Analysis of A Pressure-Flow Vapour Network

Graham W. Griffiths

Contents

Abstract

This bulletin provides an outline of the theoretical approach to the solution of pressure-flow network problems used in the SAST dynamic process simulation system OTISS™. OTISS™ offers various integration options to the user for solving the sets of differential equations involved. The user makes a choice between explicit, implicit or semi-implicit integration methods, where appropriate.

1 Introduction

The theoretical basis of methods used in OTISS™ for the solution of pressure-flow networks is described. The preferred approach is one of semi-implicit integration, which means that a combination of implicit and explicit techniques is employed. The pressures are solved using semi-implicit integration, whilst energy and compositions are integrated explicitly. This approach has two advantages over purely implicit, or equation solving methods, namely, speed and simplicity. For particularly sensitive applications, fully implicit methods are necessary in order to be able to use practical time steps.

A number of integration options are provided in OTISS™ for pressure-flow calculations, and these are discussed.

2 Linearizing The Pressure-Time Derivative

Consider a single flow w, between two pressure nodes, where,

$$w = w(p_1, p_2). \tag{1}$$

Linearizing w gives,

$$w = w(p_1 + \Delta p_1, p_2 + \Delta p_2) = w(p_1, p_2) + \frac{\partial w}{\partial p_1}\Delta p_1 + \frac{\partial w}{\partial p_2}\Delta p_2. \tag{2}$$

Now consider a pressure node with flows entering and leaving, which can be described by the *Ideal Gas Law*,

$$pV = znRT, \tag{3}$$

where,

p = pressure (bars)
V = volume (m^3)
n = number of moles (kg-mole)
z = compressibility of gas (units)
R = ideal gas constant (0.083414, bar m^3/°K/kg-mole)
T = temperature(°K)
w = mass flow (kg/h).

Converting to mass units gives,

$$p = \frac{zRT}{V\ MW}m, \tag{4}$$

where,

MW = molecular weight
m = mass (kg).

Let,

$$cc = \frac{\partial p}{\partial m} = \frac{zRT}{V\ MW}. \tag{5}$$

Thus, for small mass perturbations,

$$\Delta p = cc \times \Delta m, \tag{6}$$

where, cc represents the mass node compressibility (which should not be confused with gas compressibility).

Differentiating eqn. (4) with respect to time, neglecting the contributions made by $\mathrm{d}z/\mathrm{d}t$ and $\mathrm{d}MW/\mathrm{d}t$, gives

$$\frac{\mathrm{d}p}{\mathrm{d}t} = cc\left(\frac{\mathrm{d}m}{\mathrm{d}t} - \frac{m}{V}\frac{\mathrm{d}V}{\mathrm{d}t} + \frac{m}{T}\frac{\mathrm{d}T}{\mathrm{d}t}\right), \tag{7}$$

But, for a node having N flow connections,

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \sum_{i=1}^{N} w_i. \tag{8}$$

$$\therefore \frac{1}{cc}\frac{\mathrm{d}p}{\mathrm{d}t}\left(\sum_{i=1}^{N} w_i - \frac{m}{V}\frac{\mathrm{d}V}{\mathrm{d}t} + \frac{m}{T}\frac{\mathrm{d}T}{\mathrm{d}t}\right). \tag{9}$$

Now substituting linearised flows in accordance with eqn. (2) yields,

$$\frac{1}{cc}\frac{\mathrm{d}p}{\mathrm{d}t} = \left[\sum_{i=1}^{N}\left(w_i + \frac{\partial w_i}{\partial p}\Delta p\right) - \frac{m}{V}\frac{\mathrm{d}V}{\mathrm{d}t} + \frac{m}{T}\frac{\mathrm{d}T}{\mathrm{d}t}\right]. \tag{10}$$

For the k_{th} node in a network consisting of M nodes, having N^{kj} flow connections between the j_{th} and k_{th} nodes, this equation becomes,

$$\frac{1}{cc^k}\frac{\mathrm{d}p^k}{\mathrm{d}t} = \left[\sum_{j}^{M}\sum_{i=1}^{N^{kj}}\left(w_i^k + \sigma_i^k\frac{\partial w_i^k}{\partial p^j}\Delta p^j\right) - \frac{m^k}{V^k}\frac{\mathrm{d}V^k}{\mathrm{d}t} + \frac{m^k}{T^k}\frac{\mathrm{d}T^k}{\mathrm{d}t}\right]. \tag{11}$$

or, in vector form,

$$\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t} = \mathbf{b} + \mathbf{A}\Delta\mathbf{p}, \tag{12}$$

where,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & \cdots & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & \cdots & \cdots & a_{2M} \\ \vdots & \vdots & \ddots & & & \vdots \\ \vdots & \vdots & & a_{kk} & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ a_{M1} & a_{M2} & \cdots & \cdots & \cdots & a_{MM} \end{bmatrix}, \tag{13}$$

$$a_{kj} = cc^k\sum_{i=1}^{N^{kj}}\sigma_i^k\frac{\partial w_i^k}{\partial p^j}, (j \neq k), \tag{14}$$

$$a_{kj} = cc^k \sum_{j=1}^{M} \sum_{i=1}^{N^{kj}} \sigma_i^k \frac{\partial w_i^k}{\partial p^k}, \tag{15}$$

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \\ \vdots \\ b_M \end{bmatrix}, \tag{16}$$

$$b_{kj} = cc^k \left(\sum_{j=1}^{M} \sum_{i=1}^{N^{kj}} w_i^k - \frac{m^k}{V^k} \frac{\mathrm{d}V^k}{\mathrm{d}t} + \frac{m^k}{T^k} \frac{\mathrm{d}T^k}{\mathrm{d}t} \right). \tag{17}$$

And,

$\sigma = \pm 1$ (according to whether flow is entering or leaving the node.). This approach is known as *quasilinearization*, see [Fer-81].

2.1 Semi-Implicit Integration of Pressure

We can now integrate equation (12) using the implicit Euler method, i.e.,

$$\mathbf{p}_{n+1} = \mathbf{p}_n + \frac{\mathrm{d}\mathbf{p}_{n+1}}{\mathrm{d}t} \Delta t, \tag{18}$$

where, Δt = integration time step.

$$\therefore \mathbf{p}_{n+1} = \mathbf{p}_n + \left[\mathbf{b} + \mathbf{A} \Delta \mathbf{p}_{n+1} \right] \Delta t. \tag{19}$$

But,

$$\Delta \mathbf{p}_{n+1} = \mathbf{p}_{n+1} - \mathbf{p}_n. \tag{20}$$

$$\therefore \mathbf{p}_{n+1} = \mathbf{p}_n + \left[\mathbf{b} + \mathbf{A}(\mathbf{p}_{n+1} - \mathbf{p}_n) \right] \Delta t. \tag{21}$$

$$\therefore [\mathbf{I} - \mathbf{A}\Delta t] \mathbf{p}_{n+1} = \mathbf{b}\Delta t + [\mathbf{I} - \mathbf{A}\Delta t] \, \mathbf{p}_n. \tag{22}$$

The above represents a set of linear equations which are equivalent to,

$$\mathbf{Q}\mathbf{p}_{n+1} = \mathbf{r}, \tag{23}$$

where

$$\mathbf{Q} = [\mathbf{I} - \mathbf{A}\Delta t], \quad \text{and} \tag{24}$$

$$\mathbf{r} = \mathbf{b}\Delta t + [\mathbf{I} - \mathbf{A}\Delta t] \, \mathbf{p}_n, \tag{25}$$

which can be solved using standard elimination techniques, as **Q** and **r** are known!

The above describes the semi-implicit Euler method which has very good stability properties and is relatively computing efficient.

Because it is essential to ensure consistency between mass and pressure, a correction term is included in the pressure calculation - refer to section (6).

2.2 Fully Implicit Methods

For many problems, the semi-implicit method described above will provide a sufficiently accurate and stable solution. However, for particularly sensitive or highly non-linear problems, it may be necessary to employ a fully implicit method in order to be able to use a practical step size. Though this will usually add significantly to the computing load for each step.

Of the large number of fully implicit methods available, those based upon **backward differentiation formulas** or BDFs have been implemented in OTISS™. BDF methods are some times referred to as **Gear's method** as they form the basis of integrators developed by Gear around 1971 [Gea-71]; however, they were actually used much earlier by other workers in the field [Sha-94][pp 184].

The BDF family of integrators is defined mathematically as follows,

$$\mathbf{p}_{n+1} = \sum_{i=1}^{k} \alpha_i \mathbf{p}_{n+1-i} + \beta_0 \mathbf{f}\left(\mathbf{p}_{n+1}, t_{n+1}\right) \Delta t, \tag{26}$$

where

$$\mathbf{f} = \frac{\mathrm{d}p}{\mathrm{d}t}, \tag{27}$$

and values for α_i and β_0 are given below for orders k, up to 6 [Fer-81][pp 99]. They have the advantage that only one value of the time derivative is required at each time step. This is particularly important when solving problems with large numbers of differential equations as data handling and data storage are minimised compared with other *multi-step* methods, e.g. the **Adams-Moulton** method, which requires the use of past values of the associated *Jacobian matrix* and which also has inferior stability properties.

The *BDF* or *Gear* method has been implemented in OTISS™ up to order 6. However, it should be noted that as the order increases, the stability

Table 1: Coefficients For Gear's Method

k	1	2	3	4	5	6
b0	1	2/3	6/11	12/25	60/137	60/147
a1	1	4/3	18/11	48/25	300/137	360/147
a2		-1/3	-9/11	-36/25	-300/137	-450/147
a3			2/11	16/25	200/137	400/147
a4				-3/25	-75/137	-225/147
a5					12/137	72/147
a6						-10/147

properties of the method decrease. Thus, care has to be taken in selecting a method for a particular problem.

It will be seen from Table No:1 that the order 1 *BDF* method is identical to the implicit Euler method, and therefore has the same stability properties.

Most applications where fully implicit integration is necessary involve the solution of non-linear differential equations. This means that equation (26) has to be solved iteratively, and in OTISS™, a **Newton-Raphson** method is used. This involves linearizing the problem as for the semi-implicit method above and iterating until $\mathbf{p}_{n+1}$ converges to within the required tolerance.

2.3 Explicit Integration of Pressure

For explicit integration, the pressure nodes are effectively decoupled, and the problem becomes scalar. The pressure derivative reduces to,

$$\frac{dp^k}{dt} = b^k, \quad (k = 1 \dots M), \tag{28}$$

and, using explicit Euler

$$\therefore p^k_{n+1} = p^k_n = b^k \Delta t, \quad (k = 1 \dots M). \tag{29}$$

The use of explicit Euler integration is usually limited to calculations that seek to investigate very fast transients, when some speed advantage can be gained over implicit integration.

2.4 Trapezoidal Integration of Pressure

The Trapezoidal method is semi-implicit and uses the average of explicit and implicit derivatives in order to derive the next value, i.e.

$$\frac{d\bar{\mathbf{p}}}{dt} = \frac{1}{2}\left(\frac{d\mathbf{p}_{exp}}{dt} + \frac{d\mathbf{p}_{imp}}{dt}\right). \tag{30}$$

The average derivative therefore becomes,

$$\frac{d\bar{\mathbf{p}}}{dt} = \frac{1}{2}\left(\mathbf{b}_n + \mathbf{b}_{n+1} + \left[\mathbf{A}_n + \mathbf{A}_{n+1}\right]\Delta\mathbf{p}_{n+1}\right). \tag{31}$$

Pressure is calculated by substituting equation (31) into equation (18) and solving in exactly the same manner as for either semi-implicit or fully implicit integration. This method gives second order accuracy and better stability than explicit methods. It is usually used to obtain high accuracy when investigating fast transients, as larger step sizes can be used than would be required for explicit integration.

2.5 Local Implicit Integration of Pressure

Local implicit integration is used for slow processes that do not exhibit a significant interaction between pressure nodes. It is also useful for debugging simulation models. Because the nodes are effectively decoupled, the off-diagonal elements in A are zero. Thus, the pressure derivatives are obtained from a scalar version of equation (12), i.e.

$$\frac{dp^k}{dt} = b_k + a_{kk}\Delta p^k, \quad (k = 1 \ldots M). \tag{32}$$

This equation can be integrated implicitly without having to resort to the use of matrix methods because it can be solved algebraically. The solution therefore becomes

$$p^k_{n+1} = \frac{b_k\Delta t + [1 - a_{kk}\Delta t]p^k_n}{[1 - a_{kk}\Delta t]}, \quad (k = 1 \ldots M), \tag{33}$$

This method has a high degree of robustness and is computationally efficient. It should therefore be used whenever appropriate.

3 ENTHALPY BALANCE

The dynamic energy balance reduces to an enthalpy balance if kinetic energy and gravitational effects can be safely ignored. This is usually the case for vapour systems and, under these circumstances, the energy balance for the kth node can be described mathematically as follows

$$\frac{d(mu)^k}{dt} = \sum_{j=1}^{N^k} h^k_i w^k_i + Q^k, \tag{34}$$

where,

u = specific internal energy (kJ/kg)
h = specific enthalpy (kJ/kg)
Q = external heat flow (kJ/h) .

It is usually more convenient to describe the energy balance in terms of enthalpy, rather than internal energy. Thus, using the relationships,

$$h = u + 100\,pv, \tag{35}$$

and

$$H = U + 100\,pV, \tag{36}$$

the energy balance[1] can be expressed as,

$$\frac{dH^k}{dt} = \sum_{j=1}^{N^k} h_i^k w_i^k + Q^k + 100\frac{d(pV)^k}{dt}, \tag{37}$$

where,

v = specific volume (m^3/kg)
U = total internal energy (equal to mu) (kJ)
H = total enthalpy (equal to mh) (kJ).

An equation of this type is formulated for each node and integrated to give,

$$H_{n+1} = H_n^k + \frac{dH_n^k}{dt}\Delta t, \quad (k = 1, \ldots, M), \tag{38}$$

and, of course

$$h_{n+1}^k = \frac{H_{n+1}^k}{m_{n+1}^k}, \quad (k = 1, \ldots, M). \tag{39}$$

4 METAL MASS

The effect of vessel metal mass is accounted for in the energy balance by increasing the fluid hold-up mass by an amount that will provide an equivalent thermal effect, i.e.

$$m_e^k = m^k + M^k\frac{C_M^k}{C_p^k}, \tag{40}$$

[1]The constant "100" is a conversion factor to convert p from bar to kPa, so that $100pv$ has units (kJ).

where,

m_e = total equivalent fluid mass (kg)
M = vessel metal mass (kg)
C_p = specific heat of fluid (kJ/kg/°C)
C_M = specific heat of metal (kJ/kg/°C).

m_e is used in equation(34) and equation(39) to provide the correct overall thermal response.

5 COMPOSITION BALANCE

The composition is calculated as follows

$$\frac{dn_i^k}{dt} = \sum_{j=1}^{N^k} \frac{w_j^k x_i^k}{MW_j^k}, \quad (k = 1, \ldots, M;\ i = 1, \ldots, NC), \tag{41}$$

$$\therefore n_{i,n+1}^k = n_{i,n}^k + \frac{dn_{i,n}^k}{dt}\Delta t, \quad (k = 1, \ldots, M;\ i = 1, \ldots, NC), \tag{42}$$

$$(n_T)_{n+1}^k = \sum_{i=1}^{NC} n_{i,n+1}^k, \quad (k = 1, \ldots, M), \tag{43}$$

$$x_{i,n+1}^k = \frac{n_{i,n+1}^k}{(n_T)_{n+1}^k}, \quad (k = 1, \ldots, M;\ i = 1, \ldots, NC), \tag{44}$$

where,

NC = number of components in composition vector (units)
x_i = ith mole fraction in composition vector (0-1)
n_i = number of moles in ith component fraction (mx_i/MW)
n_T = total number of moles (m/MW).

6 MASS HOLDUP

Once the pressure and enthalpy are known for each node, then the mass hold-up can be calculated from an equation-of-state, i.e.

$$m_{n+1}^k = f(p_{n+1}^k,\ h_{n+1}^k,\ V^k), \quad (k = 1, \ldots, M). \tag{45}$$

However, for analysing real systems using a practical integration step size, this approach can lead to mass loss or mass gain over a period of time.

Therefore, in OTISS™, mass hold-up is calculated by integrating equation (8), i.e.

$$m_{n+1}^k = m_n^k + \frac{dm_n^k}{dt}\Delta t, \quad (k = 1, \ldots, M). \tag{46}$$

This results in an over determined system, which requires a correction to be made to the pressure calculation in order to ensure consistency of mass hold-up and pressure.

The pressure correction term is obtained by subtracting the integrated mass from the mass calculated from equation (45), multiplying by the node compressibility and applying a relaxation factor, as follows,

$$\varepsilon_{n+1}^k = \alpha(m_{n+1}^{*k} - m_{n+1}^k)CC, \quad (k = 1, \ldots, M), \tag{47}$$

where,

ε = pressure correction term (bars)
m^{*k} = mass hold-up calculated from equation(45) (kg)
m = mass hold-up calculated from equation(46) (kg)
α = relaxation coefficient (0-1, typically 0.2).

The above is equivalent to a mass generation term and is therefore included in equation (19).

Thus, as the calculations proceed following a transient disturbance, the mass hold-up errors asymptotically approach zero.

7 DENSITY

The node density, ρ (kg/m3), can be calculated once the mass hold-up is known, i.e.

$$\rho_{n+1}^k = \frac{m_{n+1}^k}{V^k}, \quad (k = 1, \ldots, M). \tag{48}$$

8 MOLECULAR WEIGHT

Once the mass hold-up and number of moles are known for each node, then the molecular weight can be calculated using the following algebraic relationship

$$MW_{n+1}^k = \frac{m_{n+1}^k}{(n_T)_{n+1}^k}, \quad (k = 1, \ldots, M). \tag{49}$$

References

[Gea-71] **Gear, C. W.** (1971): *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice Hall, Englewoods, New Jersey, USA.

[Sha-94] **Shampine, L. F.** (1994): *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, New York, USA.

[Fer-81] **Ferziger, J. H.** (1981): *Numerical Methods for Engineering Applications*, John Wiley, USA.

[Lee-68] **Lee, E. Stanley** (1968): *Quasilinearization and Invariant Imbedding – With Applications to Chemical Engineering and Adaptive Control*, Academic Press, New York.