1991

Modification of Newman's BAND(J) Subroutine to Multi-Region Systems Containing Interior Boundaries: MBAND

D. Fan
Texas A & M University - College Station

Ralph E. White
University of South Carolina - Columbia, white@cec.sc.edu

Publication Info

Journal of the Electrochemical Society, 1991, pages 1688-1691.

DOI: 10.1149/1.2085854
http://dx.doi.org/10.1149/1.2085854

Modification of Newman's BAND(J) Subroutine to Multi-Region Systems Containing Interior Boundaries: MBAND

D. Fan* and R. E. White**

Department of Chemical Engineering, Texas A&M University, College Station, Texas 77843

ABSTRACT

Newman's BAND(J) subroutine, which has been used widely to solve models of various electrochemical systems, is extended to solve a system of coupled, ordinary differential equations with interior boundary conditions. A set of coupled, linear ordinary differential equations is used to demonstrate the solution procedure. The results show that the extended technique has the same accuracy as that of using pentadiagonal BAND(J), but the execution speed is about five times faster than that of pentadiagonal BAND(J). Using sparse matrix solver Y12MAF to solve the same set of equations takes even longer time than pentadiagonal BAND(J).

Electrochemical systems such as batteries and fuel cells consist of multiple regions as demonstrated through several references (1-4). The phenomena that occur in these regions can be modeled mathematically. This procedure yields a coupled set of equations with interior boundary conditions. Newman's BAND(J) subroutine (5, 6) has been used widely to solve systems of coupled, nonlinear ordinary differential equation. This is done by first transforming these sets of differential equations into sets of nonlinear algebraic equations by using finite difference approximations. These sets of nonlinear algebraic equations are then solved by using the Newton-Raphson procedure with BAND(J). This procedure yields sets of linear of algebraic equations of the form $\mathbf{JC} = \mathbf{G}$ or

$$\begin{bmatrix} \mathbf{B}(1) & \mathbf{D}(1) & \mathbf{X} & & & & & \\ \mathbf{A}(2) & \mathbf{B}(2) & \mathbf{D}(2) & & & & & \\ & \cdot & \cdot & \cdot & & & & \\ & & & \mathbf{A}(j) & \mathbf{B}(j) & \mathbf{D}(j) & & \\ & & & & & \cdot & \cdot & \cdot \\ & & & & & \mathbf{Y} & \mathbf{A}(nj) & \mathbf{B}(nj) \end{bmatrix} \begin{bmatrix} \mathbf{C}(1) \\ \mathbf{C}(2) \\ \cdot \\ \mathbf{C}(j) \\ \cdot \\ \mathbf{C}(nj) \end{bmatrix} = \begin{bmatrix} \mathbf{G}(1) \\ \mathbf{G}(2) \\ \cdot \\ \mathbf{G}(j) \\ \cdot \\ \mathbf{G}(nj) \end{bmatrix} \quad [1]$$

where Newman's notation (6) is used for convenience.

Multiple Regions

Most electrochemical systems include multiple regions (*e.g.*, anode, separator, and cathode in a battery). Sets of differential equations with interior boundary conditions are used to model such devices. For example, in a two-region system, there are two differential equations within their corresponding domain of x

$$F_1(x, \mathbf{c}, \mathbf{c}', \mathbf{c}'') = 0 \quad \text{at} \quad 0 < x < l_1 \quad [2]$$

$$F_2(x, \mathbf{c}, \mathbf{c}', \mathbf{c}'') = 0 \quad \text{at} \quad l_1 < x < l_2 \quad [3]$$

where $\mathbf{c} = [c_1, c_2, \ldots, c_n]^T$ with boundary conditions of the form

$$x = 0 \quad : \quad \alpha_1 \mathbf{c} + \beta_1 \frac{\partial \mathbf{c}}{\partial x} = \mathbf{f}_1(x) \quad [4]$$

$$x = l_1 \quad : \quad \alpha_2 \frac{\partial \mathbf{c}}{\partial x}\bigg|_- = \beta_2 \frac{\partial \mathbf{c}}{\partial x}\bigg|_+ \quad [5]$$

$$x = l_2 \quad : \quad \alpha_3 \mathbf{c} + \beta_3 \frac{\partial \mathbf{c}}{\partial x} = \mathbf{f}_3(x) \quad [6]$$

Applying Newman's procedure (7) with three-point finite difference approximations used at all nodes including $x = l_1 (j = nj_1)$ yields

$$\begin{bmatrix} \mathbf{B}(1) & \mathbf{D}(1) & \mathbf{X}(1) & & & & & \\ \mathbf{A}(2) & \mathbf{B}(2) & \mathbf{D}(2) & & & & & \\ & & \mathbf{Y}(nj_1) & \mathbf{A}(nj_1) & \mathbf{B}(nj_1) & \mathbf{D}(nj_1) & \mathbf{X}(nj_1) & \\ & & & & & \mathbf{Y}(nj) & \mathbf{A}(nj) & \mathbf{B}(nj) \end{bmatrix} \times \begin{bmatrix} \mathbf{C}(1) \\ \mathbf{C}(2) \\ \cdot \\ \mathbf{C}(nj_1) \\ \cdot \\ \mathbf{C}(nj) \end{bmatrix} = \begin{bmatrix} \mathbf{G}(1) \\ \mathbf{G}(2) \\ \cdot \\ \mathbf{G}(nj_1) \\ \cdot \\ \mathbf{G}(nj) \end{bmatrix} \quad [7]$$

where the block jacobian coefficient matrix on the left side of Eq. [7] contains a row which has five elements instead of three (the row at $j = nj_1$). This is a result of using three-point forward and backward finite difference formulas at the interface boundary to maintain accuracy to $O(h^2)$. The pentadiagonal BAND(J) subroutine (8) has been used (9, 1) to solve Eq. [7], but this is costly in terms of computation time because of the unnecessary computations due to the null matrices off the tridiagonal, especially when a large number of nodal points are used. However, it is important to note that this procedure is less costly than using forward and backward two-point finite difference approximations at the interface because such approximations are only accurate to $O(h)$. It is possible to use other procedures

Table I. Averaged relative error with respect to the exact solution using 21 nodes.

Pentadiagonal BAND	Y12MAF	MBAND
4.973×10^{-5}	4.973×10^{-5}	4.973×10^{-5}

Table II. CPU time (ms) statistics for pentadiagonal BAND and MBAND.

Module	Pentadiagonal BAND	MBAND
MAIN	25 (6.94%)	24 (30.13%)
BAND	290 (82.19%)	29 (37.44%)
LUDEC	27 (7.51%)	19 (24.62%)
ZEROS	12 (3.35%)	6 (7.81%)
TOTAL	353	78

* Electrochemical Society Student Member.
** Electrochemical Society Active Member.

(10) to maintain accuracy to $O(h^2)$ at interior boundaries, but these may be difficult to apply especially to systems with several regions. A modified BAND(J), named MBAND, is presented here that reduces significantly the computation time to solve Eq. [7]. (The source code for this subroutine is available from the authors upon request.)

When **LU** decomposition is performed on the jacobian coefficient matrix in Eq. [7], the upper and lower triangular matrices that result are as follows

$$\begin{bmatrix} \mathbf{B}(1) & \mathbf{D}(1) & \mathbf{X}(1) & & & & & & & & \\ \mathbf{A}(2) & \mathbf{B}(2) & \mathbf{D}(2) & & & & & & & & \\ & \ddots & \ddots & & & & & & & & \\ & & \mathbf{Y}(nj_1) & \mathbf{A}(nj_1) & \mathbf{B}(nj_1) & \mathbf{D}(nj_1) & \mathbf{X}(nj_1) & & & & \\ & & & & \mathbf{A}(nj_1+1) & \mathbf{B}(nj_1+1) & \mathbf{D}(nj_1+1) & & & & \\ & & & & & & \ddots & \ddots & & & \\ & & & & & & \mathbf{Y}(nj) & \mathbf{A}(nj) & \mathbf{B}(nj) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{b}(1) & & & & & & \\ \mathbf{a}(2) & \mathbf{b}(2) & & & & & \\ & \ddots & \ddots & & & & \\ & \mathbf{y} & \mathbf{a}(nj_1) & \mathbf{b}(nj_1) & & & \\ & & \mathbf{a}(nj_1+1) & \mathbf{b}(nj_1+1) & & & \\ & & & \ddots & \ddots & & \\ & & & \mathbf{y}(nj) & \mathbf{a}(nj) & \mathbf{b}(nj) \end{bmatrix}$$

$$\times \begin{bmatrix} \mathbf{I} & -\mathbf{E}(1) & -\mathbf{x}(1) & & & & & \\ & \mathbf{I} & -\mathbf{E}(2) & & & & & \\ & & \ddots & \ddots & & & & \\ & & & \mathbf{I} & -\mathbf{E}(nj_1) & -\mathbf{x}(nj_1) & & \\ & & & & \mathbf{I} & -\mathbf{E}(nj_1+1) & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \mathbf{I} \end{bmatrix} \quad [8]$$

The matrix on the right side of Eq. [8] can be used to solve Eq. [7] by defining an intermediate vector **v** such that

$$\begin{bmatrix} \mathbf{I} & -\mathbf{E}(1) & -\mathbf{x}(1) & & & & & \\ & \mathbf{I} & -\mathbf{E}(2) & & & & & \\ & & \ddots & \ddots & & & & \\ & & & \mathbf{I} & -\mathbf{E}(nj_1) & -\mathbf{x}(nj_1) & & \\ & & & & \mathbf{I} & -\mathbf{E}(nj_1+1) & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{C}(1) \\ \mathbf{C}(2) \\ \vdots \\ \mathbf{C}(nj_1) \\ \mathbf{C}(nj_1+1) \\ \vdots \\ \mathbf{C}(nj) \end{bmatrix} = \begin{bmatrix} \mathbf{v}(1) \\ \mathbf{v}(2) \\ \vdots \\ \mathbf{v}(nj_1) \\ \mathbf{v}(nj_1+1) \\ \vdots \\ \mathbf{v}(nj) \end{bmatrix} \quad [9]$$

which yields

$$\begin{bmatrix} \mathbf{b}(1) & & & & & & \\ \mathbf{a}(2) & \mathbf{b}(2) & & & & & \\ & \ddots & \ddots & & & & \\ & \mathbf{y}(nj_1) & \mathbf{a}(nj_1) & \mathbf{b}(nj_1) & & & \\ & & & \mathbf{a}(nj_1+1) & \mathbf{b}(nj_1+1) & & \\ & & & & \ddots & \ddots & \\ & & & & \mathbf{y}(nj) & \mathbf{a}(nj) & \mathbf{b}(nj) \end{bmatrix} \begin{bmatrix} \mathbf{v}(1) \\ \mathbf{v}(2) \\ \vdots \\ \mathbf{v}(nj_1) \\ \mathbf{v}(nj_1+1) \\ \vdots \\ \mathbf{v}(nj) \end{bmatrix} = \begin{bmatrix} \mathbf{G}(1) \\ \mathbf{G}(2) \\ \vdots \\ \mathbf{G}(nj_1) \\ \mathbf{G}(nj_1+1 \\ \vdots \\ \mathbf{G}(nj) \end{bmatrix} \quad [10]$$

The coefficient matrices in Eq. [9] and [10] are obtained by solving Eq. [8] row by row. Solving the first row of Eq. [8] will give the values of elements in matrices **b**(1), **E**(1), and **x**(1). These values are used to calculate the elements in the next row **a**(2), **b**(2), and **E**(2). The process could be repeated until all of the elements on the right side of Eq. [8] are determined.

Next the intermediate vector **v** is obtained by using forward substitution to solve Eq. [10]. Once all of the **a**, **b**, and **E** matrices and the **v** vector are determined, the unknown vectors **c**(1), . . . ,**c**(nj) can be obtained by using backward substitution to solve Eq. [9].

The equations obtained by these procedures are given next. For the first row of Eq. [8], we obtain the following matrix equations

$$\mathbf{b}(1) \cdot \mathbf{I} = \mathbf{B}(1) \quad [11]$$

$$\mathbf{b}(1) \cdot [-\mathbf{E}(1)] = \mathbf{D}(1) \quad [12]$$

$$\mathbf{b}(1) \cdot [-\mathbf{x}] = \mathbf{X} \quad [13]$$

where **E**(1) and **x** can be determined by solving Eq. [12] and [13] by knowing **b**(1) from Eq. [11]. For the second row of Eq. [8], the formulas are slightly different

$$\mathbf{a}(2) \cdot \mathbf{I} = \mathbf{A}(2) \quad [14]$$

$$\mathbf{a}(2) \cdot [-\mathbf{E}(1)] + \mathbf{b}(2) \cdot \mathbf{I} = \mathbf{B}(2) \quad [15]$$

or

$$\mathbf{b}(2) = \mathbf{B}(2) + \mathbf{a}(2) \cdot \mathbf{E}(1) \quad [16]$$

In Eq. [16], since all matrices on the right side are known, **b**(2) can be calculated directly. This matrix can then be used to find values of **E**(2) according to Eq. [8]

$$\mathbf{a}(2) \cdot [-\mathbf{x}] + \mathbf{b}(2) \cdot [-\mathbf{E}(2)] = \mathbf{D}(2) \quad [17]$$

or

$$\mathbf{b}(2) \cdot [-\mathbf{E}(2)] = \mathbf{D}(2) + \mathbf{a}(2) \cdot \mathbf{x} \quad [18]$$

and **v**(2) according to Eq. [10]

$$\mathbf{a}(2) \cdot \mathbf{v}(1) + \mathbf{b}(2) \cdot \mathbf{v}(2) = \mathbf{G}(2) \quad [19]$$

or

$$\mathbf{b}(2) \cdot [-\mathbf{v}(2)] = -\mathbf{G}(2) + \mathbf{a}(2) \cdot \mathbf{v}(1) \quad [20]$$

Equations [18] and [20] must be solved to get the values of $\mathbf{E}(2)$ and $\mathbf{v}(2)$.

A similar technique applies to the nodes at j from 3 through $nj_1 - 1$, and from $nj_1 + 3$ through $nj - 1$

$$\mathbf{a}(j) \cdot \mathbf{I} = \mathbf{A}(j) \quad [21]$$

and

$$\mathbf{a}(j) \cdot [-\mathbf{E}(j-1)] + \mathbf{b}(j) \cdot \mathbf{I} = \mathbf{B}(j) \quad [22]$$

or

$$\mathbf{b}(j) = \mathbf{B}(j) + \mathbf{a}(j) \cdot \mathbf{E}(j-1) \quad [23]$$

and

$$\mathbf{b}(j) \cdot [-\mathbf{E}(j)] = \mathbf{D}(j) \quad [24]$$

and according to Eq. [10]

$$\mathbf{a}(j) \cdot \mathbf{v}(j-1) + \mathbf{b}(j) \cdot \mathbf{v}(j) = \mathbf{G}(j) \quad [25]$$

or

$$\mathbf{b}(j) \cdot [-\mathbf{v}(j)] = -\mathbf{G}(j) + \mathbf{a}(j) \cdot \mathbf{v}(j-1) \quad [26]$$

where $\mathbf{b}(j)$ in Eq. [23] is directly calculated and then Eq. [24] and [26] are solved to obtain $\mathbf{E}(j)$ and $\mathbf{v}(j)$. The equations at the interior boundary ($j = nj_1$) are the following (see [8])

$$\mathbf{y}(nj_1) \cdot \mathbf{I} = \mathbf{Y}(nj_1) \quad [27]$$

or

$$\mathbf{y}(nj_1) = \mathbf{Y}(nj_1) \quad [28]$$

and

$$\mathbf{y}(nj_1) \cdot [-\mathbf{E}(nj_1 - 2)] + \mathbf{a}(nj_1) = \mathbf{A}(nj_1) \quad [29]$$

or

$$\mathbf{a}(nj_1) = \mathbf{A}(nj_1) + \mathbf{y}(nj_1) \cdot \mathbf{E}(nj_1 - 2) \quad [30]$$

and

$$\mathbf{a}(nj_1) \cdot [-\mathbf{E}(nj_1 - 1)] + \mathbf{b}(nj_1) = \mathbf{B}(nj_1) \quad [31]$$

or

$$\mathbf{b}(nj_1) = \mathbf{B}(nj_1) + \mathbf{a}(nj_1) \cdot \mathbf{E}(nj_1 - 1) \quad [32]$$

and

$$\mathbf{b}(nj_1) \cdot [-\mathbf{E}(nj_1)] = \mathbf{D}(nj_1) \quad [33]$$

and

$$\mathbf{b}(nj_1) \cdot [-\mathbf{x}(nj_1)] = \mathbf{X}(nj_1) \quad [34]$$

and according to Eq. [10]

$$\mathbf{y} \cdot \mathbf{v}(nj_1 - 2) + \mathbf{a}(nj_1) \cdot \mathbf{v}(nj_1 - 1) + \mathbf{b}(nj_1) \cdot \mathbf{v}(nj_1) = \mathbf{G}(nj_1) \quad [35]$$

or

$$\mathbf{b}(nj_1) \cdot \mathbf{v}(nj_1) = \mathbf{G}(nj_1) - \mathbf{y}(nj_1) \mathbf{v}(nj_1 - 2) - \mathbf{a}(nj_1) \cdot \mathbf{v}(nj_1 - 1) \quad [36]$$

After $\mathbf{a}(j)$ is obtained by Eq. [30], $\mathbf{b}(j)$ can be determined directly by Eq. [32]. The values for $\mathbf{E}(j)$, $\mathbf{x}(j)$, and $\mathbf{v}(j)$ are determined by solving Eq. [33], [34], and [36]. Note that Eq. [33], [34], and [36] have the same coefficient matrix $\mathbf{b}(j)$ so they can be solved by one call to a matrix-solver routine.

Calculation at the node $j = nj_1 + 1$ which follows the interior boundary condition will have the same form as that of $j = 2$, *i.e.*, Eq. [16], [18], and [20]. This applies for any node which follows a boundary point or an interface point.

When at the node $j = nj$, the calculation formulas are the same as that of $j = nj_1$ except that since the matrices $\mathbf{D}(nj)$ and $\mathbf{X}(nj)$ are null, it is meaningless to solve Eq. [33] and [34], so only Eq. [36] is used to solve for $\mathbf{v}(nj)$.

Once all of the elements of matrix $\mathbf{E}$ and vector $\mathbf{v}$ have been determined, Eq. [9] can be used to find the solution vectors $\mathbf{C}$ by backward substitution. The first equation is at $j = nj$

$$\mathbf{I} \cdot \mathbf{C}(nj) = \mathbf{v}(nj) \quad [37]$$

or simply

$$\mathbf{C}(nj) = \mathbf{v}(nj) \quad [38]$$

and the rest of the values of $\mathbf{C}$ can be calculated from this value and corresponding values of $\mathbf{E}$ and $\mathbf{v}$ by

$$\mathbf{C}(j) = \mathbf{v}(j) + \mathbf{E}(j) \cdot \mathbf{C}(j+1) \quad \text{for } (2 \le j \le nj_1 - 1 \text{ and } nj_1 + 1 \le j \le nj - 1) \quad [39]$$

At the interior and the left boundaries ($j = 1$ and nj_1), the following equation is used

$$\mathbf{C}(j) = \mathbf{v}(j) + \mathbf{E}(j) \cdot \mathbf{C}(j+1) + \mathbf{x} \cdot \mathbf{C}(j+2) \quad [40]$$

Although the above equations are based on an example with only one interior boundary point, MBAND can be applied to problems containing more than one interior boundary point.

Example

To illustrate the solution procedure, a simple set of coupled ordinary differential equations (11) is used. This set of equations describes the mole fractions of the gas and liquid reactants within a gas fed porous electrode of a fuel cell at steady state. Suppose the porous electrode consists of two layers of different materials, then the following set of equations applies to the two regions [0, 0.5] and [0.5, 1.0] in the domain x

$$\frac{d^2y_1}{\partial x^2} - k_1' y_1 = 0 \quad [41]$$

$$\frac{d^2y_2}{\partial x^2} - k_2' y_1 = 0 \quad [42]$$

with boundary conditions

at $x = 0.0$

$$y_1(0) = 0.21 \quad [43]$$

$$\left.\frac{dy_2}{dx}\right|_{x=0} = 0 \quad [44]$$

at $x = 0.5$

$$\left.\frac{dy_1}{dx}\right|_{0.5-} = \left.\frac{dy_1}{dx}\right|_{0.5+} \quad [45]$$

$$\left.\frac{dy_2}{dx}\right|_{0.5-} = \left.\frac{dy_2}{dx}\right|_{0.5+} \quad [46]$$

at $x = 1.0$

$$\left.\frac{dy_1}{dx}\right|_{x=1} = 0 \quad [47]$$

$$y_2(1) = 0.127 \quad [48]$$

This set of equations can be solved analytically

$$y_1 = y_1(0) \left\{\frac{\cosh[k_1(1-x)]}{\cosh(k_1)}\right\} \quad [49]$$

$$y_2 = y_2(1) + \frac{k_2^2 y_1(0)}{k_1^2 \cosh(k_1)} \{\cosh[k_1(1-x)] - k_1(1-x)\sinh(k_1) - 1\} \quad [50]$$

$$[51]$$

where $k_1 = \sqrt{k_1'}$ and $k_2 = \sqrt{k_2'}$.

The first step in solving Eq. [41]-[48] is to transform this set of ordinary differential equations into a system of algebraic equations by applying three-point finite difference approximations to each nodal point. These procedures as well as the determination of all the jacobian elements **A, B, D, X,** and **Y** are discussed in detail by Newman (6) and White (12). According to Eq. [45] and [46], the boundary conditions at the interior boundary $j = nj_1$ are

$$F_{1j} = \frac{3C_{1,j} - 4C_{1,j-1} + C_{1,j-2}}{2\Delta x_1} - \frac{-3C_{1,j} + 4C_{1,j+1} - C_{1,j+2}}{2\Delta x_2} \quad [52]$$

$$F_{2j} = \frac{3C_{2,j} - 4C_{2,j-1} + C_{2,j-2}}{2\Delta x_1} - \frac{-3C_{2,j} + 4C_{2,j+1} - C_{2,j+2}}{2\Delta x_2} \quad [53]$$

where $C_{1,j} = y_1(x_j)$ and $C_{2,j} = y_2(x_j)$, and Δx_1 and Δx_2 are the grid sizes of regions 1 and 2, respectively

$$\Delta x_1 = \frac{0.5}{nj_1 - 1} \qquad \Delta x_2 = \frac{0.5}{nj - nj_1} \quad [54]$$

The elements of the coefficient matrices **A, B, D, X,** and **Y** at $j = nj_1$ are

$$\mathbf{Y} = \begin{bmatrix} \frac{1}{2\Delta x_1} & 0 \\ 0 & \frac{1}{2\Delta x_1} \end{bmatrix} \quad [55]$$

$$\mathbf{A} = \begin{bmatrix} \frac{-4}{2\Delta x_1} & 0 \\ 0 & \frac{-4}{2\Delta x_1} \end{bmatrix} \quad [56]$$

$$\mathbf{B} = \begin{bmatrix} \frac{3}{2(\Delta x_1 + \Delta x_2)} & 0 \\ 0 & \frac{3}{2(\Delta x_1 + \Delta x_2)} \end{bmatrix} \quad [57]$$

$$\mathbf{D} = \begin{bmatrix} \frac{-4}{2\Delta x_2} & 0 \\ 0 & \frac{-4}{2\Delta x_2} \end{bmatrix} \quad [58]$$

$$\mathbf{X} = \begin{bmatrix} \frac{1}{2\Delta x_2} & 0 \\ 0 & \frac{1}{2\Delta x_2} \end{bmatrix} \quad [59]$$

Once expressions for all of the elements for each of the coefficient matrices are known at each node, a main program is used with MBAND to solve for C_{ij}. The main program follows the same format as that required for BAND(J) routine (6). The calling sequence to MBAND is

call mband(j, ibc, nbc, nj)

where **j** is the current node, **ibc** is the boundary flag (**ibc** = **i** at **i**th boundary, **ibc** = 0 within regions), **nbc** is the number of boundaries (**nbc** = number of regions + 1), and **nj** is an integer array with a length of **nbc** which contains the number of nodes in each region. For example, for a two-region problem with 11 nodes in the first region and 21 nodes in second region, the **nj** array (length of 3) has the values of 1, 11, and 31, respectively.

The program was executed on a CRAY-YMP supercomputer, and Table I (please see p. 1688) lists the averaged relative errors[1] of Eq. [41]-[48] solved using pentadiagonal BAND(J), a sparse matrix solver Y12MAF (13), and MBAND, respectively. It can be seen that the accuracies of all three methods are identical. Table II (please see p. 1688) presents the flowtrace statistics of the pentadiagonal BAND and MBAND subroutines after execution on the CRAY-YMP supercomputer (data in the parenthesis are percentages of total CPU time). It can be seen that when the pentadiagonal BAND is used, 82.19% of the total execution time is spent on BAND whereas MBAND consumes only 37.43% of the total CPU time. It is also worth noting that ten times more CPU time (290 ms *vs.* 29 ms) was used with pentadiagonal BAND(J) compared to MBAND. A careful examination of the time consumed by the matrix solver LUDEC shows that when using pentadiagonal BAND(J), the LUDEC module used 27 ms, while when using MBAND, it took only 19 ms. This is because in pentadiagonal BAND(J) the **D** and **X** arrays have dimensions of $n \times 5$, but in MBAND their dimensions are $n \times 3$. The time savings due to the column reduction of these two arrays is about 30%. The contribution of the time reduction of LUDEC to the savings of total CPU time is significant, since this module is the most frequently called module by all of the BAND solvers. In terms of total CPU time used on the entire program, pentadiagonal BAND took 353 ms, whereas MBAND used only 78 ms, thereby resulting in about a fivefold speed increase.

[1] Averaged relative error is defined as $\bar{e} = 1/ni\ 1/nj\ \sum_i^{ni} \sum_j^{nj} |C(i,j) - y_i(x_j)/C(i,j)|$, where ni is the number of unknowns and nj is the number of nodes.

Table III. CPU time statistics of the sparse matrix solver Y12MAF.

Module	CPU time (ms)
MAIN	25 (5.49%)
REORDER	164 (36.02%)
Y12MAF	259 (56.93%)
ZEROS	7 (1.49%)
TOTAL	455

A sparse matrix solver Y12MAF was also used to solve the same equations. According to the statistics shown in Table III, Y12MAF module took 259 ms and is much slower than MBAND.

Another important factor about using MBAND is its great savings of storage space. When the number of nodes is large, storage becomes a major problem for most of the matrix solvers including routines in IMSL and LINPACK, but not for MBAND.

Acknowledgments

The authors acknowledge gratefully that this work was supported by the Jet Propulsion Laboratory under contractor No. 958344 and by Cray Research, Incorporated. The authors would also like to thank G. Pillay for many helpful discussions.

Manuscript received Aug. 13, 1990; revised manuscript received Nov. 30, 1990.

Texas A&M University assisted in meeting the publication costs of this article.

REFERENCES

1. T. V. Nguyen, R. E. White, and H. Gu, *This Journal,* **137,** 2998 (1990).
2. T. Yeu, T. V. Nguyen, and R. E. White, *ibid.*, **135,** 1971 (1988).
3. D. Bernardi and J. Newman, *ibid.*, **134,** 1309 (1987).
4. K. Tsaur and R. Pollard, *ibid.*, **131,** 975 (1984).
5. J. Newman, *Ind. Eng. Chem. Fundam.*, **7,** 514 (1968).
6. J. S. Newman, "Electrochemical Systems," Appendix C, Prentice-Hall, Inc., Englewood Cliffs, NJ (1991).
7. J. S. Newman, "Electrochemical Systems," Prentice-Hall Inc., Englewood Cliffs, NJ (1991).
8. J. Van Zee, G. Kleine, R. E. White, and J. Newman, in "Electrochemical Cell Design," R. E. White, Editor, pp. 377-389, Plenum Press, Inc., New York (1984).
9. D. Fan and R. E. White, *This Journal,* **138,** 17 (1991).
10. R. E. White, Ph.D. Thesis, University of California, Berkeley (1975).
11. R. E. White, M. A. Nicholson, L. G. Kleine, J. Van Zee, and R. Darby, *This Journal,* **131,** 268 (1984).
12. R. E. White, *Ind. Eng. Chem. Fundam.*, **17,** 367 (1978).
13. Z. Zlatev, Technical Report No. 111, Department of Computer Science, University of Aarhus, Aarhus, Denmark (1980).