THSIM User Manual

prepared by: **Wm. J. Garland, Professor**
Department of Engineering Physics
McMaster University
Hamilton, Ontario
Canada
February 1998

for the Thailand Initiative

TABLE OF CONTENTS:

LISTS:

Chapter 1 Introduction

Chapter 2 Input Format

2.1 Introduction

The input file to THSIM consists of a series of data records (each containing data fields like a database record) grouped in blocks as indicated by the following headers:

*START
*NODE
*LINK
*PUMP
*HTC
*END

The blocks can be in any order but the data records within each block follow a specific format, as given in the following sections. Within each block, there is no specified format for the input fields other than integer, float or string. The sequence of records within each block must be maintained.

2.2 *START Block

2.2.1 Input Format

The *START block contains input data for overall program control as per the following format:

*START block					
Rec	**Fld**	**Format**	**Variable**	**Description**	**Units**
1	1	float	t_start	simulation start time	seconds
	2	float	t_fin	simulation finish time	seconds
	3	float	dt_min	minimum time step	seconds
	4	float	dt_max	maximum time step	seconds
	5	float	dt_print	print time step	seconds
2	1	integer	max_iter	Maximum number of iterations for the pressure calculation in a node.	None
3	1	float	P_tol	Error tolerance for pressure	fraction
	2	float	P_scale	100% range scale for pressure error used in tolerance calculation	MPa
4	1	float	T_tol	Error tolerance for temperature	fraction
	2	float	T_scale	100% range scale for temperature error used in tolerance calculation	°C
5	1	float	h_tol	Error tolerance for specific enthalpy	fraction
	2	float	h_scale	100% range scale for density error used in tolerance calculation	kJ/kg
6	1	float	rho_tol	Error tolerance for specific enthalpy	fraction
	2	float	rho_scale	100% range scale for density error used in tolerance calculation	kg/m^3

2.2.2 Example

```
*START
0.0 1.000102 .00001 .01 .1 //t_start, t_fin, dt_min, dt_max, dt_print
25          //Maximum # of iterations
```

```
.001 10.0   //Tolerance [fraction] and scale for pressure [MPa]
.001 300.0      //Tolerance [fraction] and scale for temperature [deg C]
.001 1400.0 //Tolerance [fraction] and scale for specific enthalpy [kJ/kg]
.001 1000.0 //Tolerance [fraction] and scale for density [MPa]
1.0                     //Adjustable parameter
```

2.3 *NODE Block

2.3.1 Input Format

The *NODE block contains input data for the nodes as per the following format:

*NODE block					
Rec	**Fld**	**Format**	**Variable Name**	**Description**	**Units**
1	1	integer	num_nodes	Number of nodes = number of following records (i = 1 to num_nodes)	None
i	1	string	node[i].name	Node name (maximum of 8 characters)	None
	2	string	node[i].type	Type of node: 'STD' for standard node, 'OTANK' for open tank (surface pressure held at input value and level calculation performed), 'CTANK' for closed tank (level calculation performed)	None
	3	float	node[i].V	Node volume (liquid volume if 'OTANK')	m^3
	4	float	node[i].P	Node pressure (mid level pressure if a tank)	MPa
	5	float	node[i].T	Node temperature	°C
	6	float	node[i].rho	Node density	kg/m^3
	7	float	node[i].h	Node specific enthalpy	kJ/kg
	8	float	node[i].x	Node quality	fraction
	9	string	node[i].indvar	Specification of independent variables for the node: 'P&T', 'rho&h', 'P&h', 'P&rho	None
	10	float	node[i]. V_tank	OTANK volume if applicable, 0.0 otherwise	m
	11	float	node[i].area	Tank cross sectional area if applicable, 0.0 otherwise	m^2
	12	float	node[j]. P_surface	OTANK surface pressure boundary condition if applicable, 0.0 otherwise	MPa
	12	integer	node[i].htcid	ID number for the input record for the heat transfer coefficient correlation, 0 otherwise	None

	13	float	node[i].heat	Net fixed heat generation rate / surface heat transfer rate, 0.0 otherwise	Watts / s

2.3.2 Example

```
*NODE
3
Node1  STD  1.0  10.0  300.0  25.0  2900.0  0.0  rho&h  0.0  0.0  0.0  0.0
Node2  STD  1.0  10.0  300.0  25.0  2900.0  0.0  rho&h  0.0  0.0  0.0  0.0
Node3  STD  1.0  10.0  300.0  25.0  2900.0  0.0  rho&h  0.0  0.0  0.0  0.0
```

2.4 *LINK Block

2.4.1 Input Format

The *LINK block contains input data for the links as per the following format:

*LINK block					
Rec	**Fld**	**Format**	**Variable Name**	**Description**	**Units**
1	1	integer	num_links	Number of links = number of following records (j = 1 to num_links)	None
j	1	string	link[j].name	Link name (maximum of 8 characters)	None
	2	string	link[j].type	Type of link: 'STD' for standard link	None
	3	string	link[j].up_node	Name of upstream node	None
	4	string	link[j]. down_node	Name of downstream node	None
	5	float	link[j].A	Link cross sectional area	m^2
	6	float	link[j].D	Link hydraulic diameter	m
	7	float	link[j].L	Link length	m
	8	float	link[j].k	Link resistance coefficient	
	9	float	link[j].del_elev	Link elevation change (from inlet to outlet)	m
	10	float	link[j].W	Link flow	kg/s
	11	float	link[j].up_level	Upstream Tank connection level if applicable, 0.0 otherwise	m

	12	float	link[j]. down_level	Downstream Tank connection level if applicable, 0.0 otherwise	m
	13	integer	link[j].pumpid	ID number for the input record for the pump specification if applicable, 0 otherwise	None
	14	integer	link[j].valveid	ID number for the input record for the valve specification if applicable, 0 otherwise	None

2.4.2 Example

```
*LINK
2
Link1  STD  Node1 Node2  0.1 0.01  10.   0.5 0.0  10.0  0.0  0.0  0  0
Link2  STD  Node2 Node3  0.1 0.01  10.   0.5 0.0  10.0  0.0  0.0  0  0
```

2.5 *PUMP Block

2.5.1 Input Format

The *PUMP contains input data for the pumps referenced in the link input (field 13) as per the following format:

*PUMP block					
Rec	**Fld**	**Format**	**Variable Name**	**Description**	**Units**
1	i	integer	num_pumps	Number of pumps in the following records (k = 1 to num_pumps)	None
k	1	integer	pump[k].id	Sequential ID number for the pump = record number. Should correspond to the ID number used in field 13 of the link containing the pump.	None
	2	string	pump[k].type	Type of pump: 'STD' for standard link	None
	3	float	pump[k].h_ref	head at reference flow	metres
	4	float	pump[k].Q_ref	reference volumetric flow	m^3/s
	5	float	pump[k].a	head / h_ref = Q / Q_ref (1-a Q)^b	s/m^3
	6	float	pump[k].b		None

2.5.2 Example

```
*PUMP
1
1  STD  200.0  100.0  0.0  0.0
```

2.6 *HTC Block

2.6.1 Input Format

The *HTC contains input data for the heat transfer coefficient referenced in the node input (field 12) as per the following format:

*HTC block					
Rec	**Fld**	**Format**	**Variable Name**	**Description**	**Units**
1	1	integer	num_htcs	Number of heat transfer coefficients in the following records (m = 1 to num_htcs)	None
m	1	integer	htc[m].id	Sequential ID number for the htc. Should correspond to the ID number used in field 13 of the node containing the htc.	None
	2	string	htc[m].type	Type of htc: 'FIXED' for a constant htc 'DB' for standard Dittus Boelter correlation	None
	3	float	htc[m].h	Fixed value of h if 'FIXED' type is specified or 0.0 otherwise	Watts/ m^2 °K
	4	float	htc[m].	Unused	
	5	float	htc[m].	unused	
	6	float	htc[m].	unused	

2.6.2 Example

```
*HTC
1
1  FIXED  200.0
```

2.7 *END Block

2.7.1 Input format

This block header specifies the end of the input file. The user is free to place notes after this line if so desired. There are no input records associated with this block.

2.7.2 Example

*END

2.8 Sample Input Files

2.8.1 Test1.inp

```
THSIM.C input file:  To calculate properties of a node

*START
0.0  -1.0 0.00001 1.0 10.0 //t_start, t_fin, dt_min, dt_max, dt_print
25           //Maximum # of iterations
.001 10.0    //Tolerance [fraction] and scale for pressure [MPa]
.001 300.0   //Tolerance [fraction] and scale for temperature [deg C]
.001 1400.0  //Tolerance [fraction] and scale for specific enthalpy [kJ/kg]
.001 1000.0  //Tolerance [fraction] and scale for density [MPa]
.001 100.0   //Tolerance [fraction] and scale for flow [kg/s]
1.0                      //Adjustable parameter

*NODE
3
Node1 STD  1.0  10.0  300.0  770.0  1200.0   0.0  rho&h  0.0  0.0  0.0 0  0.0
Node2 STD  1.0  10.0  300.0  700.0  1400.0   0.0  P&T    0.0  0.0  0.0 0  0.0
Node3 STD  1.0  10.0  300.0  770.0  1500.0   0.0  P&h    0.0  0.0  0.0 0  0.0


*END
```

2.8.2 Test2.inp

```
THSIM.C input file:  To simulate a simple node-link case

*START
0.0 10.0 0.00001 1.0 0.1 //t_start, t_fin, dt_min, dt_max, dt_print
25           //Maximum # of iterations
.001 10.0    //Tolerance [fraction] and scale for pressure [MPa]
.001 300.0   //Tolerance [fraction] and scale for temperature [deg C]
.001 1400.0  //Tolerance [fraction] and scale for specific enthalpy [kJ/kg]
.001 1000.0  //Tolerance [fraction] and scale for density [MPa]
.001 100.0   //Tolerance [fraction] and scale for flow [kg/s]
1.0          //Adjustable parameter

*NODE
2
Node1 STD 1.0 10.0 300.0 500.0 1451.0  0.0  P&h   0.0   0.0   0.0  0   0.0
Node2 STD 1.0  5.0 300.0 100.0 1500.0  0.0  P&h   0.0   0.0   0.0  0   0.0

*LINK
1
Link1   STD  Node1 Node2  0.01 0.1   1. 1.5 0.0   0.0   0.0   0.0   0    0


*END
```

2.8.3 Test3.inp

```
THSIM.C input file:  To simulate a simple node-link case

*START
0.0 10.0 .00001 1. 0.1 //t_start, t_fin, dt_min, dt_max, dt_print
25           //Maximum # of iterations
.001 10.0    //Tolerance [fraction] and scale for pressure [MPa]
.001 300.0   //Tolerance [fraction] and scale for temperature [deg C]
.001 1400.0  //Tolerance [fraction] and scale for specific enthalpy [kJ/kg]
.001 1000.0  //Tolerance [fraction] and scale for density [MPa]
.001 100.0   //Tolerance [fraction] and scale for flow [kg/s]
1.0                      //Adjustable parameter

*NODE
3
Node1 STD 1.0 10.0 300.0 500.0 1451.0  0.0  P&h   0.0   0.0   0.0 0   0.0
Node2 STD 1.0  5.0 300.0 100.0 1500.0  0.0  P&h   0.0   0.0   0.0 0   0.0
Node3 STD 1.0  3.0 300.0 100.0 1500.0  0.0  P&h   0.0   0.0   0.0 0   0.0

*LINK
2
Link1   STD  Node1 Node2  0.01 0.1   1. 1.5 0.0   0.0   0.0   0.0   0    0
Link2   STD  Node2 Node3  0.01 0.1   1. 1.5 0.0   0.0   0.0   0.0   0    0


*END
```

2.8.4 Test4.inp

```
THSIM.C input file:  To simulate a simple 2 interconnected closed tank case

*START
0.0  100.0001 0.00001 1.0 10.0 //t_start, t_fin, dt_min, dt_max, dt_print
25           //Maximum # of iterations
.001 10.0    //Tolerance [fraction] and scale for pressure [MPa]
.001 300.0   //Tolerance [fraction] and scale for temperature [deg C]
.001 1400.0  //Tolerance [fraction] and scale for specific enthalpy [kJ/kg]
.001 1000.0  //Tolerance [fraction] and scale for density [kg/m^3]
.001 100.0   //Tolerance [fraction] and scale for flow [kg/s]
1.0                      //Adjustable parameter

*NODE
2
Node1 OTANK 1.0  0.2  95.0 800.0  200.0  0.0  P&T   2.0   0.1  0.10 0   0.0
Node2 OTANK 0.1  0.2  95.0 800.0  200.0  0.0  P&T   2.0   0.1  0.10 0   0.0

*LINK
1
Link1   STD  Node1 Node2  0.01 0.1 10.0 1.5 0.0   0.0   0.0   0.0   0    0


*END
```

APPENDIX 5

Extracted from "The Rate Form of the Equation of State for Two-Phase Systems" N. Garner Unpublished.

the expressions for the F and G functions can be summarized as in table 1. These expressions cover the full range from subcooled liquid to superheated steam.

Having derived the desired rate forms for the equation of state, we proceed to illustrate the utility of the approach, as indicated in the Introduction.

ANALYTICAL INVESTIGATIONS

The analytical investigation is primarily concerned with incorporating the equation of state within the describing matrix equation for the system to allow the extraction of the eigenvalues which characterize the system. First, we look at the distributed form of the governing equations.

THE DISTRIBUTED FORM

From Equation 10:

$$\frac{\partial U}{\partial t} + A \frac{\partial U}{\partial z} = B, \tag{10}$$

we define an eigenvector, X_i, which satisfies

$$(A - \lambda_i I) X_i = 0, \quad i = 1, \ldots, N \tag{63}$$

where N = rank of A.

Further, we define the modal matrix, M:

$$M = \left\{ X_1, X_2, \ldots \right\}. \tag{64}$$

Thus

$$A M = M \cdot \Lambda \quad \text{where } \Lambda = \begin{bmatrix} Y_1 & & \\ & Y_2 & 0 \\ 0 & & \ddots \end{bmatrix}. \tag{65}$$

Hence

$$M^{-1} A M = \Lambda = \text{Diagonal} \tag{66}$$

Multiplying Equation 10 thru by M^{-1} we get:

$$\mathbf{M}^{-1}\frac{\partial \mathbf{u}}{\partial t} + \mathbf{M}^{-1}\mathbf{A}\frac{\partial \mathbf{u}}{\partial z} = \mathbf{M}^{-1}\mathbf{B}. \tag{67}$$

Since $\mathbf{M}^{-1}\mathbf{A} = \Lambda\,\mathbf{M}^{-1}$ (from Equation 66) and if we define:

$$\Phi = \mathbf{M}^{-1}\,\mathbf{U} \tag{68}$$

$$\mathbf{C} = \mathbf{M}^{-1}\,\mathbf{B},$$

we get the canonical form:

$$\frac{\partial \Phi}{\partial t} + \Lambda\frac{\partial \Phi}{\partial z} = \mathbf{C}. \tag{69}$$

This is a set of N <u>independent</u> equations with solution

$$\Phi_i = \Phi_i(0)\,e^{D_i(t - z/\gamma_i)}, \tag{70}$$

where $\Phi_i(0)$ and D_i are determined form the initial and boundary conditions. Thus, the eigenvalues of Equation 69 are the velocity characteristics of the original equation set (Equation 10). If the set of equations chosen is the basic set of mass, momentum and energy equations (Equations 7-9), we find that

$$\mathbf{A} = \begin{bmatrix} v & \rho & 0 \\ 0 & v & 0 \\ 0 & 0 & v \end{bmatrix}, \tag{71}$$

which give the eigenvalues:

$$\lambda_i = \{v, v, v\}. \tag{72}$$

Thus, the basic continuity equations capture the bulk movement of fluid only.

For compressible flow, the classical way to capture more information is to apply the equation of state,

$$P = \pi(\rho, s), \text{ where } s = \text{entropy},$$

to the energy continuity equation assuming isentropic flow, $ds = (dQ/T) = 0$, to give:

$$\frac{\partial P}{\partial t} + \rho c^2\frac{\partial v}{\partial z} + v\frac{\partial P}{\partial z} = 0, \tag{73}$$

where c = speed of sound $= \sqrt{(\partial P/\partial \rho)_s}$. This replaces the energy equation to give

$$\begin{bmatrix} \rho \\ v \\ P \end{bmatrix} + \begin{bmatrix} \rho & v & 0 \\ v & 0 & 1/\rho \\ \rho c^2 & 0 & v \end{bmatrix} \begin{bmatrix} \frac{\partial \rho}{\partial z} \\ \frac{\partial v}{\partial z} \\ \frac{\partial P}{\partial z} \end{bmatrix} = \begin{bmatrix} 0 \\ F/\rho \\ 0 \end{bmatrix}. \tag{74}$$

The eigenvalues are now:

$$\lambda_i = \{v, v + c, v - c\} \tag{75}$$

This form captures the bulk velocity and the acoustic wave propogation. However, we had to assume isentropic flow and the derivation is quite obtuse. A more direct method (ref: Hinds 1981) is to recast the $\partial P/\partial z$ term of the momentum equation. Since:

$$P = \pi(e, \rho) \tag{76}$$

and

$$dP = \frac{\partial P}{\partial e} de + \frac{\partial P}{\partial \rho} d\rho \tag{77}$$

then

$$\frac{\partial P}{\partial z} = \frac{\partial P}{\partial e}\Big)_{\rho} \frac{\partial \rho}{\partial z} + \frac{\partial P}{\partial \rho}\Big)_{e} \frac{\partial \rho}{\partial z}. \tag{78}$$

Substituting into the momentum equation, the A matrix becomes

$$A = \begin{bmatrix} v & \rho & 0 \\ \frac{1}{\rho}\frac{\partial P}{\partial \rho} & v & \frac{1}{\rho}\frac{\partial P}{\partial \rho} \\ 0 & 0 & v \end{bmatrix}, \quad U = \begin{bmatrix} \rho \\ v \\ e \end{bmatrix} \tag{79}$$

whose eigenvalues are:

$$\lambda_i = \left\{ v, v + \sqrt{\left(\frac{\partial P}{\partial \rho}\right)_e}, \; v - \sqrt{\left(\frac{\partial P}{\partial \rho}\right)_e} \right\}. \tag{80}$$

Again, this captures the bulk and acoustic velocities. This method is more straightforward and intuitive and correctly shows that the primary influence of the pressure on the system is via the momentum equation.

It should be no surprise, then, that we arrive at the same conclusion by considering the rate form as previously developed:

$$\frac{\partial P}{dt} = G_1 \frac{\partial \rho}{\partial t} + G_2 \frac{\partial e}{\partial t} \tag{81}$$

2.8 Sample Input Files

2.8.1 Test1.inp

```
THSIM.C input file:  To calculate properties of a node

*START
0.0  -1.0 0.00001 1.0 10.0 //t_start, t_fin, dt_min, dt_max, dt_print
25           //Maximum # of iterations
.001 10.0    //Tolerance [fraction] and scale for pressure [MPa]
.001 300.0   //Tolerance [fraction] and scale for temperature [deg C]
.001 1400.0  //Tolerance [fraction] and scale for specific enthalpy [kJ/kg]
.001 1000.0  //Tolerance [fraction] and scale for density [MPa]
.001 100.0   //Tolerance [fraction] and scale for flow [kg/s]
1.0                      //Adjustable parameter

*NODE
3
Node1 STD  1.0  10.0  300.0  770.0  1200.0   0.0  rho&h  0.0  0.0  0.0 0  0.0
Node2 STD  1.0  10.0  300.0  700.0  1400.0   0.0  P&T    0.0  0.0  0.0 0  0.0
Node3 STD  1.0  10.0  300.0  770.0  1500.0   0.0  P&h    0.0  0.0  0.0 0  0.0


*END
```

2.8.2 Test2.inp

```
THSIM.C input file:  To simulate a simple node-link case

*START
0.0 10.0 0.00001 1.0 0.1 //t_start, t_fin, dt_min, dt_max, dt_print
25           //Maximum # of iterations
.001 10.0    //Tolerance [fraction] and scale for pressure [MPa]
.001 300.0   //Tolerance [fraction] and scale for temperature [deg C]
.001 1400.0  //Tolerance [fraction] and scale for specific enthalpy [kJ/kg]
.001 1000.0  //Tolerance [fraction] and scale for density [MPa]
.001 100.0   //Tolerance [fraction] and scale for flow [kg/s]
1.0          //Adjustable parameter

*NODE
2
Node1 STD 1.0 10.0 300.0 500.0 1451.0  0.0  P&h   0.0   0.0   0.0  0   0.0
Node2 STD 1.0  5.0 300.0 100.0 1500.0  0.0  P&h   0.0   0.0   0.0  0   0.0

*LINK
1
Link1   STD  Node1 Node2  0.01 0.1   1. 1.5 0.0   0.0   0.0   0.0   0    0


*END
```

[Note: the choice of e vs h is arbitrary, as long as consistency is maintained and the G functions appropriately redefined].

The full equation set is now:

$$\begin{bmatrix} \rho \\ v \\ e \\ P \end{bmatrix} = \begin{bmatrix} v & \rho & 0 & 0 \\ 0 & v & 0 & \frac{1}{\rho} \\ 0 & 0 & v & 0 \\ G_1 v & G_1 \rho & G_2 v & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \rho}{\partial z} \\ \frac{\partial v}{\partial z} \\ \frac{\partial e}{\partial z} \\ \frac{\partial P}{\partial z} \end{bmatrix} = \begin{bmatrix} 0 \\ F/\rho \\ Q/\rho \\ G_2 Q/\rho \end{bmatrix} \tag{82}$$

The eigenvalues are:

$$\lambda_i = \{ v, 0, v + \sqrt{G_1}, \; v - \sqrt{G_1} \; \} \tag{83}$$

where

$$G_1 = \left(\frac{\partial P}{\partial \rho} \right)_e = c^2$$

Again, the bulk and acoustic velocities have been picked up. The advantage of the rate equation approach is that the effect of pressure is kept separate and the form permits the explicit tracking of pressure on the same level as mass, momentum and energy. This would be of utility in transient fluid simulations where the choice of numerical time step could be limited by pressure phenomena. Having the rate immediately available is an asset, computationally. This will be discussed in detail later, but it is worth noting here because it is expedient to have the system equations case in one form that captures the system essence analytically and is the appropriate form for numerical computation. Eigenvalues can then be calculated as a matter of course in the numerical simulation and used to help the numerical simulation in a dynamic manner (selecting time steps, number of nodes, etc.) as well as an intuition generator.

We move on to the lumped form of the equations since this is of more interest to system modelling.

THE LUMPED FORM

To illustrate the rate form of the equation of state for lumped or network type systems, consider a two node, one link system as per Figure 11. The continuity equations for this simple system are:

Mass

$$\frac{\partial M_1}{\partial t} = -W, \quad \frac{\partial M_2}{\partial t} = W \tag{84}$$

Energy

$$\frac{\partial H_1}{\partial t} = -(H_1/M_1)\,W, \quad , \quad \frac{\partial H_2}{\partial t} = (H_1/M_1)\,W, \text{ for } W > 0 \tag{85}$$

$$\frac{\partial H_1}{\partial t} = -(H_2/M_2)\,W, \quad , \quad \frac{\partial H_2}{\partial t} = (H_2/M_2)\,W, \text{ for } W < 0\,.$$

Momentum

$$\frac{\partial W}{\partial t} = \frac{A}{L}\,(P_1 - P_2 - k|W|W)\,, \tag{86}$$

State

$$P_i = \pi\,(H_i, M_i, V_i)\,, \quad i = 1, 2\,. \tag{87}$$

or

$$\frac{\partial P_i}{\partial t} = \frac{F_{1i}\,\dfrac{\partial M_i}{\partial t} + F_{2i}\,\dfrac{\partial H_i}{\partial t} + F_{3i}\,\dfrac{\partial \forall_i}{\partial t}}{F_{4i}\,M_{vi} + F_{5i}\,M_{Li}}$$

$$\equiv G'_{1i}\,\frac{\partial M_i}{\partial t} + G'_{2i}\,\frac{\partial H_i}{\partial t}, \text{ assuming } \frac{\partial \forall_i}{\partial t} = 0 \tag{88}$$

In matrix form:

$$\frac{\partial U}{\partial t} = A\,U + B\,, \tag{89}$$

where

$$U = \begin{bmatrix} M_1 \\ H_1 \\ P_1 \\ W \\ M_2 \\ H_2 \\ P_2 \end{bmatrix} ,$$

The best specific form for the system matrix, A is not immediately obvious because the governing equations are nonlinear. The particular linear form chosen will generate its own set of eigenvalues. The "best" linearization, then, is loosely defined as that set which captures the major characteristics of the system in a robust and tractable manner.

Intuition and experience suggest that the flow is the major character in T/H systems. Inspection of Equations 84-88 shows that flow, W, appears explicitly in the continuity equations and implicitly in the pressure equation.

Eliminating $\partial M/\partial t$ and $\partial H/\partial t$ from the pressure equation gives:

$$\frac{\partial P_i}{\partial t} = \pm G'_{1i} W \pm \left(\frac{G'_{2i} H_1}{M_1} \right) W . \tag{90}$$

Thus the equation set is:

$$\begin{bmatrix} \dot{M}_1 \\ \dot{H}_1 \\ \dot{P}_1 \\ \dot{W} \\ \dot{M}_2 \\ \dot{H}_2 \\ \dot{P}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -H_1/M_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -(G'_{11} + \frac{G'_{21} H_1}{M_1}) & 0 & 0 & 0 \\ 0 & 0 & A/L & -K|W| & 0 & 0 & -A/L \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & H_1/M_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & (G'_{12} + \frac{G'_{22} H_1}{M_1}) & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} M_1 \\ H_1 \\ P_1 \\ W \\ M_2 \\ H_2 \\ P_2 \end{bmatrix} \tag{91}$$

The corresponding eigenvalues are:

$$\lambda_i = \Bigg\{0, 0, 0, 0, 0 \quad - \frac{KW}{2} \pm \sqrt{\left(\frac{KW}{2}\right)^2 - \frac{A}{L}\left(G_{11}' + G_{12}' + \frac{H_1 G_{21}'}{M_1} + \frac{A_1 G_{22}'}{M_1}\right)}\Bigg\} \tag{92}$$

Thus, the system will oscillate if the term under the square root sign is imaginary. The high incidence of "zero" roots and the "zero" columns suggests that some equations are superfluous. Inspections shows that the mass and enthalpy equations could be dropped. This results in a considerably reduced set:

$$\begin{bmatrix} P_1 \\ W \\ P_2 \end{bmatrix} = \begin{bmatrix} 0 & -(G_{11}' + G_{21}' \frac{H_1}{M_1}) & 0 \\ A/L & -K|W| & -A/L \\ 0 & (G_{12}' + G_{22}' \frac{H_1}{M_1}) & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ W \\ P_2 \end{bmatrix}$$

$$\equiv \begin{bmatrix} 0 & -C_1 & 0 \\ A/L & -K|W| & -A/L \\ 0 & C_2 & 0 \end{bmatrix} \begin{bmatrix} P_1 \\ W \\ P_2 \end{bmatrix} . \tag{93}$$

The eigenvalues for this set are:

$$\lambda_i = \Bigg\{0, \quad - \frac{KW}{2} \pm \sqrt{\left(\frac{K|W|}{2}\right)^2 - \frac{A}{L}(C_1 + C_2)}\Bigg\} . \tag{94}$$

Thus, the main system characteristics are picked up as before. This, of course, is the classic spring-mass case, as can be seen by substituting the pressure equations into the momentum equations to give:

$$\frac{\partial^2 W}{\partial t^2} = - \frac{A}{L}(C_1 + C_2) W - K|W| \frac{\partial W}{\partial t} . \tag{95}$$

The conclusion that the flow and pressure equations are the key determinators of system behaviour agress well with the findings of Bond graph theory (Paynter 1960). The results also indicate that the full Jacobi expansion of Porsching (1971) is perhaps not necessary since the mass and energy equations affect the dynamics only through the state equation.

Appendix 6

A COMPARISON OF THE RATE FORM OF THE EQUATION OF STATE TO THE JACOBIAN FORM

by

Wm. J. Garland

Department of Engineering Physics

McMaster University

Hamilton, Ontario

Canada L8S 4M1

ABSTRACT

Porsching's solution algorithm for the simulation of thermalhydraulic systems is compared to a new method based on the rate form of the equation of state. Both algorithms are developed and discussed. A direct comparison is made for a simple 2 node-1 link case to illustrate and numerically test the ideas presented.

It is shown that the final algorithms of the two methods are identical and that the rate method is more intuitive, easy to implement and permits eigenvalue extraction. The modelling of the nonlinear damping term was found to be important when large time steps were taken.

Presented at the 13th Symposium on Simulation of Reactor Dynamics and Plant Control, held at Chalk River Nuclear Laboratories, Chalk River, Ontario, April 27-28, 1987.

INTRODUCTION

One of the more successful algorithms for thermalhydraulic simulation is based on the work of Porsching (PO 69, 71). This algorithm, involving the Jacobi (or derivative of the system state matrix), is used originally in FLASH-4 (PO 69), and subsequently SOPHT (CH 77). The strength of Porsching's approach lies in the 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 converge. 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. First, as will be shown later, the matrix rate equation to be solved is not in a characteristic or eigenvalue form. Hence, it is not directly possible to extract the system eigenvalues and thus determine the stability of a state without performing a costly time solution. Secondly, the matrix rate equation involving the Jacobi 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.

Recently (GA86a, GA86b, SO85), work has been presented on the use of the rate form of the equation of state. These works 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 of the normal conservation rate equations for mass, energy and momentum plus the pressure rate equation. This forms the four cornerstone equations in thermalhydraulic systems analysis (Figure 1). It was found (GA86a) that the mass and energy equations did not contribute to the eigenvalues of the system for the simple cases studied. This agrees with the intuitive analogy of springs and masses. Further, numerical implementation prove 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 Jacobi), the question arises as to how the Rate Form compares to Porsching's method Specifically, is the pressure treatment of the two methods different? If so, what are the advantages and disadvantages of each? How do the two compare in terms of robustness, ease of implementation, clarity, stability, etc?

To investigate these questions, the following two sections are devoted to concise but explanatory derivations of Porsching's method and the Rate Form. Subsequently, a numerical test is performed on a simple system to illustrate the similarities and differences.

DERIVATION OF PORSCHING'S FORM

Following Porsching (PO71), the thermalhydraulic system equations can be written in node-link form (see Figure 2):

Momentum:

$$\dot{W}_k = f_k(t, P_i, P_j, W_k) ; \tag{1}$$

Enthalpy:

$$\dot{H}_i = \sum_{v \in T_i} \frac{H_v}{M_v} W_v - \sum_{v \in I_i} \frac{H_v}{M_v} W_v + Q_i ; \tag{2}$$

Mass:

$$\dot{M}_i = \sum_{v \in T_i} W_v - \sum_{v \in I_i} W_v ; \tag{3}$$

where

W = mass flow rate;

P = pressure;

M = mass;

H = total enthalpy;

f = some function;

Q = heat source;

t = time;

T_i = terminating node for link k;

I_i = initiating node for link k;

subscripts;

i,j = node indices;

k = link index;

v = summation index;

superscript

$\bullet = \partial/\partial t$

These equations can be written in matrix form:

$$\dot{y} = F(t,y), \tag{4}$$

where y is the column vector:

$$y = \begin{bmatrix} W_1 \\ \vdots \\ W_k \\ H_1 \\ \vdots \\ H_N \\ M_1 \\ \vdots \\ M_N \end{bmatrix} \tag{5}$$

for the case of K links and N nodes.

An implicit solution is sought for its stability advantages over explicit methods. First, we expand F via the Taylor series:

$$F(t^{n+1}, y^{n+1}) = F(t^n, y^n) + \Delta t \left[\frac{\partial F(t, y)}{\partial t} \bigg|_n + \frac{\partial F}{\partial y}(t, y) \bigg|_n \frac{\partial y}{\partial t} \right] + 0(\Delta t^2) \tag{6}$$

where the superscript, n, denotes some reference at iteration n and n + 1 denotes the time of iteration n + 1. The time difference Δt is simply $t^{n+1} - t^n$. It is assumed that F contains no explicit time dependence (i.e. terms such as at^2 or bt, where a and b are constants). Rather, the time dependency in F is through the implicit dependence of mass, energy and flow on time. That is:

$$F(t,y) \Rightarrow F(y(t)). \tag{7}$$

Thus, in equation 6,

$$\frac{\partial F}{\partial t}(t, y) \bigg|_n = 0$$

the implicit form for equation 4 is

$$\frac{y^{n+1} - y^n}{\Delta t} = F(t^{n+1}, y^{n+1}). \tag{8}$$

Substituting in the expansion of equation 6:

$$\dot{y} = F(t^n, y^n) + \Delta t\, J\, \dot{y}, \tag{9}$$

where the Jacobi is:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & \cdots \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & \cdots \\ \vdots & \vdots & \end{bmatrix} \tag{10}$$

Rearranging equation 9:

$$\dot{y} = [I - \Delta t\, J]^{-1} F(t^n, y^n) = \frac{y^{n+1} - y^n}{\Delta t} \tag{11}$$

or

$$y^{n+1} = y^n + \Delta t[I - \Delta t\, J]^{-1} F(t^n, y^n) \tag{12}$$

or

$$\Delta y \equiv y^{n+1} - y^n = \Delta t[I - \Delta t\, J]^{-1} F(t^n, y^n) \quad . \tag{13}$$

Equations 12 and 13 are the general forms for the implicit method. Note that the function, F, is general. It is only required that it be differentiable. In practice, it is required that F be smooth so that there are no discontinuities in J. Discontinuities tend to play havoc on the numerical stability of simulations. For this reason, the steam tables employed in the simulation must have continuous derivatives (see also GA 86c).

The Jacobi, J, is of size K + 2N by K + 2N. For practical simulations (50 or more nodes), this gives rather large matrices to invert. To reduce the cost of inversion, Porsching utilized the fact that flow is the major parameter and eliminated M^{n+1} and H^{n+1} from the matrix equation 12 (or 13) by backsubstituting, leaving a matrix equation implicit in W, but not in M and H. To illustrate, consider a two node, one link case (figure 3). The governing equations are:

Mass:

$$\frac{\partial M_1}{\partial t} = -W \,, \qquad \frac{\partial M_2}{\partial t} = W \,. \tag{14}$$

Energy:

$$\begin{aligned} \frac{\partial H_1}{\partial t} &= -(H_1/M_1)W \,, \qquad \frac{\partial H_2}{\partial t} = (H_1/M_1)W \,, \quad \text{for } W > 0 \,, \\ \frac{\partial H_1}{\partial t} &= -(H_2/M_2)W \,, \qquad \frac{\partial H_2}{\partial t} = (H_2/M_2)W \,, \quad \text{for } W < 0 \,. \end{aligned} \tag{15}$$

Momentum:

$$\frac{\partial W}{\partial t} = \frac{A}{L}(P_1 - P_2) - K|W|\,W \,. \tag{16}$$

State:

$$P_i = n(H_i, M_i, V_i), \quad i = 1, 2. \tag{17}$$

where V = volume.

The Jacobi for this case is:

$$J = \begin{bmatrix} -2K|W| & \frac{A}{L}\frac{\partial P_1}{\partial H_1} & -\frac{A}{L}\frac{\partial P_2}{\partial H_2} & \frac{A}{L}\frac{\partial P_1}{\partial M_1} & -\frac{A}{L}\frac{\partial P_2}{\partial M_2} \\ -H_1/M_1 & -\frac{W}{M_1} & 0 & +\frac{H_1 W}{M_1^2} & 0 \\ +H_1/M_1 & +\frac{W}{M_1} & 0 & -\frac{H_1 W}{M_1^2} & 0 \\ -1 & 0 & 0 & 0 & 0 \\ +1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{18}$$

When flow reverses, the source node is 2 rather than 1, and the Jacobi becomes:

$$J = \begin{bmatrix} -2K|W| & \frac{A}{L}\frac{\partial P_1}{\partial H_1} & -\frac{A}{L}\frac{\partial P_2}{\partial H_2} & \frac{A}{L}\frac{\partial P_1}{\partial M_1} & -\frac{A}{L}\frac{\partial P_2}{\partial M_2} \\ -H_2/M_2 & 0 & -\frac{W}{M_2} & 0 & +\frac{H_2 W}{M_2^2} \\ +H_2/M_2 & 0 & \frac{W}{M_2} & 0 & -\frac{H_2 W}{M_2^2} \\ -1 & 0 & 0 & 0 & 0 \\ +1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{19}$$

Thus, the matrix equation to be solved is:

$$[I - \Delta t\, J]\, \Delta y = \Delta t\, F(t^n, y^n), \tag{13}$$

where $[I - \Delta t\, J]$ is given by:

$$\begin{bmatrix} 1+2\Delta t K|W| & -\frac{A}{L}\frac{\partial P_1}{\partial H_1}\Delta t & \frac{A}{L}\frac{\partial P_2}{\partial H_2}\Delta t & -\frac{A}{L}\frac{\partial P_1}{\partial M_1}\Delta t & -\frac{A}{L}\frac{\partial P_2}{\partial M_2}\Delta t \\ H_1/M_1\,\Delta t & 1+\frac{W}{M_1}\Delta t & 0 & -\frac{H_1 W}{M_1^2}\Delta t & 0 \\ -H_1/M_1\,\Delta t & -\frac{W}{M_1}\Delta t & 1 & \frac{H_1 W}{M_1^2}\Delta t & 0 \\ (+1)\Delta t & 0 & 0 & 1 & 0 \\ (-1)\Delta t & 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

and F is:

$$\begin{bmatrix} \frac{A}{L}(P_1-P_2)-K|W|W \\ -H_1/M_1\,W \\ +H_1/M_1\,W \\ -W \\ +W \end{bmatrix} \quad (21)$$

for the case where $W>0$. Since the same arguments apply for the case where $W<0$, further discussion will confine itself to the case where $W>0$.

Multiply through in equation 13, using equations 20 and 21, noting that:

$$\Delta y = \begin{bmatrix} \Delta W_1 \\ \Delta H_1 \\ \Delta H_2 \\ \Delta M_1 \\ \Delta M_2 \end{bmatrix} , \quad (22)$$

we find:

$$(1+2\Delta t\, K|W|)\Delta W - \frac{A}{L}\frac{\partial P_1}{\partial H_1}\Delta t\,\Delta H_1 + \frac{A}{L}\frac{\partial P_2}{\partial H_2}\Delta t\,\Delta H_2 - \frac{A}{L}\frac{\partial P_1}{\partial M_1}\Delta t\,\Delta M_1$$

$$+ \frac{A}{L}\frac{\partial P_2}{\partial M_2}\Delta t\,\Delta M_2 = \Delta t\left[\frac{A}{L}(P_1 - P_2) - K|W|W\right], \tag{23a}$$

$$\frac{\Delta W H_1}{M_1}\Delta t + \left(1 + \frac{W}{M_1}\Delta t\right)\Delta H_1 - \frac{\Delta t H_1 W \Delta M_1}{M_1^2} = \Delta t\left(-\frac{H_1 W}{M_1}\right), \tag{23b}$$

$$-\frac{\Delta W H_1}{M_1}\Delta t - W\frac{\Delta H_1}{M_1}\Delta t + \Delta H_2 + \frac{\Delta t H_1 W \Delta M_1}{M_1^2} = \frac{\Delta t H_1 W}{M_1}, \tag{23c}$$

$$\Delta W\,\Delta t + \Delta M_1 = -W\Delta t, \tag{23d}$$

and

$$\Delta W\,\Delta t + \Delta M_2 = W\Delta t. \tag{23e}$$

Thus, from equations 23d and 23e:

$$\Delta M_1 = -(W + \Delta W)\Delta t \tag{24a}$$

$$\Delta M_2 = (W + \Delta W)\,\Delta t \tag{24b}$$

$$\Delta H_1 = -\frac{\Delta t\, H_1 W/M_1 - \Delta t\,\Delta W H_1/M_1 + \Delta t\, H_1 W\,\Delta M_1/\Delta M_2^2}{1 + W/M_1\,\Delta t}$$

$$= -\Delta t\,\frac{H_1}{M_1}(W + \Delta W) \quad \text{after simplification.} \tag{24c}$$

and

$$\Delta H_2 = \Delta t\frac{H_1}{M_1}(W + \Delta W). \tag{24d}$$

Note that mass and enthalpy are conserved.

We note also that the expressions for ΔM and ΔH are similar to what you would obtain by the straightforward application of implicit forward differencing of the orginal equations 14 and 15, i.e:

$$\frac{M_1^{n+1}-M_1^n}{\Delta t} = -W^{n+1}, \qquad \frac{M_2^{n+1}-M_2^n}{\Delta t} = +W^{n+1},$$

$$\frac{H_1^{n+1}-H_1^n}{\Delta t} = -\frac{H_1^{n+1}}{M_1^{n+1}} W^{n+1}, \qquad \frac{H_2^{n+1}-H_1^n}{\Delta t} = \frac{H_1^{n+1}}{M_1^{n+1}} W^{n+1}. \tag{25}$$

The only difference between equations 24 and 25 is the treatment of the H_1/M_1 factor. To arrive at 24c and d, the mass equations (24a and b) were used and indeed, implicit treatment of M and H was used. The result (24a to d) appears explicit in M and H when compared to equation 25. This does not mean that the Jacobi form leads to a semi-implicit method (implicit in flow, explicit in mass and enthalpy). It means that the implicit contributions cancel out. (Note, however, it can be shown that for the general case, Porsching's method is fully implicit in its treatment of the mass equation but not for the enthalpy equation.) Thus we conclude that there is little merit in carrying the mass and enthalpy equations in implicit form. This agrees with earlier observations (GA 86a) that the eigenvalues of a thermal-hydraulic system are associated with the flow and pressure equations, not with the mass and energy equations.

To finish off the derivation of Porsching's approach, we use equations 24a to d in equation 23a to give:

$$\left\{1+2\Delta t\, K|W| + \frac{A}{L}\Delta t^2\left[\frac{H_1}{M_1}\left(\frac{\partial P_1}{\partial H_1}+\frac{\partial P_2}{\partial H_2}\right)+\left(\frac{\partial P_1}{\partial M_1}+\frac{\partial P_2}{\partial M_2}\right)\right]\right\}\Delta W$$

$$= \Delta t\left\{\frac{A}{L}(P_1-P_2)-K|W|W-\Delta t\frac{A}{L}W\left[\frac{H_1}{M_1}\left(\frac{\partial P_1}{\partial H_1}+\frac{\partial P_2}{\partial H_2}\right)+\left(\frac{\partial P_1}{\partial M_1}+\frac{\partial P_2}{\partial M_2}\right)\right]\right\} \tag{26}$$

Setting:

$$C_1 = \frac{H_1}{M_1}\frac{\partial P_1}{\partial H_1}+\frac{\partial P_1}{\partial M_1} \tag{27}$$

and

$$C_2 = \frac{H_1}{M_1}\frac{\partial P_2}{\partial H_2}+\frac{\partial P_2}{\partial M_2} \tag{28}$$

we have

$$\left\{1+2\Delta t\, K|W| + \frac{A}{L}\Delta t^2(C_1+C_2)\right\}\Delta W$$

$$= \Delta t\left\{\frac{A}{L}(P_1-P_2) - K|W|W - \Delta t\frac{A}{L}W(C_1+C_2)\right\} \tag{29}$$

Of course, the general case of N nodes and K links would follow the same substitution route with greatly increased complexity (see PO71). It is not evident from Porsching's general expressions that the resulting flow equations (equation 26 for the simple case of 2 nodes and 1 link) are more dependent on pressure and its derivatives (with respect to mass and energy) than it is on the mass and the enthalpy themselves. This observation, once pointed out, is obvious and undeniable. Yet, no existing thermalhydraulic code for system simulation takes full advantage of this observation.

Does a formulation which contains the appropriate implicit treatment without the large overhead of the general perturbation approach of Porsching's method? A clue exists in equation 17:

$$P = \pi(M, H, V) \tag{17}$$

This can be rewritten:

$$dP = \frac{\partial P}{\partial M}\,dM + \frac{\partial P}{\partial H}\,dH + \frac{\partial P}{\partial V}\,dV. \tag{30}$$

The form of equation 30 suggests that equation (26) contains some version of the total pressure derivative. Indeed we shall see in the next section that the rate form of the equation of state yields the same expression as Porsching's method, without the large overhead.

DERIVATION OF THE RATE FORM OF THE EQUATION OF STATE (from GA86a)

The determination of pressure from known values of other thermodynamic properties is not direct since interpolation and iteration is required because the independent (known) parameters are temperature, T, and pressure, P. Unfortunately, T and P are rarely the independent parameters in system dynamics since the numerical solution of the conservation equations yield mass and energy as a function of time. Hence, from the point of view of the

Pages 12-14 deleted.

$$G_2(P,x) = \frac{F_2(P)}{xF_4(P) + (1-x)F_5(P)}$$

$$= \frac{v_g - v_f}{\left[x\frac{dh_g}{dP} + (1-x)\frac{dh_f}{dP}\right](v_g - v_f) - \left[x\frac{dv_g}{dP} + (1-x)\frac{dv_f}{dP}\right](h_g - h_f)} \quad (46)$$

The F functions are smooth, slowly varying functions of pressure provided good curve fits are used. The latest steam tables (HA84) were used to fit saturated properties to less than 1/4% accuracy using low order polynomials and exponentials (GA86c). Considerable effort was spent on obtaining accuracy and continuous derivatives over the full pressure range. The fact that good fits are available means that the F functions are well behaved which in turn makes the rate form of the equation of state extremely well behaved, as shown later. The G functions are also well behaved for the same reasons.

In general, the equation of state can be written in rate form for all situations (GA86a). We adopt the general form:

$$\frac{dP}{dt} = \frac{F_1\frac{dM}{dt} + F_2\frac{dH}{dt} + F_3\frac{dV}{dt} + M_v F_6\frac{dT_v}{dt} + M_l F_7\frac{dT_l}{dt}}{M_v F_4 + M_l F_5}$$

$$= G_1\frac{d\rho}{dt} + G_2\frac{dh}{dt} + G_3\frac{dT_v}{dt} + G_4\frac{dT_l}{dt} \quad (47)$$

The expressions for the F and G functions are summarized as in GA86a. These expressions cover the full range from subcooled liquid to superheated steam.

Thus, in addition to the system conservation equations 14 to 16, we have two pressure rate equations:

$$\frac{\partial P_1}{\partial t} = \frac{F_{11}\frac{\partial M_1}{\partial t} + F_{12}\frac{\partial H_1}{\partial t}}{M_{1v}F_{14} + M_{1\ell}F_{15}} \equiv G_{11}\frac{\partial M_1}{\partial t} + G_{12}\frac{\partial H_1}{\partial t} \quad (48)$$

$$\frac{\partial P_2}{\partial t} = \frac{F_{21}\frac{\partial M_2}{\partial t} + F_{22}\frac{\partial H_2}{\partial t}}{M_{2v}F_{14} + M_{2\ell}F_{25}} \equiv G_{21}\frac{\partial M_1}{\partial t} + G_{22}\frac{\partial H_2}{\partial t} \quad (49)$$

Substituting in equations 48 and 49 for dM/dt and dH/dt we have:

$$\frac{\partial P_1}{\partial t} = -\left(G_{1i}' + G_{1i}' \frac{H_1}{M_1}\right)W \tag{50}$$

$$\frac{\partial P_2}{\partial t} = +\left(G_{2i}' + G_{2i}' \frac{H_1}{M_1}\right)W \tag{51}$$

But G_{11}' is simply $\partial P_1/\partial M_1$, G_{12}' is simply $\partial P_1/\partial H_1$, etc. Thus, we see that equations 50 and 51 can be rewritten as:

$$\frac{\partial P_1}{\partial t} = -C_1 W \tag{52}$$

$$\frac{\partial P_2}{\partial t} = +C_2 W \tag{53}$$

In matrix form, considering just the flow and pressure equations, we have:

$$\frac{\partial U}{\partial t} = A(U, t)\, U(t) + B \tag{54}$$

$$U = \begin{bmatrix} W \\ P_1 \\ P_2 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

$$A \equiv \begin{bmatrix} -K|W| & \frac{A}{L} & -\frac{A}{L} \\ -C_1 & 0 & 0 \\ +C_2 & 0 & 0 \end{bmatrix}. \tag{55}$$

A typical implicit treatment would be:

$$\frac{U^{n+1} - U^n}{\Delta t} = A(U^{n+1}, t)\, U^{n+1}. \tag{56}$$

Often, to simplify, A is treated explicitly, i.e $A \rightarrow A(U^n, t)$. This yields:

$$(I - \Delta t A)\, U^{n+1} = U^n \tag{57}$$

to be solved for U^{n+1} by matrix inversion.

In this case, since A contains a major nonlinearity in flow (the term K|W|), it is best to treat it more carefully.

Consider the fully implicit form of the flow equation of (54):

$$\frac{W^{n+1}-W^n}{\Delta t} \equiv \frac{\Delta W}{\Delta t} = -K|W^{n+1}|W^{n+1} + \frac{A}{L}(P_1^{n+1}-P_2^{n+1})$$

$$= -K|W^n+\Delta W|(W^n+\Delta W) + \frac{A}{L}(P_1^n-P_2^n) - \frac{A}{L}\Delta t(C_1+C_2)W^{n+1}$$

$$= -K|W^n|W^n - 2K|W^n|\Delta W + 0(\Delta W^2) + \frac{A}{L}(P_1^n-P_2^n)$$

$$-\frac{A}{L}\Delta t(C_1+C_2)W^n - \frac{A}{L}\Delta t(C_1+C_2)\Delta W. \qquad (58)$$

Collecting terms:

$$\left\{1+2K|W^n|\Delta t + \frac{A}{L}\Delta t^2(C_1+C_2)\right\}\Delta W$$

$$= \Delta t\left\{\frac{A}{L}(P_1-P_2) - K|W|W - \Delta t\frac{A}{L}W(C_1+C_2)\right\}. \qquad (59)$$

This is identical to equation 29 obtained from Porsching's method.

Thus, to answer the question posed in the beginning: Yes, a formulation, which contains the appropriate implicit treatment without the large overhead of Porsching's method, does exist. That formulation is obtained via the rate form of the equation of state. The solution algorithm is now straight forward. All the needed partial derivatives are contained in the coefficients, C_1 and C_2, and once coded, can be used for all thermodynamic phases, from single phase subcooled liquid through to superheated steam, and for all the thermal-hydraulic models, from the simple HEM through to the six-equation model. Case dependent system Jacobi's are not required. The system equations are of the simple form of 54 and the user is free to choose from the existing spectrum of numerical schemes, depending on the user's needs. The next section explores a few such schemes to test the importance of implicitly modelling the nonlinear damping term K|W|.

Odd as an operator

$$\nabla \equiv \frac{\partial}{\partial x}\hat{i} + \frac{\partial}{\partial y}\hat{j} + \frac{\partial}{\partial z}\hat{k} = \partial_i \text{ or } \frac{\partial}{\partial x_i}$$

$$\therefore \nabla\phi = \frac{\partial\phi}{\partial x}\hat{i} + \frac{\partial\phi}{\partial y}\hat{j} + \frac{\partial\phi}{\partial z}\hat{k} = \partial_i\phi \quad \text{(vector)}$$

($\nabla\phi$: scalar)

$$\nabla\cdot\bar{V} = \left(\frac{\partial}{\partial x}\hat{i} + \frac{\partial}{\partial y}\hat{j} + \frac{\partial}{\partial z}\hat{k}\right)\left(V_1\hat{i} + V_2\hat{j} + V_3\hat{k}\right)$$

$$= \frac{\partial V_1}{\partial x} + \frac{\partial V_2}{\partial y} + \frac{\partial V_3}{\partial z} = \partial_i V_i \quad \text{(scalar)}$$

$$\nabla\times\bar{V} \equiv \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ V_1 & V_2 & V_3 \end{vmatrix} = \epsilon_{ijk}\,\partial_j V_k$$

ODD (−1): (1 3 2) Even (+1)

ϵ with repeated indices = 0

$$\nabla^2 = \nabla\cdot\nabla = \left(\frac{\partial}{\partial x}\hat{i} + \frac{\partial}{\partial y}\hat{j} + \frac{\partial}{\partial z}\hat{k}\right)\cdot\left(\frac{\partial}{\partial x}\hat{i} + \frac{\partial}{\partial y}\hat{j} + \frac{\partial}{\partial z}\hat{k}\right)$$

$$= \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} = \partial_i\partial_i = \partial_i^2$$

$\nabla\bar{V}$ = DYAD (9 components)

$$\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix}\left(V_1 \quad V_2 \quad V_3\right) \Rightarrow \begin{pmatrix} \frac{\partial V_1}{\partial x} & \frac{\partial V_2}{\partial x} & \frac{\partial V_3}{\partial x} \\ \frac{\partial V_1}{\partial y} & \frac{\partial V_2}{\partial y} & \frac{\partial V_3}{\partial y} \\ \frac{\partial V_1}{\partial z} & \frac{\partial V_2}{\partial z} & \frac{\partial V_3}{\partial z} \end{pmatrix}$$

$$\left[\bar{A}\,\bar{B} = A_i B_j\right]$$

$\bar{A} \cdot \bar{\bar{\tau}} = A_i \tau_{ij}$ (vector), thus $\nabla \cdot \bar{\bar{\tau}} = \partial_i \tau_{ij}$

$\bar{\bar{\tau}} \cdot \bar{A} = \tau_{ij} A_j$ (vector)

note: $\bar{A} \cdot \bar{\bar{\tau}} \neq \bar{\bar{\tau}} \cdot \bar{A}$ in general

$\bar{A} \times \bar{\bar{\tau}} = \epsilon_{iKl} A_K \tau_{jl}$

$\bar{\bar{\tau}} \times \bar{A} = \epsilon_{iKl} \tau_{jK} A_l$ ~~start~~ from back (l)

$\bar{\bar{\sigma}} \cdot \bar{\bar{\tau}} = \sigma_{ij} \tau_{jK}$

$\bar{\bar{\sigma}} : \bar{\bar{\tau}} = \sigma_{ij} \tau_{ji}$

$\begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots \\ \sigma_{21} & \cdots & \end{pmatrix} : \begin{pmatrix} \tau_{11} & \tau_{12} & \cdots \\ \tau_{21} & & \end{pmatrix}$

$= \sigma_{11}\tau_{11} + \sigma_{12}\tau_{21} + \sigma_{13}\tau_{31}$
$+ \sigma_{21}\tau_{12} + \sigma_{22}\tau_{22} + \sigma_{23}\tau_{32}$
$+ \sigma_{31}\tau_{13} + \sigma_{32}\tau_{23} + \sigma_{33}\tau_{33}$

$\bar{\bar{\tau}} : \nabla \bar{v} = \tau_{ij} \partial_j v_i$

How to tell the rank of an operation:

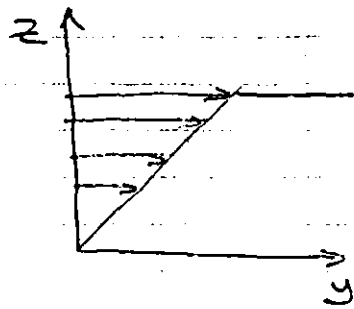
Σ = sum of individual ranks, ie $\bar{\bar{\tau}}$ is rank 2
$\bar{v}$ is rank 1
ϕ is rank 0

thus Σ for $\bar{\bar{\tau}} : \nabla \bar{v}$ is $2 + 1 + i = 4$
$\bar{\bar{\sigma}} : \bar{\bar{\tau}}$ is $2 + 2 = 4$
$\bar{A} \times \bar{\bar{\tau}}$ is $1 + 2 = 3$
$\nabla \phi$ is $1 + 0 = 1$

then if symbol = $\begin{Bmatrix} \text{nothing} \\ \cdot \\ : \\ \times \end{Bmatrix}$ rank = Σ minus $\begin{Bmatrix} 0 \\ -2 \\ -4 \\ -1 \end{Bmatrix}$

thus $\bar{\bar{\tau}} : \nabla \bar{v}$ is $4 - 4 - 0 = 0$ (: scalar)
$\bar{\bar{\sigma}} : \bar{\bar{\tau}}$ is $4 - 4 = 0$ (: scalar)
$\bar{A} \times \bar{\bar{\tau}}$ is $3 - 1 = 2$ (tensor)
$\nabla \phi$ is $1 - 0 = 1$ (vector)

Shear stress



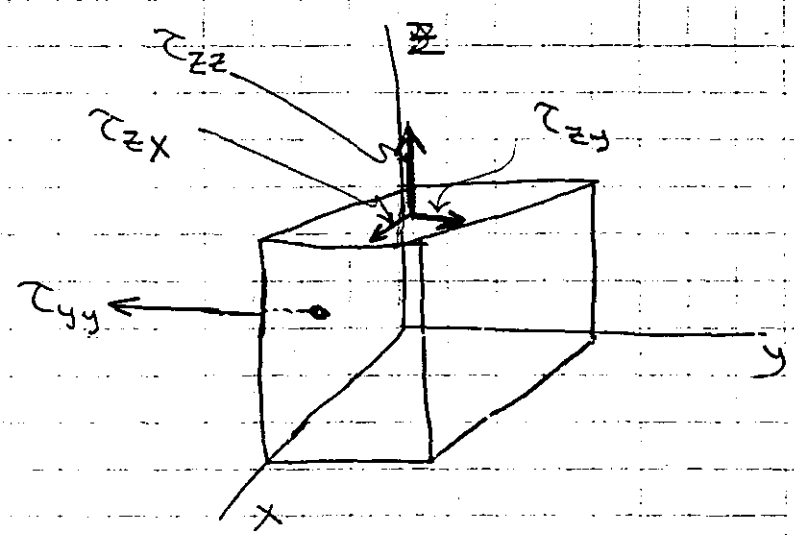
$$\frac{F}{Area} \propto \frac{dV_y}{dz}$$

$$-\mu \frac{dV_y}{dz} \equiv \tau_{zy}$$

(μ ← viscosity)

(i.e. deformation is proportion to the force/unit area)

$$\bar{\bar{\tau}} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix}$$



τ_{ij}

↑ acts on a plane perpendicular to i in the j direction

SOLUTIONS FOR $\dot{\underset{\sim}{y}} = \underset{\sim}{F}(t, \underset{\sim}{y})$ (where $\underset{\sim}{y} = f_n(t)$)

Explicit:

$$\frac{\underset{\sim}{y}^{n+1} - \underset{\sim}{y}^{n}}{\Delta t} = \underset{\sim}{F}(t^n, \underset{\sim}{y}^n)$$

$$\therefore \underset{\sim}{y}^{n+1} = \underset{\sim}{y}^n + \Delta t \, \underset{\sim}{F}(t^n, \underset{\sim}{y}^n)$$

Advantages: 1) Minimum amount of core required.
2) Straightforward to program.
3) Cheap per iteration.
4) Robust

Disadvantages: 1) Stability requirements usually restrict maximum Δt, $\therefore$ # of iterations can be large $\therefore$ expensive. Not a problem if accuracy requirements keep Δt small.

Implicit

$$\frac{\underset{\sim}{y}^{n+1} - \underset{\sim}{y}^{n}}{\Delta t} = \underset{\sim}{F}(t^{n+1}, \underset{\sim}{y}^{n+1})$$

Expand $\underset{\sim}{F}$ via Taylor Series:

$$\underset{\sim}{F}(t^{n+1}, y^{n+1}) = \underset{\sim}{F}(t^n, y^n) + \Delta t \left[\frac{\partial \underset{\sim}{F}(t,y)}{\partial t}\bigg|_n + \frac{\partial \underset{\sim}{F}(t,y)}{\partial \underset{\sim}{y}}\bigg|_n \frac{\partial \underset{\sim}{y}}{\partial t} \right] + O(\Delta t^2)$$

($=0$ if no explicit dependence on t, i.e. $F(t,y) = F(y)$)

$$\therefore \dot{\underset{\sim}{y}} = \underset{\sim}{F}(t^n, y^n) + \Delta t \, \underset{\approx}{J} \, \dot{\underset{\sim}{y}}$$

$\underset{\approx}{J}$ = Jacobi $= \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} \\ \frac{\partial f_2}{\partial y_1} & \end{bmatrix}$

$$\therefore \dot{\underset{\sim}{y}} = \left[\underset{\approx}{I} - \Delta t \, \underset{\approx}{J}\right]^{-1} \underset{\sim}{F}(t^n, \underset{\sim}{y}^n) = \frac{\underset{\sim}{y}^{n+1} - \underset{\sim}{y}^{n}}{\Delta t}$$

$$\therefore \underset{\sim}{y}^{n+1} = \underset{\sim}{y}^n + \Delta t \left[\underset{\approx}{I} - \Delta t \, \underset{\approx}{J}\right]^{-1} \underset{\sim}{F}(t^n, \underset{\sim}{y}^n)$$

This can be shown to be a very stable algorithm even for large Δt (see Porsching, Num. & Eng., 43, p218(1971)

Advantages: 1) Stable
2) Total cost < explicit, usually due to large Δt capability.

Disadvantages 1) Requires more core space
2) Requires matrix inversion
3) Requires analytical setup of Jacobi.
4) Not as easy to program as explicit.

87-02-19

But this is not the only way to use the Jacobian.

We have:

$$\frac{\underset{\sim}{y}^{n+1} - \underset{\sim}{y}^{n}}{\Delta t} = \underset{\sim}{F}\left(t^{n+1}, \underset{\sim}{y}^{n+1}\right)$$

$$= \underset{\sim}{F}(t^n, y^n) + \left[\frac{\partial \underset{\sim}{F}(t,y)}{\partial t}_{0} + \frac{\partial \underset{\sim}{F}(t,\underset{\sim}{y})}{\partial \underset{\sim}{y}}_{\equiv \underline{\underline{J}}}\right]\left(y^{n+1} - y^n\right)$$

$$= \left[\underset{\sim}{F}(t^n, y^n) - \underline{\underline{J}}\,\underset{\sim}{y}^n\right] + \underline{\underline{J}}\,\underset{\sim}{y}^{n+1}$$

$$\dot{\underset{\sim}{y}} = \underline{B} + \underline{\underline{J}}\,\underset{\sim}{y}^{n+1}$$

(similar to $\dot{y} = Ay + B$)

In this sense, $\underline{\underline{J}}$ contains the eigenvalues of the sy

But this does not give the eigenvalues of the sy

since $\underset{\sim}{y}^{n+1} \neq \underset{\sim}{y}$

87-03-26

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

$$\dot{y} = f(y_0,t_0) + \Delta t\, \dot{y}$$

$$= f(y_0,t_0) + J\,\Delta y$$

$$\Delta\dot{y} = f(y_0,t_0) + J\,\Delta y$$

$$\Delta y = y - y_0$$

$$\dot{y}_0 = f(y_0,t_0)$$

↑

This is eigenvalue form.

∴ J contain system eigenvalues.

87-04-21

above true but get λ's for perturbed system (ie about a S.S.)

p formulation gives λ's wrt present state (not necessarily S.S.). Big difference.