1994

A Finite-Difference Method for Pseudo-Two-Dimensional Boundary Value Problems

Z. Mao
University of South Carolina - Columbia

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

Publication Info

Journal of the Electrochemical Society, 1994, pages 151-156.

Publisher's link: http://dx.doi.org/10.1149/1.2054675
DOI: 10.1149/1.2054675

A Finite-Difference Method for Pseudo-Two-Dimensional Boundary Value Problems

Z. Mao* and R. E. White*

Department of Chemical Engineering, University of South Carolina, Columbia, South Carolina 29208

ABSTRACT

A finite-difference method is presented for solving pseudo-two-dimensional boundary-value problems. The sparse and nearly block tridiagonal properties of the matrices generated by using the finite-difference method for problems of this type are fully utilized and maintained, which yields a method that is highly efficient in the use of storage space and computation. An example shows that the central process unit time required by the method is significantly less than that required by an alternative method.

Pseudo-two-dimensional boundary-value problems often appear in chemical and electrochemical reactor analysis.[1-3] They can also appear in adsorption bed design problems.[4] In this type of problem, there exist multiple phases or regions where different transport processes occur in each phase. The transport processes in one phase interact with those in another phase only at the interface between the phases. The schematic presented in Fig. 1 represents these kinds of problems. In region I, the transport processes are usually described by n partial differential equations which are normally simplified to be functions of the spatial coordinate x and time t only, but depend on the values of the variables of the Y phase at the interface ($y = 0$). These equations are of the form

$$F_{\mathrm{I},i}\left[\frac{\partial^2 \mathbf{C}_\mathrm{I}}{\partial x^2}, \frac{\partial \mathbf{C}_\mathrm{I}}{\partial x}, \frac{\partial \mathbf{C}_\mathrm{I}}{\partial t}, \mathbf{C}_\mathrm{I}, \left.\frac{\partial \mathbf{C}_\mathrm{Y}}{\partial y}\right|_{y=0}, \mathbf{C}_\mathrm{Y}|_{y=0}\right] = 0 \quad i = 1, 2, \ldots, n \tag{1}$$

where $\mathbf{C}_\mathrm{I}$ and $\mathbf{C}_\mathrm{Y}$ represent vectors of variables in regions I and Y

$$\mathbf{C}_\mathrm{I} = (C_{\mathrm{I}1}, C_{\mathrm{I}2}, \ldots, C_{\mathrm{I}n})^\mathrm{T} \tag{2}$$

$$\mathbf{C}_\mathrm{Y} = (C_{\mathrm{Y}1}, C_{\mathrm{Y}2}, \ldots, C_{\mathrm{Y}m})^\mathrm{T} \tag{3}$$

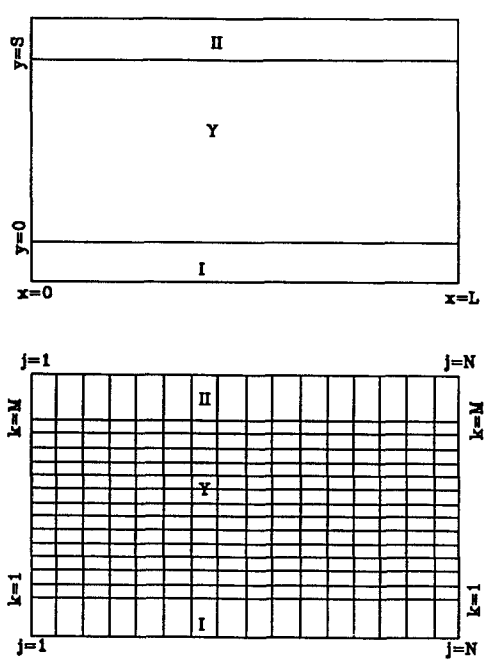


Fig. 1. (a, top) Schematic view of model regions, (b, bottom) finite difference grids for the model regions.

* Electrochemical Society Active Member.

where n and m represent the number of dependent variables in regions I and Y, respectively. Similarly, n_1 general equations are given for region II

$$F_{II,i}\left[\frac{\partial^2 \mathbf{C}_{II}}{\partial x^2}, \frac{\partial \mathbf{C}_{II}}{\partial x}, \frac{\partial \mathbf{C}_{II}}{\partial t}, \mathbf{C}_{II}, \left.\frac{\partial \mathbf{C}_\mathrm{Y}}{\partial y}\right|_{y=s}, \mathbf{C}_\mathrm{Y}|_{y=s}\right] = 0$$

$$i = 1, 2, \ldots, n_1 \tag{4}$$

where $\mathbf{C}_{II}$ represents a vector with n_1 dependent variables in region II

$$\mathbf{C}_\mathrm{II} = (C_{II1}, C_{II2}, \ldots, C_{IIn_1}) \tag{5}$$

The transport processes that are important within region Y are normally assumed to occur in the y direction only and are described by m equations for the m dependent variables

$$F_{\mathrm{Y},i}\left[\frac{\partial^2 \mathbf{C}_\mathrm{Y}}{\partial y^2}, \frac{\partial \mathbf{C}_\mathrm{Y}}{\partial y}, \frac{\partial \mathbf{C}_\mathrm{Y}}{\partial t}, \mathbf{C}_\mathrm{Y}\right] = 0 \quad i = 1, 2, \ldots, m \tag{6}$$

The boundary conditions at $x = 0$ and at $x = L$ for Eq. 1 and 4 are generally given as follows

$$f_{\mathrm{I},i}\left[\frac{\partial \mathbf{C}_\mathrm{I}}{\partial x}, \frac{\partial \mathbf{C}_\mathrm{I}}{\partial t}, \mathbf{C}_\mathrm{I}\right] = 0 \quad i = 1, 2, \ldots, n \tag{7}$$

$$f_{\mathrm{II},i}\left[\frac{\partial \mathbf{C}_{II}}{\partial x}, \frac{\partial \mathbf{C}_\mathrm{II}}{\partial t}, \mathbf{C}_{II}\right] = 0 \quad i = 1, 2, \ldots, n_1 \tag{8}$$

The boundary conditions at $y = 0$ and $y = S$ for Eq. 6 contain local variables in regions I and II and derivatives of $\mathbf{C}_\mathrm{Y}$ at these interfaces

$$f_{\mathrm{Y},i}\left[\mathbf{C}_\mathrm{I}, \left.\frac{\partial \mathbf{C}_\mathrm{Y}}{\partial y}\right|_{y=0}, \mathbf{C}_\mathrm{Y}|_{y=0}\right] = 0 \quad i = 1, 2, \ldots, m \tag{9}$$

$$f_{\mathrm{Y},i}\left[\mathbf{C}_{II}, \left.\frac{\partial \mathbf{C}_\mathrm{Y}}{\partial y}\right|_{y=S}, \mathbf{C}_\mathrm{Y}|_{y=S}\right] = 0 \quad i = 1, 2, \ldots, m \tag{10}$$

Some examples of the physical regions represented by regions I, Y, and II are given next. For a packed-bed chemical reactor, *e.g.*, the n variables in region I would represent reactant and product concentrations, pressure, temperature, etc. in the axial direction, and the m variables in region Y would represent reactant and product concentrations and temperature within the porous catalyst particles. Region II in Fig. 1 would not be needed to model this type of reactor. For electrochemical reactors (*e.g.*, batteries), region I would represent an electrolyte phase within a porous electrode with the dependent variables being concentrations of ionic species, potential, electrolyte velocity, *etc.*, region Y would represent a thin layer of insoluble solid active material on a metal substrate in which diffusion and migration of holes, electrons, and protons may occur simultaneously,[5] and region II would represent the metal substrate. In a fuel-cell model, regions I and II would represent the anode and cathode gas channels, respectively, and region Y would represent the cross section of the cell (anode/separator/cathode, see Ref. 6, *e.g.*). These examples are referred to here as pseudo-two-dimensional boundary-value problems. To the best of our knowledge, there is no efficient

numerical method that could be easily programmed and used for this type of problem. Therefore, an algorithm specifically for this type of problem was written and programmed in a subroutine named PTWO. The algorithm is presented below, and the subroutine is available upon request.

Although there are many general-purpose software packages such as DSS/2 [7] and Speedup,[8] and subroutines such as LSARG in IMSL [9] that may be suitable for solving this type of problem, the special characteristics of this problem are probably not utilized by these packages to reduce the computer storage and central processing unit (CPU) time. Consequently, large amounts of computer storage space and computer time are required to solve accurately a typical problem of this type using these packages. The method presented by Nguyen and White[10] can be applied to this problem, but their general method cannot take full advantage of the special properties of this problem; consequently, their method is not as efficient for this type of problem as the method presented here.

Numerical Procedure

The governing equations 1, 4, 6 and the boundary conditions 7-10 are usually nonlinear and coupled. The first step in solving this system of equations using the finite-difference method is to descritize the equations with finite-difference expressions for the first and second derivatives. The next step is to solve the resulting sets of nonlinear coupled algebraic equations by using the Newton-Raphson method.[11] The structure of the coefficient matrix for this system of equations depends on the definition of the vector of variables. In order to obtain a block, nearly tridiagonal coefficient matrix, the following form was chosen for this vector of unknown vectors

$$\mathbf{C}(j) = (C_{\mathrm{I}}(j), \mathbf{C}_{\mathrm{Y}}(1, j), \ldots, \mathbf{C}_{\mathrm{Y}}(k, j), \ldots, \mathbf{C}_{\mathrm{Y}}(M, j), \mathbf{C}_{\mathrm{II}}(j))^{\mathrm{T}} \quad j = 1, 2, \ldots, N \quad [11]$$

where j represents the jth node in the x direction, k represents the kth node in the y direction, and N and M represent the numbers of nodes in the x and y directions, respectively, as shown in Fig. 1b, where regions I and II are shown, for ease of illustration, as if they have a finite thickness, which is not the case. $\mathbf{C}_{\mathrm{I}}(j)$ represents the vector of unknowns given by Eq. 2 at each node point j and $y = 0$ ($k = 1$), and $\mathbf{C}_{\mathrm{II}}(j)$ represents the vector of unknowns given by Eq. 5 at each node point j and $y = S$ ($k = M$). $\mathbf{C}_{\mathrm{Y}}(k, j)$ represents the vector of unknowns in region Y as given by Eq. 3 at the kth node in the y direction and the jth node in the x direction. Using three-point finite-difference expressions for the first and second derivatives and this definition of $\mathbf{C}(j)$, the resulting system of linearized equations can be written in matrix form as[a]

$$\begin{pmatrix} B_1 & D_1 & X & & & & \\ A_2 & B_2 & D_2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & A_j & B_j & D_j & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & A_{N-1} & B_{N-1} & D_{N-1} \\ & & & & Y & A_N & B_N \end{pmatrix} \begin{pmatrix} \mathbf{C}(1) \\ \mathbf{C}(2) \\ \vdots \\ \mathbf{C}(j) \\ \vdots \\ \mathbf{C}(N) \end{pmatrix} = \begin{pmatrix} \mathbf{G}(1) \\ \mathbf{G}(2) \\ \vdots \\ \mathbf{G}(j) \\ \vdots \\ \mathbf{G}(N) \end{pmatrix} \quad [12]$$

where $\mathbf{G}(j)$ represents the constant vector for the system of the linearized equations written as follows for $j = 1, 2, \ldots, N$

$$\mathbf{G}(j) = [\mathbf{G}_{\mathrm{I}n}(j), \mathbf{G}_{\mathrm{Y}m}(1, j), \ldots, \mathbf{G}_{\mathrm{Y}m}(k, j), \ldots, \mathbf{G}_{\mathrm{Y}m}(M, j), \mathbf{G}_{\mathrm{II}n_1}(j)]^{\mathrm{T}} \quad [13]$$

where the subscripts show the length of each subvector. In Eq. 12 A_j ($j = 2, 3, \ldots, N$), D_j ($j = 1, \ldots, N-1$), X, and Y are block matrices $[(n + mM + n_1) \times (n + mM + n_1)]$, that contain only nn and $n_1 n_1$ nonzero elements which are the Jacobian coefficients of the n and n_1 equations in region I and II with respect to $\mathbf{C}_{\mathrm{I}}(3)$ and $\mathbf{C}_{\mathrm{II}}(3)$ for X, with respect to $\mathbf{C}_{\mathrm{I}}(j-1)$ and $\mathbf{C}_{\mathrm{II}}(j-1)$ for A_j, with respect to $\mathbf{C}_{\mathrm{I}}(j+1)$ and $\mathbf{C}_{\mathrm{II}}(j+1)$ for D_j, and with respect to $\mathbf{C}_{\mathrm{I}}(N-2)$ and $\mathbf{C}_{\mathrm{II}}(N-2)$ for Y. These nonzero blocks are located at the top left corner and at the bottom right corner of a common block matrix form with D_j given as an example

$$\mathbf{D}_j = \begin{pmatrix} d_{nn}(j) & & \\ & & \\ & & d_{n_1 n_1}(j) \end{pmatrix} \quad [14]$$

where all of the other blocks in $\mathbf{D}_j$ contain zeros. The block matrices on the diagonal $\mathbf{B}_j$ ($j = 1, 2, \ldots, N$) have a nearly block tridiagonal structure as follows

$$\mathbf{B}_j = \begin{pmatrix} b_{nn} & b_{nm}(1) & b_{nm}(2) & b_{nm}(3) & & & \\ & b_y(1,1) & b_y(2,1) & b_y(3,1) & & & \\ & b_y(1,2) & b_y(2,2) & b_y(3,2) & & & \\ & \ddots & \ddots & \ddots & & & \\ & & b_y(1,k) & b_y(2,k) & b_y(3,k) & & \\ & & & \ddots & \ddots & \ddots & \\ & & & b_y(1,M_1) & b_y(2,M_1) & b_y(3,M_1) & \\ & & & b_y(1,M) & b_y(2,M) & b_y(3,M) & b_{mn_1} \\ & & & b_{n_1 m}(1) & b_{n_1 m}(2) & b_{n_1 m}(3) & b_{n_1 n_1} \end{pmatrix} \quad [15]$$

where $M_1 = M - 1$, b_{nn} and $b_{nm}(i)$ ($i = 1, 2, 3$) are $n \times n$ and $n \times m$ block matrices, representing the Jacobian coefficient matrices of the n equations in region I at the jth node in the x direction with respect to $\mathbf{C}_{\mathrm{I}}(j)$ and $\mathbf{C}_{\mathrm{Y}}(k, j)$, ($k = 1, 2, 3$), respectively. The block matrices b_{mn} ($m \times n$) and $b_y(i, 1)$ ($i = 1, 2, 3$) represent the Jacobian coefficients matrices of the m equations at the interface between regions I and Y with respect to $\mathbf{C}_{\mathrm{I}}(j)$ and $\mathbf{C}_{\mathrm{Y}}(k, j)$ ($k = 1, 2, 3$); $b_y(i, k)$ ($i = 1, 2, 3$, $k = 2, \ldots, M-1$) are $m \times m$ block matrices representing the Jacobian coefficient matrices for the m-equations at the kth node in the y direction with respect to $\mathbf{C}_{\mathrm{Y}}(k-1, j)$, $\mathbf{C}_{\mathrm{Y}}(k, j)$, and $\mathbf{C}_{\mathrm{y}}(k+1, j)$; the block matrices $b_y(i, M)$ ($i = 1, 2, 3$) and b_{mn_1} represent the Jacobian coefficient matrices of the m equations at the interface between regions Y and II with respect to $\mathbf{C}_{\mathrm{Y}}(M-2, j)$, $\mathbf{C}_{\mathrm{Y}}(M-1, j)$, $\mathbf{C}_{\mathrm{Y}}(M, j)$, and $\mathbf{C}_{\mathrm{II}}(j)$, respectively, and the matrices $b_{n_1 m}(i)$ ($i = 1, 2, 3$) and $b_{n_1 n_1}$ represent the Jacobian coefficient matrices of the n_1 equations in the x direction in region II with respect to $\mathbf{C}_{\mathrm{Y}}(M-2, j)$, $\mathbf{C}_{\mathrm{Y}}(M-1, j)$, $\mathbf{C}_{\mathrm{Y}}(M, j)$, and $\mathbf{C}_{\mathrm{II}}(j)$.

Since the coefficient matrix in Eq. 12 is nearly block tridiagonal, LU factorization of the matrix will not result in any nonzero fill-ins beyond those blocks shown in the matrix. In addition, such decomposition causes only small fill-ins within the block matrices (B_j and D_j, $j = 1, 2, \ldots, N$), as discussed in detail later. Therefore, LU factorization is used to solve Eq. 12. That is, by decomposing the coefficient matrix in Eq. 12 into lower and upper triangular matrices, Eq. 12 can be rewritten as

[a] Please see White[12] for definitions of the Jacobian elements given in Eq. 12 for the case with only one region.

$$\begin{pmatrix} B_1 & & & & \\ A_2 & B_2^{\mathrm{L}} & & & \\ & \ddots & \ddots & & \\ & & A_{N-1} & B_{N-1}^{\mathrm{L}} & \\ & & Y & A_N^{\mathrm{L}} & B_N^{\mathrm{L}} \end{pmatrix} \begin{pmatrix} I & D_1^{\mathrm{U}} & X^{\mathrm{U}} & & \\ & I & D_2^{\mathrm{U}} & & \\ & & \ddots & \ddots & \\ & & & I & D_{N-1}^{\mathrm{U}} \\ & & & & I \end{pmatrix} \begin{pmatrix} \mathbf{C}(1) \\ \mathbf{C}(2) \\ \vdots \\ \mathbf{C}(j) \\ \vdots \\ \mathbf{C}(N) \end{pmatrix} = \begin{pmatrix} \mathbf{G}(1) \\ \mathbf{G}(2) \\ \vdots \\ \mathbf{G}(j) \\ \vdots \\ \mathbf{G}(N) \end{pmatrix} \quad [16]$$

where

$$B_1 D_1^{\mathrm{U}} = D_1 \quad [17]$$

$$B_1 X^{\mathrm{U}} = X \quad [18]$$

$$B_2^{\mathrm{L}} D_2^{\mathrm{U}} = D_2 - A_2 X^{\mathrm{U}} \quad [19]$$

where the dimensions of the block matrices d_{nn}^{u}, $d_{nn_1}^{\mathrm{r}}$, $d_{n_1n}^{\mathrm{l}}$, and $d_{n_1n_1}^{\mathrm{u}}$ are carried by their subscripts, and the block matrices $d_i^{\mathrm{l}}(k, j)$ and $d_i^{\mathrm{r}}(k, j)$ $(k = 1, 2, \ldots, M)$ are $m \times n$ and $m \times n_1$, respectively. The matrices A_j $(j = 2, 3, \ldots, N-1)$, and Y in the lower tridiagonal matrix in Eq. 16 are the same as those in Eq. 14; B_1 in Eq. 16 is the same as B_1 in Eq. 15; B_j^{L} is different from B_j only at the four corners of the matrix

$$\mathbf{B}_j^{\mathrm{L}} = \begin{pmatrix} b_{nn}^{\mathrm{l}} & b_{nm}(1) & b_{nm}(2) & b_{nm}(3) & & & b_{\mathrm{r}} \\ b_{mn} & b_y(1,1) & b_y(2,1) & b_y(3,1) & & & \\ & b_y(1,2) & b_y(2,2) & b_y(3,2) & & & \\ & & \ddots & \ddots & & & \\ & & b_y(1,k) & b_y(2,k) & b_y(3,k) & & \\ & & & \ddots & \ddots & & \\ & & & b_y(1,M_1) & b_y(2,M_1) & b_y(3,M_1) & \\ & & & b_y(1,M) & b_y(2,M) & b_y(3,M) & b_{nm_1} \\ b_{\mathrm{l}} & & & b_{n_1m}(1) & b_{n_1m}(2) & b_{n_1m}(3) & b_{n_1n_1}^{\mathrm{l}} \end{pmatrix} \quad [30]$$

$$B_j^{\mathrm{L}} = B_j - A_j D_{j-1}^{\mathrm{U}} \quad j = 2, 3, \ldots, N-1 \quad [20]$$

$$B_j^{\mathrm{L}} D_j^{\mathrm{U}} = D_j \quad j = 3, 4, \ldots, N-1 \quad [21]$$

$$A_N^{\mathrm{L}} = A_N - Y D_{N-2}^{\mathrm{U}} \quad [22]$$

$$B_N^{\mathrm{L}} = B_N - A_N^{\mathrm{L}} D_{N-1}^{\mathrm{U}} \quad [23]$$

Equation 16 is then solved for $\mathbf{C}(j)$ $(j = 1, 2, \ldots, N)$ by the following forward calculations (Eq. 24-26) and backward substitutions (Eq. 27 and 28)

$$B_1 \mathbf{C}^*(1) = G(1) \quad j = 1 \quad [24]$$

$$B_j^{\mathrm{L}} \mathbf{C}^*(j) = G(j) - A_j \mathbf{C}^*(j-1) \quad j = 2, 3, \ldots, N-1 \quad [25]$$

$$B_N^{\mathrm{L}} \mathbf{C}(N) = G(N) - Y \mathbf{C}^*(N-2) - A_N^{\mathrm{L}} \mathbf{C}^*(N-1) \quad [26]$$

$$\mathbf{C}(j) = \mathbf{C}^*(j) - D_j^{\mathrm{U}} \mathbf{C}(j+1) \quad \mathbf{j} = N-1, N-2, \ldots, 2 \quad [27]$$

$$\mathbf{C}(1) = \mathbf{C}^*(1) - D_j^{\mathrm{U}} \mathbf{C}(2) - X^{\mathrm{U}} \mathbf{C}(3) \quad [28]$$

where $\mathbf{C}^*(j)$ is the usual intermediate vector in the LU factorization method.

Since the block matrices X, D_j $(j = 1, 2, \ldots, N-1)$ in Eq. 12 have only two small block nonzero elements at two corners as shown in Eq. 14, the LU factorization via Eq. 17 through 23 will cause only small two block column fill-ins in the block matrices X^{U} and D_j^{U} in the upper tridiagonal matrix in Eq. 16. For example, D_j^{U} is of the form

$$D_j^{\mathrm{U}} = \begin{pmatrix} d_{nn}^{\mathrm{u}}(j) & d_{nn_1}^{\mathrm{r}}(j) \\ d_i^{\mathrm{l}}(1,j) & d_i^{\mathrm{r}}(1,j) \\ \vdots & \vdots \\ d_i^{\mathrm{l}}(M,j) & d_i^{\mathrm{r}}(M,j) \\ d_{n_1n}^{\mathrm{l}}(j) & d_{n_1n_1}^{\mathrm{u}}(j) \end{pmatrix} \quad [29]$$

where for $j = 2, 3, \ldots, N-1$

$$b_{nn}^{\mathrm{l}} = b_{nn} - a_{nn} d_{nn}^{\mathrm{u}}(j-1) \quad [31]$$

$$b_{\mathrm{r}} = -a_{nn} d_{nn_1}^{\mathrm{r}}(j-1) \quad [32]$$

$$b_{\mathrm{l}} = -a_{n_1n_1} d_{n_1n}^{\mathrm{l}}(j-1) \quad [33]$$

$$b_{n_1n_1}^{\mathrm{l}} = b_{n_1n_1} - a_{n_1n_1} d_{n_1n_1}^{\mathrm{u}}(j-1) \quad [34]$$

and for $j = N$

$$b_{nn}^{\mathrm{l}} = b_{nn} - a_{nn}^{\mathrm{l}} d_{nn}^{\mathrm{u}}(N-1) - a_{nn_1}^{\mathrm{r}} d_{n_1n}^{\mathrm{l}}(N-1) \quad [35]$$

$$b_{\mathrm{r}} = -a_{nn}^{\mathrm{l}} d_{nn_1}^{\mathrm{r}}(N-1) - a_{nn_1}^{\mathrm{r}} d_{n_1n_1}^{\mathrm{u}}(N-1) \quad [36]$$

$$b_{\mathrm{l}} = -a_{n_1n}^{\mathrm{l}} d_{nn}^{\mathrm{u}}(N-1) - a_{n_1n_1}^{\mathrm{l}} d_{n_1n}^{\mathrm{l}}(N-1) \quad [37]$$

$$b_{n_1n_1}^{\mathrm{l}} = b_{n_1n_1} - a_{n_1n}^{\mathrm{l}} d_{nn_1}^{\mathrm{r}}(N-1) - a_{n_1n_1}^{\mathrm{l}} d_{n_1n_1}^{\mathrm{u}}(N-1) \quad [38]$$

A_N^{L} in Eq. 16 is also different from A_N in Eq. 12 and has two small block fill-ins at the top right and bottom left corners

$$A_N^{\mathrm{L}} = \begin{pmatrix} a_{nn}^{\mathrm{l}} & a_{nn_1}^{\mathrm{r}} \\ a_{n_1n}^{\mathrm{l}} & a_{n_1n_1}^{\mathrm{l}} \end{pmatrix} \quad [39]$$

where

$$a_{nn}^{\mathrm{l}} = a_{nn} - y_{nn} d_{nn}^{\mathrm{u}}(N-2) \quad [40]$$

$$a_{nn_1}^{\mathrm{r}} = -y_{nn} d_{nn}^{\mathrm{u}}(N-2) \quad [41]$$

$$a_{n_1n}^{\mathrm{l}} = -y_{n_1n_1} d_{n_1n}^{\mathrm{l}}(N-2) \quad [42]$$

$$a_{n_1n_1}^{\mathrm{l}} = a_{n_1n_1} - y_{n_1n_1} d_{n_1n_1}^{\mathrm{u}}(N-1) \quad [43]$$

In order to determine the nonzero blocks in X^{U}, D_j^{U} $(j = 1, 2, \ldots, N)$, and the intermediate solution vectors $\mathbf{C}^*(j)$ $(j = 1, 2, \ldots, N-1)$, and $\mathbf{C}(N)$, Eq. 17, 18, 21, and 24 to 26 have to be solved. To do this, LU decomposition is applied to B_1 and B_j^{L} $(j = 2, 3, \ldots, N)$. For example, decomposition of B_j^{L} yields the following lower and upper matrices

$$\mathbf{B}_j^L = \begin{pmatrix} b_{nn}^l & & & & & & & & & \\ b_{mn} & b_{nm}(1,1) & & & & & & & & \\ & b_y(1,2) & b_y^l(2,2) & & & & & & & \\ & & \ddots & \ddots & & & & & & \\ & & & b_y(1,k) & b_y^l(2,k) & & & & & \\ & & & & \ddots & \ddots & & & & \\ & & & & & b_y(1,M) & b_y^l(2,M) & b_y^l(3,M) & & \\ b_1 & b_f^l(1) & b_f^l(M_3) & b_{n_1m}^l(1) & b_{n_1m}^l(2) & b_{n_1m}^l(3) & b_{n_1n_1}^* & & b_r^u & \\ I_{nn} & b_{nm}^u(1) & b_{nm}^u(2) & b_{nm}^u(3) & & & & & b_f^r(1) & \\ & I_{mm} & b_y^u(2,1) & b_y^u(3,1) & & & & & b_f^r(2) & \\ & & I_{mm} & b_y^u(3,2) & & & & & \vdots & \\ & & & \ddots & \ddots & & & & b_f^r(k) & \\ & & & & I_{mm} & b_y^u(3,k) & & & \vdots & \\ & & & & & \ddots & \ddots & & & \\ & & & & & & I_{mm} & b_y^u(3,M_1) & b_f^r(M_1) & \\ & & & & & & & I_{mm} & b_{mn_1}^u & \\ & & & & & & & & I_{n_1n_1} & \end{pmatrix} \quad [44]$$

where $M_3 = M - 3$, $b_f^l(i)$ $(i = 1, 2, \ldots, M - 3)$ and $b_f^r(i)$ $(i = 1, 2, \ldots, M - 1)$ are $n_1 \times m$ and $m \times n_1$ block matrices that result from this factorization. To determine those nonzero block matrices in the upper matrix of Eq. 44, many sets of equations with the coefficient matrices on the diagonal (b_{nn}^l, $b_y^l(1, 1)$, $b_y^l(2, i)$ $(i = 2, 3, \ldots, M - 1)$, $b_y^l(3, M)$, and $b_{n_1n_1}^*$ and in the lower triangular matrix of Eq. 44 have to be solved, as shown in Appendix A. Once B_j^L has been decomposed into the lower and upper parts as given in Eq. 44, X^U, D_j^U, and $\mathbf{C}^*(j)$ can be easily obtained from Eq. 17, 18, 21, and 24 and 25; and, subsequently, $\mathbf{C}(j)$ can be obtained from Eq. 26 to 28.

It can be seen from the procedure presented above that the special arrangement of the variable vectors given by Eq. 11 for a pseudo-two-dimensional boundary-value problem leads to a nearly block tridiagonal structure for both the main coefficient matrix in Eq. 12 and those submatrices on the diagonal (B_j, $j = 1, 2, \ldots, N$) and also creates the nearly empty matrices X, D_j's, and A_j's. Use of LU factorization for the main coefficient matrix in Eq. 12 and for B_1 and B_j^L $(j = 2, 3, \ldots, N)$ causes only small fill-ins of nonzero elements, and, consequently, requires much less computation time to solve the system of the equations than a method that does not utilize these properties of the matrices.

It is worth mentioning that if some other method such as orthogonal collocation were to be used instead of finite differences, a similar algorithm could be developed using the same concepts as those presented here for finite differences.

Programming Procedure

It should be noted in the above procedure in solving Eq. 12 via 16 that the matrices B_j and A_j are used only once and that only those nonzero elements in the upper matrices D_j^U and X^U and the intermediate solution $\mathbf{C}^*(j)$ are needed during the forward calculations. Therefore, the calling sequence used by Newman[13] in his one-dimensional subroutine BAND is used here in the programming scheme in order to reduce storage space. That is, the nonzero elements in D_j^U and the intermediate solution $\mathbf{C}^*(j)$ $(j = 1, 2, \ldots, N - 1)$ are determined and stored at each node j. When the last node N in the x direction is reached, the solution of the system is calculated by backward substitution using the intermediate solution, and the upper matrices D_j^U and X^U. Therefore, this sequence requires storage space for the nonzero elements in each D_j^U $(j = 1, 2, \ldots, N - 1)$ as shown in Eq. 29, those in X^U, and the intermediate solution $\mathbf{C}^*(j)$ $(j = 1, 2, \ldots, N - 1)$, only. The storage space needed for nonzero elements in A_j, B_j, $G(j)$, and for those fill-ins in B_j^L and A_N^L are temporary because these coefficients are renewed at each node j.

An Example

The example presented here is a simple model for a porous TiS_2 electrode with an ideal binary electrolyte.[2,14] This example was chosen because it is a simple example of a pseudo-two-dimensional problem and because it has been solved by using the method presented here and by the method of Nguyen and White,[10] so that the effectiveness of the method presented here can be determined. The problem has only two phases (electrolyte and solid) and involves diffusion and migration in the electrolyte phase in one direction only (x) and diffusion in the solid phase in one direction only (y). The third region indicated by II in Fig. 1 does not exist in this problem. The governing equations for the electrolyte phase consist of a mass balance and a charge balance

$$\frac{\partial \epsilon C}{\partial t} = D + \frac{\partial^2 C}{\partial x^2} + D + \frac{F}{RT}\left(\frac{\partial C}{\partial x}\frac{\partial \phi}{\partial x} + C\frac{\partial^2 \phi}{\partial x^2}\right) - gD_s\left(\frac{\partial C_s}{\partial y}\right)_{y=S} \quad [45]$$

$$F\left[(D_+ - D_-)\frac{\partial^2 C}{\partial x^2} + (D_+ + D_-)\frac{F}{RT}\left(\frac{\partial C}{\partial x}\frac{\partial \phi}{\partial x} + C\frac{\partial^2 \phi}{\partial x^2}\right)\right] - FgD_s\left(\frac{\partial C_s}{\partial y}\right)_{y=S} = 0 \quad [46]$$

where the symbols C and ϕ represent the dependent variables in the electrolyte phase and C_s represents the dependent variable in the solid phase, and the other symbols are constants.

The boundary conditions for Eq. 45 and 46 are as follows

$$\left(\frac{\partial C}{\partial x}\right)_{x=0} = 0 \quad [47]$$

$$\left(\frac{\partial \phi}{\partial x}\right)_{x=0} = 0 \quad [48]$$

$$-F\left[(D_+ - D_-)\frac{\partial C}{\partial x} + \frac{F}{RT}(D_+ + D_-)C\frac{\partial \phi}{\partial x}\right]_{x=L} = i_{app} \quad [49]$$

$$C|_{x=L} = C_o \quad [50]$$

where i_{app} and C_o are constants.

The governing equations for the solid phase in the y direction is Fick's diffusion equation in cylindrical coordinates

$$\frac{\partial C_s}{\partial t} = D_s\left[\frac{\partial^2 C_s}{\partial y^2} + \frac{1}{y}\frac{\partial C_s}{\partial y}\right] \quad [51]$$

The boundary conditions for this equation are

$$\left.\frac{\partial C_s}{\partial y}\right|_{y=0} = 0 \quad [52]$$

$$(\pi - \phi)|_{y=S} = (\pi - \phi)_o + \frac{RT}{F}\left[\ln\left(\frac{C_s^* - C_s}{C_s}\right) + \ln\frac{C}{C_o} - f\left(\frac{C_s}{C_s^*} - 0.5\right)\right]_{y=S} \quad [53]$$

where the symbols $(\pi - \phi)_o$, C_s^* and f represent constants. The initial conditions for the above problem are $C_s = C_s^o$, and $C = C_o$ for all x and y.

Table I. Comparison of CPU times.

Method	CPU time[a] (s) CRAY-YMP[b]	VAX-8650 (DP)[c]	VAX-8650 (QP)[d]	SUN/SPARC	No. of nodes
PTWO	0.5725×10^{-2}	0.1000	0.3200	6.000×10^{-2}	$N = 21$
BAND	4.4100×10^{-2}	1.6400	4.5200	7.7500	$M = 21$
PTWO	2.8309×10^{-2}	0.5100	1.5900	0.3100	$N = 51$
BAND	0.8306	—	136.22	61.590	$M = 51$
PTWO	6.0667×10^{-2}	1.1700	2.6600	0.6300	$N = 101$
BAND	1.2023	—	257.82	161.34	$M = 51$

[a] Estimated by using the IMSL function CTIME().
[b] The programs were run with single precision.
[c] The programs were run with double precision.
[d] The programs were run with quadruple precision.

The governing equations for the electrolyte phase (Eq. 45 and 46) contain a derivative of the variable in the solid phase at the solid surface ($y = S$), the governing equation for the solid phase (Eq. 51) does not contain any variable of the electrolyte phase, but its boundary condition (Eq. 53) is strongly coupled with the variables of the electrolyte phase. The equations in this problem can be solved easily using the algorithm presented above. First, the model equations are descritized using three-point finite-difference equations for both the first and second derivatives and the Crank-Nicolson approximation is used to obtain $(\Delta t)^2$ accuracy in time. The resulting sets of nonlinear, coupled algebraic equations are solved using the Newton-Raphson method and the procedure presented here.

Table I presents a comparison of CPU times for one iteration in solving the above problem with the subroutines BAND using the method of Nguyen and White[10] and with PTWO on three different computers: a SUN workstation, a VAX-8650, and a CRAY-YMP. Although the differences in the CPU times used by PTWO and by BAND depend on the number of nodes, as shown in Table I, PTWO required less CPU time than BAND by at least one hundred times on the SUN workstation. The CPU time spent by PTWO on the VAX-8650 was fourteen to ninety-seven times less than that by BAND when the programs were run with quadruple precision. When the number of nodes was increased and the programs were run with double precision on the VAX-8650, an error occurred in BAND due to overflow, which did not occur in PTWO. It is interesting to note that both BAND and PTWO required less CPU time on the SUN workstation than on the VAX-8650. On the CRAY-YMP, the CPU time used by PTWO was seven to twenty-nine times less than that by BAND. For the example problem given above, it was found that use of a large number of nodes is essential to obtain an accurate solution, and an actual simulation would require approximately three thousand time steps.[15] Consequently, the CPU time would be prohibitively large if the problem were to be solved using BAND on a SUN workstation or on a VAX-8650, whereas PTWO provided accurate results for an affordable CPU time.

Conclusions

Pseudo-two-dimensional problems can be solved by using the finite-difference method presented in this work with much less computation time and storage space than that required by a full two-dimensional solver.[10] This is because in this method the vectors of variables are arranged so that the Jacobian coefficient matrix for the system of equations and the submatrices for each node are nearly block tridiagonal. The highly sparse and nearly block tridiagonal properties of these matrices are utilized to reduce the storage and CPU time by using LU factorization to solve these systems of equations.

Acknowledgments

The authors are grateful for the support of this work by the Office of Research & Development, Hughes Aircraft Company, the Center for Space Power at Texas A&M University, and for the computer time on the Cray-YMP supercomputer at Texas A&M received through a grant from Cray Research, Inc.

Manuscript submitted July 8, 1993; revised manuscript received Sept. 27, 1993.

The University of South Carolina assisted in meeting the publication costs of this article.

APPENDIX A

LU factorization of B_j^l.

The block matrices b_{nn}^l, b_{mn}, $b_y(1, k)$ ($k = 2, 3, \ldots, M$), and b_1 in the lower triangular matrix in Eq. 44 are the same as those in Eq. 30. The other nonzero block matrices in the lower and upper block triangular matrices are calculated as follows

$$b_{nn}^l b_{nm}^u(i) = b_{nm}(i) \quad i = 1, 2, 3 \qquad [A\text{-}1]$$

$$b_{nn}^l b_r^u = b_r \qquad [A\text{-}2]$$

for $k = 1$

$$b_y^l(1, 1) = {}_y(1, 1,) - b_{mn} b_{nm}^u(1) \qquad [A\text{-}3]$$

$$b_y^l(1, 1) b_y^u(i, 1) = b_y(i, 1) - b_{mn} b_{nm}^u(i) \quad i = 2, 3 \qquad [A\text{-}4]$$

$$b_y^l(1, 1) b_f^r(i, 1) = -b_{mn} b_r^u \qquad [A\text{-}5]$$

for $k = 2$

$$b_y^l(2, k) = b_y(2, k) - b_y(1, k) b_y^u(2, k - 1) \qquad [A\text{-}6]$$

$$b_y^l(2, k) b_y^u(3, k) = b_y(3, k) - b_y(1, k) b_y^u(3, k - 1) \qquad [A\text{-}7]$$

$$b_y^l(2, k) b_f^r(k) = -b_y(1, k) b_f^r(k - 1) \qquad [A\text{-}8]$$

for $3 \le k \le M$

$$b_y^l(2, k) = b_y(2, k) - b_y(1, k) b_y^u(3, k - 1) \qquad [A\text{-}9]$$

$$b_y^l(2, k) b_y^u(3, k) = b_y(3, k) \qquad [A\text{-}10]$$

$$b_y^l(2, k) b_f^r(k) = -b_y(1, k) b_f^r(k - 1) \qquad [A\text{-}11]$$

for $k = M$

$$b_y^l(2, k) = b_y(2, k) - b_y(1, k) b_y^u(3, k - 2) \qquad [A\text{-}12]$$

$$b_y^l(3, k) = b_y(3, k) - b_y^l(2, k) b_y^u(3, k - 1) \qquad [A\text{-}13]$$

$$b_y^l(3, k) b_{mn_1}^u = b_{mn_1} - b_y(1, k) b_f^r(k - 2) - b_y^l(2, k) b_f^r(k - 1) \qquad [A\text{-}14]$$

$$b_f^l(1) = -b_l b_{nm}^u(1) \qquad [A\text{-}15]$$

$$b_f^l(2) = -b_l b_{nm}^u(2) - b_f^l(1) b_y^u(2, 1) \qquad [A\text{-}16]$$

$$b_f^l(3) = -b_l b_{nm}^u(3) - b_f^l(1) b_y^u(3, 1) - b_f^l(2) b_y^u(3, 2) \qquad [A\text{-}17]$$

$$b_f^l(i) = -b_f^l(i - 1) b_y^u(3, i - 1) \quad i = 4, 5, \ldots, M - 3 \qquad [A\text{-}18]$$

$$b_{n_1 m}^l(1) = b_{n_1 m}(1) - b_f^l(M - 3) b_y^u(3, M - 3) \qquad [A\text{-}19]$$

$$b_{n_1 m}^l(2) = b_{n_1 m}(2) - b_{n_1 m}^l(1) b_y^u(3, M - 2) \qquad [A\text{-}20]$$

$$b_{n_1 m}^l(3) = b_{n_1 m}^l(3) - b_{n_1 m}^l(2) b_y^u(3, M - 1) \qquad [A\text{-}21]$$

$$b_{n_1 n_1}^* = b_{n_1 n_1}^l - b_l b_r^u - \sum_{k=1}^{M-3} b_f^l(k) b_f^r(k) - b_{n_1 m}^l(1) b_f^r(M - 2) - b_{n_1 m}^l(2) b_f^r(M - 1) - b_{n_1 m}^l(3) b_{mn_1}^u \qquad [A\text{-}22]$$

Equations A-1, A-2, A-4, A-5, A-7, A-8, A-10, A-11, and A-14 in this Appendix have to be solved to obtain the

nonzero elements in the upper tridiagonal block matrix in Eq. 44. LU factorization is used here to solve these equations.

REFERENCES

1. G. F. Froment and K. B. Bischoff, *Chemical Reactor Analysis and Design,* 2nd ed., p. 477, John Wiley & Sons, Inc., New York (1990).
2. K. West, T. Jacobsen, and S. Atlung, *This Journal,* **129,** 1480 (1982).
3. J. Bouet, F. Richard, and Ph. Blanchard, in *Proceedings of the Symposium on Nickel Hydroxide Electrode,* D. A. Corrigan and A. H. Zimmerman, Editors, PV 90-4, p. 260, The Electrochemical Society Softbound Proceedings Series, Pennington, NJ (1990).
4. C. D. Holland and A. I. Liapis, *Computer Methods for Solving Dynamic Separation Problems,* p. 389, McGraw-Hill, Inc., New York (1983).
5. M. Sinha, Ph.D. Thesis, University of California, Los Angeles (1982).
6. T. E. Springer, T. A. Zawodzinski, and S. Gottesfeld, in *Proceedings of the Symposium on Modeling of Batteries and Fuel Cells,* R. E. White, M. W. Verbrugge, and J. F. Stockel, Editors, PV 91-10, p. 209, The Electrochemical Society Proceedings Series, Pennington, NJ (1991).
7. W. E. Schiesser, *DSS/2: Fortran 77 Code for Systems of Ordinary and One, Two, and Three-Dimensional Partial Differential Equations,* Lehigh University and Naval Air Development Center, Leigh, PA (1988).
8. T. Hailu, MS Thesis, Texas A&M University, College Station, TX (1990).
9. IMSL, MATH/LIBRARY User's Manual, p. 11, IMSL, Houston, TX (1987).
10. T. V. Nguyen and R. E. White, *Comput. Chem. Engng.,* **11,** 543 (1987).
11. B. A. Finlayson, *Nonlinear Analysis in Chemical Engineering,* p. 11, McGraw-Hill, Inc., New York (1980).
12. R. E. White, *Ind. Engn. Chem. Fundam.,* **17,** 367 (1978).
13. J. S. Newman, *Electrochemical Systems,* Appendix C, 2nd Ed., Prentice-Hall, Englewood Cliffs, NJ (1991).
14. R. M. Spotnitz, D. Zuckerbrod, S. L. Johnson, J. T. Lundquist, and R. E. White, in *Proceedings of the Symposium on Modeling of Batteries and Fuel Cells,* R. E. White, M. W. Verbrugge, and J. F. Stockel, Editors, PV 91-10, p. 88, The Electrochemical Society Proceedings Series, Pennington, NJ (1991).
15. Z. Mao and R. E. White, *J. Power Sources,* **43,** 181 (1993).

Study of the Redox Process of Poly (2-Naphthol) Film Using *In Situ* Multiple Internal Reflection FTIR Spectroscopy

Minh-Chau Pham and Pierre-Camille Lacaze

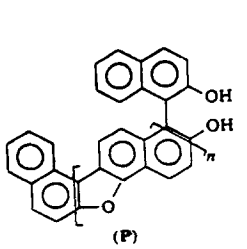
Institut de Topologie et de Dynamique des Systemés de l'Universite Paris 7, 75005 Paris, France

ABSTRACT

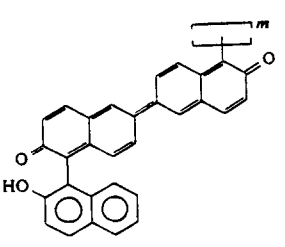
Poly(NAP-2) is a novel conducting polymer film obtained by electrochemical oxidation of 2-naphthol. The polymer structure consists of polymers P and oligomers Q. The P structure comprises alternate naphthylene and furan rings while the Q structure bears quinonoid groups in the chain. The electrochemical oxidation-reduction of the two forms P and Q in poly(NAP-2) film was studied. The redox process of the furan groups in the polymer structure P is detected in neutral acetonitrile while that involving Q/H_2Q couple in Q structure is observed in acidic acetonitrile solution.

Recently, we showed that a novel electroactive and conducting polymer film, poly(2-naphthol) [poly(NAP-2)], was obtained by electrochemical oxidation of 2-naphthol in acetonitrile solution. The polymer structure and the electropolymerization mechanism were elucidated by *in situ* multiple internal reflection Fourier transform infrared spectroscopy (MIRFTIRS)[1] and *in situ* electron spin resonance (ESR) study.[2]

The proposed structure for poly(NAP-2) film is composed of two forms, P and Q. The polymer P comprises alternate naphthylene and furan rings analogously to poly(NAP-1) film[3] while the Q structure involves oligomers bearing quinonoid groups



(P)



(Q)

Polymer films can be deposited on Pt or glassy carbon electrodes at constant potential (1.5 or 2 V *vs.* SCE) or by potential cycling between 0.2 and 1.5 V or between 0.2 and 2 V *vs.* SCE.

When the potential limit is 2 V, the formation of the quinonoid structure Q is favored while P structure formation is predominant when the potential limit is 1.5 V.[2]

In the present paper, an investigation of the mechanism of the electrochemical oxidation-reduction process of poly(NAP-2) film is performed by electrochemical and MIRFTIRS studies.

Experimental

Materials.—2-Naphthol (Aldrich Chemical Company) was sublimed before use. The electrolytes, NBu_4ClO_4 and $LiAsF_6$ (Fluka), were used without further purification. Acetonitrile was provided by Aldrich (spectrophotometric grade).

Preparation of poly(NAP-2) film.—For the double potential step experiments, poly(NAP-2) films were prepared by electropolymerization of 2-naphthol (0.1*M*) in an acetonitrile solution containing 0.1*M* of the electrolyte ($LiAsF_6$ or $LiClO_4$) by potential scanning between 0.2 and 1.5 V or 2 V *vs.* SCE. The film thickness was measured by scanning electron microscopy (SEM). A strip of polymer was scratched out from a polymer film deposited on a Pt substrate to reveal a cross section of the film.

In the MIRFTIRS studies, in order to investigate the redox process of the film in the whole film thickness, the film