Chapter 4

Computation fluid dynamics (CFD) technique

4.1 General

Computation fluid dynamics (CFD) technique, as defined by Anderson [1995], is the art of replacing the integrals or partial derivatives in conservation equation of fluid flow (such as described in Chapter 3) with discretised algebraic form, which in turn are solved to obtain "numbers" for the flow field value at discrete point/volume in time and/or space. The end product of CFD is indeed a collection of numbers which are the main objective of most engineering analyses, the quantitative description of the problem.

There are many potential CFD softwares available for academic and commercial application. In this study, a CFD software "PHOENICS 2.1" is employed. This software provide specific finite volume discretization technique with equation solver suitable for multiphase flow problem called IPSA (Inter-Phase Slip Algorithm). This IPSA is employed to handle two phase pipe flow conservation equations described in Chapter 3.

The material of this Chapter is confined to the IPSA and CFD subject related. For basic concept and complete subject of finite volume CFD, the work of Patanka [1980] is recommended. PHOENICS 2.1 Lecture note provided in the software give the complete knowledge of its CFD code is frequently referred in this chapter. The original paper of IPSA written by Spalding [1980] is also the main source of material in this chapter.

4.2 Definition of grid, node and control volume.

In flow volume considered at which pressure and other scalar dependent variables prevails, if this volume is subdivided into many "finite" cell volume and the dependent variables locate "within" and "represent for" each cell volume is computed and added together, the flow field values of the whole space-time domain are therefore obtained. Diagram shown in Figure 4.1 illustrate this concept, however, it must not be supposed that cells are always equal in size , or rectangular in shape. The grid and node system use in this CFD code is called "staggered grid" as diagram shown in Figure 4.2 . The diagram illustrates a single cell with four of its neighbors. Temperature, pressure and concentration are calculated and then located at the "grid node" P, N, E, W which lie within cell but west to east velocities are evaluated for the cell wall locations like w and e. and south to north velocities are evaluated for the cell faces s and n. For three dimension cell, additional high and low neighbors at the node H and L and velocities evaluated at cell faces h and l are presented.

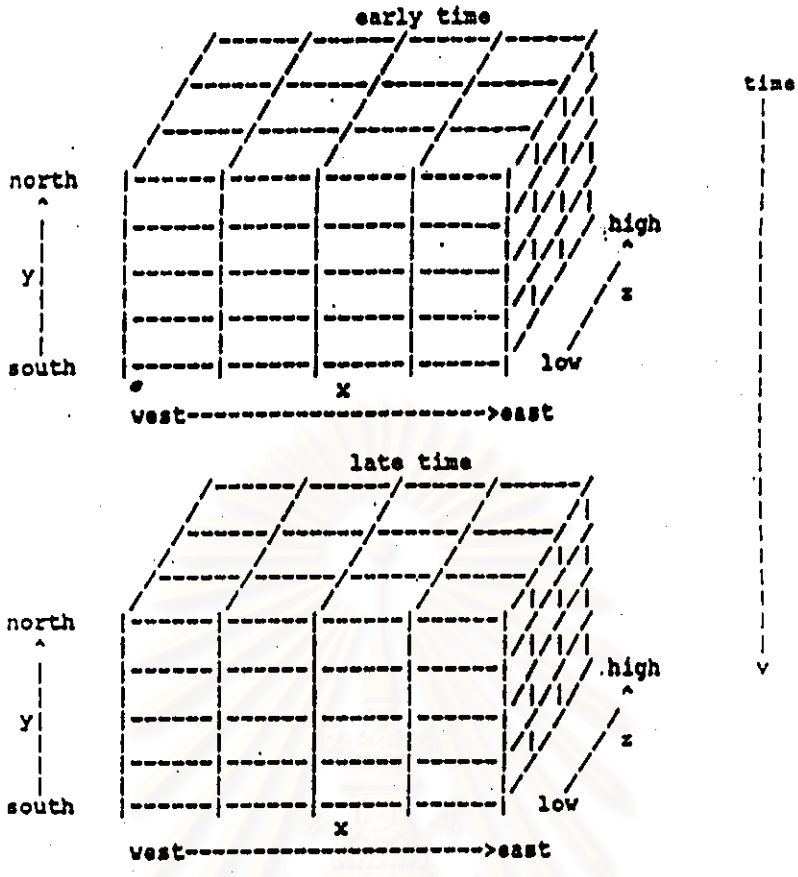

Figure 4.1
Discretised control volume

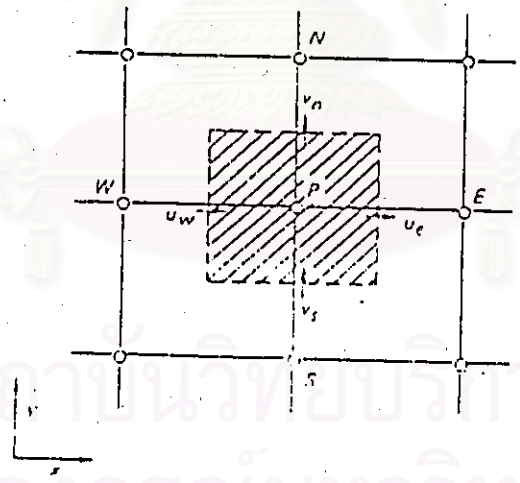

Figure 4.2
Staggered grid arrangement

The cylindrical coordinate require grid transformation. In this CFD code grid transformation between cylindrical and Cartesian coordinate (conceptually the same as method in Appendix A1) is available. User can specify cylindrical grid geometry in to the code which automatically transform the geometry and information to Cartesian coordinate for ease of computation.

A discretized control volume in cylindrical coordinate is shown in Figure 4.3. The notation in the Figure is analogous to Figure 4.3 which will be discussed later.

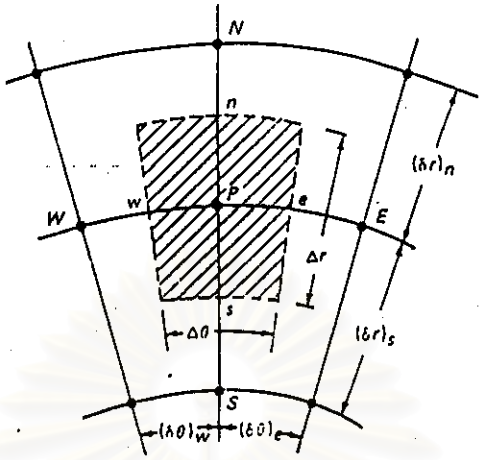

Figure 4.3
Discretized control volume in cylindrical coordinate

4.3 Finite volume equation

In this section the formulation of discrete finite volume equation for IPSA is discussed. Consider a general form of differential conservation equation :

$$\frac{\partial}{\partial t}\left(\rho_i\gamma_i\varphi_i\right)+\nabla.\left(\rho_i\gamma_i v_i\varphi_i\right)=\nabla.\left(T_\varphi\gamma_i\nabla\varphi_i\right)+\Phi_i \qquad (4.1)$$

φ_i stand for any conserved property for phase i.T_φ stand for diffusivity of φ . Integration of equation (4.1) over control volume V yield:

$$\iiint_{\mathcal{V}}\left[\frac{\partial}{\partial t}\left(\rho_i\gamma_i\varphi_i\right)+\nabla.\left(\rho_i\gamma_i v_i\varphi_i-T_\varphi\gamma_i\nabla\varphi_i\right)-\Phi_i\right]d\mathcal{V}=0 \qquad (4.2)$$

From divergence theorem

$$\iiint_{\mathcal{V}}\nabla.\left(\rho_i\gamma_i v_i\varphi_i-T_\varphi\gamma_i\nabla\varphi_i\right)d\mathcal{V}=\iint_{\mathcal{S}}\left(\rho_i\gamma_i v_i\varphi_i-T_\varphi\gamma_i\nabla\varphi_i\right)_n d\mathcal{S} \qquad (4.3)$$

Substitute expression (4.3) into (4.2)

$$\iiint_{\mathcal{V}}\left[\frac{\partial}{\partial t}\left(\rho_i\gamma_i\varphi_i\right)-\Phi_i\right]d\mathcal{V}+\iint_{\mathcal{S}}\left(\rho_i\gamma_i v_i\varphi_i-T_\varphi\gamma_i\nabla\varphi_i\right)_n d\mathcal{S}=0 \qquad (4.4)$$

Integral solution of equation (4.4), for steady flow, transient term is set to zero, can be written in discrete-algebraic form as follows;

The terms $\iint_S \left(\rho\gamma_i v_i \varphi_i - T_\varphi \gamma_i \nabla\varphi_i\right)_n dS$, integration of convection-diffusion terms, yield sum for all faces of face area multiply with area and time average for the face of outward component of convection-diffusion term which can be written as ;

$$\iint_S \left(\rho\gamma_i v_i \varphi_i - T_\varphi \gamma_i \nabla\varphi_i\right)_n dS = \sum_A^{All\,A} A_k \left(\rho\gamma_i v_i \varphi_i - T_\varphi \gamma_i \nabla\varphi_i\right)_k^+ \qquad (4.5)$$

The discrete integration of other source term yield total source term in control volume

$$\forall \phi_i = \Phi_i \qquad (4.6)$$

Substitute Expression (4.5), (4.6) in (4.4), yield Discretized finite volume equation. For steady flow, the equation is written as;

$$\sum_A^{All\,A} A_k \left(\rho\gamma_i v_i \varphi_i - D_\varphi \gamma_i \nabla\varphi_i\right)_k^+ - \Phi_i = 0 \qquad (4.7)$$

4.4 Interpolation rules

Before finite volume equations such as equation (4.7) are solved, some interpolation rules need to be established. These interpolation rules are unnecessary for analytical equation solving which yield exact solution, on the other hand, in solving discretized equation, the piecewise solutions are given which are somewhat differ from the continuos exact solutions. In order to yield "physical realistic" piecewise solutions, interpolation rules of discretized equation are therefore required.

For ease of illustration, one dimensional Discretized control volume is shown in Figure 4.4.

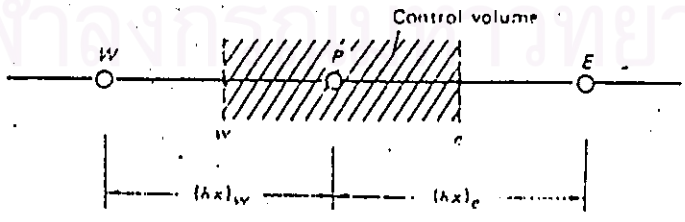

Figure 4.4
One dimensional Discretized control volume.

Interpolation rule (i)

Consider a time step or space marching step for which the given (or known) value is denoted as φ^0_P and the new (or unknown) value is denoted as φ^1_P, then :

$$\varphi^0_P = \varphi^1_P \tag{4.8}$$

This rule will be called Fully Implicit scheme, implies that new value of φ_P prevails over the entire time or space marching step for which any value depended on φ_P should be iteratively recalculated from φ^1_P.

Interpolation rule (ii)

Let $F_{i,e}=\rho_i v_{i,e}$ indicate the strength of convection and $D_{i,e}=T_e/\delta x_e$ indicate the strength of diffusion.

The Upwind scheme is defined by following assumption:

The value of φ_I at an interface is equal to the value of φ_i at the node on upwind side. Thus writing in notation given in Figure 4.4

$$\varphi_{i,e} = \varphi_{i,p} \qquad \text{if } F_{i,e} < 0 \quad (v_{i,e} \text{ is in outward direction}) \tag{4.9a}$$

$$\varphi_{i,e} = \varphi_{i,E} \qquad \text{if } F_{i,e} > 0 \quad (v_{i,e} \text{ is in inward direction}) \tag{4.9b}$$

It should be noted that no minus sign is attached to P,N, E, ...etc. for upwind scheme.

The Central difference scheme is defined by following assumption:

$$\varphi_e = \frac{1}{2}\left(\varphi_E + \varphi_P\right)$$
$$\varphi_w = \frac{1}{2}\left(\varphi_P + \varphi_W\right) \tag{4.10}$$

The central difference scheme interpolates φ_i between the interface in straight -forward manner using arithmetic mean. The above expressions are based on the assumption that face e lie midway between P and E.

Patankar (1980) suggested that each scheme yields better physical realistic solution within each range of flow parameter. The Hybrid scheme is therefore developed to provide good physical realistic solution in all range of flow parameter. Hybrid scheme characterize the combination of upwind scheme and central difference. However, it is best to consider the hybrid scheme as the three lines approximation to the exponential curve of exact solution. The hybrid scheme can be simply represented by the following criteria :

Define dimensionless Peclet number

$$P_{i,e} = \frac{F_{i,e}}{D_{i,e}} \tag{4.11}$$

Hybrid scheme is the combination of

Central difference scheme If $-2 \le P_{i,e} \le 2$ (4.12a)

Upwind scheme If $P_{i,e}<-2$ or $P_{i,e}>2$ (4.12b)

For brevity, the discretization treatment in the next section will be done on the basis of upwind scheme, since it is used as a default for IPSA, eventhough the hybrid scheme is provided as a default in single phase flow for CFD code employed.

Interpolation rule (iii)

For diffusion term in equation (4.7), consider one dimensional discretized control volume as shown in Figure 4.4, the diffusion quantity through face area e is

$$\rho_i A_e \gamma_i T_\varphi \frac{\partial \varphi}{\partial x} = \frac{\rho_i A_e \left(\gamma_i T_\varphi\right)_e \left(\varphi_p - \varphi_E\right)}{\delta x_e} \tag{4.13}$$

The question is arisen out of what value of $\gamma_{i,e}T_{\varphi,e}$ is used for the right hand side of equation (4.13). Interpolation of diffusivity term is therefore the essence of this rule. Consider the control volume in Figure 4.4, assume that the volume surrounding point P contain the material of uniform diffusivity $T_{\varphi,P}$ and for point E of uniform diffusivity $T_{\varphi,E}$, keep in mind that diffusion flux is constant and face e lie midway between P and E. Diffusion flux balance of composite slab between E to e and e to P yield diffusivity at interface $T_{\varphi,e}$:

$$\frac{1}{\gamma_{i,e} T_{\varphi_i,e}} = 0.5\left(\frac{1}{\gamma_{i,P} T_{\varphi_i,P}} + \frac{1}{\gamma_{i,E} T_{\varphi_i,E}}\right)$$

$$\text{or} \quad \gamma_{i,e} T_{\varphi_i,e} = 2\frac{\left(\gamma_{i,P} T_{\varphi_i,P}\right)\left(\gamma_{i,E} T_{\varphi_i,E}\right)}{\left(\gamma_{i,P} T_{\varphi_i,P} + \gamma_{i,E} T_{\varphi_i,E}\right)} \tag{4.14}$$

Equation (4.14) show that $T_{\varphi i,e}$ is the "harmonic mean" of $T_{\varphi i,P}$ and $T_{\varphi i,E}$.

Interpolation rule (iv)

The source terms Φ_i in equation (4.7) must be linear function. If Φ_i are nonlinear function of φ_i, source term linearization is strictly required. In this CFD code, the linearized expression of φ-dependent source term is written as:

$$\Phi_{i,P} = -a_s\left(\varphi_{i,P} - \varphi_{rf}\right) \tag{4.15}$$

Where a_s is source term coefficient and φ_{rf} is reference value of φ_i.

These four interpolation rules are applied to conserve scalar property at interface $\varphi_{i,e}$. The next two interpolation rules are in order for velocity equation based upon displaced control volumes arisen out of staggered grid for which the velocities are located at cell interface.

Interpolation rule (v)

For displaced control volume, it is preferable to take arithmetic mean for convection terms ($\rho_i\gamma_i v_i$) between two opposite cell face to obtain connection term at the grid node P,E,...etc.. As a result:

$$\left(\rho_i\gamma_i v_i\right)_P = \frac{\left(\rho_i\gamma_i v_i\right)_e + \left(\rho_i\gamma_i v_i\right)_w}{2} \tag{4.16}$$

Interpolation rule (vi)

The pressure gradient term should be obtained from the difference of the pressures between two grid node which lie on cell face of displaced control volume.

4.5 Final form of discretized finite volume equation

In this section, final form of Discretized finite volume equation of various conserved properties are described. The word "final" form implies that the general discretized equation such as equation (4.7) is manipulated by application of appropriate interpolation rules. These final form equations are prepared by CFD code at the commencement of solution procedure.

4.5.1 Scalar equation

Application of the i-iv interpolation rules to equation (4.5) result in final form of discretized scalar property equation. The most important scalar property is enthalpy which is out of the scope of this study. However, the enthalpy equation has to be derived first in a generic form to be used as a basis for next equation derivation.

In order the equation can be written in compact form, some notations are introduced as follows:

((f(x))) stand for logical expression that ((f(x))) equal f(x) if f(x) is positive and equal to zero if f(x) is negative.

Thus, convection-diffusion coefficient :

$$a_N = \left\{\left(\left(-\rho_{i,N}\gamma_{i,N}v_{i,n}\right)\right) + \frac{\gamma_{i,n}T_{i,n}}{\delta x_n}\right\}A_n \qquad (4.17a)$$

and correspondingly

$$b_N = \left\{\left(\left(\rho_{i,P}\gamma_{i,P}v_{i,n}\right)\right) + \frac{\gamma_{i,n}T_{i,n}}{\delta x_n}\right\}A_n \qquad (4.17b)$$

The above expressions imply that the convection part of b_N is zero when a_N is finite, and vice versa. Thus, the upwind scheme is satisfied. For source term, coefficient can be directly applied from equation (4.15). The final form of scalar equation is derived by applying interpolation rule i-iv and expression (4.17a and b) to equation 4.7 resulting :

$$\varphi_{i,P} = \frac{a_N\varphi_{i,N} + a_S\varphi_{i,S} + a_E\varphi_{i,E} + a_W\varphi_{i,W} + a_\Phi\varphi_{rf}}{b_N + b_s + b_E + b_W + a_\Phi} \qquad (4.18)$$

4.5.2 Volume fraction equation

Final form of γ_i has a similar form to that of φ_i equation except that a's and b's in equation (4.17) must be defined without γ_i's. As a result :

$$a_{\gamma,N} = \left\{\left(\left(-\rho_{i,N}v_{i,n}\right)\right) + \frac{T_{i,n}}{\delta x_n}\right\}A_n \qquad (4.19a)$$

$$b_{\gamma,N} = \left\{\left(\left(\rho_{i,P}v_{i,n}\right)\right) + \frac{T_{i,n}}{\delta x_n}\right\}A_n \qquad (4.19b)$$

Thus, final discretized γ_i equation, written on the same form as equation (4.18)

$$\gamma_{i,P} = \frac{a_{\gamma,N}\gamma_{i,N} + a_{\gamma,S}\gamma_{i,S} + a_{\gamma,E}\gamma_{i,E} + a_{\gamma,W}\gamma_{i,W} + a_{\gamma,\Phi}\gamma_{rf}}{b_{\gamma,N} + b_{\gamma,s} + b_{\gamma,E} + b_{\gamma,W} + a_{\gamma,\Phi}} \qquad (4.20)$$

It should be noted that $\gamma_{i,p}$ for each phase is solved separately, so that non physical values of $\gamma_{i,P}$ can be generated , for example, $\gamma_{i,P}$ exceed to unity or become negative or sum for all phases of $\gamma_{i,P}$ may not equal to unity. Therefore, some "trap" are required during solution procedure in order to force computed value into constraint.

4.5.3 Velocity equation

Final form of discretized velocity equation take a similar form to that equation (4.18) but is written on displaced control volume. Especial interest attached to the roles of the pressure and interphase momentum transfer terms which form part of momentum source term but need special treatment.

A displaced control volume on staggered grid for x- and y- velocity are represented in Figure 4.5 . The displaced control volume faces now lie on the grid line where grid nodes of main control volume are located. This layout realized one of the main advantage of staggered grid : pressure gradient p_P-p_E can directly represent for x-direction momentum driving force for displaced control volume considered. The other term which need special treatment is interphase momentum transfer source term which is written as:

$$\Phi_{i,\text{interphase}} = \sum_{j\neq i}^{\text{all j}} \xi_{ji}\left(v_j - v_i\right) \tag{4.21}$$

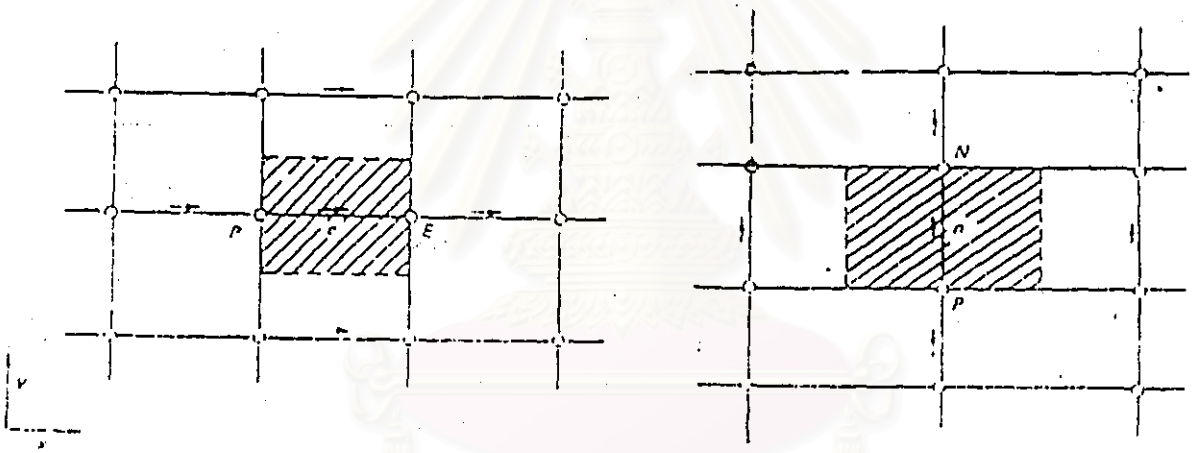

Figure 4.5
Displaced control volume for x-velocity (e-w) and y-velocity (s-n)

Interphase momentum transfer ξ_{ij},, for example, the relaxation time in momentum equation, is a function of $|v_j - v_i|$. This make the multiphase velocity equation tightly couple. The arithmetic mean interpolation of phase bulk density between grid nodes is employed to determine ξ_{ij} at the e's location.

The convection and diffusion terms for velocity equation is somewhat different from those of scalar equation as it depend on direction of the flow. For example, only the convection terms of the left and right neighbor velocities of x-direction (e-w) velocities shown in Figure 4.5 transfer into/out of the displaced control volume, and only shear stress terms due to viscous diffusion of upper and lower neighbor transfer in to/ out of the displaced control volume. The interpolation rule (v) is applied in order to determine the convection term at the grid node face area. Final form of x-direction velocity equation is then written as :

$$v_{i,e} = \frac{\left(\sum_{nb} a_{nb} v_{i,nb}\right) + \Phi_{i,e} + \gamma_{i,e} A_e \left(p_P - p_E\right) + \sum_{j \neq i}^{All\,j} \xi_{ij} v_{j,e}}{\sum_{b} b_e + \sum_{j \neq i}^{All\,j} \xi_{ij}} \tag{4.22}$$

Coefficient a_{nb} and b_e can be expressed as follows;

If $v_{i,nb}$ is on the same grid line as $v_{i,e}$ in x-direction

$$a_{nb} = \left(\left(\gamma_i \rho_i v_i\right)\right)_{nb}$$

and correspondingly (4.23)

$$b_e = \left(\left(-\gamma_i \rho_i v_i\right)\right)_e$$

If $v_{i,nb}$ is on the same grid line as $v_{i,e}$ in y-direction

$$a_{nb} = \left(\left(-\frac{\gamma_i \rho_i \mu}{\Delta y}\right)\right)_{nb}$$

and correspondingly (4.24)

$$b_e = \left(\left(\frac{\gamma_i \rho_i \mu}{\Delta y}\right)\right)_e$$

The momentum source term considered in this study is gravity term which also require arithmetic mean interpolation between grid node value.

4.5.4 Overall continuity equation

Final form of discretized overall continuity equation start from the definition that summation of all phase mass diffusion terms equal to zero and net mass source in control volume must be zero . Therefore, sum for all phase of (sum for all faces of (face area multiply with outward normal component of $\gamma_i \rho_i v_i$)) equal to zero. Writing in final form of discretized equation :

$$\sum_{i=1}^{i=n} \left(\beta_{i,n} v_{i,n} + \beta_{i,s} v_{i,s} + \beta_{i,e} v_{i,e} + \beta_{i,w} v_{i,w}\right) = 0 \tag{4.25}$$

Here the definition of $\beta_{i,n}$...can be deduced from above descriptive expression.

4.5.5 Boundary condition

There are many ways to introduce boundary value to flow field equation. The direct method is to store value in the boundary grid node which lie upon the boundary line.

the half control volume annex to boundary line is therefore create for which its face lie mid way between boundary grid node and adjacent flow field grid node. The other method as treated in this CFD codes, treat a boundary value as source term of variable in question. Boundary value is therefore not inserted directly, on the other hand, the boundary value is inserted as a source term at the cell node instead at the boundary line. Of course, near-boundary cells must be made small enough for the shift of location to be unimportant. Boundary value is computed by the same equation form as of source term equation (4.15), except that $\varphi_{S,}$, the source term value in cell, become $\varphi_{boundary}$ and the coefficient belong to boundary condition.

4.6 Solution procedure

Once the final form of discretized finite volume equations were derived in Section 4.5, its solution procedure are now discussed. The scope of discussion is confined to which will be directly applied to this study. The major difficulty of multiphase flow equation solving is arisen out of its non-linearity and interphase coupling. Therefore, the solution procedure must be performed by iterative ie. "guess and adjustment" means. In this section IPSA solution procedure is discussed to illustrate the solution procedure used in this study.

4.6.1 Solution technique for two phase pipe flow

Most of the high Reynolds number pipe flow problem without obstacle can be classified as parabolic flow problem as the influence of downstream condition to upstream is very small. But if back flows occur so that the influence of downstream condition to upstream is large, the flow is classified as elliptic flow problem.

In this CFD code, slabwise parabolic is provided for which each slab (along pipe axis) variables are calculated for many iteration. After the slabwise iterations are completed, the next adjacent slab variables along the axial direction are calculated. This "marching integration" is carried on until reaching outlet boundary. The slabwise operation, conducted in sequence from lowest slab to highest will be called "sweep". Parabolic solver is a "single sweep" operation. Since each slab is visited only one time in calculation, it is usually necessary to conduct several slabwise iteration sufficiently in order to reduce the imbalances in the equation.

Elliptic pipe flow solver, require no limited region or direction of marching solution. The entire domain of flow field are solved "whole field" in each sweep. Because the equation for values at one slab ordinarily make reference to values at next-higher slab, later adjustment made at the higher adjustment will invalidate, to some extent, the adjustment which have just been made at the lower one. For this reason many sweeps must be needed, until further adjustments are unnecessary.

In this study, parabolic-elliptic mixed solver is employed. The solution procedure start with a sweep of parabolic computation and then repeated for many sweep similar to elliptic computation. This mixed solver has been proven for its ability to accelerate convergence for the flow in which the flow direction is entirely

positive from lowest slab to highest, and the cell is long and thin along the flow direction.

During each slabwise or sweep computation, the equations are solved by certain computation technique. ADI (Alternated Direction Implicit) technique is employed in this study. ADI technique solve the two dimensional discretized finite volume equation using Thomas' algorithm (TDMA) as described in Appendix A2. TDMA is capable for solving linear one dimensional finite volume equation since

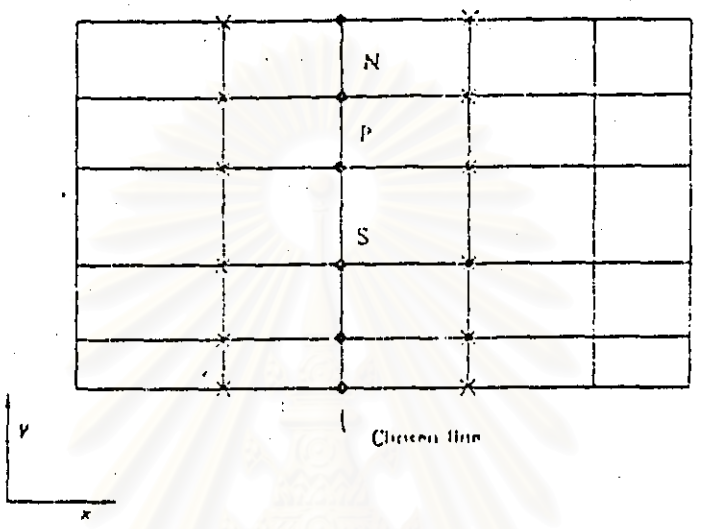

Figure 4.6
Representation for line-by-line, ADI method

only grid node variable and the other two neighbors are allowed to form tridiagonal metrix (for example $(\varphi_E, \varphi_P, \varphi_W) \equiv (u_1, u_2, u_3)$, compare with notation in Appendix 2). The first step of ADI is to solve the variable in one spatial coordinate (i.e. x-direction (e-w)) at the fixed values of the other (y-direction (s-n)). Once a grid line is chosen, the neighboring line values are assumed to be its latest (or guessed) values, as shown in Figure 4.6 (dotted for chosen node ,crosses for neighboring node). Assume that there are N grid nodes along x-direction lying on grid line y. TDMA will compute value on all N grid nodes simultaneously once. After completion, N grid nodes along x direction on the next y-direction grid line, say, y+1, are then computed . If there are M grids along y-direction, M times of TDMA are required. The alternating direction of computation will adjust all values within domain to satisfy its boundary conditions.

A special aspect of multiphase flow is that the interaction coefficient between the phases is often large. The solution procedure designed for single phase flow only could lead very slow convergence. The remedy for any tightly coupled pair of equation caused by interphase interaction is provided in this CFD code called PEA (Partial Elimination Algorithm). This algorithm form part in IPSA to manipulate pair of coupled equations such as temperature, velocity in to the form of which the interaction terms are eliminated. The detail of algorithm is described in Appendix A3.

As the computation technique necessitate for solution procedure has been outlined, the solution procedure of each variable equation will be described in the next section.

4.6.2 Solution procedure for scalar property equation

In this study, enthalpy equation is not included. In the presence of enthalpy equation, it will be solved first by PEA technique because of the possibility of tight linkage between the enthalpies of the various phase. After PEA eliminate the tight interphase linkage, the equation can then be solved by conventional ADI scheme. For the other scalar property which is influenced by velocity such as turbulence kinetic energy and its dissipation rate, PEA is not necessary for such scalar quantity since PEA is applied for velocity equation and interphase coupling is almost eliminated.

4.6.3 Solution procedure for volume fraction equation

Equation of the form (4.20) can be solved in straight-forward manner. There is one such equation for every phases and for every grid nodes but , as a rule, it is preferable to solve all but one of the phases; the γ_i's of the remaining phase is then deduced from the knowledge that all the γ_i's must sum to unity at every nodes. However, γ_i's are calculated independently so that "sum to unity" constraint may be violated cause divergence of solution. In IPSA, a trap is provided to force all γ_i's into "sum to unity" constraint. This "trap" is called BETTER equation.

Equation (4.20) can be written in more compact form:

$$\gamma_{i,P} = \frac{\sum\left(a_N \gamma_{i,N}\right) + \dot{m}}{\sum b_N} \tag{4.26}$$

For overall volume fraction

$$\sum_{i=1}^{all\,i} \frac{\sum\left(a_N \gamma_{i,N}\right) + \dot{m}}{\sum b_N} = 1 \tag{4.27}$$

If the above equation set are solved in straight-forward manner, divergence is likely possible. The BETTER equation is then formulated by introduce a guessed correction factor f_i which will equal to unity at the final solution :

$$f = \frac{\left[\dot{m} + \frac{\sum \overline{\gamma}_N}{a_N}\right]^1}{\overline{\gamma}_{i,P} b_P} \tag{4.28}$$

The $\overline{\gamma}$ terms in equation (4.28) denotes a guessed (or previous cycle) volume fraction

Finally, BETTER equation is written as

$$\gamma_{i,P} = \frac{\bar{\gamma}_{i,P} f}{\sum_{i=1}^{all\,i} \left(\bar{\gamma}_{i,P} f\right)} \tag{4.29}$$

The BETTER equation is developed from experience, not from mathematical logic. This equation accelerate convergence by way that if $f_i>1$, $\gamma_{i,P}$ will be larger than $\bar{\gamma}_{i,P}$ cause the computation move to the right direction. $f_i = 1$ in final solution.

4.6.4 Solution procedure for velocity equation

In order to solve the velocity equation, it is necessary to ascribe value to pressure . the most straight forward of which is to use those pressure value which are already in store and which result from the pressure correction operation conducted at the end of previous cycle.

The velocity equation (4.22) , if solved in straight forward manner, yield a slow converge solution or sometime, diverge solution due to interphase coupling effect. PEA technique in appendix A3 is therefore necessary. The final form after applying PEA technique , yield an equation for $v_{i,e}$ whilst $v_{j,e}$ are eliminated as shown in equation (A3.5). Writing in the notation of equation (4.22) resulting in:

$$v_{i,e} = \frac{\left[\left(\sum_{i,nb}\left(a_{nb}v_{i,nb}\right) + \Phi_{i,e} + \gamma_i A_e\left(p_P - p_E\right)\right) + \frac{\left(\sum_{j=1}^{All\,i}\sum_{i,nb}\left(a_{nb}v_{i,nb}\right) + \sum_{j=1}^{All\,i}\Phi_{i,e} + A_e\left(p_P - p_E\right)\right)\xi_{ij}}{\sum_{j\neq i}^{All\,i}\sum_{i,e} b_e}\right]}{\left[\sum_{i,e} b_e + \frac{\left(\sum_{j=1}^{All\,i}\sum_{i,e} b_{i,e}\right)\xi_{ij}}{\sum_{j\neq i}^{All\,i}\sum_{i,e} b_e}\right]} \tag{4.30}$$

Consider equation (4.30) if ξ_{ij} is very large (for example, in solid phase equation of dilute gas solid flow). the equation is reduced to

$$v_{i,e} = \frac{\left(\sum_{i=1}^{All\,i}\sum_{i,nb}\left(a_{nb}v_{i,nb}\right)_i\right) + \sum_{i=1}^{All\,i}\Phi_{i,e} + A_e\left(p_P - p_E\right)}{\left(\sum_{i=1}^{All\,i}\sum_{i,e} b_{i,e}\right)} \tag{4.31}$$

If ξ_{ij} is very small, the velocity equation is approximately in the same form as equation (4.22) but dropping out interphase momentum transfer term.

For ease of derivation, velocity equation (4.31) after PEA operation can be written in generic form :

$$v_{i,e} = \frac{\eta_i + \zeta_i(p_P - p_E)}{\omega_i} \tag{4.32}$$

4.6.5 Velocity and pressure correction

The velocity equations can be solved only when the pressure field is given or is somehow estimated. Unless the correction field is employed, the resulting velocity field will not satisfy the continuity equation. Such an imperfect velocity field based on a guessed pressure field p^* will be denoted by $v^*_{i,e}, v^*_{i,n}$. This starred velocity field will result from the equation writing in the form of (4.32):

$$v^*_{i,e} = \frac{\eta^*_i + \zeta_i(p^*_P - p^*_E)}{\omega_i} \tag{4.33}$$

Let propose that the correction pressure p' is obtained from

$$p = p^* + p' \quad ; \quad v_{i,e} = v^*_{i,e} + v'_{i,e} \tag{4.34}$$

Subtract equation (4.33) from (4.32)

$$v'_{i,e} = \frac{\left(\sum_{i=1}^{All\,i}\sum_{i,nb} a_{nb} v'_{i,nb}\right) + \zeta_i(p'_P - p'_E)}{\omega_i} \tag{4.35}$$

IPSA is an extension of semi-implicit type of operation in which the term $\left(\sum_{i=1}^{All\,i}\sum_{i,nb} a_{nb} v'_{i,nb}\right)$ is dropped out. As a result ;

$$v'_{i,e} = \frac{\zeta_i(p'_P - p'_E)}{\omega_i} \tag{4.36}$$

Equation (4.36) will be called the velocity equation formula which can be also written as:

$$v_{i,e} = v^*_{i,e} + \frac{\zeta_i(p'_P - p'_E)}{\omega_i} \tag{4.37}$$

The remaining task is to obtain a discretization equation for p′ which will be called pressure correction equation. The velocity fields which have been generated by the solution of the momentum equation (4.33) based on an incorrect pressure field cannot satisfy overall continuity equation (4.25): when combine with the in-store values of the γ_i's which are appropriate to the pressure, there is found to be a net accumulation, or loss, of mass from each cell. IPSA is developed to conquer this difficulty, by replacing the zero on the right hand side of equation (4.25) by E_p, the symbol stand for mass continuity error for cell P.

It is now supposed that all the pressures in the flow field will be modified so as to change the bulk densities and the velocities, and thereby to bring about the elimination of errors. Differentiation of appropriate terms of equation (4.25) lead to

$$\sum_{i=1}^{All\,i}\left(\begin{array}{l}\beta_{i,n}\left(\frac{\partial v_{i,n}}{\partial p_N}p'_N+\frac{\partial v_{i,n}}{\partial p_P}p'_P\right)+\beta_{i,s}v_{i,s}\left(\frac{\partial v_{i,s}}{\partial p_S}p'_S+\frac{\partial v_{i,s}}{\partial p_P}p'_P\right)\\ +\beta_{i,n}\left(\frac{\partial v_{i,n}}{\partial p_N}p'_N+\frac{\partial v_{i,n}}{\partial p_P}p'_P\right)+\beta_{i,s}v_{i,s}\left(\frac{\partial v_{i,s}}{\partial p_S}p'_S+\frac{\partial v_{i,s}}{\partial p_P}p'_P\right)\end{array}\right)=-E_P \tag{4.38}$$

The differentiating coefficient can be derived from differentiate velocity equation (4.32) with respect to pressure :

$$\frac{\partial v_{i,e}}{\partial p_P}=-\frac{\partial v_{i,e}}{\partial p_E}=\frac{\zeta_i}{\omega_i} \tag{4.39}$$

Equation (4.38) will be called pressure correction equation.

4.6.5 IPSA solution procedure.

IPSA can be regarded as SIMPLE (Semi IMplicit Pressure Linked Equation) algorithm (Patankar and Spalding [1972]) supplement for solving the coupled momentum and mass conservation equation. IPSA procedure can be summarized as follows:

1) Enthalpy equation and other scalar property equations are solved first by the procedure discussed in Section 4.6.2.

2) Volume fraction equation is then solved by the procedure described in Section 4.6.3.

3) Pressure field is guessed (which lead to velocities which do not satisfy the overall continuity equation).

4) Solve for v^* velocity equation as desribed in Section 4.6.4.

5) Solve p' equation (4.38).

6) Compute p from $p^* + p' = p$, this corrected pressure will be treated as new guessed pressure in 3) on the next computation cycle.

7) Compute velocity field using velocity correction equation (4.37).

8) Repeat the next cycle from 1)

Solution is considered converge when $-E_p$ in (4.38) and all other error remaining in all equations are acceptably small.

4.6.6 Solution control device

Although solution procedure described above is invented to combat divergence, it still possibly occur when solving highly non-linear or tightly coupled equation. In order to prevent divergence or improve convergence, some factor need to be introduced into algebraic equation during solution process. For ADI scheme, it is preferable to introduce underrelaxation factor. The word "underrelaxation" implies that it modulate the computed value and cause slower convergence. There are many ways of introducing relaxation factor. However, Any computation scheme must possess this property; the final converged solution, although obtained through the use of arbitrary relaxation factor, must still satisfy the original discretization equation.