A PEDAGOGICAL LOOK AT FULLY- AND SEMI-IMPLICIT SOLUTIONS TO THERMALHYDRAULIC SYSTEM EQUATIONS

by

Wm. J. Garland
Department of Engineering Physics
McMaster University
Hamilton, Ontario
Canada L8S 4M1

ABSTRACT

Porsching's classical solution algorithm for the simulation of thermalhydraulic systems is revisited with a view to pedagogy. The general linearized system equations are used to develop a fully-implicit, back-substituted solution. Matrix notation is used and the solution algorithm is explored using a simple example. Porsching's algorithm and other approximations are derived as special cases.

KEYWORDS: Thermalhydraulics, Systems, Porsching, Jacobian, Implicit, Semi-implicit.

INTRODUCTION

One of the more successful algorithms for thermalhydraulic simulation is based on the work of Porsching 1969, 1971. This algorithm, involving the Jacobian (derivative of the system state matrix), is used originally in the computer program FLASH-4 (Porsching 1969) and subsequently in the Ontario Hydro program SOPHT (Chang 1977) and evolved into forms used in RETRAN (Agee 1982).

The strength of Porsching's approach lies in its recognition of flow as the most important dependent parameter and, hence, its fully implicit treatment of flow. This leads to excellent numerically stability, consistency and convergence. Further, the Jacobian permits a generalized approach to the linearization of nonlinear systems. This allows the development of a system state matrix which contains all the system dynamics in terms of the dependent parameters of mass, energy and flow. Back substitution finally gives a matrix rate equation in terms of the system flow (the unknown) and the system derivatives. While this approach is certainly a proven and successful one, it has some disadvantages. The matrix rate equation involving the Jacobian is as complicated as it is general. The resulting expressions are somewhat obtuse and it is difficult to obtain an intuitive feel for the system. This complexity also hinders implementation in a simulation code and makes error tracking a tedious process. The pervasiveness and obtuseness of the algorithm begs a revisit so as to distil the salient features, leaving them exposed for pedagogy and further scrutiny.

Recently, (Garland 1987), work has been presented on the use of the Rate Form of the equation of state. This work showed that by casting the equation of state in the form of a rate equation rather than the normal algebraic form, the system state matrix can be more logically formed from the normal conservation rate equations for mass, energy and momentum plus the pressure rate equation. These form the four cornerstone equations in thermalhydraulic systems analysis (Figure 1). Numerical implementation of the rate form proved to be very successful, leading to roughly a factor of 10 improvement over the algebraic form of the equation of state, largely due to the iterative nature of the algebraic form. Incorporating the implicit pressure dependency in the numerical method also drastically improved the numerical stability.

Since Porsching's method also carried the pressure dependency implicitly (via the Jacobian), the question arises as to how the Rate form compares the Porsching's method. This paper is devoted to an explanatory derivation of the fully-implicit back-substituted form (FIBS), which is a more general than the Rate form. It is shown that the Porsching form is identical to the Rate form and is a subset of the fully-implicit back-substituted form and is easily derived from it. The FIBS form thus offers an alternative to Porsching, is found to be of some pedagogical usefulness and is far more intuitive and easier to code.

DERIVATION OF FIBS

Following Porsching (Porsching 1971), the general form of system equations can be written

$$\dot{y} = F(t,y) \tag{1}$$

which is linearized, assuming no explicit t dependence to give:

$$\dot{y} = F^t + \Delta t\ J\ \dot{y} \tag{2}$$

or

$$\Delta y = \Delta t\ F^t + \Delta t\ J\ \Delta y \tag{3}$$

to give

$$[I - \Delta t\ J]\ \Delta y = \Delta t\ F^t \tag{4}$$

where J is the system Jacobian.

For typical thermalhydraulic systems using the node-link notation[1]:

$$\frac{dW_j}{dt} = \frac{A_j}{L_j}\left(P_u + S_{WP}\,\Delta P_u - P_d - S_{WP}\,\Delta P_d\right) - k_j\left(W_j + S_{WW}\,\Delta W_j\right)^2 + b_{wj}$$

$$= \frac{\Delta W_j}{\Delta t} \tag{5}$$

Typically $b_{wj} = (A_j/L_j)\ (h_j \rho_j g + \Delta P_{pump})$ where h_j = height,

$$\frac{dM_i}{dt} = \sum_{k \epsilon d_i}\left(W_k + S_{MW}\,\Delta W_k\right) - \sum_{k \epsilon u_i}\left(W_k + S_{MW}\,\Delta W_k\right) = \frac{\Delta M_i}{\Delta t} \tag{6}$$

$$\frac{dH_i}{dt} = \sum_{k \epsilon d_i}\left(W_k + S_{HW}\,\Delta W_k\right)\frac{\left(H_k + S_{HH}\,\Delta H_k\right)}{\left(M_k + S_{HM}\,\Delta M_k\right)} - \sum_{k \epsilon u_i}\left(W_k + S_{HW}\,\Delta W_k\right)\frac{\left(H_k + S_{HH}\,\Delta H_k\right)}{\left(M_k + S_{HM}\,\Delta M_k\right)} + Q_i$$

$$- \sum_{k \epsilon d_i}\left(\frac{W_k H_k}{M_k} + \frac{S_{HW} H_k}{M_k}\,\Delta W_k + \frac{S_{HH} W_k}{M_k}\,\Delta H_k - \frac{S_{HM} W_k H_k}{M_k^2}\,\Delta M_k\right) -$$

$$\sum_{k \epsilon u_i}\left(\frac{W_k H_k}{M_k} + \frac{S_{HW} H_k}{M_k}\,\Delta W_k + \frac{S_{HH} W_k}{M_k}\,\Delta H_k - \frac{S_{HM} W_k H_k}{M_k^2}\,\Delta M_k\right) + Qi$$

---

[1] Porsching actually uses U, total energy rather than H, total enthalpy in a hybrid form:

$$\dot{U}_i = \sum_{k \epsilon I_i}\left(S_k/M_k\right) W_k - \sum_{k \epsilon I_i}\left(H_k/M_k\right) W_k + Q_i$$

There is no advantage to tracking both H and U in a simulation; thus in this paper, H is used throughout.

$$- \frac{\Delta H_i}{\Delta t} \tag{7}$$

$$\Delta P_i = \frac{\partial P_i}{\partial M_i} \Delta M_i + \frac{\partial P_i}{\partial H_i} \Delta H_i + \frac{\partial P_i}{\partial V_i} \Delta V_i \tag{8}$$

or

$$\frac{\Delta Pi}{\Delta t} = C_{1i} \frac{\Delta M_i}{\Delta t} + C_{2i} \frac{\Delta H_i}{\Delta t} \quad \text{for constant volume}$$

where $k \epsilon d_i$ indicates a sum over links for which the node i is a downstream node (i.e. links are sources) and $k \epsilon u_i$ is for upstream nodes.

Switches, S, are used to provide user control over degree of implicitness:

0 = explicit

1 = implicit.

The system unknowns to be solved for are ΔW, ΔM, ΔH and ΔP using equations (5), (6), (7) and (8). The general strategy is to reduce the number of unknowns so that the size of the matrices to be inverted in the simuitaneous solution of these equations is reduced. The mass equation (6) is simple and is used to eliminate ΔM in terms of ΔW. Flow is chosen as the prime variable since it is the main actor in thermalhydraulic systems. The enthalpy equation poses a problem as it is too complex to permit a simple substitution. Porsching surmounts this by setting $S_{HH} = S_{HM} = 0$, ie making the solution explicit in specific enthalpy. However, we need not make this assumption; by casting the equations in matrix notation, the full implicitness can be retained while still allowing the back substitutions to be made.

Proceeding then, using matrix notation:

$$\Delta M = \Delta t \, A^{MW} \, [W^t + S_{MW} \, \Delta W] \tag{6a}$$

where, for a 4 node - 5 link example (Figure 2):

$$A^{MW} = \begin{array}{c} \text{links} \rightarrow \\ \begin{matrix} 1 \\ 1 \\ 0 \\ 0 \end{matrix} \end{array} \begin{array}{c} \begin{matrix} \quad 1 \quad & \quad 2 \quad & \quad 3 \quad & \quad 4 \quad & \quad 5 \quad \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 \end{bmatrix} \end{array} \begin{array}{l} \text{nodes} \\ \downarrow \\ \\ \\ \\ \end{array} \tag{9}$$

This matrix contains the total system geometry. It is constructed by the following procedure:

For each column (link), insert -1 for the upstream node and +1 for the downstream node for that link since the link supplies (adds) flow to the downstream node and takes it away from

the upstream node. Flow reversal is handled automatically since the sign of W will take care of mass accounting properly.

The form of other matrices in the following are derivable from A^{MW}. This can be used to advantage in coding. The input data for each link need only contain pointers to the upstream node and the downstream node for that link. This allows A^{MW} to be created. In short, the upstream node and downstream node for each link completely defines the geometry and this can be used to programming advantage.

The flow equation is:

$$\Delta W = \Delta t \{ A^{WP} [P^t + S_{WP} \Delta P] + A^{MW} [W^t + 2S_{WW} \Delta W] + B^W \} \quad (5a)$$

Where:

$$A^{WW} = \begin{bmatrix} -k_1|W_1| & & & \\ & -k_2|W_2| & 0 & \\ & & \ddots & \\ & 0 & & -k_5|W_5| \end{bmatrix} \quad (10)$$

$$A^{WP} = \begin{bmatrix} A_1/L_1 & -A_1/L_1 & 0 & 0 \\ 0 & A_2/L_2 & -A_2/L_2 & 0 \\ 0 & 0 & A_3/L_3 & -A_3/L_3 \\ -A_4/L_4 & 0 & 0 & A_4/L_4 \\ 0 & -A_5/L_5 & 0 & A_5/L_5 \end{bmatrix} \quad (11)$$

note that A^{WP} is formed easily from A^{MW} by the following procedure:

First multiply (A^{MW} by $-A_1/L_1, -A_2/L_2, \ldots A_5/L_5)^{-1}$

Then transpose the resulting matrix to give A^{WP}.

$$B^W = \begin{bmatrix} A_1/L_1(h_1\rho_1 g + \Delta P_{pump1}) \\ A_2/L_2(h_2\rho_1 g + \Delta P_{pump2}) \\ \vdots \\ \vdots \end{bmatrix} \quad (12)$$

Finally:

$$\Delta H = \Delta t \{ A^{HW} [W^t + S_{HW} \Delta W] + S_{HH} A^{HH*} \Delta H* - S_{HM} A^{HM*} \Delta M^* + B^H \} \quad (7a)$$

Where ΔH^* and ΔM^* refer to the enthalpy and mass associated with upstream properties of the links (ie the transported properties). Thus

$$A^{HH^*} = \begin{bmatrix} -W_1/M_1 & 0 & 0 & +W_4/M_4 & 0 \\ W_1/M_1 & -W_2/M_2 & 0 & & +W_5/M_4 \\ 0 & W_2/M_2 & -W_3/M_3 & 0 & 0 \\ 0 & 0 & W_3/M_3 & -W_4/M_4 & -W_5/M_4 \end{bmatrix}$$

$$\Delta H^* = \begin{bmatrix} \Delta H_1 \\ \Delta H_2 \\ \Delta H_3 \\ \Delta H_4 \\ \Delta H_4 \end{bmatrix}, \quad \Delta M^* = \begin{bmatrix} \Delta M_1 \\ \Delta M_2 \\ \Delta M_3 \\ \Delta M_4 \\ \Delta M_4 \end{bmatrix} \qquad (13 \text{ \& } 14)$$

$$A^{HW} = \begin{bmatrix} -H_1/M_1 & 0 & 0 & +H_4/M_4 & 0 \\ H_1/M_1 & -H_2/M_2 & 0 & 0 & H_4/M_4 \\ 0 & H_2/M_2 & -H_3/M_3 & 0 & 0 \\ 0 & 0 & H_3/M_3 & -H_4/M_4 & -H_4/M_4 \end{bmatrix} \qquad (15)$$

For each link, the elements of the column are formed from the link flow, W_k and the upstream properties (H and M). Each link has a sink and source node.

Similarly

$$A^{HM^*} = \begin{bmatrix} -W_1H_1/M_1^2 & 0 & 0 & W_4H_4/M_4^2 & 0 \\ W_1H_1/M_1^2 & -W_2H_2/M_2^2 & 0 & 0 & W_5H_4/M_4^2 \\ 0 & W_2H_2/M_2^2 & -W_3H_3/M_3^2 & 0 & 0 \\ 0 & 0 & W_3H_3/M_3^2 & -W_4H_4/M_4^2 & -W_5H_4/M_4^2 \end{bmatrix} \qquad (16)$$

We wish to write the matrix equations eliminating the * parameters, ie convert ΔH^* to ΔH, ΔM^* to ΔM. To do this we introduce a transfer matrix, I^{LN} so that

$$\Delta H^* = I^{LN} \Delta H \qquad (17)$$

where

$$I^{LN} = \begin{array}{c} \text{nodes} \rightarrow \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \quad \begin{array}{l} \text{links} \\ \text{|} \\ \text{v} \end{array} \qquad (18)$$

where I^{LN} is formed by entering 1 for the node that is the upstream or source node for each link.

Now, we can define:

$$A^{HH^*} \Delta H^* = A^{HH^*} I^{LN} \Delta H \qquad (19)$$

$$= A^{HH} \Delta H$$

and $$A^{HM^*} \Delta M^* = A^{HM^*} I^{LN} \Delta M \qquad (20)$$

$$= A^{HM}\ \Delta M.$$

Thus

$$\Delta H = \Delta t\ \{A^{HW}\ (W^t + S_{HW}\ \Delta W) + S_{HH}\ A^{HH}\ \Delta H - S_{HM}\ A^{HM}\ \Delta M + B^H\} \tag{21}$$

Substituting in the mass equation (6a):

$$\Delta H = \Delta t\ \{\ A^{HW}\ (W + S_{HW}\ \Delta W) + S_{HH}\ A^{HH}\ \Delta H - \Delta t\ S_{HM}\ A^{HM}\ A^{MW}\ (W^t + W^{MW}\ \Delta W) + B^H\ \} \tag{22}$$

Solving for ΔH:

$$\Delta H = \Delta t[I - \Delta t\ S_{HH}\ A^{HH}]^{-1}\ \{A^{HW}\ (W^t + S_{HW}\ \Delta W) - \Delta t\ S_{HM}\ A^{HM}\ A^{MW}(W^t + S_{MW}\ \Delta W) + B^H\} \tag{23}$$

So now we have ΔM and ΔH in terms of ΔW. Recalling (8), in matrix notation, we have:

$$\Delta P = C_1\ \Delta M + C_2\ \Delta H, \tag{8a}$$

where

$$C_1 = \begin{bmatrix} C_{11} & & & \\ & C_{12} & 0 & \\ & & C_{13} & \\ 0 & & & C_{14} \end{bmatrix} \tag{24}$$

Similarly for C_2.

We can back-substitute ΔM and ΔH into (8) and the result into the flow equation to leave a matrix equation in ΔW only, which can be solved by traditional numeric means. Hence,

$$\begin{aligned} \Delta P &= \Delta t\ C_1\ A^{MW}\ (W^t + S_{MW}\ \Delta W) + \Delta t\ C_2\ [I - \Delta t\ S_{HH}\ A^{HH}]^{-1}\ [A^{HW}\ (W^t + S_{HW}\ \Delta W) \\ &\quad - \Delta t\ S_{HM}\ A^{HM}\ A^{MW}\ (W^t + S_{MW}\ \Delta W) + B^H] \\ &= \Delta t\ A^{PW1}\ W^t + \Delta t\ A^{PW2}\ \Delta W + \Delta t\ B^P \end{aligned} \tag{25}$$

where : $A^{PW1} = C_1\ A^{MW} + C_2\ [I - \Delta t\ S_{HH}\ A^{HH}]^{-1}\ [A^{HW} - \Delta t\ S_{HM}\ A^{HM}\ A^{MW}]$ (26)

$$A^{PW2} = S_{MW}\ C_1\ A^{MW} + C_2\ [I - \Delta t\ S_{HH}\ A^{HH}]^{-1}\ [S_{HW}\ A^{HW} - \Delta t\ S_{HM}\ S_{MW}\ A^{HM}\ A^{MW}] \tag{27}$$

$$B^P = C_2\ [I - \Delta t\ S_{HH}\ A^{HH}]^{-1}\ B^H \tag{28}$$

Thus:

$$\Delta W = \Delta t \{ A^{WP} [P^t + \Delta t S_{WP} (A^{PW1} W^t + A^{PW2} \Delta W + B^P)] + A^{WW} [W^t + 2S_{WW} A^{WW} \Delta W] + B^W \} \quad (29)$$

Collecting terms in ΔW:

$$[I - \Delta t(2 S_{WW} A^{WW} + \Delta t S_{WP} A^{WP} A^{PW2})] \Delta W - \Delta t \{[A^{WW} + \Delta t S_{WP} A^{WP} A^{PW1}] W^t + B^W + A^{WP} [P^t + \Delta t S_{WP} B^P]\} \quad (30)$$

which is of the form

$$A \Delta W = B$$

which can be solved by conventional means to yield ΔW. Then we can directly calculate ΔM, ΔH and ΔP using equations 6a, 7a (or 21), and 8a. Associated changes in temperature can be obtained as for pressure, using the appropriate equation of state coefficients.

Special cases

To summarize, the general solution is given by the following equations:

$$A^{PW1} = C_1 A^{MW} + C_2 [I - \Delta t S_{HH} A^{HH}]^{-1} [A^{HW} - \Delta t S_{HM} A^{HM} A^{MW}] \quad (26)$$

$$A^{PW2} = S_{MW} C_1 A^{MW} + C_2 [I - \Delta t S_{HH} A^{HH}]^{-1} [S_{HW} A^{HW} - \Delta t S_{HM} S_{MW} A^{HM} A^{MW}] \quad (27)$$

$$B^P = C_2 [I - \Delta t S_{HH} A^{HH}]^{-1} B^H \quad (28)$$

$$[I - \Delta t(2 S_{WW} A^{WW} + \Delta t S_{WP} A^{WP} A^{PW2})] \Delta W = \Delta t \{ [A^{WW} + \Delta t S_{WP} A^{WP} A^{PW1}] W^t + B^W + A^{WP} [P^t + \Delta t S_{WP} B^P] \} \quad (30)$$

$$\Delta M = \Delta t A^{MW} [W^t + S_{MW} \Delta W] \quad (6a)$$

$$\Delta H = \Delta t \{ A^{HW} (W^t + S_{HW} \Delta W) + S_{HH} A^{HH} \Delta H - S_{HM} A^{HM} \Delta M + B^H \} \quad (21)$$

$$\Delta P = C_1 \Delta M + C_2 \Delta H \quad (8a)$$

Special cases of this general algorithm are as follows:

Fully explicit: all S's = 0

$$A^{PW1} = C_1 A^{MW} + C_2 A^{HW} \tag{26}$$

$$A^{PW2} = 0 \tag{27}$$

$$B^P = C_2 B^H \tag{28}$$

$$\therefore \Delta W = \Delta t \{ A^{WW} W^t + B^W + A^{WP} P^t \} \tag{30}$$

$$\Delta M = \Delta t A^{MW} W^t \tag{6a}$$

$$\Delta H = \Delta t \{ A^{HW} W^t + B^H \} \tag{7a}$$

$$\Delta P = C_1 \Delta M + C_2 \Delta H, \tag{8a}$$

as expected.

Porsching's semi-implicit ($S_{HH} = 0$ and $S_{HM} = 0$, all other S's = 1)

$$A^{PW1} = C_1 A^{MW} + C_2 A^{HW} \tag{26}$$

$$A^{PW2} = C_1 A^{MW} + C_2 A^{HW} \tag{27}$$

$$B^P = C_2 B^H \tag{28}$$

$$[I - \Delta t(2 A^{WW} + \Delta t A^{WP} A^{PW2})] \Delta W$$

$$= \Delta t \{ [A^{WW} + \Delta t A^{WP} A^{PW1}] W^t + B^W + A^{WP} [P^t + \Delta t B^P] \} \tag{30}$$

$$\Delta M = \Delta t A^{MW} [W^t + \Delta W] \tag{6a}$$

$$\Delta H = \Delta t \{ A^{HW} (W^t + \Delta W) + B^H \} \tag{21}$$

$$\Delta P = C_1 \Delta M + C_2 \Delta H \tag{8a}$$

Fully Implicit: All S's = 1

$$A^{PW1} = C_1 A^{MW} + C_2 [I - \Delta t A^{HH}]^{-1} [A^{HW} - \Delta t A^{HM} A^{MW}] \tag{26}$$

$$A^{PW2} = C_1 A^{MW} + C_2 [I - \Delta t A^{HH}]^{-1} [A^{HW} - \Delta t A^{HM} A^{MW}] \tag{27}$$

$$B^P = C_2 [I - \Delta t A^{HH}]^{-1} B^H \tag{28}$$

$$[I - \Delta t(2 A^{WW} + \Delta t A^{WP} A^{PW2})] \Delta W$$

$$= \Delta t \{ [A^{WW} + \Delta t A^{WP} A^{PW1}] W^t + B^W + A^{WP} [P^t + \Delta t B^P] \} \tag{30}$$

$$\Delta M = \Delta t \, A^{MW} \, [W^t + \Delta W] \tag{6a}$$

$$\Delta H = \Delta t \, \{ \, A^{HW} \, (W^t + \Delta W) + A^{HH} \, \Delta H - A^{HM} \Delta M + B^H \, \} \tag{21}$$

$$\Delta P = C_1 \, \Delta M + C_2 \, \Delta H \tag{8a}$$

PROGRAMMING NOTES

It should be noted that the full system geometry is contained in A^{MW}. All other matrices are derived from this matrix and node/link properties. Programming is thus very straightforward. In addition, the switches, S, can be varied at will to control the degree of implications of the system variables, W, M, H and P.

The fully-implicit method is more complicated than the semi-implicit method in that it requires the addition and multiplication of more matrices as well as a matrix inversion. The effect of these additional operations is quite costly, especially when a large number of nodes is needed. In one case study (Hoskins 1989), for 9 nodes and links, the cost is a 50% increase in iteration time. But this becomes a 250% increase as one approaches the 36 node/link case. By handling the matrix operations as efficiently as possible, some increase in speed should be attainable for both models. Using efficient assembly routines (rather than FORTRAN) for the matrix operations yielded a 10 to 20% reduction (increasing from 9 nodes to 36 nodes) in the time per iteration for the semi-implicit method and a 15 to 25% reduction in the fully-implicit case.

Usually the matrices contain mostly zeros and, in the case of a circular loop, may be diagonally dominant in nature (i.e. non-zero elements occupy one, two or three stripes through the matrix). By writing routines specific to the nodal layout for handling the matrix operations, significant gains in speed may be possible. However, the simulator will no longer be general in nature and the routines may have to be changed if the nodal layout is altered.

If the multiplication of two large matrices is desired, say NxN in dimension, the time to carry out the operation (N^3 multiplications and N^3 additions) can be very significant. However, it is possible to reduce the number of individual operations without losing the generality of the method. Take, for example, the multiplication of A^{WP} and A^{PW}. The rows in the former term pertain to links and the columns to nodes. Each row will only contain two terms located in the columns corresponding to the upstream and downstream nodes of that particular link. Thus, knowing which are the upstream and downstream nodes for every link, it is only necessary to do two multiplications and one addition to obtain each element of the product matrix ($2N^2$ multiplications and N^2 additions). By taking

advantage of having only two elements in each row of the former term or only two elements in each column of the latter term wherever possible, significant savings in time may be observed. With this improvement in the code, a cut in time by a factor of two for 18 nodes and by a factor of three for 36 nodes, regardless of the method (semi- or fully-implicit) was obtained. The cost of the fully-implicit method is reduced slightly to a 32% increase in iteration time over the semi-implicit method when 9 nodes and 9 links are used. This becomes a 214% increase as one approaches the 36 node case.

Since the focus of this paper is to provide a less obtuse and more general derivation of thermalhydraulic system equations than Porsching's method, a full comparison of the performance of the fully- and semi-implicit methods will not be made. Suffice it to say that, in general, the semi-implicit method has a Courant limit on the maximum time step that can be taken in order to ensure stability. The fully-implicit method does not have this limitation. As the Courant time step limit is determined by the nodal residence time, the time step limit is dependant on the node sizes and the flows through the nodes. Practical simulations have a further time step constraints such as: the tracking of movement of valves, the maintenance of accuracy, synchronizing of report times, etc. Thus, the choice between the semi- or fully-implicit method depends on the time per iteration multiplied by the number of iterations required to reach the largest time step permitted by the simulation problem. For example, for a 9 node case, the semi-implicit method required 0.10 seconds per iteration and required 2 iterations to meet the report time of 1.0 seconds. The fully-implicit method meet the report time in one iteration which took 0.14 seconds. At 36 nodes however, the semi-implicit method took 2 x 0.71 seconds while the fully-implicit method took 2.12 seconds. Clearly, one method is not superior to the other in all cases.

Pressure determination involves the use of property derivatives. To avoid the numerical problems associated with discontinuities, smooth functions for properties must be used, such as those derived by [Garland 1988 and 1989]. FORTRAN Source code for the water properties as described in these works is available on a MS-DOS diskette from the author. These functions and routines permit the quick and fast evaluation of ΔP and ΔT given ΔM and ΔH for all water phases. Automatic adjustment is provided to prevent P and T drift from values consistent with current M and H values. These routines are non-iterative, essential for real-time simulation.

FORTRAN source code illustrating the FIBS algorithm is available on MS-DOS diskette from the author.

CONCLUSION

The FIBS approach for thermalhydraulic system simulation has been compared to the classic work of Porsching. Porsching's algorithm is derived as a subset of the fully implicit approach. Focusing on the system Jacobian, as Porsching did, focuses on the perturbation of the system as a whole. Although general, it tends to obscure the interaction of the main players in typical thermalhydraulic systems: flow and pressure. The FIBS form is shown to be more general than Porsching's method, yet less obtuse. The interplay of flow and pressure is clarified and coding is simplified.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the financial support of the Natural Sciences and Engineering Research Council of Canada and the assistance of R.J.Wilson, EACS.

REFERENCES

AG82 L.J. Agee, "RETRAN Thermalhydraulic Analyses: Theory and Applications", Progress in Nuclear Energy, Vol. 10, No. 1, pp. 19-67 (1982).

CH77 C.Y.F. Chang and J. Skears, "SOPHT - A Computer Model for CANDU-PHWR Heat Transport Networks and Their Control", Nucl. Technol. 35, 591 (1977).

GA87 W.J. Garland and R. Sollychin, "The Rate Form of State for Thermalhydraulic Systems: Numerical Considerations", Engineering Computations, Vol. 4, December 1987.

GA88 W.J. Garland and J.D. Hoskins, "Approximate Functions for the Fast Calculation of Light Water Properties at Saturation", Int'l J. of Multiphase Flow, Vol. 14, No. 3, pp 333-348 (1988).

GA89 W.J. Garland and B.J. Hand, "Simple Functions for the Fast Approximation of Light Water Thermodynamic Properties", Nuclear Engineering & Design, Vol. 113, pp. 21-34, (1989).

HO89 J. D. Hoskins, "A Heat Transport System Operator Companion for a CANDU Nuclear Reactor", Masters Thesis, McMaster University, Hamilton, Ontario, Canada, July, 1989.

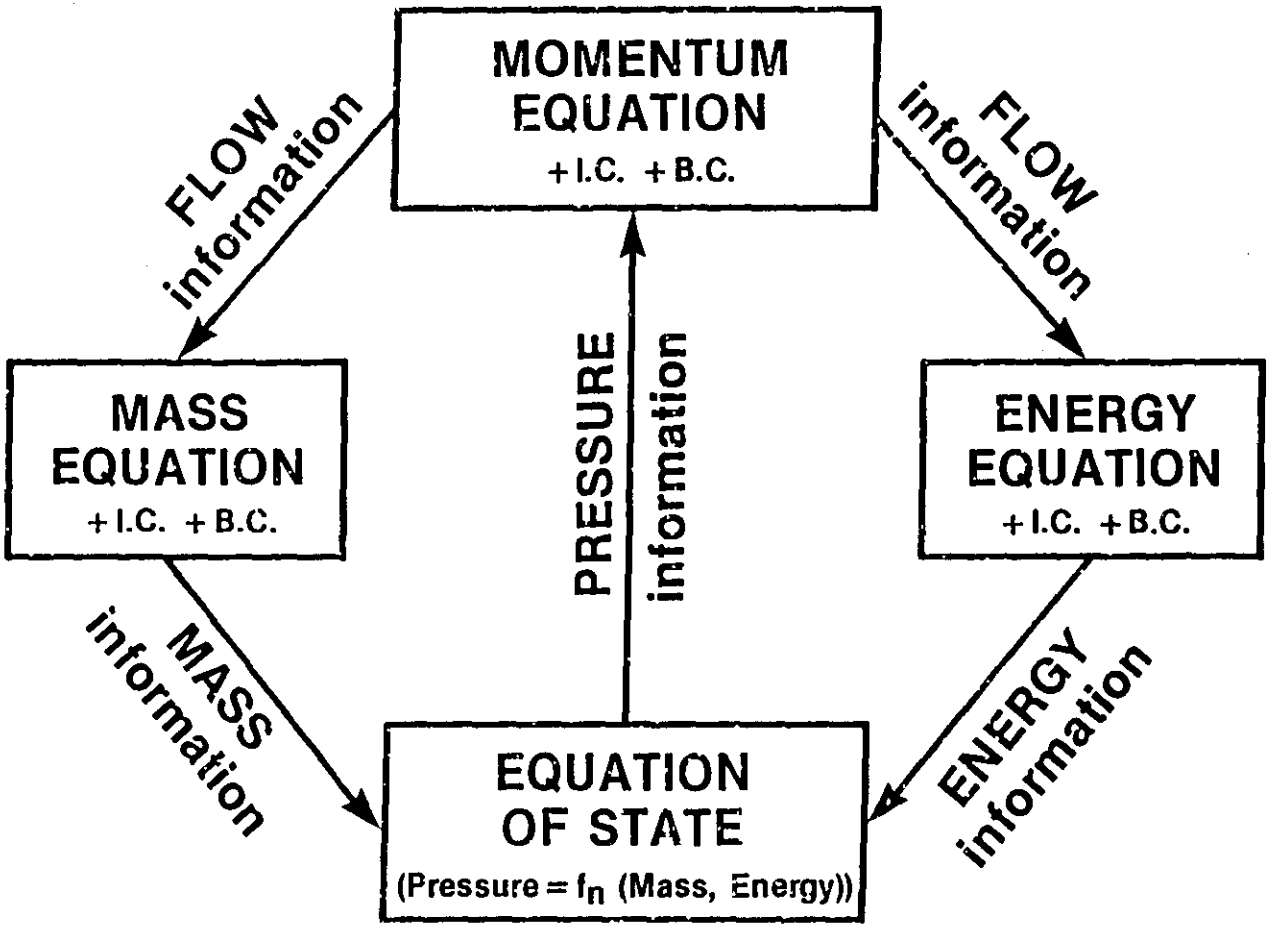
PO69 T.A. Porsching, J.H. Murphy, J.A. Redfield, and V.C. Davis, "FLASH-4: A Fully Implicit FORTRAN IV Program for the Digital Simulation of Transients in a Reactor Plant", WAPD-TM-840, Mar. 1969.

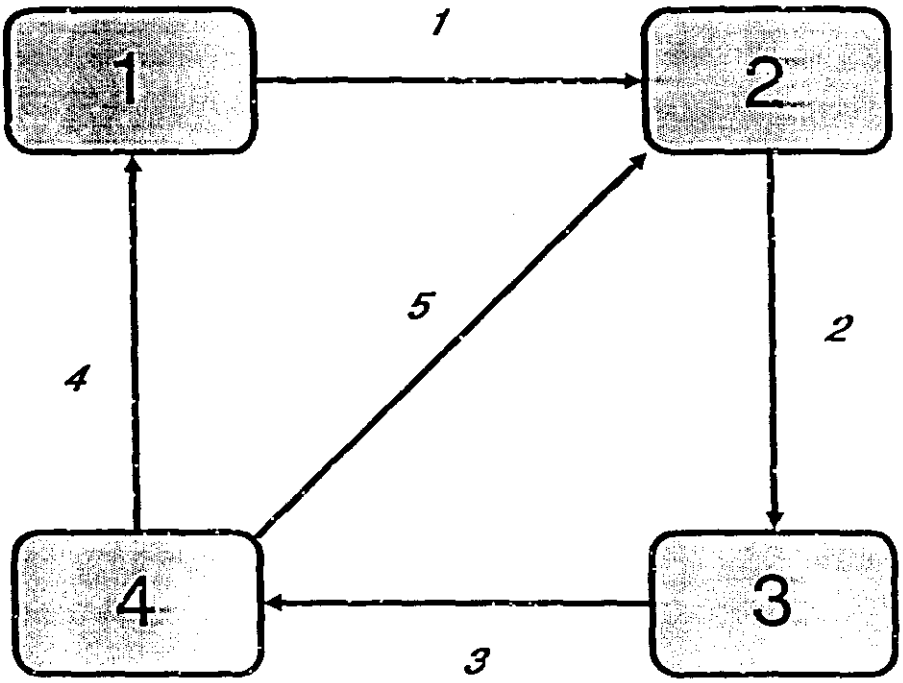
PO71 T.A. Porsching, J.H. Murphy, J.A. Redfield, "Stable Numerical Integration of Conservation Equations for Hydraulic Networks", Nuc. Sci. and Eng. 43, p 218-225 (1971).

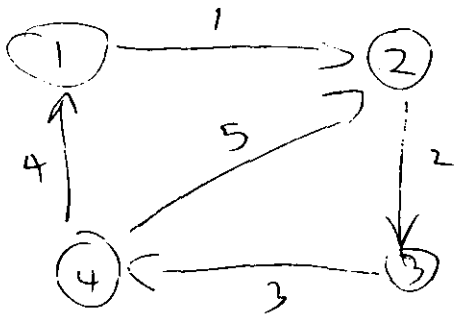
Figure Captions:

Figure 1: The four cornerstone equations for thermalhydraulic system simulation and the flow of information between them.

Figure 2: The simple 4 node - 5 link example.


MOMENTUM EQUATION
+I.C. +B.C.
FLOW information
FLOW information
MASS EQUATION
+I.C. +B.C.
ENERGY EQUATION
+I.C. +B.C.
PRESSURE information
MASS information
ENERGY information
EQUATION OF STATE
(Pressure = f_n (Mass, Energy))


1
2
3
4
1
2
3
4
5



$$\begin{bmatrix} \Delta m_1 \\ \Delta m_2 \\ \Delta m_3 \\ \Delta m_4 \end{bmatrix} = \Delta t \begin{bmatrix} -1 & & & & \\ -1 & & & +1 & \\ +1 & -1 & & & +1 \\ & +1 & -1 & & \\ & & +1 & -1 & -1 \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \end{bmatrix}$$

node

number

$$\dot{y} = F(t, y)$$

$$= F(t_0, y_0) + \Delta t \frac{\partial F}{\partial t} + \Delta y \frac{\partial F}{\partial y}$$

$$= F(t_0, y_0) + \Delta t \, \underline{\underline{J}} \, \dot{y}$$

or

$$\frac{\Delta y}{\Delta t} = F^{t} + J \Delta y$$