System Codes for Reactor Licensing – Part 2: Code Description

Keith Ardron
UK Licensing Manager ,
AREVA NP UK

AREVA

Imperial College – Nuclear Thermalhydraulics Course: February 2014

Contents

Contents

- Evolution of codes from Homogeneous Eqilibrium 3 Equation Model to Two-Fluid Six Equation Model
- Finite Differecne Solution Methods
- Explicit and Implicit Solutions
- Techniques for Solving Implicit Equations

Homogenous Equilibrium Model (HEM)

- Early approach used in system codes such as RELAP3 was to use the HEM which treated steam water mixture as pseudo-fluid with phases traveling at same velocity and in thermal equilibrium
- Conservation equations derived from those for compressible flow of gas mixtures
- HEM is a good approximation for high velocity dispersed flow regimes where phases are well mixed
- HEM not adequate for flows where gas and liquid streams are separated (e.g. stratified flows, annular flows, counter-current flows) or flows where mean phase temperatures are different (sub-cooled boiling, dispersed flow film boiling etc)

1-D Homogeneous Equilibrium Model

Mass Conservation	$\frac{\partial \rho}{\partial t} + \frac{1}{A}\frac{\partial}{\partial z}(A\rho u) = 0$
Momentum Conservation	$\frac{\partial}{\partial t}(\rho u) + \frac{1}{A}\frac{\partial}{\partial z}(A\rho u^2) + \frac{\partial p}{\partial z}$ $+\rho g \sin\gamma = \frac{\tau_{wz} P_w}{A \cos\beta}$
Energy Conservation	$\frac{\partial}{\partial t}\rho(\varepsilon + \frac{u^2}{2}) + \frac{1}{A}\frac{\partial}{\partial z}A\rho u(h + \frac{u^2}{2})$ $-\frac{1}{A}\frac{\partial}{\partial z}(\kappa A \frac{\partial \theta}{\partial z}) + \rho u g \sin\gamma = \frac{\dot{q}_L}{A}$

Table 2.3 One-Dimensional Conservation Equations for Duct Flow

Duct Area
A
Density
$\rho = \alpha_g \rho_g + \alpha_f \rho_f$
Enthalpy
$h = X_g h_g + X_f h_f$
Internal energy
$\varepsilon = X_g \varepsilon_g + X_f \varepsilon_f$

Mass Fraction X_g and Volume Fraction α_g related by
$$\left(\frac{1-\alpha_g}{\alpha_g}\right) = \left(\frac{1-X_g}{X_g}\right)\left(\frac{\rho_g}{\rho_f}\right)$$

Duct Axis Upward Inclination
γ
Duct Expansion Angle
β
Temperature
θ
Conductivity
κ
Perimeter
P_w

Two-Fluid 6-Equation Model (TFM)

- Requirement to model separated flows or flows where the bulk phases were at different temperatures led to the move to the 2-fluid six equation model in system codes (TRAC-PF1, RELAP5, CATHARE)
- In the TFM the mass, momentum and energy equations are solved for the phases separately, using constitutive equations to specify transfer of mass, momentum and energy between the phases and between the phases and the duct wall.
- Early concern was that some two-fluid equation formulations were non-hyperbolic, implying that no solution existed (ill-posedness). Found that this was usually due to neglect of physical terms which stabilized the equations and rendered them well-posed, without necessarily impacting the solution of most problems

	Conservation Equation	Interfacial Balance Equation
Mass	$\frac{\partial}{\partial t}(C_{1k}\alpha_k\rho_k) + \frac{1}{A}\frac{\partial}{\partial z}(A\ C_{2k}\alpha_k\rho_k u_{kz}) = \Gamma_k$	$\sum_k \Gamma_k = 0$
Momentum	$\frac{\partial}{\partial t}(C_{2k}\alpha_k\rho_k u_{kz}) + \frac{1}{A}\frac{\partial}{\partial z}(AC_{3k}\alpha_k\rho_k u_{kz}^2) + \alpha_k\frac{\partial p_k}{\partial z} + (p_k - p_{ki})\frac{\partial \alpha_k}{\partial z} + \alpha_k\ \rho_k g\ \sin\gamma = \tau_{wkz}P_{wk}/A\ \cos\beta + u_{kiz}\Gamma_k + M_{kz}^d$	$\sum_k (u_{kiz}<\Gamma_k> + p_{ki}\frac{\partial \alpha_k}{\partial z} + M_{kz}^d) = 0$
Energy	$\frac{\partial}{\partial t}(C_{6k}\alpha_k\rho_k h_k)\ \frac{1}{A}\frac{\partial}{\partial z}(AC_{3k}\alpha_k\rho_k u_{kz}h_k) + \frac{\partial}{\partial t}(C_{3k}\alpha_k\rho_k u_{kz}^2/2) + \frac{1}{A}\frac{\partial}{\partial z}AC_{5k}\alpha_k\rho_k u_{kz}^3/2) - \frac{\partial}{\partial t}C_{4k}\alpha_k p_k + p_{ki}\frac{\partial \alpha_k}{\partial t} + C_{2_k}\alpha_k\ \rho_k u_{kz} g\ \sin\beta = \frac{Q_k^w}{A} + \Gamma_k(h_{ki} + u_{ki}^2/2) + E_k^Q + \frac{1}{A}u_{kiz}M_{kz}^d$	$\sum_k [\ \Gamma_k(h_{ki} + u_{ki}^2/2) - p_{ki}\frac{\partial \alpha_k}{\partial t} + E_k^Q + u_{kiz}M_{kz}^d\ ] = 0$

Table 2.5 One-dimensional Time-Average Two-Fluid Equations.

AREVA

Contents

- Evolution of codes from Homogeneous Eqilibrium 3 Equation Model to Two-Fluid Six Equation Model
- **Finite Difference Solution Methods**
- Explicit and Implicit Solutions
- Techniques for Solving Implicit Equations

Finite Difference Methods

- Finite Difference Methods have been the methods most commonly used in system codes for solving transient equations for steam water flows
- Challenge for analyst has been to find differencing schemes which produce accurate solutions with rapid convergence & acceptable computer run times
- Basic finite differencing approach will be described

Finite Difference Method

- Equations first written as quasi linear first order partial differential equations:

Accumulation term

Advection term

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial z} = c \qquad (1)$$

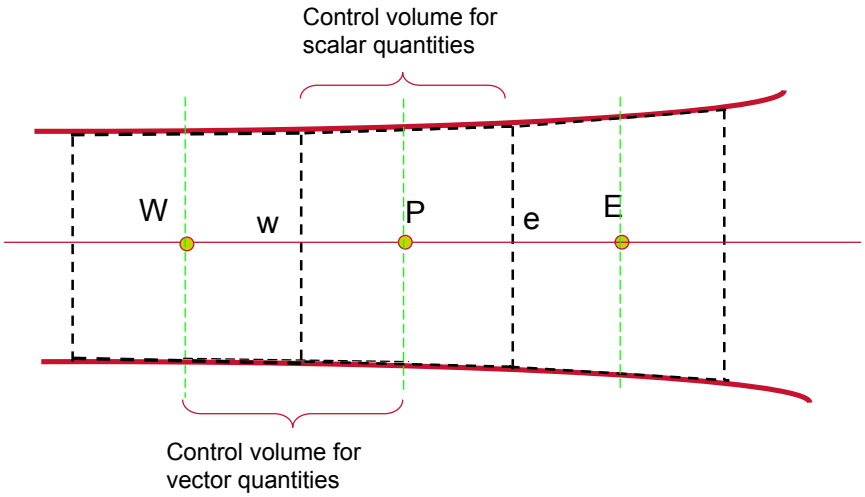
Source term

φ is a conserved quantity which is a function of the chosen dependent variables y_i

c is a sum or terms that depend on the y_i and their partial derivatives in space and time

- The dependent variables of the problem y_i can be selected in many ways
- Equation (1) is written in discretised form on a finite difference mesh

Finite Difference 'Staggered' Mesh (TRAC, RELAP, CATHARE)



W, P, E = cell centres for evaluation of scalar quantities (ρ, p, h..)

w, e = cell centres for evaluation of vector quantities (u) (=scalar cell boundaries)

Discretised Equations

►Equations (1) are written in discretised form using upwind differencing for the advection terms. For a scalar variable at mesh point P

$$\frac{(\underline{\phi}_P - \underline{\phi}_P^0)}{\Delta t} + u_w \frac{(\phi_P^* - \underline{\phi}_W^*)}{\Delta z} = c_P^* \qquad u > 0$$

$$\frac{(\underline{\phi}_P - \underline{\phi}_P^0)}{\Delta t} + u_e \frac{(\phi_E^* - \phi_P^*)}{\Delta z} = c_P^* \qquad u < 0$$

(2)

Advection term always derived using properties of cell upstream of cell P. Also referred to as donor cell differencing

Contents

- Evolution of codes from Homogeneous Eqilibrium 3 Equation Model to Two-Fluid Six Equation Model
- Finite Differecne Solution Methods
- Explicit and Implicit Solutions
- Techniques for Solving Implicit Equations

Explicit v Implicit Time Advancement

- In solving Equation (2) φ^* and c^* can be defined either at the new timestep or the previous timestep
- Defining the dependent variables at the previous timestep is termed *explicit* advancement. It is simple to implement numerically as the equations can be solved one by one in a single step.
- Evaluating the dependent variables at the new time level is called *implicit* advancement. It is difficult to implement as it involves iterative solution of non-linear simultaneous equations in the dependent variables
- But an implicit or semi-implicit method turns out to be the only practical approach in most cases…

Problem of Explicit Integration

▶To examine the accuracy and stability of the different approaches, consider the basic form of equation (2), treating c and the value of φ in the donor cell as constants. In the limit Δt→0: the equation can be expressed as on ordinary differential equation in time:

$$\frac{df}{dt} + |u|\frac{(f - f_u)}{\Delta z} = c$$

f_u = value of f in upstream cell

which can be written as

$$\frac{df}{dt} = -af + b \qquad a > 0 \qquad (3)$$

where

$$a = \frac{|u|}{\Delta z} \qquad b = -\frac{|u| f_u}{\Delta z} + c$$

Model equation

$$\frac{df}{dt} = -af + b$$

Exact solution (after n steps)

$$f_n = f_2 + (f_1 - f_2)e^{-an\Delta t}$$

$$f(0) = f_1$$

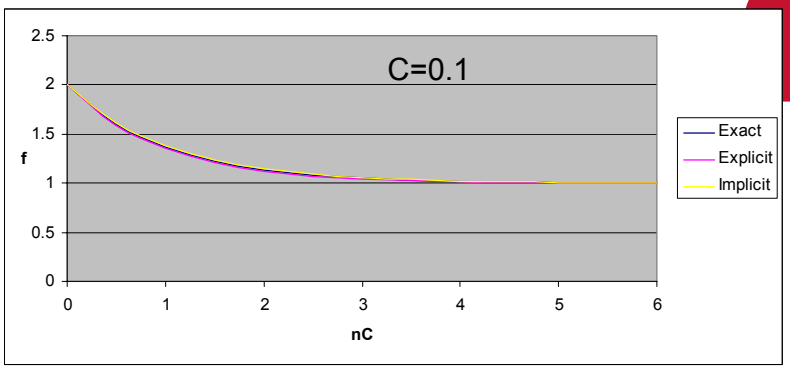
$$f(\infty) = f_2$$

Explicit Solution (after n steps)

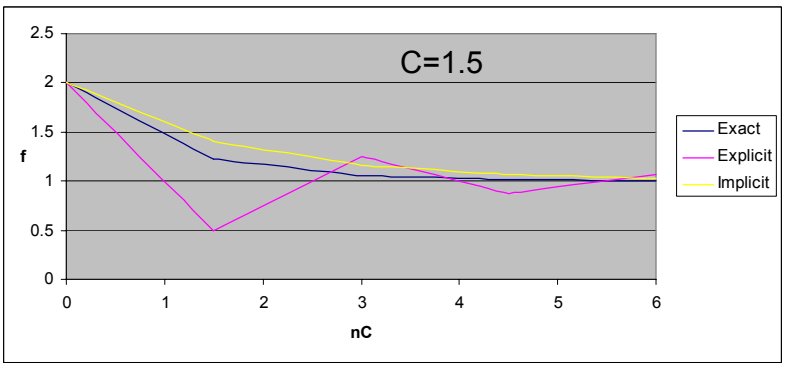
$$f_n = f_2 + (f_1 - f_2)(1 - a\Delta t)^n$$

Implicit Solution (after n steps)

$$f_n = f_2 + (f_1 - f_2)\left(\frac{1}{1 + a\Delta t}\right)^n$$



$f_1 = 2 \qquad f_2 = 1 \qquad C = a\Delta t = u\Delta t / \Delta z$



Timestep Limit in Explicit Integration

- Explicit integration requires time steps below Courant limit for stability.

$$\Delta t_{max} = \Delta z / u$$

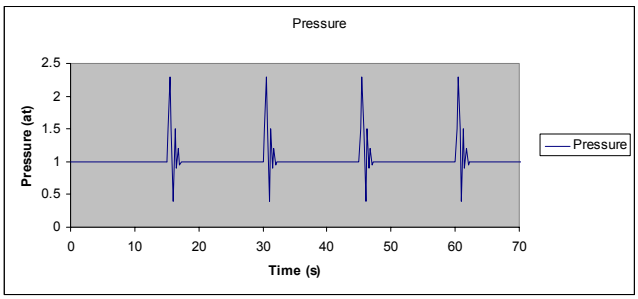
- Penalty in computer time not acceptable for large system codes
- Note that terms arise in conservation equations analogous to the advection terms but coefficients of order of the transport velocities of sonic or thermal waves. Such terms may cause even more restrictive timestep limitations
- General rule is to avoid purely explicit integration in developing a system code although some codes use a mixture or implicit and explicit integration to reduce computational effort at each step

Note: Equation (3) confirms that if advection term had been calculated with downwind differencing the coefficient a would be negative. Hence solutions would diverge for both implicit and explicit method, for all time-steps. Upwind differencing is thus necessary for stability

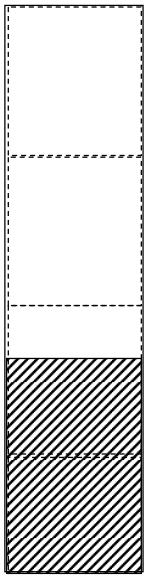
Choice of Dependent Variables- Mass Errors

- Most obvious choice of solution variables $\underline{y}$ is the conserved quantities of mass, momentum and energy –suggesting dependent variables ρ, (ρu) and h.
- This choice is sensible in gas dynamics but does not work well for gas-liquid flow
- Following this approach cell pressure p would have to be derived from ρ and h using state equations. When the gas phase disappears from a cell, small numerical errors in density in the liquid equation of state can result in a large errors in pressure, due to the liquid being largely incompressible (water-packing problem)
- Therefore p is usually chosen as a solution variable and ρ derived from p and h using state equations.
- *The disadvantage with this approach is that mass conservation errors arise due to approximations inherent in state equations. These errors must be calculated and maintained at acceptable levels by timestep control*
- *Some approaches involve a final solution step in which the density is corrected using the conservative form of the mass conservation equation*

Choice of Dependent Variables- Mass Errors



Filling of steam filled pipe with saturated water at atmospheric pressure showing water packing problem when liquid level crosses the cell boundaries



Forced flow boundary condition

Implicit Solution Approaches in System Codes

Approaches to Solving the Finite Difference Equations Implicitly

- Typical dependent variables set chosen for 6 equation model (CATHARE example): α_g, p, u_g, u_f, h_g, h_f, X_{nc}

- Conservation equations written in implicit finite difference form, using upwind differencing

$$\underline{F}(\underline{y}) = 0 \qquad (4)$$

where $\underline{F}$ is column vector of equations and $\underline{y}$ is a the vector of the dependent variables

The total number of equations and variables is

$$(4+N_c)\,N_s + 2N_v$$

where

N_c= number of non-condensable gases ; N_s= number of scalar control volumes; N_v= number of vector control volumes

Method of Successive Substitution

- This is the simplest approach. Eq (4) is written as

$$H\,\underline{y} = \underline{b} \qquad (5)$$

where H is a square matrix of coefficients and b is a column vector of source terms. Each row of H corresponds to a conservation equation for a particular cell. The only non-zero coefficients are the cell values and those for the small number of connected cells. Hence the matrix is sparse.

- A sparse matrix solver can be used to solve (5) for $\underline{y}$. Calculated values of $\underline{y}$ can then be used to recalculate H and b and the process repeated until convergence is achieved
- The process is equivalent to solving $\underline{y}=\underline{g}(\underline{y})$ by successive substitution

$$\underline{y}^{k+1}=\underline{g}(\underline{y}^{k}) \quad \text{where } \underline{g}=H^{-1}\underline{b}$$

However convergence of this process is not guaranteed. Convergence is often improved by applying under-relaxation:

$$y_i^{k+1} = \theta g_i(\underline{y}^{k}) + (1-\theta) y_i^{k}$$

where θ is a relaxation parameter (<1 in most problems)

- Successive substitution method used in PHEONICS code

Newton Raphson Method (e.g. CATHARE)

- Method uses the generalization of the Newton method to a system on equations. The Newton method for a single variable states that if y^k is an approximate solution to F(y)=0 a better approximation will be given by:

$$y^{k+1} = y^k - F(y^k)/F'(y^k)$$

- The Newton method can be generalized to a system of non-linear equations $\underline{F}(\underline{y})=0$

$$\sum_{j=1}^{N}\left(\frac{\partial F_i}{\partial y_j}\right)^k (y_j^{k+1} - y_j^k) = -F_i^k \qquad (6)$$

- The matrix $J_{ij}=\partial F_i/\partial y_j$ is referred to as the Jacobian Matrix. In matrix notation (6) can be written:

$$\underline{y}^{k+1} = \underline{y}^k - J^{-1}\underline{F}^k$$

- The Jacobian elements are derived by numerical differentiation
- Newton method gives rapid convergence but requires an initial guess close to the solution

Example of discretised form of energy equation (CATHARE)

$$A\frac{\partial}{\partial t}\left[\alpha_k.\rho_k.(H_k + \frac{V_k^2}{2})\right] + \frac{\partial}{\partial z}\left[A.\alpha_k.\rho_k.V_k.(H_k + \frac{V_k^2}{2})\right] - A.\alpha_k.\frac{\partial P}{\partial t}$$
$$= A.q_{ke} + \chi_c.q_{pk} + (-1)^k.A.\Gamma.\left(H_k + \frac{1}{2}w_i^2\right) + A.\alpha_k.\rho_k.V_k.g_z$$

$$VOL_o.\frac{(\alpha_k)_o^{n+1}.(\rho_k)_o^{n+1}.\left[(H_k)_o^{n+1} + \frac{1}{2}.\left|\overline{V}_k^2\right|_o^{n+1}\right] - (\alpha_k)_o^n.(\rho_k)_o^n.\left[(H_k)_o^n + \frac{1}{2}.\left|\overline{V}_k^2\right|_o^n\right]}{t^{n+1} - t^n}$$
$$+ A_{x_+}.(\tilde{\alpha}_k)_{x_+}^{n+1}.(\tilde{\rho}_k)_{x_+}^{n+1}.(V_k)_{x_+}^{n+1}.\left[(\tilde{H}_k)_{x_+}^{n+1} + \frac{1}{2}.(V_k^2)_{x_+}^{n+1}\right]$$
$$- A_{x_-}.(\tilde{\alpha}_k)_{x_-}^{n+1}.(\tilde{\rho}_k)_{x_-}^{n+1}.(V_k)_{x_-}^{n+1}.\left[(\tilde{H}_k)_{x_-}^{n+1} + \frac{1}{2}.(V_k^2)_{x_-}^{n+1}\right]$$
$$- VOL_o.(\alpha_k)_o^{n+1}.\frac{\left[P_o^{n+1} - P_o^n\right]}{t^{n+1} - t^n}$$
$$= VOL_o.(q_{ke})_o^{n+1} + (\chi_e)_o.(q_{pk})_o^{n+1}$$
$$+ (-1)^k.VOL_o.\Gamma_o^{n+1}.\left[(H_k)_o^{n+1} + \frac{1}{2}.(w_i^2)_o^{n+1}\right]$$
$$+ VOL_o.(\alpha_k)_o^{n+1}.(\rho_k)_o^{n+1}.(\overline{V}_k)_o^{n+1}.(g_z)_o$$

Notation

o= scalar cell value

X =vector cell value

~ = donor (upstream) scalar cell value)

n+1=current timestep value

n=previous time step value

Semi-Implicit Methods (1/2)

- To reduce computer time associated with iterative solutions of large equations systems, some codes carry out a two-step iteration on some or all the equations.
- Examples are TRAC-PF1 code and RELAP5 codes.
- TRAC-PF1
  - two step iteration on the momentum equations first used to derive the new-time velocities u^{k+1} using old time densities and estimated new time pressures
  - Equations for mass and energy conservation then solved iteratively for new time values dependent variables α, p, T_g, T_f (Newton Raphson method)
  - Final step is to solve for new time density and energy using conservative form of the mass and energy equations
  - Hence mixed/implicit explicit method with only partial iteration

Semi-Implicit Methods (2/2)

- RELAP5 code
  - Mass & momentum equations written in sum and difference form
  - Dependent variables used: α_g, p, u_g, u_f, ε_g, ε_f, X_{nc}
  - Two step process used to find new time values
  - The only equation system solved simultaneously is NxN linear equation system for cell pressures (solved with sparse matrix solver)
  - Final step is to solve for new time density and energy using conservative form of the mass and energy equations
  - Solution scheme has been criticized as non-linear equations are solved *without iteration*
  - Probably OK at small timesteps