_7, e_4)$ ; choose  $W = (e_1 + e_4 + e_7)$ . Therefore, in this case the work to compute  $J$  satisfies  $\omega(J) \sim 3 \cdot \omega(F)$ . Note that some elements can be determined twice, e.g.,  $J_{11}$ .

However, the matrix in Figure 3.2 shows how to obtain the nonzeros of  $J$ , using substitution, in work proportional to  $2 \cdot \omega(F)$ . Let  $V = W = (e_1 + e_4 + e_7)$ . Let  $p_1$  be the (forward) computed vector  $p_1 = JV$ ; let  $p_2^T$  be the (reverse) computed row

vector  $p_2^T = W^T J$ . Then

$$p_1 = \begin{pmatrix} \square_{11} \\ \square_{21} \\ \square_{31} \\ \square_{41} + \triangle_{44} \\ \square_{54} \\ \square_{64} \\ \square_{74} + \triangle_{77} \\ \square_{87} \\ \square_{97} \\ \square_{10,7} \end{pmatrix}, \quad p_2^T = (\triangle_{11} + \square_{41}, \triangle_{12}, \triangle_{13}, \triangle_{44} + \square_{74}, \triangle_{45}, \triangle_{46}, \triangle_{77}, \triangle_{78}, \triangle_{79}).$$

Most of the nonzero elements are determined directly (no conflict). The remaining elements can be resolved,

$$\square_{74} = p_1(7) - p_2(7); \quad \triangle_{44} = p_2(4) - \square_{74}; \quad \square_{41} = p_1(4) - \triangle_{44}.$$

It is easy to extend this example so that the difference between the number of groups needed, between substitution and direct determination, increases with the dimension of the matrix. For example, a block generalization is illustrated in Figure 3.3: if we assume  $l > 2w$  it is straightforward to verify that in the optimal partition the number of groups needed for direct determination will be  $3w$  and determination by substitution requires  $2w$  groups.

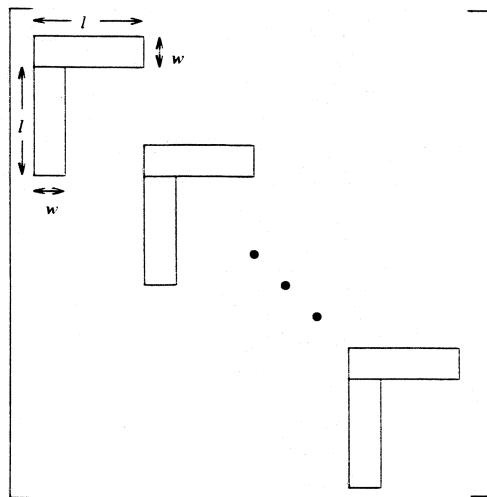


FIG. 3.3. Block example.

**DEFINITION 3.** A bipartition  $(G_R, G_C)$  of a matrix  $A$  is consistent with determination by substitution if there exists an ordering  $\pi$  on elements  $a_{ij}$  such that for every nonzero  $a_{ij}$  of  $A$ , either column  $j$  is in a group where all nonzeros in row  $i$ , from other columns in the group, are ordered lower than  $a_{ij}$  or row  $i$  is in a group where all the nonzeros in column  $j$ , from other rows in the group, are ordered lower than  $a_{ij}$ .

In the usual way we can construct a matrix  $V$  from the column grouping  $G_C$  and a matrix  $W$  from the row grouping  $G_R$ : for example, to construct the columns of  $V$  associate with each group in  $G_C$  a Boolean vector, with unit entries indicating

membership of the corresponding columns. We can now state our main problem(s) more precisely.

*The bipartition problem (direct).* Given a matrix  $A$ , obtain a bipartition  $(G_R, G_C)$  consistent with direct determination such that total number of groups,  $|G_R| + |G_C|$ , is minimized.

*The bipartition problem (substitution).* Given a matrix  $A$ , obtain a bipartition  $(G_R, G_C)$  consistent with determination by substitution such that the total number of groups  $|G_R| + |G_C|$  is minimized.

The bipartition problems can also be expressed in terms of graphs and graph coloring. This graph view is important in that it more readily exposes the relationship of the bipartition problems with the combinatorial approaches used in the sparse finite-differencing literature, e.g., [3, 4, 5, 7, 6]. However, we note that the remainder of this paper, with the exception of the error analysis in section 6, does not rely directly on this graph interpretation.

To begin, we need the usual notion of a coloring of the vertices of a graph, the definition of a bipartite graph, and the concept of *path* coloring [3, 7] specialized to the bipartite graph case.

A  $p$ -coloring of a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  is the set of vertices or nodes and  $\mathcal{E}$  is the set of edges, is a function

$$\phi : \mathcal{V} \rightarrow \{1, 2, \dots, p\}$$

such that  $\phi(u) \neq \phi(v)$  if  $(u, v) \in \mathcal{E}$ . The *chromatic number*  $\chi(\mathcal{G})$  is the smallest  $p$  for which  $\mathcal{G}$  has a  $p$ -coloring. A  $p$ -coloring  $\phi$  of  $\mathcal{G}$  induces a partition of vertices  $\mathcal{V}$  into  $p$  groups  $G_k, k = 1, 2, \dots, p$ , such that

$$G_k = \{u \in \mathcal{V} : \phi(u) = k\}.$$

Given a matrix  $A \in \mathbb{R}^{m \times n}$ , define a bipartite graph  $\mathcal{G}_b(A) = ([\mathcal{V}_1, \mathcal{V}_2], \mathcal{E})$  where  $\mathcal{V}_1 = \{r_1, r_2, \dots, r_m\}$ ,  $\mathcal{V}_2 = \{c_1, c_2, \dots, c_n\}$ ,  $c_j$  corresponds to the  $j$ th column of  $A$ , and  $r_i$  corresponds to the  $i$ th row of  $A$ . There is an edge,  $(r_i, c_j) \in \mathcal{E}$  if  $a_{ij}$  is nonzero in  $A$ .

In [3, 7] a *path  $p$ -coloring* of a graph is defined to be a vertex coloring using  $p$  colors with the additional property that every path of at least three *edges* uses at least three colors. Here we need a slight modification of that concept appropriate for the direct determination problem. We note that "color 0" is distinguished in that it corresponds to the lack of a true color assignment: i.e.,  $\phi(i) = 0$  indicates that vertex  $i$  is not assigned a color.

DEFINITION 4. Let  $\mathcal{G}_b = ([\mathcal{V}_1, \mathcal{V}_2], \mathcal{E})$  be a bipartite graph. A mapping  $\phi : [\mathcal{V}_1, \mathcal{V}_2] \rightarrow \{0, 1, \dots, p\}$  is a bipartite path  $p$ -coloring of  $\mathcal{G}_b$  if the following conditions hold.

1. Adjacent vertices have different assignments, i.e., if  $(i, j) \in \mathcal{E}$  then  $\phi(i) \neq \phi(j)$ .
2. The set of positive colors used by vertices in  $\mathcal{V}_1$  is disjoint from the set of positive colors used by vertices in  $\mathcal{V}_2$ , i.e.,  $i \in \mathcal{V}_1, j \in \mathcal{V}_2 \Rightarrow \{\phi(i) \neq \phi(j) \text{ or } \phi(i) = \phi(j) = 0\}$ .
3. If vertices  $i$  and  $j$  are adjacent to vertex  $k$  with  $\phi(k) = 0$ , then  $\phi(i) \neq \phi(j)$ .
4. Every path of three edges uses at least three colors.

The smallest number for which graph  $\mathcal{G}_b$  is bipartite path  $p$ -colorable is denoted by  $\chi_p(\mathcal{G}_b)$ . Figure 3.4 shows a valid bipartite path  $p$ -coloring. Numbers adjacent to the vertices denote colors. We note that Hossain and Steihaug [14] define a similar



concept. However, their definition of path  $p$ -coloring does not allow for the “uncolor assignment”; i.e.,  $\phi(i) = 0$ . Consequently, a technique to remove empty groups is needed [14].

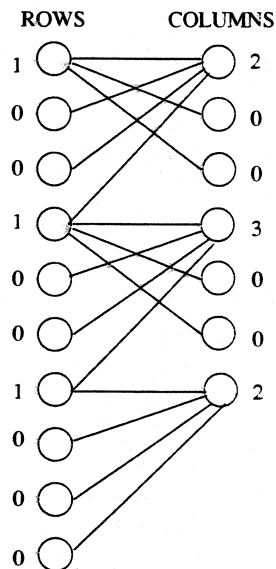


FIG. 3.4. A valid bipartite path coloring.

We are now in position to state the graph analogy to the concept of a bipartition consistent with direct determination.

**THEOREM 3.1.** *Let  $A$  be an  $m$ -by- $n$  matrix with corresponding bipartite graph  $\mathcal{G}_b(A) = ([\mathcal{V}_1, \mathcal{V}_2], \mathcal{E})$ . The mapping  $\phi : [\mathcal{V}_1, \mathcal{V}_2] \rightarrow \{0, 1, \dots, p\}$  induces a bipartition  $(G_R, G_C)$ , with  $|G_R| + |G_C| = p$ , consistent with direct determination if and only if  $\phi$  is a bipartite path  $p$ -coloring of  $\mathcal{G}_b(A)$ .*

*Proof.* ( $\Leftarrow$ ) Assume that  $\phi$  is a bipartite path  $p$ -coloring of  $\mathcal{G}_b(A)$ , inducing a bipartition  $(G_R, G_C)$  of rows and columns of  $A$ . If this bipartition is not consistent with direct determination, then there is a nonzero element  $a_{ij}$  in the matrix for which the definition “either column  $j$  is in a group of  $G_C$  which has no other column having a nonzero in row  $i$ , or row  $i$  is in a group of  $G_R$  which has no other rows having a nonzero in column  $j$ ” doesn’t hold. This can happen only if one of the following cases holds:

- $\phi(r_i) = 0, \phi(c_j) \neq 0$  and there exists a column  $q$  with  $a_{iq} \neq 0$  such that  $\phi(c_j) = \phi(c_q)$ . But this contradicts condition 3.
- $\phi(c_j) = 0, \phi(r_i) \neq 0$  and there exists a row  $p$  with  $a_{pj} \neq 0$  such that  $\phi(r_i) = \phi(r_p)$ . But this contradicts condition 3.
- $\phi(r_i) \neq 0, \phi(c_j) \neq 0$ . There exists a column  $q$  and a row  $p$  such that columns  $j$  and  $q$  are in the same group with  $a_{iq} \neq 0$  and rows  $i$  and  $p$  are in the same group with  $a_{pj} \neq 0$ . This implies  $\phi$  is a 2-coloring of the path  $(r_p - c_j - r_i - c_q)$ , a contradiction of condition 4.

( $\Rightarrow$ ) Conversely, assume that  $\phi$  induces a bipartition consistent with direct determination of  $A$ . It is clear that conditions 1–3 must be satisfied. It remains for us to establish condition 4: i.e., every path of three edges uses at least three colors. Suppose there is a bicolored path:  $r_i - c_j - r_k - c_l$ , where  $\phi(r_i) = \phi(r_k)$ ,  $\phi(c_j) = \phi(c_l)$ . Clearly by condition 3 the two colors on this path are positive. It is easy to see that element

$a_{jk}$  cannot be determined directly: there is a conflict in row group  $\phi(r_i) = \phi(c_j)$  and there is a conflict in column group  $\phi(c_k) = \phi(c_l)$ , and these are the only two chances to determine  $a_{jk}$ .  $\square$

To capture the substitution notion the cyclic  $p$ -coloring definition [3] is modified slightly and applied to a bipartite graph.

**DEFINITION 5.** Let  $A$  be an  $m$ -by- $n$  matrix with corresponding bipartite graph  $\mathcal{G}_b(A) = ([\mathcal{V}_1, \mathcal{V}_2], \mathcal{E})$ . A mapping  $\phi : [\mathcal{V}_1, \mathcal{V}_2] \rightarrow \{0, 1, \dots, p\}$  is a bipartite cyclic  $p$ -coloring of  $\mathcal{G}_b$  if the following conditions hold.

1. Adjacent vertices have different assignments; i.e., if  $(i, j) \in \mathcal{E}$  then  $\phi(i) \neq \phi(j)$ .
2. The set of positive colors used by vertices in  $\mathcal{V}_1$  is disjoint from the set of positive colors used by vertices in  $\mathcal{V}_2$ ; i.e.,  $i \in \mathcal{V}_1, j \in \mathcal{V}_2 \Rightarrow \{\phi(i) \neq \phi(j) \text{ or } \phi(i) = \phi(j) = 0\}$ .
3. If vertices  $i$  and  $j$  are adjacent to vertex  $k$  with  $\phi(k) = 0$ , then  $\phi(i) \neq \phi(j)$ .
4. Every cycle uses at least at least three colors.

The smallest number for which graph  $\mathcal{G}_b$  is bipartite cyclic  $p$ -colorable is denoted by  $\chi_c(\mathcal{G}_b)$ . Figure 3.5 shows a valid bipartite cyclic  $p$ -coloring. Note that only two colors are necessary, whereas the bipartite path  $p$ -coloring in Figure 3.4 requires three colors. The notion of a bipartition consistent with determination via substitution can now be cleanly stated in graph-theoretic terms.

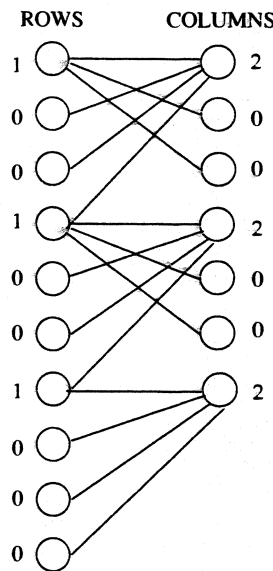


FIG. 3.5. A valid bipartite cyclic coloring.

**THEOREM 3.2.** Let  $A$  be an  $m$ -by- $n$  matrix with corresponding bipartite graph  $\mathcal{G}_b(A) = ([\mathcal{V}_1, \mathcal{V}_2], \mathcal{E})$ . The mapping  $\phi : (\mathcal{V}_1, \mathcal{V}_2) \rightarrow \{0, 1, \dots, p\}$  induces a bipartition  $(G_R, G_C)$ , with  $|G_R| + |G_C| = p$ , consistent with determination by substitution if and only if  $\phi$  is a bipartite cyclic  $p$ -coloring of  $\mathcal{G}_b(A)$ .

*Proof.*  $(\Rightarrow)$  Assume  $\phi$  induces a bipartition consistent with determination by substitution but  $\phi$  is not a bipartite cyclic  $p$ -coloring of  $\mathcal{G}_b(A)$ . Clearly conditions 1-2 must hold; it is easy to see that if condition 3 doesn't hold then not all nonzero elements can be determined. The only nontrivial violation is condition 4: there is a cycle which has only two colors; i.e., all the vertices  $\in \mathcal{V}_1$  in the cycle have the same

color  $c_1$ , and all the vertices  $\in V_2$  in the cycle have the same color  $c_2$ . Note that neither  $c_1$  nor  $c_2$  can be equal to 0, since a node colored 0 in a cycle would imply that its adjacent vertices are both colored differently, implying that there are at least three colors. Consider the submatrix  $A_s$  of  $A$ , corresponding to this cycle. Submatrix  $A_s$  has at least two nonzeros in each row and in each column, since each vertex has degree 2 in the cycle. But since we are considering substitution methods only, at least one element of  $A_s$  needs to be computed directly. Clearly there is no way to get any element of this submatrix directly, a contradiction.

( $\Leftarrow$ ) Conversely, assume that  $\phi$  is a bipartite cyclic  $p$ -coloring of  $G_b(A)$  but that bipartition  $(G_R, G_C)$  induced by  $\phi$  is not consistent with determination by substitution. But edges (nonzeros) with one end assigned color "0" can be determined directly: by the definition of bicoloring there will be no conflict. Moreover, every pair of positive colors induces a forest (i.e., a collection of trees); therefore, as shown in [3], the edges (nonzeros) in the induced forest can be resolved via substitution.  $\square$

The two bipartition problems can now be simply stated in terms of optimal bipartite path and cyclic  $p$ -colorings.

*The bipartite path  $p$ -coloring problem.* Determine a bipartite path  $p$ -coloring of  $\mathcal{G}_b(A)$  with the smallest possible value of  $p$ ; i.e.,  $p = \chi_p(\mathcal{G}_b)$ .

*The bipartite cyclic  $p$ -coloring problem.* Determine a bipartite cyclic  $p$ -coloring of  $\mathcal{G}_b(A)$  with the smallest possible value of  $p$ ; i.e.,  $p = \chi_c(\mathcal{G}_b)$ .

The graph theoretic view is useful for both analyzing the complexity of the combinatorial problem and suggesting possible algorithms, exact or heuristic. In fact, using the  $p$ -coloring notions discussed above, and an approach similar to that taken in [3], it is easy to show that corresponding decision problems are NP-complete. Garey and Johnson [9] provide an excellent introduction to complexity theory.

*Bipartite cyclic  $p$ -coloring decision problem (CCDP).* Given an integer  $p \geq 3$  and an arbitrary bipartite graph  $\mathcal{G}$ , is it possible to assign a cyclic  $p$ -coloring to nodes of  $\mathcal{G}$ ?

*Bipartite path  $p$ -coloring decision problem (PCDP).* Given an integer  $p \geq 3$  and an arbitrary bipartite graph  $\mathcal{G}$ , is it possible to assign a bipartite path  $p$ -coloring to nodes of  $\mathcal{G}$ ?

The proofs are a straightforward adaptation of those in [3], and we omit them here. The upshot of these (negative) complexity results is that in practice we must turn our attention to (fast) heuristics to approximately solve the cyclic and path coloring problems. In the next section we present simple, effective, and "easy-to-visualize" heuristics for these two combinatorial problems.

Finally, it is easy to establish a partial ordering of chromatic numbers:

$$(3.2) \quad \chi_c(\mathcal{G}_b(A)) \leq \chi_p(\mathcal{G}_b(A)) \leq \min(\chi(\mathcal{G}(A^T)), \chi(\mathcal{G}(A))),$$

where  $\mathcal{G}(M)$  refers to the column intersection graph of matrix  $M$ ;  $\chi(\mathcal{G}(M))$  is the (usual) chromatic number of graph  $\mathcal{G}(M)$ .

The first inequality in (3.2) holds because if  $\phi$  is a bipartite path  $p$ -coloring then  $\phi$  is a bipartite cyclic  $p$ -coloring; the second inequality holds because a trivial way to satisfy conditions 1–4 of Definition 4 is to assign "0" to all the row (column) nodes and then use positive colors on all the column (row) nodes to satisfy condition 3. This ordering supports the tenet that use of bipartition/bicoloring is never worse than one-sided calculation and that a substitution approach is never worse than a direct approach (in principle).

**4. Bicoloring.** The two combinatorial problems we face, corresponding to direct determination and determination by substitution, can both be approached in the

following way. First, permute and partition the structure of  $J$ :  $\bar{J} = P \cdot J \cdot Q = [J_C | J_R]$ , as indicated in Figure 4.1. The construction of this partition is crucial; however, we postpone that discussion until after we illustrate its utility. Assume  $P = Q = I$  and  $J = [J_C | J_R]$ .

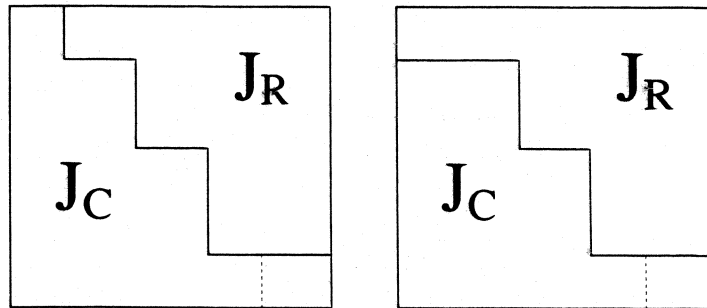


FIG. 4.1. Possible partitions of the matrix  $\bar{J} = P \cdot J \cdot Q$ .

Second, define appropriate intersection graphs  $\mathcal{G}_C^I, \mathcal{G}_R^I$  based on the partition  $[J_C | J_R]$ ; a coloring of  $\mathcal{G}_C^I$  yields a partition of a subset of the columns  $G_C$ , which defines matrix  $V$ . Matrix  $W$  is defined by a partition of a subset of rows  $G_R$ , which is given by a coloring of  $\mathcal{G}_R^I$ . We call this double coloring approach *bicoloring*. The difference between the direct and substitution cases is in how the intersection graphs  $\mathcal{G}_C^I, \mathcal{G}_R^I$  are defined and how the nonzeros of  $J$  are extracted from the pair  $(W^T J, JV)$ .

**4.1. Direct determination.** In the direct case the intersection graph  $\mathcal{G}_C^I$  is defined:  $\mathcal{G}_C^I = (\mathcal{V}_C^I, \mathcal{E}_C^I)$  where

- vertex  $j \in \mathcal{V}_C^I$  if  $nnz(\text{column } j \cap J_C) \neq 0$ ;
- $(r, s) \in \mathcal{E}_C^I$  if  $r \in \mathcal{V}_C^I, s \in \mathcal{V}_C^I, \exists k$  such that  $J_{kr} \neq 0, J_{ks} \neq 0$  and either  $(k, r) \in J_C$  or  $(k, s) \in J_C$ .

The key point in the construction of graph  $\mathcal{G}_C^I$ , and why  $\mathcal{G}_C^I$  is distinguished from the usual column intersection graph, is that columns  $r$  and  $s$  are said to intersect if and only if their nonzero locations partially overlap in  $J_C$ : i.e., columns  $r$  and  $s$  intersect if  $J_{kr} \cdot J_{ks} \neq 0$  and either  $(k, r) \in J_C$  or  $(k, s) \in J_C$  for some  $k$ .

The “transpose” of the procedure above is used to define  $\mathcal{G}_R^I = (\mathcal{V}_R^I, \mathcal{E}_R^I)$ . Specifically,  $\mathcal{G}_R^I = (\mathcal{V}_R^I, \mathcal{E}_R^I)$  where

- vertex  $i \in \mathcal{V}_R^I$  if  $nnz(\text{row } i \cap J_R) \neq 0$ ;
- $(r, s) \in \mathcal{E}_R^I$  if  $r \in \mathcal{V}_R^I, s \in \mathcal{V}_R^I, \exists k$  such that  $J_{rk} \neq 0, J_{sk} \neq 0$  and either  $(r, k) \in J_R$  or  $(s, k) \in J_R$ .

In this case the reason graph  $\mathcal{G}_R^I$  is distinguished from the usual row intersection graph is that rows  $r$  and  $s$  are said to intersect if and only if their nonzero locations partially overlap in  $J_R$ : rows  $r$  and  $s$  intersect if  $J_{rk} \cdot J_{sk} \neq 0$  and either  $(r, k) \in J_R$  or  $(s, k) \in J_R$  for some  $k$ .

The bipartition  $(G_R, G_C)$ , induced by coloring of graphs  $\mathcal{G}_R^I$  and  $\mathcal{G}_C^I$ , is consistent with direct determination of  $J$ . To see this consider a nonzero element  $(i, j) \in J_C$ : column  $j$  is in a group of  $G_C$  (corresponding to a color) with the property that no other column in  $G_C$  has a nonzero in row  $i$ . Hence, element  $(i, j)$  can be directly determined. Analogously, consider a nonzero element  $(r, s) \in J_R$ : row  $r$  will be in a group of  $G_R$  (corresponding to a color) with the property that no other row in  $G_R$  has a nonzero in column  $s$ . Hence, element  $(r, s)$  can be directly determined. Since every nonzero of  $J$  is covered, the result follows.

*Example.* Consider the example Jacobian matrix structure shown in Figure 4.2 with the partition  $(J_C, J_R)$  shown.

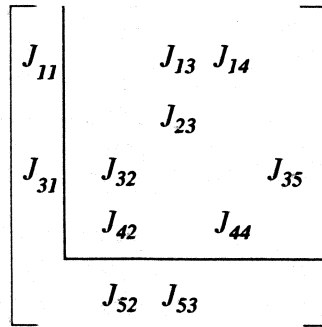


FIG. 4.2. Example partition.

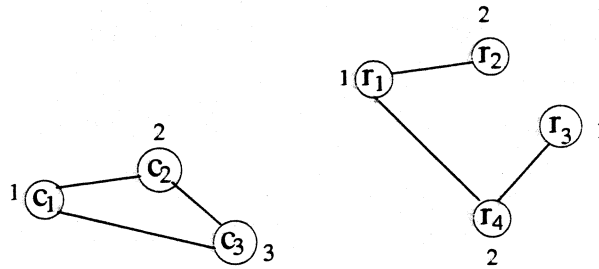


FIG. 4.3. Graphs  $\mathcal{G}_C$  and  $\mathcal{G}_R$  (direct approach).

The graphs  $\mathcal{G}_C$  and  $\mathcal{G}_R$  formed by the algorithm outlined above are given in Figure 4.3. Coloring  $\mathcal{G}_C$  requires three colors, while  $\mathcal{G}_R$  can be colored in two. Boolean matrices  $V$  and  $W$  can be formed in the usual way: each column corresponds to a group (or color), and unit entries indicate column (or row) membership in that group:

$$V = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad JV = \begin{pmatrix} J_{11} & 0 & \times \\ 0 & 0 & \times \\ J_{31} & \times & \times \\ 0 & \times & 0 \\ 0 & J_{52} & J_{53} \end{pmatrix},$$

$$W = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad W^T J = \begin{pmatrix} \times & J_{32} & J_{13} & J_{14} & J_{35} \\ \times & J_{42} & J_{23} & J_{44} & 0 \end{pmatrix}.$$

Clearly, all nonzero entries of  $J$  can be identified in either  $JV$  or  $W^T J$ .

**4.2. Determination by substitution.** The basic advantage of determination by substitution in conjunction with partition  $J = [J_C | J_R]$  is that sparser intersection

graphs  $\mathcal{G}_C^I, \mathcal{G}_R^I$  can be used. Sparser intersection graphs mean thinner matrices  $V, W$  which, in turn, result in reduced cost.

In the substitution case the intersection graph  $\mathcal{G}_C^I$  is defined as  $\mathcal{G}_C^I = (\mathcal{V}_C^I, \mathcal{E}_C^I)$  where

- vertex  $j \in \mathcal{V}_C^I$  if  $nnz(\text{column } j \cap J_C) \neq 0$ , and
- $(r, s) \in \mathcal{E}_C^I$  if  $r \in \mathcal{V}_C^I, s \in \mathcal{V}_C^I, \exists k$  such that  $J_{kr} \neq 0, J_{ks} \neq 0$  and both  $(k, r) \in J_C, (k, s) \in J_C$ .

Note that the intersection graph  $\mathcal{G}_C^I = (\mathcal{V}_C^I, \mathcal{E}_C^I)$  captures the notion of two columns intersecting if there is overlap in nonzero structure in  $J_C$ : columns  $r$  and  $s$  intersect if  $J_{kr} \cdot J_{ks} \neq 0$  and both  $(k, r) \in J_C, (k, s) \in J_C$  for some  $k$ . It is easy to see that  $\mathcal{E}_C^I$  is a subset of the set of edges used in the direct determination case.

The “transpose” of the procedure above is used to define  $\mathcal{G}_R^I = (\mathcal{V}_R^I, \mathcal{E}_R^I)$ . Specifically,  $\mathcal{G}_R^I = (\mathcal{V}_R^I, \mathcal{E}_R^I)$  where

- vertex  $i \in \mathcal{V}_R^I$  if row  $i \in J_R$  and  $nnz(\text{row } i \cap J_R) \neq 0$ , and
- $(r, s) \in \mathcal{E}_R^I$  if  $r \in \mathcal{V}_R^I, s \in \mathcal{V}_R^I, \exists k$  such that  $J_{rk} \neq 0, J_{sk} \neq 0$  and both  $(r, k) \in J_R, (s, k) \in J_R$ .

The intersection graph  $\mathcal{G}_R^I = (\mathcal{V}_R^I, \mathcal{E}_R^I)$  captures the notion of two rows intersecting if there is overlap in nonzero structure in  $J_R$ : rows  $r$  and  $s$  intersect if  $J_{rk} \cdot J_{sk} \neq 0$  and  $(r, k) \in J_R$  and  $(s, k) \in J_R$  for some  $k$ . It is easy to see that  $\mathcal{E}_R^I$  is a subset of the set of edges used in direct determination.

All the elements of  $J$  can be determined from  $(W^T J, J V)$  by a substitution process. This is evident from the illustrations in Figure 4.4.

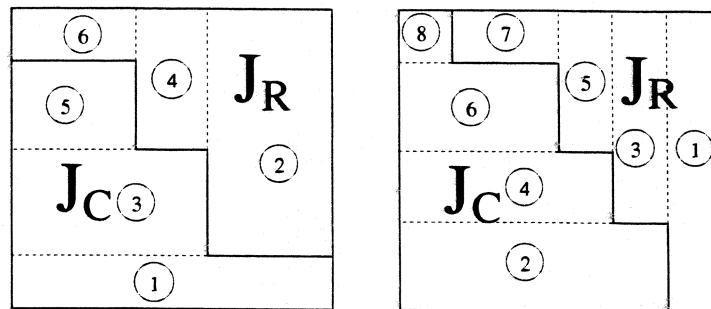


FIG. 4.4. Substitution orderings.

Figure 4.4 illustrates two of four possible nontrivial types of partitions. In both cases it is clear that nonzero elements in the section labelled “1” can be solved for directly—by the construction process they will be in different groups. Nonzero elements in “2” can either be determined directly or will depend on elements in section “1”. But elements in section “1” are already determined (directly) and so, by substitution, elements in “2” can be determined after “1”. Elements in section “3” can then be determined, depending only on elements in “1” and “2”, and so on until the entire matrix is resolved.

*Example.* Consider again the example Jacobian matrix structure shown in Figure 4.2. Column and row intersection graphs corresponding to substitution are given in Figure 4.5. Note that  $\mathcal{G}_C$  is disconnected and requires two colors;  $\mathcal{G}_R$  is a simple chain and also requires two colors.

The coloring of  $\mathcal{G}_C$  and  $\mathcal{G}_R$  leads to the following matrices  $V, W$  and the resulting

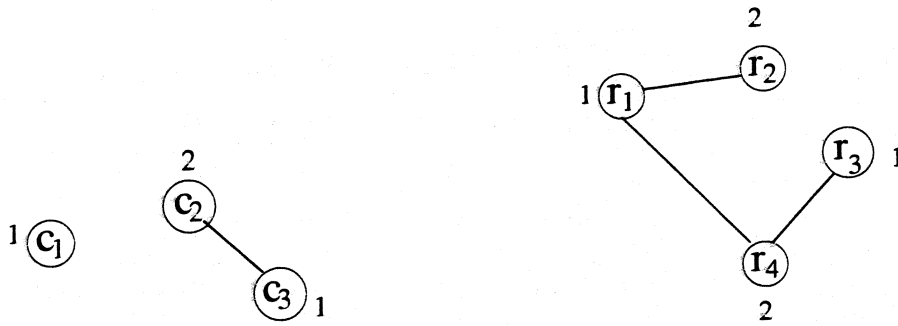


FIG. 4.5. Graphs  $\mathcal{G}_C$  and  $\mathcal{G}_R$  for substitution process.

computation of  $JV, W^T J$ :

$$V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \quad JV = \begin{pmatrix} J_{11} + J_{13} & 0 \\ \times & 0 \\ J_{31} & \times \\ 0 & \times \\ J_{53} & J_{52} \end{pmatrix},$$

$$W = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad W^T J = \begin{pmatrix} \times & J_{32} & J_{13} & J_{14} & J_{35} \\ \times & J_{42} & J_{23} & J_{44} & 0 \end{pmatrix}.$$

It is now easy to verify that all nonzeros of  $J$  can be determined via substitution.

**4.3. How to partition  $J$ .** We now consider the problem of obtaining a useful partition  $[J_C|J_R]$  and corresponding permutation matrices  $P, Q$ , as illustrated in Figure 4.1. A simple heuristic is proposed based on the knowledge that the subsequent step, in both the direct and the substitution method, is to color intersection graphs based on this partition.

Algorithm MNCO builds partition  $J_C$  from the bottom up and partition  $J_R$  from right to left. At the  $k$ th major iteration either a new row is added to  $J_C$  or a new column is added to  $J_R$ ; the choice depends on considering a lower bound effect:

$$\rho(J_R^T) + \max(\rho(J_C), \text{nnz}(r)) < (\rho(J_C) + \max(\rho(J_R^T), \text{nnz}(c))),$$

where  $\rho(A)$  is the maximum number of nonzeros in any row of matrix  $A$ ,  $r$  is a row under consideration to be added to  $J_C$ , and  $c$  is a column under consideration to be added to  $J_R$ . Hence, the number of colors needed to color  $\mathcal{G}_C^I$  is bounded below by  $\rho(J_C)$ ; the number of colors needed to color  $\mathcal{G}_R^I$  is bounded below by  $\rho(J_R^T)$ .

In algorithm MNCO, matrix  $M = J(R, C)$  is the submatrix of  $J$  defined by row indices  $R$  and column indices  $C$ :  $M$  consists of rows and columns of  $J$  not yet assigned to either  $J_C$  or  $J_R$ .

**MINIMUM NONZERO COUNT ORDERING (MNCO)**

1. Initialize  $R = (1 : m), C = (1 : n), M = J(R, C)$
2. Find  $r \in R$  with fewest nonzeros in  $M$
3. Find  $c \in C$  with fewest nonzeros in  $M$

```

4. Repeat Until  $M = \emptyset$ 
   if  $\rho(J_R^T) + \max(\rho(J_C), nnz(r)) < (\rho(J_C) + \max(\rho(J_R^T), nnz(c)))$ 
   (LB)
      $J_C = J_C \cup (r \cap C)$ 
      $R = R - \{r\}$ 
   else
      $J_R = J_R \cup (c \cap R)$ 
      $C = C - \{c\}$ 
   end if
    $M = J(R, C)$ 
end repeat

```

Note that, upon completion,  $J_R, J_C$  have been defined; the requisite permutation matrices are implicitly defined by the ordering chosen in MNCO.

**5. Bicoloring performance.** In this section we present results of numerical experiments. The work required to compute the sparse Jacobian matrix is the work needed to compute  $(W^T J, J V)$  which, in turn, is proportional to the work to evaluate the function  $F$  times the sum of the column dimensions of the Boolean matrices  $V \in \mathbb{R}^{n \times t_V}, W \in \mathbb{R}^{m \times t_W}$ . The column dimension sum  $t_V + t_W$  is equal to the number of colors used in the bicoloring. In our experiments we compare the computed coloring numbers required for the direct and substitution approaches. We also compute the number of colors required by one-sided schemes: a column partition alone corresponds to the construction of  $V$  based on coloring the column intersection graph of  $J$ ; a row partition alone corresponds to the construction of  $W$  based on coloring the row intersection graph of  $J$ . The latter case leads to the application of the reverse mode of AD (alone), whereas the former case leads to use of the forward mode.

Both the direct and substitution methods require colorings of their respective pairs of intersection graphs  $\mathcal{G}_C^I, \mathcal{G}_R^I$ . Many efficient graph coloring heuristics are available: in our experiments we use the incidence degree (ID) ordering [6].

We use three sources of test matrices: a linear programming testbed with results reported in Table 5.1 and summarized in Table 5.2; the Harwell-Boeing sparse matrix collection, with results reported in Tables 5.3 and 5.4; self-generated  $m$ -by- $n$  "grid matrices" with results given in Tables 5.5 and 5.6. A grid matrix is constructed in the following way. First, approximately  $\sqrt{n}$  of the columns are chosen, spaced uniformly. Each chosen column is randomly assigned  $DENS \cdot m$  nonzeros. Second, approximately  $\sqrt{m}$  of the rows are chosen, spaced uniformly. Each chosen row is randomly assigned  $DENS \cdot n$  nonzeros. We vary  $DENS$  as recorded in Table 5.5.

For each problem we cite the dimensions of the matrix  $A$  and the number of nonzeros ( $nnz$ ). The experimental results we report are the number of colors required by our bicoloring approach, both direct and substitution, and the number of colors required by one-sided schemes.

**5.1. Observations.** First, we observe that the bicoloring approach is often a significant improvement over one-sided determination. Occasionally, the improvement is spectacular, e.g., "cannes 715". Improvement in the Harwell-Boeing problems are generally more significant than on the LP collection in the sense that bicoloring significantly outperforms both one-sided possibilities. This is partially due to the fact that the matrices in the LP collection are rectangular, whereas the matrices in the Harwell-Boeing collection are square: calculation of the nonzeros of  $J$  from  $W^T J$  alone can be quite attractive when  $J$  has relatively few rows. The grid collection



TABLE 5.1  
*LP constraint matrices* (<http://www.netlib.org/lp/data/>).

Name	m	n	nnz	Bicoloring		One-sided	
				Direct	Substitution	Column	Row
standata	359	1274	3230	9	7	745	10
scagr25	471	671	1725	8	5	10	9
scagr7	129	185	465	8	5	10	9
stair	356	620	4021	36	29	36	36
blend	74	114	522	16	14	29	16
vtp.base	198	347	1052	12	10	38	12
agg	488	615	2862	19	13	43	19
agg2	516	758	4740	26	21	49	43
agg3	516	758	4756	27	21	52	43
bore3d	233	334	1448	28	24	73	28
israel	174	316	2443	61	49	119	136
boeing1	351	726	3827	32	28	315	32
boeing2	166	305	1358	23	18	93	23
tuff	333	630	4563	21	16	114	25
adlittle	56	138	424	11	10	27	11

TABLE 5.2  
*Totals for LP collection.*

Bicoloring		1-sided coloring	
Direct	Substitution	Column	Row
337	270	1753	452

displays the advantage of bicoloring to great effect—grid matrices are ideal bicoloring candidates.

In general the advantage of substitution over direct determination is not as great as the difference between bicoloring and one-sided determination. Nevertheless, fewer colors are almost always needed, and for expensive functions  $F$  this can be important. For most problems the gain is about 20% although it can approach 50%, e.g., “watt2”.

**5.2. Interface with ADOL-C.** We have interfaced our coloring and substitution routines with the ADOL-C software. The C++ package ADOL-C [12] facilitates the evaluation of first- and higher-order derivatives of vector function, defined by programs written in C or C++.

We compare the time needed on a sample problem with respect to five approaches:

- AD/bicoloring (direct),
- AD/bicoloring (substitution),
- AD/column coloring (forward mode),
- AD/row coloring (reverse mode),
- FD (sparse finite differencing based on column coloring).

The test function  $F$  we use is a simple nonlinear function: define  $f(a) = a^4 + 5 \cdot a$ , let  $N(i)$  be the index set of nonzeros in row  $i$  of the Jacobian matrix, and define  $F_i(x) = \sum_{j \in N(i)} f(x_j)$ . The Jacobian matrix (and thus the sparsity pattern) is a 10-by-3 block version of Figure 3.3; i.e.,  $l = 10$ ,  $w = 3$ . Problem dimensions  $n = 100, 200, 400$ , as indicated in Figure 5.1, were used in the experiments.

Our results, portrayed in Figure 5.1, suggest the following order of execution time requirement by different techniques:

$$FD > AD/row > AD/column > AD/bicoloring(direct) \\ > AD/bicoloring(substitution).$$

TABLE 5.3  
The Harwell-Boeing collection (ftp from orion.cerfacs.fr).

Name	m	n	nnz	Bicoloring		1-sided coloring	
				Direct	Substitution	Column	Row
watt2	1856	1856	11550	20	12	128	65
cannes 256	256	256	2916	32	23	83	83
cannes 268	268	268	1675	18	12	33	33
cannes 292	292	292	2540	17	17	35	35
cannes 634	634	634	7228	28	21	28	28
cannes 715	715	715	6665	22	18	105	105
cannes 1054	1054	1054	12196	31	23	35	35
cannes 1072	1072	1072	12444	32	24	35	35
chemimp/impcolc	137	137	411	6	4	8	9
chemimp/impcold	425	425	1339	6	5	11	11
chemimp/impcole	225	225	1308	21	14	21	31
chemwest/west0067	67	67	294	9	7	9	12
chemwest/west0381	381	381	2157	12	9	29	50
chemwest/west0497	497	497	1727	22	19	28	55
smtape/gent113	113	113	655	19	13	20	27
smtape/arc130	130	130	1282	25	23	124	124

TABLE 5.4  
Totals for Harwell-Boeing collection.

Bicoloring		1-sided coloring	
Direct	Substitution	Column	Row
320	244	732	738

Note that FD requires more time than AD/column even though the same coloring is used for both. This is because the work estimate  $t_V \cdot \omega(F)$  is actually an upper bound on the work required by the forward mode where  $t_V$  is the number of columns of  $V$ . This bound is often loose in practice, whereas  $t_V \cdot \omega(F)$  is tight for finite differencing since the subroutine to evaluate  $F$  is actually called (independently)  $t_V$  times.

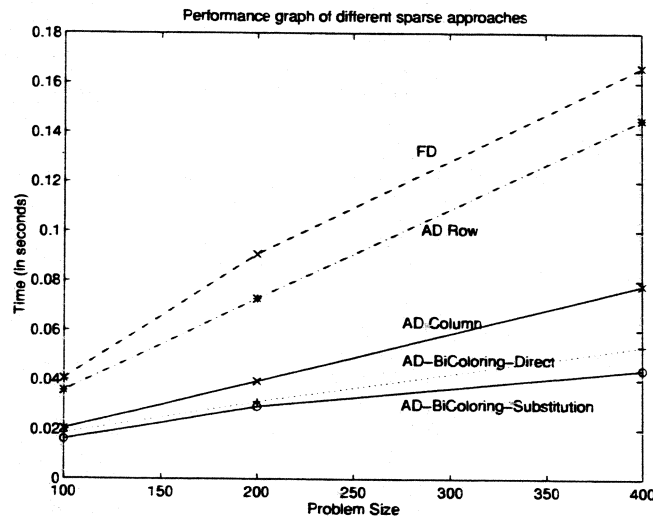


FIG. 5.1. A comparison of different sparse techniques.

Another interesting observation is that the reverse mode calculation (AD/row) is

TABLE 5.5  
*Grid matrices.*

m	n	DENS	Bicoloring		1-sided coloring	
			Direct	Substitution	Column	Row
10	10	0.44	5	4	6	7
10	10	0.75	6	5	7	8
10	10	0.98	6	6	9	9
10	20	0.55	8	8	12	8
10	20	0.89	8	8	19	10
10	20	1.00	8	8	20	10
25	25	0.52	11	10	18	15
25	25	0.61	11	10	21	19
25	25	0.99	10	10	25	25
25	100	0.56	15	16	61	24
25	100	0.67	15	15	81	24
25	100	1.00	15	15	100	25
100	100	0.52	20	20	84	74
100	100	0.64	20	20	95	93
100	100	1.00	20	20	100	100
100	400	0.53	30	30	282	98
100	400	0.64	30	30	352	100
100	400	1.00	30	30	400	100

TABLE 5.6  
*Totals for grid matrices.*

Bicoloring		1-sided coloring	
Direct	Substitution	Column	Row
268	265	1692	749

about twice as expensive as the forward calculation (AD/column). This is noteworthy because in this example, based on the structure Figure 3.1, the column dimensions of  $V$  and  $W$  are equal. This suggests that it may be practical to weigh the cost of the forward calculation of  $Jv$  versus the calculation of  $w^T J$ , where  $w, v$  are vectors. We comment further on this aspect in section 7.

**6. Substitution and roundoff.** In general, the substitution approach requires fewer colors and therefore is more efficient,<sup>3</sup> in principle, than direct determination. However, there is a possibility of increased round-off error due to the substitution process. In fact an analogous issue arises in the sparse Hessian approximation context [3, 7, 15] where, indeed, there is considerable cause for concern. The purpose of this section is to examine this question in the AD context. The bottom line here is that there is less to worry about in this case. In the sparse Hessian approximation case significant error growth occurs when the finite-difference (FD) stepsize varies over the set of FD directions; however, in our current setting this is not an issue since the "stepsize" is equal to unity in all cases.

First we consider the number of substitutions required to determine any nonzero of  $J$  from  $(W^T J, JV)$  where matrices  $W, V$  are chosen using our substitution strategy. There is good news: similar to the sparse Hessian approximation situation [3, 7, 15], the number of dependencies, or substitutions, to resolve a nonzero of  $J$  can be bounded above by  $\frac{1}{2}[(m+n-2)]$ .

<sup>3</sup>Of course a substitution method does incur the extra cost of performing the substitution calculation. However, this can be done very efficiently, and the subsequent cost is usually negligible.

**THEOREM 6.1.** *Let  $\phi$  be a bipartite cyclic  $p$ -coloring of  $G_b(J)$ . Then,  $\phi$  corresponds to a substitution determination of  $J$  and each unknown in  $J$  is dependent on at most  $m + n - 2$  unknowns. Moreover, it is possible to order the calculations so that the maximum number of substitutions is less than or equal to  $\frac{1}{2}[(m + n - 2)]$ .*

*Proof.* First, edges (nonzeros) with one end assigned color "0" can be determined directly: by the definition of bicoloring there will be no conflict. Second, every pair of positive colors induces a forest (i.e., a collection of trees in  $G_b(J)$ ); therefore, the edges (nonzeros) in the induced forest can be resolved via substitution [3]. Hence, all edges (nonzeros) can be resolved either directly or by a substitution process and the worst case corresponds to a tree with  $m + n - 1$  edges yielding an upper bound of  $m + n - 2$  substitutions. However, it is easy to see that the substitution calculations can be ordered to halve the worst-case bound yielding at most  $\frac{1}{2}[(m + n - 2)]$  substitutions.  $\square$

Next we develop an expression to bound the error in the computed Jacobian. Except for the elements that can be resolved directly, the nonzero elements of the Jacobian matrix can be solved for by considering each subgraph induced by two positive colors (directions), one color corresponding to a subset of rows, one color corresponding to a subset of columns. Let us look at the subgraph  $\mathcal{G}_{p,q}$  induced by colors  $p$  (columns),  $q$  (rows). Let  $z_p = Jv_p$ ,  $y_q^T = w_q^T J$ ,  $R_q$  be the set of rows colored  $q$ , and  $C_p$  be the set of columns colored  $p$ , where

$$w_q = \sum_{i \in R_q} e_i, \quad v_p = \sum_{i \in C_p} e_i.$$

Let

$$\hat{z}_p = z_p + \epsilon_p^c, \quad \hat{y}_q = y_q + \epsilon_q^r$$

denote the quantities computed via AD. Note that the errors introduced here are only due to the AD process and are typically very small.

In the solution process an element  $J_{ij}$  is determined:

$$J_{ij} = \hat{z}_{p_i} - \sum_{k \in N(r_i)} J_{ik} \quad \text{or} \quad J_{ij} = \hat{y}_{q_j} - \sum_{k \in N(c_j)} J_{kj}$$

depending on whether a column equation (of form  $Jv$ ) or a row equation (of form  $w^T J$ ) is used. Here,  $N(r_i)$  denotes set of neighbors of row  $i$  in  $\mathcal{G}_{p,q}$ , and  $N(c_j)$  denotes set of neighbors of column  $j$  in  $\mathcal{G}_{p,q}$ .

Assume that  $J^{actual} = F'(x)$  denotes the actual Jacobian matrix; hence,

$$J_{ij}^{actual} = z_{p_i} - \sum_{k \in N(r_i)} J_{ik}^{actual} \quad \text{or} \quad J_{ij}^{actual} = y_{q_j} - \sum_{k \in N(c_j)} J_{kj}^{actual}.$$

Define an error matrix  $E = J - J^{actual}$  and let  $\epsilon_{ij}$  be the difference

$$\epsilon_{ij} = \hat{z}_{p_i} - \sum_{k \in N(r_i) \cup \{j\}} J_{ik} \quad \text{or} \quad \epsilon_{ij} = \hat{y}_{q_j} - \sum_{k \in N(c_j) \cup \{i\}} J_{kj}$$

depending on the way element  $J_{ij}$  was computed.

We take into account the effect of *rounding errors* by letting  $\hat{\epsilon}_{ij}$  be equal to  $\epsilon_{ij}$  plus the contribution from rounding errors due to use of the equation that determines

$J_{ij}$ . We can now write

$$(6.1) \quad \hat{\epsilon}_{ij} = E_{ij} + \sum_{k \in N(r_i)} E_{ik} \quad \text{or} \quad \hat{\epsilon}_{ij} = E_{ij} + \sum_{k \in N(c_j)} E_{kj}$$

again, depending on the way  $J_{ij}$  is calculated.

Moreover, we let  $\epsilon_{max}$  be the constant:

$$\epsilon_{max} = \max_{i,j} |\hat{\epsilon}_{ij}|.$$

Note that  $\epsilon_{max}$  has no contribution from stepsizes, unlike results for finite differencing [3, 15].

**THEOREM 6.2.** *If  $J$  is obtained by our substitution process then*

$$|E_{ij}| \leq \frac{1}{2} [(m+n)] \cdot \epsilon_{max}.$$

*Proof.* From equation (6.1),

$$(6.2) \quad \hat{\epsilon}_{ij} = E_{ij} + \sum_{k \in N(r_i)} E_{ik}$$

or

$$(6.3) \quad \hat{\epsilon}_{ij} = E_{ij} + \sum_{k \in N(c_j)} E_{kj}.$$

Let us assume, without loss of generality, that equation (6.2) holds. This implies a bound

$$|E_{ij}| \leq |\hat{\epsilon}_{ij}| + \sum_{k \in N(r_i)} |E_{ik}|.$$

But the same decomposition can be applied recursively to each  $E_{ik}$ , and using Theorem 6.1, the result follows.  $\square$

There are two positive aspects to Theorem 6.2. First, unlike the sparse FD substitution method for Hessian matrices [3, 7, 15], there is no dependence on a variable stepsize: in AD the stepsize is effectively uniformly equal to unity. Second, there is no cumulative dependence on  $nnz(J)$  but rather just on the matrix dimensions,  $m+n$ . However, there is one unsatisfactory aspect of the bound in Theorem 6.2: the bound is expressed in terms of  $\epsilon_{max}$ , but  $\epsilon_{max}$  is not known to be restricted in magnitude. A similar situation arises in the [3, 7, 15]. Nevertheless, as illustrated in the example discussed below,  $\epsilon_{max}$  is usually modest in practice.

We conclude this section with a small experiment where we inspect final accuracies of the computed Jacobian matrices. The test function  $F$  is a simple nonlinear function as described in section 5.2.

In Table 6.1 "FD1" is the sparse FD computation [6] using a fixed stepsize  $\alpha = \sqrt{\epsilon} \approx 10^{-8}$ ; "FD2" refers to the sparse FD computation [6] using a variable stepsize:  $\alpha$  is uniformly varied in the range  $[\frac{\sqrt{\epsilon}}{4}, 4 \cdot \sqrt{\epsilon}]$ . The column labelled "Rel error" records  $\|ERR\|_2$ , where the nonzeros of  $ERR$  correspond to the nonzeros of  $J$ : for  $J_{ij} \neq 0$ ,  $ERR_{ij} = \frac{J_{ij}^{actual} - J_{ij}^{computed}}{J_{ij}^{actual}}$ .

The general trends we observe are the following. First, similar to the results reported in [1] for forward-mode direct determination, the Jacobian matrices determined by our bicoloring/AD approach are significantly and uniformly more accurate than the FD approximations. This is true for both direct determination and the substitution approach. Second, the direct approach is uniformly more accurate than the substitution method; however, the Jacobian matrices determined via substitution are sufficiently accurate for most purposes, achieving at least 10 digits of accuracy and usually more. Finally, on these problems there is relatively little difference in accuracy between the fixed step (FD1) method and the variable step (FD2) method. However, as illustrated in [5], it is easy to construct examples where the variable step (FD2) approach produces unacceptable accuracy.

TABLE 6.1  
Errors (sample nonlinear problem).

	Direct	Substitution	FD1	FD2
Size	Rel error	Rel error	Rel error	Rel error
100	$1.30 \times 10^{-15}$	$7.63 \times 10^{-13}$	$7.06 \times 10^{-6}$	$2.06 \times 10^{-5}$
200	$2.93 \times 10^{-15}$	$4.19 \times 10^{-12}$	$1.06 \times 10^{-5}$	$6.72 \times 10^{-5}$
400	$9.62 \times 10^{-15}$	$2.02 \times 10^{-11}$	$4.77 \times 10^{-5}$	$9.81 \times 10^{-5}$
800	$4.57 \times 10^{-14}$	$1.45 \times 10^{-10}$	$9.21 \times 10^{-5}$	$8.49 \times 10^{-4}$

**7. Concluding remarks.** We have proposed an effective way to compute a sparse Jacobian matrix  $J$  using AD. Our proposal uses a new graph technique, bicoloring, to divide the differentiation work between the two modes of AD, forward and reverse. The forward mode computes the product  $JV$  for a given matrix  $V$ ; the reverse mode computes the product  $W^T J$  for a given matrix  $W$ . We have suggested ways to choose thin matrices  $V, W$  so that the work to compute the pair  $(W^T J, JV)$  is modest and so that the nonzero elements of  $J$  can be readily extracted.

Our numerical results strongly support the view that bicoloring/AD is superior to one-sided computations (both AD and FD) with respect to the order of work required. Of course AD approaches offer additional advantages over FD schemes: significantly better accuracy, no need to heuristically determine a stepsize rule, e.g., [10], and the sparsity pattern need not be determined a priori [12].

Implicit in our approach is the assumption that the cost to compute  $Jv$  by forward mode AD is equal to the cost of computing  $w^T J$  by reverse mode AD, where  $v, w$  are vectors. This is true in the order of magnitude sense—both computations take time proportional to  $\omega(F)$ —but the respective constants may differ widely. It may be pragmatic to estimate “weights”  $w_1, w_2$ , with respect to a given AD tool, reflecting the relative costs of forward and reverse modes. It is very easy to introduce weights into algorithm *MNCO* (section 4.3) to heuristically solve a “weighted” problem,  $\min w_1 \chi_1 + w_2 \chi_2$ , where  $\chi_1$  is the number of row groups (or colors assigned to the rows) and  $\chi_2$  is the number of column groups (or colors assigned to the columns). The heuristic *MNCO* can be changed to address this problem by simply changing the conditional (*LB*) to the following:

$$\text{if } w_1 \cdot \rho(J_R^T) + w_2 \cdot \max(\rho(J_C), \text{nnz}(r)) < w_1 \cdot \rho(J_C) + w_2 \cdot \max(\rho(J_R^T), \text{nnz}(c)).$$

Different weights produce different allocations of work between forward and reverse modes, skewed to reflect the relative costs. For example, consider a 50-by-50 grid matrix with  $DENS = 1$  (see section 5), and let us vary the relative weighting of

forward versus reverse mode:  $w_1 = 0 : .25 : 1$ , and  $w_2 = 1 - w_1$ . The results of our weighted bicoloring approach are given in Table 7.1.

TABLE 7.1  
Weighted problem results.

$w_1$	$\chi_1$	$\chi_2$
0.00	50	0
0.25	16	7
0.50	10	10
0.75	7	16
1.00	0	50

Finally we note that the bicoloring ideas can sometimes be used to efficiently determine relatively dense Jacobian matrices provided structural information is known about the function  $F$ . For example, suppose  $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is a partially separable function,  $F = F_1 + F_2 + \dots + F_t$ , where  $F_i : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $i = 1, \dots, t$ , and each component function  $F_i$  typically depends on only a few components of  $x$ . Clearly each Jacobian function  $J_i$  is sparse while the summation  $J = \sum_{i=1}^t J_i$  may or may not be sparse depending on the sparsity patterns. However, if we define a "stacked" function  $\tilde{F}$ ,

$$\tilde{F}(x) = \begin{pmatrix} F_1(x) \\ F_2(x) \\ \vdots \\ F_t(x) \end{pmatrix},$$

then the Jacobian of  $\tilde{F}$  is

$$\tilde{J} = \begin{pmatrix} J_1 \\ J_2 \\ \vdots \\ J_t \end{pmatrix}.$$

Clearly  $\tilde{J}$  is sparse and the bicoloring/AD technique can be applied to  $\tilde{J}$ , possibly yielding a dramatic decrease in cost. Specifically, if  $J$  is dense (a possibility) then the work to compute  $J$  without exploiting structure is  $n \cdot \omega(F)$ , whereas the cost of computing  $\tilde{J}$  via bicoloring/AD is approximately  $\chi_c(\mathcal{G}_b(\tilde{J})) \cdot \omega(F)$  flops, where  $\chi_c(\mathcal{G}_b(\tilde{J}))$  is the minimum number of colors required for a bipartite cyclic coloring of graph  $\mathcal{G}_b(\tilde{J})$ . Typically,  $\chi_c(\mathcal{G}_b(\tilde{J})) \ll n$ . The idea of applying the bicoloring/AD technique in a structured way is not restricted to partially separable functions.

**Acknowledgments.** We are very grateful to Andreas Griewank, his student Jean Utke, and his colleague David Juedes for helping us with the use of ADOL-C.

#### REFERENCES

- [1] B. M. AVERICK, J. J. MORÉ, AND C. H. BISCHOF, *Computing Large Sparse Jacobian Matrices Using Automatic Differentiation*, Preprint MCS-P348-0193, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, April 1993
- [2] C. H. BISCHOF, A. BOUARICHA, P. M. KHADEMI, AND J. J. MORÉ, *Computing gradients in large-scale optimization using automatic differentiation*, SIAM J. Alg. Discrete Methods, 7 (1986), pp. 221–235.

- [3] T. F. COLEMAN AND J.-Y. CAI, *The cyclic coloring problem and estimation of sparse Hessian matrices*, SIAM J. Alg. Discrete Methods, 7 (1986), pp. 221–235.
- [4] T. F. COLEMAN, B. S. GARBOW, AND J. J. MORÉ, *Software for estimating sparse Jacobian matrices*, ACM Trans. Math. Software, 10 (1984), pp. 329–345.
- [5] T. F. COLEMAN, B. S. GARBOW, AND J. J. MORÉ, *Software for estimating sparse Hessian matrices*, ACM Trans. Math. Software, 11 (1985), pp. 363–377.
- [6] T. F. COLEMAN AND J. J. MORÉ, *Estimation of sparse Jacobian matrices and graph coloring problems*, SIAM J. Numer. Anal., 20 (1983), pp. 187–209.
- [7] T. F. COLEMAN AND J. J. MORÉ, *Estimation of sparse Hessian matrices and graph coloring problems*, Math. Programming, 28 (1984), pp. 243–270.
- [8] A. R. CURTIS, M. J. D. POWELL, AND J. K. REID, *On the estimation of sparse Jacobian matrices*, J. Inst. Math. Appl., 13 (1974), pp. 117–119.
- [9] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman, San Francisco, 1979.
- [10] P. E. GILL, W. MURRAY, M. A. SAUNDERS, AND M. H. WRIGHT, *Computing forward-difference intervals for numerical optimization*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 310–321.
- [11] A. GRIEWANK, *Some bounds on complexity of gradients, Jacobians and Hessians*, in Complexity in Numerical Optimization, P. M. Pardalos, ed., World Scientific Publishing Co., River Edge, NJ, 1993.
- [12] A. GRIEWANK, D. JUEDES, AND J. UTKE, *ADOL-C: A package for automatic differentiation of algorithms written in C/C++*, ACM Trans. Math. Software, 22 (1996), pp. 131–167.
- [13] A. GRIEWANK, *Direct calculation of Newton steps without accumulating Jacobians*, in Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., SIAM, Philadelphia, 1990, pp. 115–137.
- [14] A. K. M. S. HOSSAIN AND T. STEIHAUG, *Computing a Sparse Jacobian Matrix by Rows and Columns*, Tech. Report 109, Department of Informatics, University of Bergen, Bergen, 1995.
- [15] M. J. D. POWELL AND PH. L. TOINT, *On the estimation of sparse Hessian matrices*, SIAM J. Numer. Anal., 16 (1979), pp. 1060–1074.