

1 of 2

**FORCE2 - A Multidimensional Flow Program  
for Gas Solids Flow  
Theory Guide**

DEC 06 1993  
OSTI

**Topical Report**

**S.W. Burge**

**May 1991**

Work Performed Under Contract No.: DE-AC21-89MC24193

*W-31109-ENG-28*

For  
U.S. Department of Energy  
Office of Fossil Energy  
Morgantown Energy Technology Center  
Morgantown, West Virginia

By  
Argonne National Laboratory  
Energy Systems Division  
Argonne, Illinois

**MASTER**

**DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED**

## **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available at (615) 576-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161; phone orders accepted at (703) 487-4650.

**FORCE2 - A Multidimensional Flow Program  
for Gas Solids Flow  
Theory Guide**

**Topical Report**

**S.W. Burge**

**Work Performed Under Contract No.: DE-AC21-89MC24193**

For  
U.S. Department of Energy  
Office of Fossil Energy  
Morgantown Energy Technology Center  
P.O. Box 880  
Morgantown, West Virginia 26507-0880

By  
Argonne National Laboratory  
Energy Systems Division  
9700 South Cass Avenue  
Argonne, Illinois 60439-4815

**May 1991**

**MASTER**

*hp*  
**DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED**

## TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1.0	INTRODUCTION -----	1
2.0	HYDRODYNAMIC MODELS -----	2
	2.1 OVERVIEW -----	2
	2.2 CONTINUITY EQUATIONS -----	3
	2.3 MOMENTUM EQUATIONS -----	3
3.0	CONSTITUTIVE RELATIONS -----	8
	3.1 OVERVIEW -----	8
	3.2 INTERPHASE DRAG -----	8
	3.3 SOLID STRESS -----	10
	3.4 EQUATION OF STATE -----	11
	3.4.1 Ideal Gas Law -----	11
	3.4.2 User Function for Gas Specific Volume -----	11
	3.4.3 User Function for Solids Density -----	12
	3.5 MEAN FREE PATH FOR PARTIAL SLIP CONDITION -----	12
4.0	FORCE2 FINITE DIFFERENCE EQUATIONS -----	14
	4.1 OVERVIEW -----	14
	4.2 CONTROL VOLUME AND EQUATION CONVENTIONS -----	14
	4.3 EQUATIONS FOR THE TRANSIENT SIMULATION -----	22
	4.3.1 Gas Void Fraction from the Solids Continuity Equation -----	22
	4.3.2 Gas Mass Residual -----	24
	4.3.3 X-Momentum Equations -----	25
	4.3.4 Y-Momentum Equations -----	33
	4.3.5 Z-Momentum Equations -----	38
	4.3.6 Pressure Correction Equation -----	42
	4.4 EQUATIONS FOR THE STEADY SIMULATION -----	43
	4.4.1 Overview -----	43
	4.4.2 Gas Void Fraction Equation -----	44
	4.4.3 X-Momentum Equation -----	50
	4.4.4 Y-Momentum Equation -----	55
	4.4.5 Z-Momentum Equation -----	58
	4.4.6 Pressure Correction Equation -----	61
	4.5 BOUNDARY CONDITIONS -----	65
	4.5.1 Inflow Boundaries -----	65
	4.5.2 Outflow Boundaries -----	66

TABLE OF CONTENTS (continued)

<u>Section</u>		<u>Page</u>
4.5.2.1	Continuitive Outflow for Transient Mode -----	66
4.5.2.2	Continuitive Outflow for Steady Mode -----	66
4.5.2.3	Constant Pressure, Transient Mode Only -----	70
4.5.3	Velocities Normal to Solid Surfaces -----	70
4.5.4	Velocities Parallel to Solid Surfaces -----	70
4.5.4.1	No Slip -----	71
4.5.4.2	Partial Slip -----	71
4.5.4.3	Full Slip or Symmetry Boundary -----	72
4.5.5	Mass Sources & Sinks -----	72
4.6	INITIAL CONDITIONS -----	72
4.7	RESIDUES AND RELAXATION OF STEADY EQUATIONS -----	73
4.7.1	Equation Residue -----	73
4.7.2	Relaxation of Steady Equations -----	74
4.7.2.1	Under-relaxation -----	74
4.7.2.2	Inertia Relaxation -----	75
4.7.3	Summary -----	76
5.0	TRANSIENT SOLUTION PROCEDURE -----	77
5.1	OVERVIEW -----	77
5.2	OVERALL TRANSIENT SOLUTION METHOD -----	78
6.0	SOLUTION PROCEDURE FOR STEADY SIMULATION -----	83
6.1	OVERVIEW -----	83
6.2	OVERALL SOLUTION PROCEDURE -----	84
7.0	STRUCTURE OF FORCE2 -----	88
7.1	OVERVIEW -----	88
7.2	PROGRAM ORGANIZATION AND FLOW -----	88
7.3	GEOMETRIC QUANTITIES -----	91
7.4	FIELD VARIABLES -----	100
7.5	VARIABLE NAMING CONVENTIONS -----	102
7.6	MODULE INTERFACING -----	103
7.6.1	Input Routine -----	104
7.6.2	Calculation Routine -----	106
7.6.3	Output Routine -----	106
7.6.4	Source-Term Calculation Routine -----	106

TABLE OF CONTENTS (continued)

<u>Section</u>		<u>Page</u>
7.7	FORCE2 SOURCE CODE FILES -----	108
7.8	UTILITY ROUTINES -----	108
7.8.1	Transport Coefficient Calculation -----	108
7.8.2	Tilde Term Calculation -----	109
7.8.3	Error Subroutine -----	109
7.8.4	Field-Variable Definition -----	110
7.8.5	Field-Variable Access -----	110
7.8.6	Page Heading -----	113
7.8.7	Input Processing -----	113
7.9	VARIABLE NAME DICTIONARY -----	114
7.10	SUBROUTINE LIST -----	114
8.0	REFERENCES -----	129
	APPENDIX A - COUPLED TRI-DIAGONAL MATRIX SOLVER -----	A-1

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
4.1	MAIN AND MOMENTUM CONTROL VOLUME ARRANGEMENT - FORCE2 THEORY -----	16
4.2	VELOCITY LOCATIONS IN THE I,J,K FINITE DIFFERENCE GRID - FORCE2 THEORY -----	17
4.3	GENERAL CONTROL VOLUME ARRANGEMENT - FORCE2 THEORY -----	18
4.4	OUTFLOW PARALLEL TO THE X-DIRECTION - FORCE2 THEORY -----	67
4.5	FLOW IN THE Y-DIRECTION ADJACENT TO A SOLID BOUNDARY - FORCE2 THEORY -----	67
7.1	OVERALL FORCE2 EXECUTION FLOW - FORCE2 THEORY -----	89
7.2	FORCE2 INPUT PHASE - FORCE2 THEORY -----	90
7.3	CALCULATION PHASE FOR A STEADY SIMULATION IN FORCE2 - FORCE2 THEORY -----	92
7.3(A)	EXECUTION FLOW IN ROUTINE CNTRL2 - FORCE2 THEORY -----	92
7.3(B)	EXECUTION FLOW IN ROUTINE FLOW2 - FORCE2 THEORY -----	92
7.3(C)	EXECUTION FLOW IN ROUTINE MODSMP - FORCE2 THEORY -----	93
7.4	CALCULATION PHASE FOR A TRANSIENT SIMULATION IN FORCE2 - FORCE2 THEORY -----	94

## LIST OF FIGURES (continued)

<u>Figure</u>		<u>Page</u>
7.4(A)	EXECUTION FLOW IN ROUTINE CNTRLT - FORCE2 THEORY -----	94
7.4(B)	EXECUTION FLOW IN ROUTINE FLOW2T - FORCE2 THEORY -----	95
7.4(C)	EXECUTION FLOW IN ROUTINE USTART - FORCE2 THEORY -----	96
7.5	FINAL OUTPUT PHASE OF FORCE2 - FORCE2 THEORY -----	97
7.6	FORCE2 GRID SYSTEM - FORCE2 THEORY -----	98
7.7	GEOMETRIC QUANTITIES IN FORCE2 - FORCE2 THEORY -----	99
7.8	INTERPOLATION FACTORS IN FORCE2 - FORCE2 THEORY -----	99
7.9	MAIN CONTROL VOLUME INTERPOLATION IN FORCE2 - FORCE2 THEORY	99
7.10	INPUT PROCESSING IN FORCE2 - FORCE2 THEORY -----	105
7.11	STORAGE MANAGEMENT IN FORCE2 - FORCE2 THEORY -----	112

## LIST OF TABLES

<u>Table</u>		<u>Page</u>
3.1	TYPICAL SOLIDS ELASTIC MODULUS PARAMETERS -----	11
4.1	DEFINITION OF TERMS IN THE GENERAL TRANSPORT EQUATION -----	21
4.2	TERMS IN THE GENERAL FORCE STEADY TRANSPORT EQUATION -----	74
7.1	FORCE2 GEOMETRIC VARIABLES -----	115
7.2	GEOMETRIC VARIABLES FOR EACH CELL IN THE FORCE2 GRID SYSTEM	115
7.3	VARIABLE NAME DICTIONARY FOR FORCE2 -----	116
7.4	SUBROUTINES IN FORCE2 -----	122

## NOMENCLATURE

<u>Symbol</u>	<u>Description</u>
A	cross sectional flow area
a	transport coefficient or coefficient in user-defined functions for gas specific volume or solids density
C	coefficient, convection part of tilde term, or compaction modulus
D	diffusion coefficient
DG	gas mass residual
DS	solids mass residual
DU	parameter in steady pressure correction equation
$d_p$	particle diameter
F	convecting flow rate
G	solids elastic modulus
$g_x, g_y, g_z$	acceleration due to gravity in the x, y, and z-directions, respectively
L	characteristic length or mean free path
M	molecular weight
M	mass source
P	pressure
Q	convecting volumetric flow rate
q	volumetric flow rate
R	equation residual
$\bar{R}$	universal gas constant in the Ideal Gas Law
$R_x, R_y, R_z$	flow resistance coefficients for flow in the x, y, and z-directions, respectively
Re	Reynolds Number
r	microscopic density ratio, $\rho_\phi / (\rho_\phi)_R$
$r'$	normalized macroscopic density, $\rho'_\phi / (\rho_\phi)_R$
S	source term

## NOMENCLATURE (continued)

<u>Symbol</u>	<u>Description</u>
T	temperature
t	time
u, v, w	velocities in the x, y, and z-directions, respectively
v	fluid velocity
v	volume
x, y, z	coordinate directions
$\beta_x, \beta_y, \beta_z$	coefficients for interphase drag in the x, y, and z-directions, respectively
$\gamma_v$	volume porosity
$\gamma_x, \gamma_y, \gamma_z$	surface permeabilities in the x, y, and z-directions, respectively.
$\Delta L$	characteristic length
$\Delta t$	timestep
$\Delta x, \Delta y, \Delta z$	control volume dimensions in the x, y, and z-directions, respectively
$\epsilon$	void fraction
$\zeta$	a parameter used to select the Hydrodynamic Model or coordinate perpendicular to surface with origin on surface
$\theta$	under-relaxation factor
$\Lambda$	characteristic velocity
$\lambda$	coefficient of bulk viscosity
$\mu$	microscopic viscosity
$\mu'$	macroscopic viscosity, $(\epsilon_\phi \cdot \mu_\phi)$
$\xi$	a parameter used to implement the upwind difference scheme and defined as 1 for a positive velocity and 0 for a negative velocity
$\rho$	microscopic density
$\rho'$	macroscopic density, $(\epsilon_\phi \cdot \rho_\phi)$

NOMENCLATURE (continued)

<u>Symbol</u>	<u>Description</u>
$\tau$	diffusion part of tilde term
$\phi$	a dependent solution variable or sphericity (also called shape factor) of solids particle
$\psi$	a dependent solution variable
$\omega$	under-relaxation parameter in the pressure correction equation for the transient solution

Subscripts

A	hydrodynamic model A
B	control volume in back of volume P or hydrodynamic model B
E	control volume east of volume P
F	control volume in front of volume P
N	Control volume north of volume P
NET	net amount
P	control volume of interest
0	a subscript used to denote quantities at the center of Main Control Volumes or velocities calculated with the Momentum Control Volumes
R	reference quantity
S	control volume south of volume P
W	control volume west of volume P
b	back face
e	east face
f	front face
g	gas phase
i, j, k	indices used to define locations in the flow domain
max	maximum value
min	minimum value

## NOMENCLATURE (continued)

<u>Symbol</u>	<u>Description</u>
n	north face
nom	nominal value
p	particle
q	a subscript used to denote quantities on the faces of a Main or Momentum Control Volume
rel	relative
s	solids phase or south face
sw	solids velocity parallel to the wall at the wall surface (solid boundary)
u	unrelaxed term
v	volumetric
w	west face
x,y,z	in the x, y, or z direction
$\beta$	relating to interphase drag
$\epsilon_u$	unrelaxed terms in the steady void equation
$\phi$	identifies phase, $\phi = g$ or $s$
$\phi_0$	identifies other phase, i.e., when $\phi=g$ , $\phi_0=s$ and when $\phi=s$ , $\phi_0=g$
$\Delta P$	pressure gradient term

### Superscripts

'	macroscopic quantity such as $\rho'$ or correction quantity
iter	steady iteration number
0	evaluated at the beginning of the timestep for a transient solution or at the beginning of the problem for a steady solution
*	trial or assumed value

## 1.0 INTRODUCTION

This report describes the theory and structure of the FORCE2 flow program. The manual describes the governing model equations, solution procedure and their implementation in the computer program.

FORCE2 is an extension of an existing B&W multidimensional, two-phase flow program. FORCE2 was developed for application to fluid beds by implementing a gas-solids modeling technology derived, in part, during a joint government - industry research program, "Erosion of FBC Heat Transfer Tubes," coordinated by Argonne National Laboratory. The development of FORCE2 was sponsored by ASEA-Babcock, an industry participant in this program.

This manual is the principal documentation for the program theory and organization. Program usage and post-processing of code predictions with the FORCE2 post-processor are described in a companion report, FORCE2 - A Multidimensional Flow Program for Fluid Beds, User's Guide. This manual is segmented into sections to facilitate its usage. In section 2.0, the mass and momentum conservation principles, the basis for the code, are presented. In section 3.0, the constitutive relations used in modeling gas-solids hydrodynamics are given. The finite-difference model equations are derived in section 4.0 and the solution procedures described in sections 5.0 and 6.0. Finally, the implementation of the model equations and solution procedure in FORCE2 is described in section 7.0.

## 2.0 HYDRODYNAMIC MODELS

### 2.1 OVERVIEW

The hydrodynamic approach to fluidization which was started by Davidson (1961) is the basis for the models implemented in FORCE2. All the solid particles with identical densities and diameters form a continuum, a particulate phase. The gas and solids phases are then treated as interpenetrating fluids in an Eulerian formulation. Conservation of mass and momentum are then applied to each phase (a total of two or more) to derive the hydrodynamic model. Both single and multiple particle phases have been simulated with this approach, Gidaspow (1986), Syamlal (1985). The current FORCE2 model considers only two phases: one gas phase and one solids phase. The capabilities of several computer codes utilizing this approach were recently reviewed by Smoot (1984) and Gidaspow (1986). Work at the Illinois Institute of Technology (IIT) using the FLUFIX computer code to model a small-scale "two dimensional" fluidized bed, Gidaspow (1986), has provided partial validation of the hydrodynamic model. Because FORCE2 contains many features (three-dimensional cartesian, variable control volume size, etc.) needed to model large beds, validation for many industrial applications is now possible.

Two hydrodynamic models, called Models A and B according to Lyczkowski (1989), have been implemented in FORCE2. They are extensions of the models developed by Lyczkowski (1989, 1990) for the FLUFIX code. The models have been extended in FORCE2 to include:

- o Three-dimensional cartesian geometries
- o Volume porosities and surface permeabilities to account for volume and surface obstructions in the flow field.

The models consist of mass and momentum conservation equations and are described in the following sections. A cartesian coordinate system ( $x, y, z$ ) is used with  $u$ ,  $v$ , and  $w$  denoting fluid velocities in the  $x$ ,  $y$ , and  $z$ -directions. This convention and others are described in the nomenclature.

## 2.2 CONTINUITY EQUATIONS

Conservation of mass in a phase is expressed by

$$\frac{\partial}{\partial t}(\rho'_\phi \gamma_v) + \frac{\partial}{\partial x}(\rho'_\phi \gamma_x u_\phi) + \frac{\partial}{\partial y}(\rho'_\phi \gamma_y v_\phi) + \frac{\partial}{\partial z}(\rho'_\phi \gamma_z w_\phi) = \gamma_v \dot{m}_\phi \quad (2.1)$$

where the subscript,  $\phi$ , denotes the phase (i.e.,  $\phi=g$  for the gas phase and  $\phi=s$  for the solids phase) and

- $\rho'$  = macroscopic density
- $\epsilon_\phi$  =  $\epsilon_\phi \cdot \rho_\phi$
- $\epsilon_g + \epsilon_s$  = 1
- $\dot{m}_\phi$  = rate of mass addition [mass/(volume\*time)] from a mass source
- $\gamma_x, \gamma_y, \gamma_z$  = surface permeabilities in the x, y, and z directions. They are geometric parameters.
- $\gamma_v$  = flow area available to the fluids/total area
- $\gamma_v$  = volume porosity
- $\gamma_v$  = volume available to the fluids/total volume

Other symbols are defined in the Nomenclature.

## 2.3 MOMENTUM EQUATIONS

Treatment of the pressure gradient term in the gas and solids phase momentum equations results in two hydrodynamic models, called Models A and B (Lyczkowski 1989). In Model A, this term is included in both phases whereas it is included only in the gas phase in Model B. As noted by Lyczkowski (1989,1978), the treatment in Model A results in an initial value problem that is ill-posed. This situation leads to a conditionally stable numerical solution. In Model B the pressure gradient term is included only in the gas phase momentum equations resulting in a well-posed problem.

The momentum equations for both models are as follows:

- Conservation of Momentum in the X-Direction -

$$\begin{aligned}
 & \frac{\partial}{\partial t} (\rho'_\phi \gamma_v u_\phi) + \\
 & \frac{\partial}{\partial x} (\rho'_\phi u_\phi \gamma_x u_\phi) + \frac{\partial}{\partial y} (\rho'_\phi u_\phi \gamma_y v_\phi) + \frac{\partial}{\partial z} (\rho'_\phi u_\phi \gamma_z w_\phi) = \\
 & - \zeta_\phi \gamma_v \epsilon_\phi \frac{\partial p}{\partial y} + \gamma_v \rho'_\phi g_x - \gamma_v R X_\phi \frac{\rho'_\phi u_\phi u_\phi}{2} \\
 & + \gamma_v \beta_x (u_{\phi 0} - u_\phi) + \gamma_v G_\phi \frac{\partial \epsilon}{\partial x} g \\
 & + \frac{\partial}{\partial x} (\gamma_x \mu'_\phi \frac{\partial u_\phi}{\partial x}) + \frac{\partial}{\partial y} (\gamma_y \mu'_\phi \frac{\partial u_\phi}{\partial y}) + \frac{\partial}{\partial z} (\gamma_z \mu'_\phi \frac{\partial u_\phi}{\partial z}) \\
 & + \frac{\partial}{\partial x} (\gamma_x \mu'_\phi \frac{\partial u_\phi}{\partial x}) + \frac{\partial}{\partial x} (\gamma_x \epsilon_\phi \lambda_\phi \operatorname{div} v_\phi) \\
 & + \frac{\partial}{\partial y} (\gamma_y \mu'_\phi \frac{\partial v_\phi}{\partial x}) + \frac{\partial}{\partial x} (\gamma_z \mu'_\phi \frac{\partial w_\phi}{\partial x}) \tag{2.2}
 \end{aligned}$$

- Conservation of Momentum in the Y-Direction

$$\begin{aligned}
 & \frac{\partial}{\partial t} (\rho'_\phi \gamma_v v_\phi) + \\
 & \frac{\partial}{\partial x} (\rho'_\phi v_\phi \gamma_x u_\phi) + \frac{\partial}{\partial y} (\rho'_\phi v_\phi \gamma_y v_\phi) + \frac{\partial}{\partial z} (\rho'_\phi v_\phi \gamma_z w_\phi) = \\
 & - \zeta_\phi \gamma_v \epsilon_\phi \frac{\partial p}{\partial y} + \gamma_v \rho'_\phi g_y - \gamma_v R Y_\phi \frac{\rho'_\phi v_\phi v_\phi}{2} \\
 & + \gamma_v \beta_y \cdot (v_{\phi 0} - v_\phi) + \gamma_v G_\phi \frac{\partial \epsilon}{\partial y} g
 \end{aligned}$$

$$\begin{aligned}
& + \frac{\partial}{\partial x} (\gamma_x \mu'_\phi \frac{\partial v_\phi}{\partial x}) + \frac{\partial}{\partial y} (\gamma_y \mu'_\phi \frac{\partial v_\phi}{\partial y}) + \frac{\partial}{\partial z} (\gamma_z \mu'_\phi \frac{\partial v_\phi}{\partial z}) \\
& + \frac{\partial}{\partial y} (\gamma_y \mu'_\phi \frac{\partial v_\phi}{\partial y}) + \frac{\partial}{\partial y} (\gamma_x \varepsilon_\phi \lambda_\phi \operatorname{div} v_\phi) \\
& + \frac{\partial}{\partial x} (\gamma_x \mu'_\phi \frac{\partial u_\phi}{\partial y}) + \frac{\partial}{\partial z} (\gamma_z \mu'_\phi \frac{\partial w_\phi}{\partial y}) \tag{2.3}
\end{aligned}$$

- Conservation of Momentum in the Z-Direction -

$$\begin{aligned}
& \frac{\partial}{\partial t} (\rho'_\phi \gamma_v w_\phi) + \\
& \frac{\partial}{\partial x} (\rho'_\phi w_\phi \gamma_x u_\phi) + \frac{\partial}{\partial y} (\rho'_\phi w_\phi \gamma_y v_\phi) + \frac{\partial}{\partial z} (\rho'_\phi w_\phi \gamma_z w_\phi) = \\
& - \zeta_\phi \gamma_v \varepsilon_\phi \frac{\partial p}{\partial z} + \gamma_v \rho'_\phi g_z - \gamma_v R Z_\phi \frac{\rho'_\phi w_\phi}{2} \\
& + \gamma_v \beta_z (w_{\phi 0} - w_\phi) + \gamma_v G_\phi \frac{\partial \varepsilon}{\partial z} \\
& + \frac{\partial}{\partial x} (\gamma_x \mu'_\phi \frac{\partial w_\phi}{\partial x}) + \frac{\partial}{\partial y} (\gamma_y \mu'_\phi \frac{\partial w_\phi}{\partial y}) + \frac{\partial}{\partial z} (\gamma_z \mu'_\phi \frac{\partial w_\phi}{\partial z}) \\
& + \frac{\partial}{\partial z} (\gamma_z \mu'_\phi \frac{\partial w_\phi}{\partial z}) + \frac{\partial}{\partial z} (\gamma_z \varepsilon_\phi \lambda_\phi \operatorname{div} v_\phi) \\
& + \frac{\partial}{\partial x} (\gamma_x \mu'_\phi \frac{\partial u_\phi}{\partial z}) + \frac{\partial}{\partial y} (\gamma_y \mu'_\phi \frac{\partial v_\phi}{\partial z}) \tag{2.4}
\end{aligned}$$

where

$$\operatorname{div} v_\phi = \frac{\partial u_\phi}{\partial x} + \frac{\partial v_\phi}{\partial y} + \frac{\partial w_\phi}{\partial z}$$

$\lambda_\phi$  = coefficient of bulk viscosity for phase  $\phi$

and the subscript  $\phi_0$  indicates the "other" phase, i.e., for  $\phi=g$ ,  $\phi_0=s$  and  $\phi=s$  and  $\phi_0=g$ .

The parameter,  $\zeta_\phi$ , is used to select the hydrodynamic model according to

<u>Model</u>	<u><math>\phi</math></u>	<u><math>\zeta_\phi</math></u>
A	g,s	1.0
B	g	$1.0/\varepsilon_g$
B	s	0.0

The solids elastic modulus,  $G_s$ , is used to calculate the normal component of the solids stress through the relation

$$G_s \cdot \partial \varepsilon_g$$

similar to what is done in solids mechanics. The primary computational function of the solids stress term is to keep the bed from compacting below the defluidized or packed bed state. Any solids stress model that accomplishes this is adequate. For the gas phase equations, the modulus,  $G_g$ , is set to 0.0.

The last seven terms in each momentum equation are the compressible formulation for the fluid stresses. All these terms are included when the above equations are solved (i.e., a transient simulation). The equations used for a steady simulation are derived from the above by deleting the temporal terms ( $\partial/\partial t$ ) and by neglecting the last four viscous terms in each equation. The resulting set is called a steady, "incompressible" formulation.

For a three-dimensional problem, the flow field is described by eight nonlinear, coupled partial differential equations for eight dependent variables: gas void fraction,  $\varepsilon_g$ ; the pressure,  $P$ ; gas velocities in the three directions,  $u_g$ ,  $v_g$ ,  $w_g$ ; and solids velocities in the three directions  $u_s$ ,  $v_s$ ,  $w_s$ . The drag coefficients,  $\beta_x$ ,  $\beta_y$ ,  $\beta_z$  and solids elastic modulus,  $G_s$ , are defined in terms of these variables using correlations in the next section.

The flow resistance coefficients,  $RX_\phi$ ,  $RY_\phi$ ,  $RZ_\phi$ , are input to the models and, similarly, could be defined with correlations.

## 3.0 CONSTITUTIVE RELATIONS

### 3.1 OVERVIEW

The hydrodynamic models described in the previous section include a coefficient for interphase drag, a solids elastic modulus for calculating the normal component of the solids stresses, and the microscopic densities of each phase. These parameters are determined as functions of the fluid velocities, pressure and void fraction using the correlations given in this section. In addition, the mean free path used to develop the partial slip boundary condition for the solids phase is defined.

### 3.2 INTERPHASE DRAG

Standard correlations are used to evaluate fluid-particle friction (drag). However, the drag coefficients,  $\beta_x$ ,  $\beta_y$ ,  $\beta_z$ , will depend on the hydrodynamic model to be used. Lyczkowski (1989) performs a simple analysis based on the work of Gidaspow (1986) to relate the coefficients for the two models according to

$$\beta_B = \beta_A / \varepsilon_g$$

where

$\beta_A$  = coefficient for Model A

$\beta_B$  = coefficient for Model B

The coefficient for Model A is then obtained from standard correlations, Gidaspow and Ettehadieh (1983), Ettehadieh et al., (1984). Below a gas void fraction of 0.8,  $\beta_A$  is given by the Ergun equation (Ergun 1952); above 0.8, the expression given by Wen and Yu (Wen and Yu 1966), as discussed by Gidaspow (1986) is used. These expressions may be summarized as follows:

For  $\epsilon_g \leq 0.8$

$$\beta_A = 150 \frac{(1 - \epsilon_g)^2 \mu_g}{\epsilon_g (d_p \phi_s)^2} + 1.75 \frac{\rho_g |v_{rel}| (1 - \epsilon)}{d_p \phi_s}$$

For  $\epsilon_g > 0.8$

$$\beta_A = \frac{3}{4} C_d \frac{|v_{rel}| \rho_g (1 - \epsilon_g)}{d_p \phi_s} \epsilon_g^{-1.7}$$

$$C_d = 24(1 + 0.15 Re^{0.687})/Re \quad \text{for } Re \leq 1000$$

and

$$C_d = 0.44 \quad \text{for } Re > 1000$$

where the Reynolds number is given by

$$Re = \frac{|v_{rel}| d_p \phi_s \rho_g \epsilon}{\mu_g}$$

The particle sphericity (also called shape factor)  $\phi_g$ , is defined as the ratio of the actual surface area of the particle to the surface area of a spherical particle of diameter  $d_p$ .

The following two options are available for evaluating the relative velocity,  $v_{rel}$ :

#### Scalar Drag Formulation

The relative velocity is taken as the difference in the magnitude of the gas and solids velocity vectors, i.e.,

$$V_{rel} = \sqrt{(u_g - u_s)^2 + (v_g - v_s)^2 + (w_g - w_s)^2}$$

This results in isentropic coefficients.

### Vector Drag Formulation

The relative velocity is evaluated as the difference in the magnitude of the gas and solids velocities in the direction of interest. For example, for the X-momentum equations, the relative velocity would be defined as

$$V_{rel} = u_g - u_s$$

This results in non-isentropic coefficients.

The drag coefficients,  $\beta_x$ ,  $\beta_y$ ,  $\beta_z$ , are then evaluated as:

<u>Coefficient</u>	<u>Model</u>	
	A	B
$\beta_x$	$\beta_{Ax}$	$\beta_{Bx}$
$\beta_y$	$\beta_{Ay}$	$\beta_{By}$
$\beta_z$	$\beta_{Az}$	$\beta_{Bz}$

### 3.3 SOLID STRESS

As noted in Section 2.3, the solids phase momentum equations contain a normal component of solids-phase stress due to particle-to-particle interactions. A general formulation for such a term would include the effects of porosity, pressure, and the displacement tensors of solids velocity, gas velocity, and relative velocity. No such formulation is currently available. This stress is physically necessary to prevent the particles from compacting to unreasonably low gas volume fractions. The stress only becomes important as the particles contact each other, i.e., below the minimum fluidization condition. Lyczkowski (1989) summarizes research that has been directed to developing suitable expression(s) to model the particle-to-particle interaction.

Based on both experimental data and fluid bed simulations with the FLU FIX code, Lyczkowski (1989) has derived a generalized solids elastic modulus coefficient,  $G(\varepsilon)$ , of the form

$$G(\varepsilon) = G_0 \cdot \text{EXP}[-C(\varepsilon - \varepsilon^*)]$$

where  $C$  is the compaction modulus and  $\varepsilon^*$  the compaction gas volume fraction. The normalizing units factor,  $G_0$ , has been taken as 1.0 Dyne/cm<sup>2</sup>. The compaction modulus and volume fractions will depend on parameters such as particle size and density and are chosen to yield reasonable predictions of bed porosity near a packed state. They are inputs to FORCE2. Typical values are given in Table 3.1.

Table 3.1  
TYPICAL SOLIDS ELASTIC MODULUS PARAMETERS

Compaction Modulus (C)	Compaction Gas Volume Fraction ( $\varepsilon^*$ )	Reference
600	.376	Lyczkowski, et al., 1986, 1987
500	.422	Gidaspow and Syamlal, 1985

### 3.4 EQUATION OF STATE

For the gas phase, density may be determined from the Ideal Gas Law or from a user-defined function for gas specific volume as a function of temperature. For the solids phase, solids density is determined as a function of temperature only using a function specified by the user. The formulations for both phases are given below.

#### 3.4.1 Ideal Gas Law

Gas density is determined by

$$P = \rho \cdot \frac{\bar{R}}{M} \cdot T$$

where

$M$  = gas molecular weight

$R$  = universal gas constant

The molecular weight and gas constant are inputs to FORCE2.

### 3.4.2 User Function for Gas Specific Volume

Gas density is determined by

$$\frac{1}{\rho} = \frac{1}{\rho_0} + \sum_{i=1}^4 a_{1i} \cdot T^i$$

where

$a_{1i}$  = user defined coefficients

### 3.4.3 User Function for Solids Density

Solids density is determined by

$$\rho = \rho_0 + \sum_{i=1}^4 a_{2i} \cdot T^i$$

where

$a_{2i}$  = user defined coefficients

## 3.5 MEAN FREE PATH FOR PARTIAL SLIP CONDITION

At solid surfaces or boundaries, three solids velocity conditions are considered: i) zero velocity (no-slip), ii) velocity equal to the velocity in the adjacent active cell (free-slip) and iii) a "wall velocity" based on kinetic theory (partial slip). The solids velocity for the partial slip condition is specified (FLU FIX 1989) according to

$$v_{sw} = L \frac{\partial v_s}{\partial \zeta}$$

where

$v_s$  = solids velocity parallel to the surface

$v_{sw}$  = solids velocity at the surface

$L$  = mean free path from continuum theory

$\zeta$  = coordinate perpendicular to surface with origin on surface

In FORCE2, the mean free path is determined by

$$L = d_p \cdot \phi_p / (6 \sqrt{2} \cdot \epsilon_s)$$

where

$d_p$  = particle diameter

$\phi_p$  = particle sphericity

$\epsilon_s$  = solids void fraction

## 4.0 FORCE2 FINITE DIFFERENCE EQUATIONS

### 4.1 OVERVIEW

The finite difference equations that are solved to predict fluid bed hydrodynamics are developed in this section. These expressions are derived by dividing the flow domain into a collection of control volumes and then integrating the governing momentum and continuity equations, presented in Section 2.0, over these volumes. This results in a set of coupled, non-linear equations describing the velocities, void fractions, and pressures within the volumes.

In Section 4.2, the control volume arrangement and conventions for the difference equations are described.

In Sections 4.3 and 4.4 the finite difference equations for the transient and steady simulations are developed. The transient simulation is an extension of the method used in the 2-D FLUFIX code (Lyczkowski and Bouillard, 1989). It was installed without modifications in FORCE2. The steady simulation, however, is based on the existing FORCE2 approach. Because the two methods differ in the treatment of the convection and diffusion terms, slightly different finite difference equations result.

In Section 4.5, the finite difference equations are modified to account for boundary conditions. In Section 4.6, initial conditions are considered and in Section 4.7, relaxation is applied to the finite difference equations for the steady simulation.

### 4.2 CONTROL VOLUME AND EQUATION CONVENTIONS

The flow (or computational) domain is divided into a collection of cells or control volumes in a Cartesian coordinate system. Scalar quantities such as pressure and void fraction are calculated at the centers of these volumes denoted Main Control Volumes. Gas and solids velocities are calculated on the Main Volume faces utilizing a second set of control volumes that are "staggered" with respect to the Main Volumes. This second set of volumes connects the centers of the Main Volumes and are called Momentum Control

Volumes. The collection of Main and Momentum Volumes is the conventional "Staggered" mesh used in most finite difference formulations (Patankar 1980).

Two schemes are used in this report to describe the flow field: i) an "i,j,k" designation and ii) a general scheme. The Main and Momentum Volumes are shown in Figure 4.1 in the domain described by the "i,j,k" indices. Centers of the Main cells are identified by the integers i,j,k. The Momentum Volumes are named X-, Y-, or Z-Momentum Volumes depending on their orientation. The flow field is further described in Figure 4.2 where the faces of the Main Volume (also the location where velocities will be calculated with the Momentum Volumes) are identified by half-integer values. For example, in Figure 4.2, location  $i+1/2, j, k$  is the upper-X face of the Main Control Volume centered at  $i, j, k$ . Alternately, a more general and compact scheme is used to identify control volumes (Main or Momentum) in the flow domain as shown in Figure 4.3. In this scheme the control volume of interest is designated by the upper case letter P. The neighboring volumes are identified as points on the compass as

E - the neighbor that is east of volume P  
W - the neighbor that is west of volume P  
N - the neighbor that is north of volume P  
S - the neighbor that is south of volume P  
F - the neighbor that is in front of volume P  
B - the neighbor that is in back of volume P

The faces of volume P are identified in a similar manner except using lower case letters, i.e., e, w,...f, and b.

The x-, y- and z-dimensions of the main control volumes are inputs to the model. Areas and volumes associated with both the main and momentum volumes are based on these dimensions and on the user-specified volume porosities and surface permeabilities.

A general transport equation (representing conservation of mass or momentum) may then be written using the general control volume arrangement as

$$a_p \phi_p = \sum a_q \phi_q + b_p \psi_p + s_c \quad 4.1$$

where  $\phi_p$  and  $\psi_p$  are dependant solution variables and

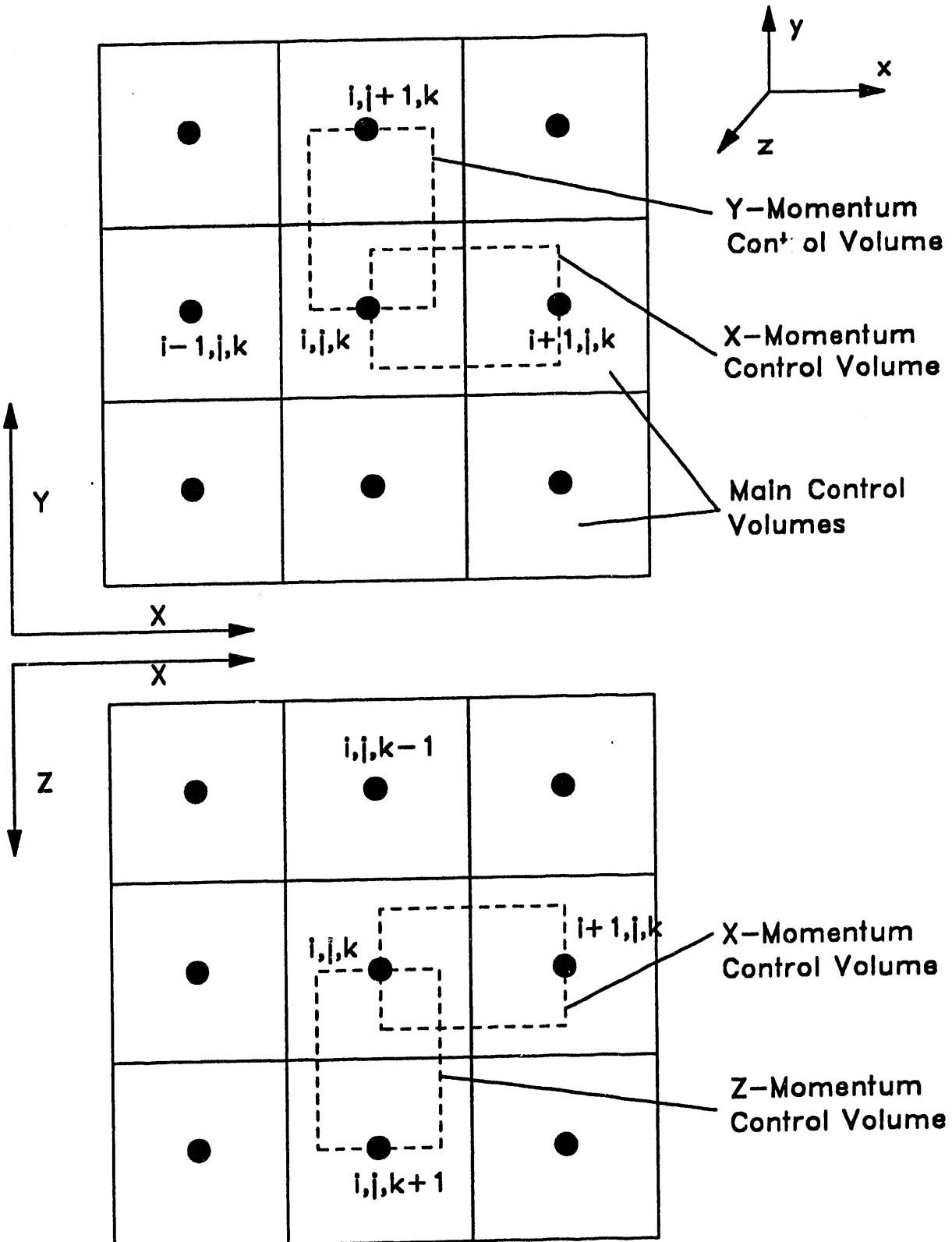


Figure 4.1. Main and momentum control volume arrangement - FORCE2 theory.

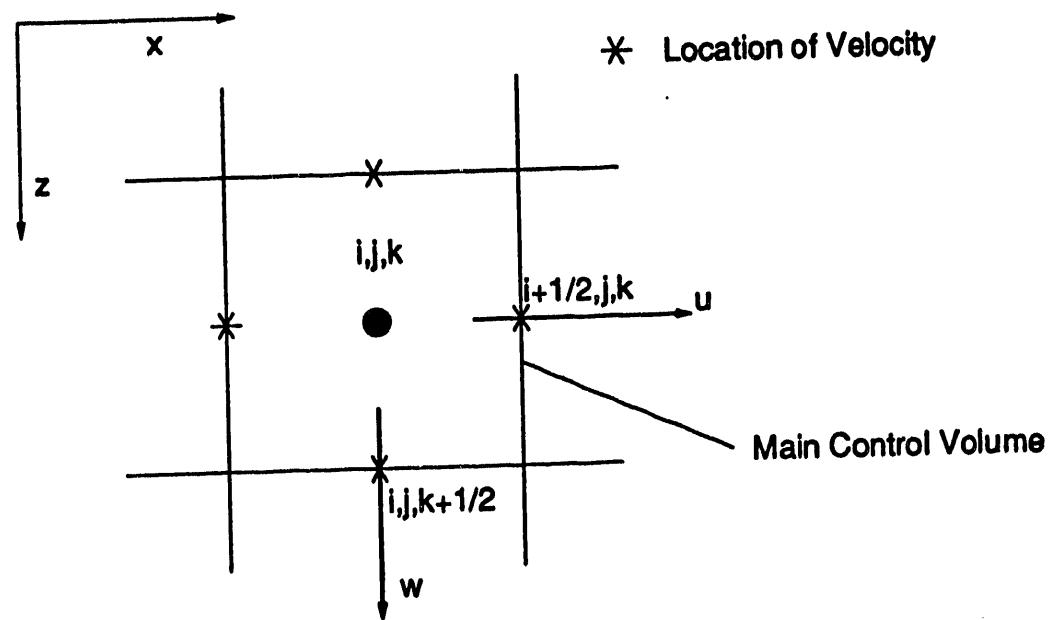
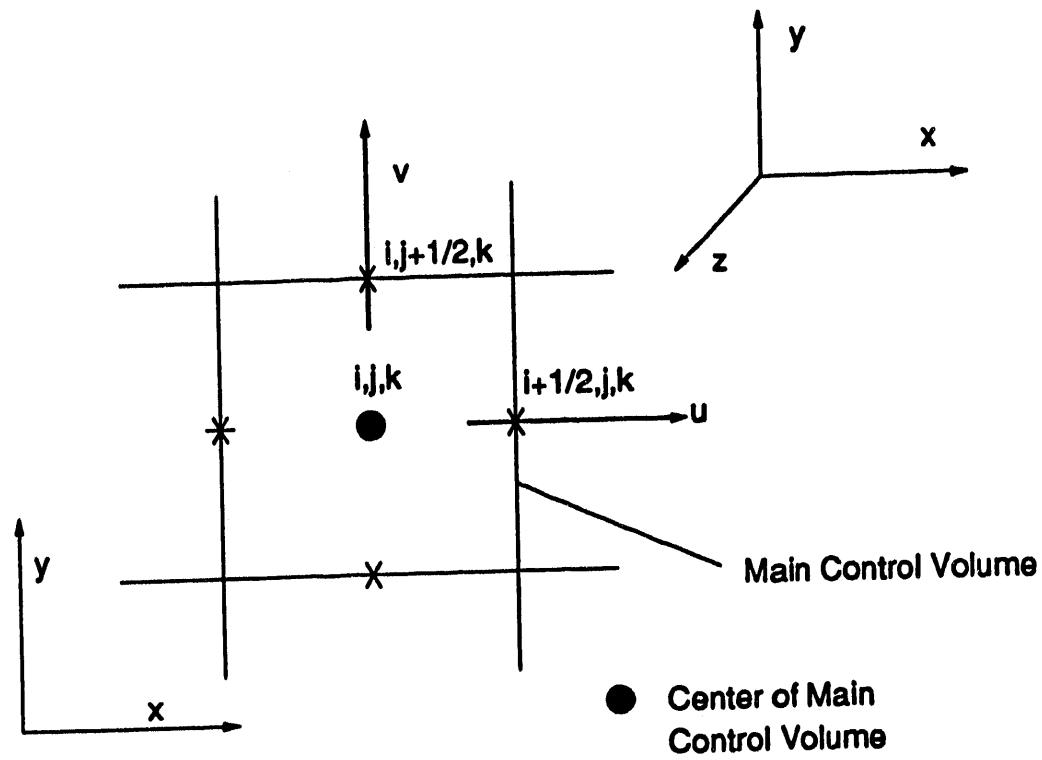


Figure 4.2. Velocity locations in the  $i, j, k$  finite difference grid - FORCE2 theory.

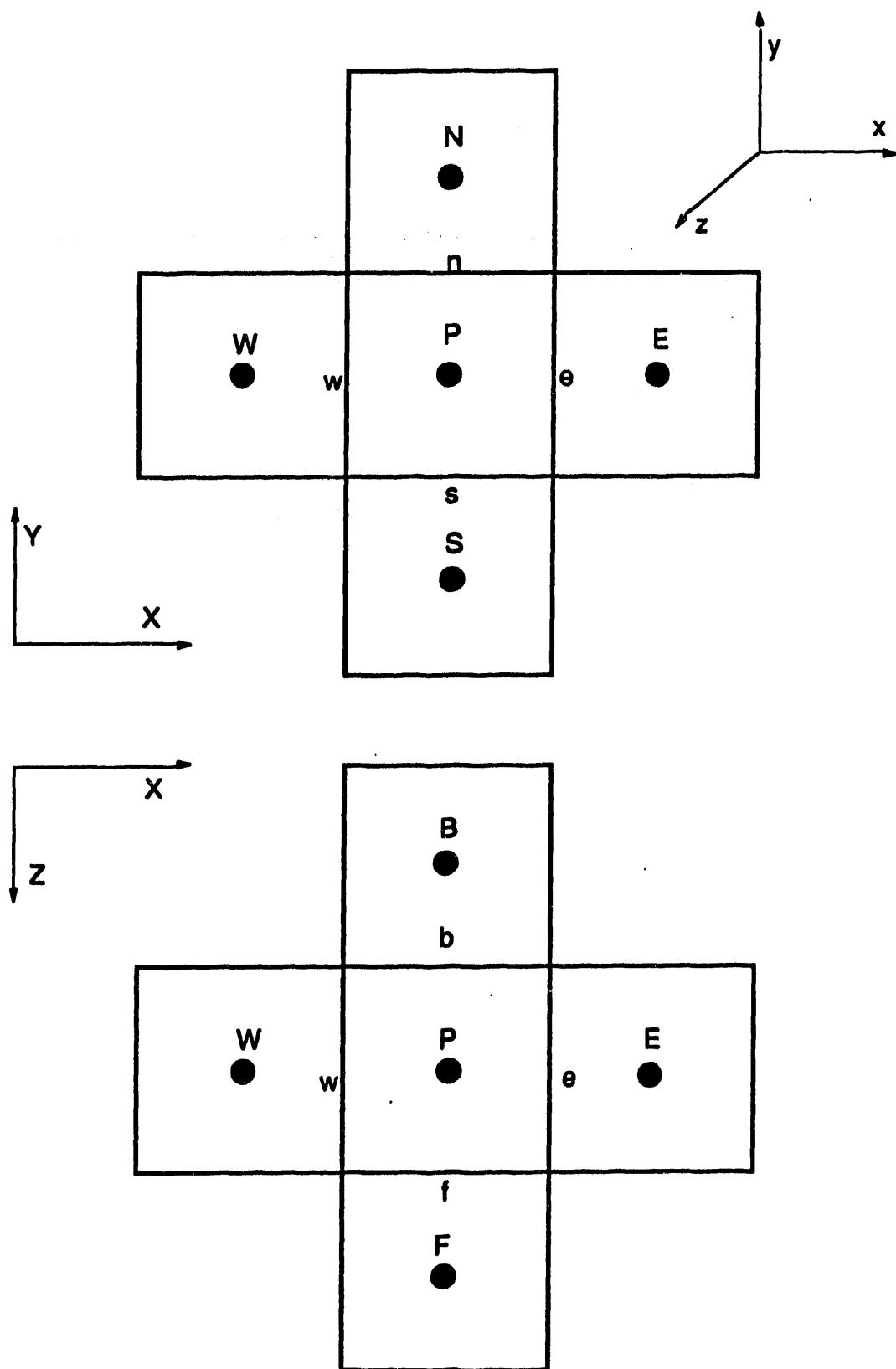


Figure 4.3. General control volume arrangement - FORCE2 theory.

$$a_q = \text{transport coefficients on face } q \text{ of control volume } P$$

$$\sum a_q \phi_0 = a_e \phi_E + a_w \phi_W + a_n \phi_N + a_s \phi_S + a_f \phi_F + a_b \phi_B$$

$\phi_N, \phi_S, \phi_E, \phi_W, \dots, \phi_B$  = The values of  $\phi$  in the cells that are North ( $\phi_N$ ), South ( $\phi_S$ ), East ( $\phi_E$ ), West ( $\phi_W$ ), Front of ( $\phi_F$ ), and Back of ( $\phi_B$ ) the considered cell with value

$$a_p = \sum a_q + b_p + S_p$$

$$S_c = \text{a source of } \phi_p$$

$$S_p = \text{a positive coefficient for a source of } \phi_p$$

$$b_p = \text{a positive coefficient linking the two dependent variables } \phi_p \text{ and } \psi_p$$

These terms are further outlined in Table 4.1 for each transport equation.

The coefficients,  $a_q$ , link neighboring variables in the mesh and are based on convection and diffusion. Finally, the control volumes in Figure 4.3 represent the Main Control volumes when equation 4.1 is a continuity equation and represent the Momentum volumes when equation 4.1 is a momentum equation. For example, if equation 4.1 represents the X-direction momentum equation, the  $\phi$ 's are as follows:

$$\phi_p = u_{i+1/2, j, k} \quad \phi_W = u_{i-1/2, j, k}$$

$$\phi_N = u_{i+1/2, j+1, k} \quad \phi_F = u_{i+1/2, j, k+1}$$

$$\phi_S = u_{i+1/2, j-1, k} \quad \phi_B = u_{i+1/2, j, k-1}$$

$$\phi_E = u_{i+3/2, j, k}$$

where  $u$  is the gas or solids velocity in the X-direction. The transport coefficients,  $a$ 's, correspond to the following:

$$a_n = a_{i+1/2, j+1/2, k} \quad a_f = a_{i+1/2, j, k+1/2}$$

$$a_s = a_{i+1/2, j-1/2, k} \quad a_b = a_{i+1/2, j, k-1/2}$$

$$a_e = a_{i+1,j,k}$$

$$a_w = a_{i,j,k}$$

Finally, a factor of 1/2 appears quite frequently in the coefficients of the finite difference equations. For example, the coefficient of the velocity in the transient momentum equation in the X-direction,  $a_{px}$  equation 4.5, includes the storage term,

$$(\overline{\rho' V}) = [(\rho'_e V)_e + (\rho'_w V)_w]/2$$

For this term, as well as many others, the factor of 1/2 arises due to approximate integration over the control volume, not due to a central difference approximation, i.e.,

$$\int_V \frac{\partial(\rho' u)}{\partial t} dV = \left[ \int_{V_e} \rho' dV + \int_{V_w} \rho' dV \right] \frac{\partial u}{\partial t}$$
$$= (\overline{\rho' V}) \cdot \frac{\partial u}{\partial t}$$

then

$$\int_{V_e} \rho' dV \approx \frac{1}{2}(\rho' V_e)$$

$$\int_{V_w} \rho' dV \approx \frac{1}{2}(\rho' V_w)$$

where  $V_e$  and  $V_w$  denote volumes of the main control volumes connected by the momentum volume, i.e.,

$$V_e = V_{i+1,j,k}$$

$$V_w = V_{i,j,k}$$

Table 4-1  
DEFINITION OF TERMS IN THE GENERAL TRANSPORT EQUATION

Transport Equation	$\Phi_P$	$\psi_P$	$b_P$	$s_c$	$s_P$
X, Y, or Z-Momentum	X, Y, or Z-gas velocity	Solids velocity in same direction as gas	Interphase drag	Pressure, body forces storage term (transient)	Distributed resistance storage term (transient)
X, Y, or Z-Momentum	X, Y, or Z-solids velocity	Gas velocity in same direction as solids	Interphase drag	Pressure, body forces solids stresses storage term (transient)	Distributed resistance storage term (transient)
Solids Continuity	Void fraction	0.0	0.0	Source/sink for solids	0.0
Gas Continuity	Void fraction	0.0	0.0	Source/sink for gas	0.0

#### 4.3 EQUATIONS FOR THE TRANSIENT SIMULATION

The governing finite difference equations for the transient simulation are developed in this section. Equations for the gas void fraction and gas mass residue are derived based on the solids and gas continuity equations respectively. The expression for the gas residue is used to check solution convergence and used to derive a pressure correction equation for the transient solution scheme, Section 5.0.

Equations for the gas and solids velocities are developed based on the momentum equations for each phase. The viscous and convection terms are included explicitly, unlike the steady formulation in which a hybrid differencing scheme (Patankar, 1980) is used to combine these terms.

##### 4.3.1 Gas Void Fraction from the Solids Continuity Equation

By approximating the time derivative as a simple difference, the solids continuity equation, Equation 2.1, is integrated over the Main Control volume depicted in Figure 4.3 to yield the following expression for the solids macroscopic density

$$\begin{aligned}
 (\epsilon_s \rho_s)_P &= (\epsilon_s \rho_s)_P^0 \\
 &\quad - \frac{\Delta t}{V_p} [(u_s A \rho'_s)_E - (u_s A \rho'_s)_W] \\
 &\quad - \frac{\Delta t}{V_p} [(v_s A \rho'_s)_N - (v_s A \rho'_s)_S] \quad 4.2 \\
 &\quad - \frac{\Delta t}{V_p} [(w_s A \rho'_s)_F - (w_s A \rho'_s)_B] \\
 &\quad + \dot{m}_s \cdot \Delta t
 \end{aligned}$$

where the subscript, P, indicates the macroscopic density at the center of the Main Control Volume and the subscripts e, w, n, .... and b indicate quantities evaluated on the East, West, .... and Back Faces of the Main Control (see Figure 4.3 in section 4.2), i.e., in the "i,j,k" domain.

$P \Rightarrow i, j, k$   
 $e \Rightarrow i + 1/2, j, k$   
 $w \Rightarrow i-1/2, j, k$   
 $n \Rightarrow i, j+1/2, k$   
 .  
 .  
 .  
 $b \Rightarrow i, j, k-1/2$

and

$E \Rightarrow i+1, j, k$   
 $W \Rightarrow i-1, j, k$   
 .  
 .  
 .  
 $B \Rightarrow i, j, k-1$

Some additional terms are defined as

$\Delta t$  = problem timestep  
 $V_p$  = volume in the Main Control Volume available for the gas and solids phases  
 $= V_v \cdot \Delta x_p \cdot \Delta y_p \cdot \Delta z_p$   
 $A$  = cross sectional flow areas on the associated face of the main control volume  
 $\Delta x_p, \Delta y_p, \Delta z_p$  = main control volume dimensions in the X, Y, and Z directions, respectively  
 $(\epsilon_s \cdot \rho_s)_p^0$  = macroscopic density at the beginning of the timestep

The macroscopic densities on the faces of the Main Volume are evaluated using an upwind approximation (Patankar 1980) as follows

$$(\rho'_s)_e = \xi_e \cdot (\rho'_s)_P + (1-\xi_e) (\rho'_s)_E$$

where

$$\xi_e = \begin{cases} 1 & (u_s)_e \geq 0 \\ 0 & (u_s)_e < 0 \end{cases}$$

and similar formulations for densities on the other faces of the Volume.

The gas void fraction is then determined using the macroscopic solids density from Equation 4.2 as follows

$$\epsilon_g = (\rho_s \epsilon_s) / \rho_s$$

and finally

$$\epsilon_g = 1 - \epsilon_s \quad 4.3$$

Some important observations include:

- o All the terms in Equation 4.2, with the exception of  $(\epsilon_s \rho_s)^0$ , are evaluated at the end of the timestep.
- o An iterative solution method is used for the transient simulation resulting in Equation 4.2 being solved several times at each Main Control Volume during a single timestep. This formulation in which macroscopic densities at node P can appear on the right-hand side is used in FLUFIX and is the default formulation in FORCE2. An optional method in which Equation 4.2 is rearranged like Equation 4.1 in which all macroscopic densities at P are collected onto the left-hand side is also available in FORCE2. Both formulations are installed in subroutine VFRAC; however, the FLUFIX form is currently used.

#### 4.3.2 Gas Mass Residual

During the solution over a timestep, nodal pressures are adjusted until gas mass is conserved. Convergence of the solution is determined by the magnitude of gas mass residue (the amount of gas that is "created" or

"destroyed" due to non-convergence) and is determined by integrating the gas continuity equation, Equation 2.1, over the Main Control Volume as follows:

$$\begin{aligned}
 \text{DG}_P &= \frac{V_p}{\Delta t} \cdot (\rho'_g - \rho'^0_g)_P \\
 &+ [(u_g^A \rho'_g)_e - (u_g^A \rho'_g)_v] \\
 &+ [(v_g^A \rho'_g)_n - (v_g^A \rho'_g)_s] \\
 &+ [(w_g^A \rho'_g)_f - (w_g^A \rho'_g)_b] \\
 &- \dot{M}_g
 \end{aligned} \tag{4.4}$$

where P indicates the center of the main volume and e, v, ... indicate parameters on the faces as described in the previous section. The convected macroscopic gas densities on the faces, i.e.,  $\rho'_{ge}$ ,  $\rho'_{gv}$ , ..., are determined using the upwind scheme shown above based on gas velocities on the cell faces and

$\text{DG}$  = gas mass residue

$\dot{M}_g$  = gas source  
 $= V_p \cdot \dot{m}_g$

$\rho'^0_g$  = macroscopic density at the beginning of the timestep

#### 4.3.3 X-Momentum Equations

The X-direction momentum equation, Equ 2.2, is integrated over an X-Momentum Volume such as that depicted in Figure 4.1 and Figure 4.2 to give

$$s_{Px}(u_\phi)_P = (\overline{\rho'_\phi u'_\phi v})_P - \zeta_\phi v_P \cdot \frac{p_e - p_w}{\Delta x_p}$$

$$\begin{aligned}
 & + (\beta_x^0 \cdot v \cdot u_{\phi 0})_p \\
 & + (\tilde{G}_{\phi} \tilde{v})_p \cdot \frac{(e_x^e - e_x^w)}{\Delta x_p} \quad \phi = g \text{ or } s
 \end{aligned} \tag{4.5}$$

where the subscripts indicate

$$p \Rightarrow i+1/2, j, k$$

$$e \Rightarrow i+1, j, k$$

$$w \Rightarrow i, j, k$$

It should be noted that  $e$  and  $w$  indicate the centers of the main control volumes that are connected and by the X-momentum volume

and

$$\begin{aligned}
 v_p & = \text{volume of the momentum volume} \\
 & = (v_e + v_w)/2 \\
 & = (v_{i+1,j,k} + v_{i,j,k})/2
 \end{aligned}$$

$$\begin{aligned}
 \Delta x_p & = \text{length of the momentum volume in the X-direction} \\
 & = (\Delta x_{i+1} + \Delta x_i)/2
 \end{aligned}$$

$$\Delta x_i, \Delta x_{i+1} = \text{lengths of the } i \text{ and } i+1 \text{ main control volumes in the X-direction, an input to the model}$$

$(\tilde{G}_{\phi} \tilde{v})$  is denoted the "tilde" term (Rivard and Torrey 1977) and contains the gravity, convection and diffusion terms. It is evaluated at the beginning of the timestep. The other terms are based on parameters evaluated at the end of the timestep unless denoted with the superscript 0, i.e.,  $(\ )^0$ . These terms are further defined below.

$(\beta_x^0)_p$  = Drag coefficient evaluated at the beginning of the timestep and at the face of the main control volume. Drag coefficients are calculated based on conditions at the center of the main volume. Cell centered velocities are taken as the average of velocities on the faces of the main volume. The drag coefficient for the momentum volume,  $(\beta_x^0)_p$ , is then found by interpolation.

$$(\beta_x^0)_p = f_1(\beta_x^0)_e + (1-f_1)(\beta_x^0)_w$$

$$\text{with } f_1 = \Delta x_w / (\Delta x_w + \Delta x_e)$$

The coefficient,  $a_{px}$ , is given by

$$a_{px} = (\bar{\rho'_V}) / \Delta t^+ (\beta_x + k_{\phi x}^0)_p \cdot v_p$$

where

$$(\bar{\rho'_V}) = [(\rho'_V)_e + (\rho'_V)_w] / 2$$

$$k_{\phi x}^0 = (R x_\phi \cdot \rho'_\phi \cdot u_\phi / 2)_p^0$$

with  $(\rho'_\phi)_p^0$  determined by interpolation as

$$(\rho'_\phi)_p^0 = f_1(\rho'_\phi)_e^0 + (1-f_1)(\rho'_\phi)_w^0$$

#### Solid Stress Term:

The solid stress term, which is present only in the solids equations, is evaluated based on gas void fraction at the end of the timestep according to

$$(\bar{G'_V}) = [(\bar{G'_V})_e + (\bar{G'_V})_w] / 2$$

where

$$G_\phi = G_\phi(\epsilon_g)$$

#### Tilde Term:

As noted above, the tilde term is evaluated only at the beginning of timestep. Consequently, in the equations below, all parameters are evaluated

at the start of the timestep: the superscript  $\circ$ ,  $(\ )^\circ$  is not included to simplify the notation. The tilde term is made up of body force, convection and diffusion components as follows:

$$(\overline{\rho'_\phi \mathbf{u}_\phi})_p = (\overline{\rho'_V})_p \cdot (\mathbf{u}_\phi)_p / \Delta t$$

$$+ B_x + C_x + \tau_x$$

The density-volume product,  $(\overline{\rho'_V})_p$ , is as defined earlier, except it is evaluated at the beginning of the timestep.

Gravity Term:  $B_x$

$$B_x = (\overline{\rho'_V})_p \cdot g_x$$

Convection Term:  $C_x$

The convection term is defined as

$$C_x = - \int_V \frac{\partial}{\partial x} (\gamma_x u_\phi \rho'_\phi u_\phi) dV$$

$$- \int_V \frac{\partial}{\partial y} (\gamma_y v_\phi \rho'_\phi u_\phi) dV - \int_V \frac{\partial}{\partial z} (\gamma_z w_\phi \rho'_\phi u_\phi) dV$$

and may be expressed as

$$C_x = - \sum Q_q \cdot (\rho'_\phi u_\phi)_Q - Q_p \cdot (\rho'_\phi u_\phi)_p$$

where

$\sum$  = summation over the faces of the X-momentum control volume according to

$$\sum Q_q (\rho'_\phi u'_\phi)_Q = Q_e (\rho'_\phi u'_\phi)_E + Q_w (\rho'_\phi u'_\phi)_W \\ + \dots Q_b (\rho'_\phi u'_\phi)_B$$

with

$$E \Rightarrow i+3/2, j, k$$

$$W \Rightarrow i-1/2, j, k$$

$$B \Rightarrow i+1/2, j, k-1$$

$$b \Rightarrow i+1/2, j, k-1/2$$

Expressions for the convecting volumetric flow rates,  $Q_q$ , are developed by considering the volumetric rates on each face and by expressing the convected momenta,  $(\rho'_\phi u'_\phi)_q$ , on each face in terms of the momenta in neighboring cells  $(\rho'_\phi u'_\phi)_Q$ , using the upwind approximation. This approach results in

$$q_e = (Au_\phi)_e$$

$$q_w = (Au_\phi)_w$$

$$q_n = (Av_\phi)_n$$

$$q_s = (Av_\phi)_s$$

$$q_f = (Aw_\phi)_f$$

$$q_b = (Aw_\phi)_b$$

where  $q_e$ ,  $q_w$ , ..., are the convecting flow rates on the faces of the X-momentum volume and

$$(Au_\phi)_e = [(Au_\phi)_p + (Au_\phi)_z]/2$$

$$(Au_\phi)_w = [(Au_\phi)_p + (Au_\phi)_n]/2$$

$$(Av_\phi)_n = [(Av_\phi)_{i+1,j+1/2,k} + (Av_\phi)_{i,j+1/2,k}]/2$$

The flow rates on the south, front and back face,  $q_s$ ,  $q_b$  and  $q_f$ , are defined similar to  $q_n$ . The convection flow rates,  $Q_q$ , are now defined as

$$Q_e = (1 - \xi_e) q_e$$

$$Q_w = - \xi_w \cdot q_w$$

$$Q_n = (1 - \xi_n) \cdot q_n$$

$$Q_s = - \xi_s \cdot q_s$$

$$Q_t = (1 - \xi_t) \cdot q_t$$

$$Q_b = - \xi_b \cdot q_b$$

$$Q_p = \xi_e q_e - (1 - \xi_w) \cdot q_w + \xi_n \cdot q_n - (1 - \xi_s) \cdot q_s + \xi_t \cdot q_t - (1 - \xi_b) \cdot q_b$$

The  $\xi$  terms arise due to the upwind approximation and are defined as

$$\xi_e = \begin{cases} 1 & (u_\phi)_e \geq 0 \\ 0 & (u_\phi)_e < 0 \end{cases}$$

$$\xi_w = \begin{cases} 1 & (u_\phi)_w \geq 0 \\ 0 & (u_\phi)_w < 0 \end{cases}$$

$$\xi_n = \begin{cases} 1 & (v_\phi)_n \geq 0 \\ 0 & (v_\phi)_n < 0 \end{cases}$$

and similarly on the other faces with

$$(u_\phi)_e = [(u_\phi)_p + (u_\phi)_s]/2$$

$$(u_\phi)_w = [(u_\phi)_p + (u_\phi)_n]/2$$

$$(v_\phi)_n = f_1 \cdot (v_\phi)_{i+1,j+1/2,k} + (1-f_1) \cdot (v_\phi)_{i,j+1/2,k}$$

The equivalent velocities on the other faces such as  $(v_\phi)_s$ ,  $(v_\phi)_t$  and  $(v_\phi)_b$ , are also determined by interpolation.

Finally, the macroscopic density in the convected momentum term is also determined by interpolation. For example,

$$(\rho'_\phi)_p = f_1 (\rho'_\phi)_e + (1-f_1) (\rho'_\phi)_w$$

$$= f_1 (\rho'_\phi)_{i+1,j,k} + (1-f_1) (\rho'_\phi)_{i,j,k}$$

Diffusion Term:  $\tau_x$

The diffusion term is made up of the last six terms in the X-momentum equation, Equation 2.2, and is expressed as

$$\tau_x = \sum D_q \cdot (u_\phi)_Q - D_p \cdot (u_\phi)_P + \tau_c$$

where

$$\sum D_q \cdot (u_\phi)_Q = D_e \cdot (u_\phi)_E + D_w \cdot (u_\phi)_W + \dots + D_b \cdot (u_\phi)_B$$

$$D_p = D_e + D_w + \dots + D_b$$

and  $\tau_c$  is as defined below. The diffusion coefficients on each face,  $D_q$ , are defined as follows.

$$D_e = \frac{2(A\mu'_\phi)_e}{\Delta x_e}$$

$$D_w = \frac{2(A\mu'_\phi)_w}{\Delta x_w}$$

$$D_n = \frac{(A\mu'_\phi)_n}{\Delta y_n}$$

• •

$$D_b = \frac{(A\mu'_\phi)_b}{\Delta z_b}$$

where

$$\mu'_\phi = \mu_\phi \cdot \varepsilon_\phi$$

$$\Delta x_e = \Delta x_{i+1}$$

$$\Delta x_w = \Delta x_i$$

$$\Delta y_n = [\Delta y_j + \Delta y_{j+1}] / 2$$

$$\Delta z_b = [\Delta z_k + \Delta z_{k-1}] / 2$$

$$(A\mu'_\phi)_e = [(A_x)_P + (A_x)_E] \cdot (\mu'_\phi)_e / 2$$

$$(A\mu'_\phi)_w = [(A_x)_P + (A_x)_W] \cdot (\mu'_\phi)_w / 2$$

$$(A\mu'_\phi)_n = [(A_y \cdot \mu'_\phi)_{i+1,j+1/2,k} + (A_y \mu'_\phi)_{i,j+1/2,k}] / 2$$

$$(A_z \mu'_\phi)_b = [(A_z \cdot \mu'_\phi)_{i+1,j,k-1/2} + (A_z \mu'_\phi)_{i,j,k-1/2}] / 2$$

with the macroscopic viscosities  $(\mu'_\phi)_{i+1,j+1/2,k}$ ,  $(\mu'_\phi)_{i,j+1/2,k}$ ,  $(\mu'_\phi)_{i+1,j,k-1/2}$ ,  $(\mu'_\phi)_{i,j,k-1/2}$  determined by interpolation. The macroscopic viscosities,  $(\mu'_\phi)_e$  and  $(\mu'_\phi)_w$  are at main control volume centers.

The last term,  $\tau_c$ , is made up of the last three terms in Equation 2.2 and is given by:

$$\tau_c = (A \lambda'_\phi \operatorname{div} V)_e - (A \lambda'_\phi \operatorname{div} V)_w$$

$$+ (A \mu'_\phi \frac{\partial v_\phi}{\partial x})_n - (A \mu'_\phi \frac{\partial v_\phi}{\partial x})_s$$

$$+ (A \mu'_\phi \frac{\partial w_\phi}{\partial x})_f - (A \mu'_\phi \frac{\partial w_\phi}{\partial x})_b$$

with

$$\lambda'_\phi = \lambda_\phi \cdot \varepsilon_\phi$$

The derivatives in each term are expressed in terms of known velocities on the faces of the volume. For example,

$$\left( \frac{\partial v_\phi}{\partial x} \right)_n = [(v_\phi)_{i+1,j+1/2,k} - (v_\phi)_{ij+1/2,k}] / \Delta x_p$$

The macroscopic viscosities on the faces are determined by interpolation as described above.

- Solution for  $u_g$  and  $u_s$  -

The gas and solids momentum equations, Equation 4.5, are solved simultaneously to predict velocities at the same location. The two equations,  $\phi=g$  and  $s$  in Equation 4.5, may be written as:

$$a_g u_g = C_\beta \cdot u_s + S_g$$

$$a_s u_s = C_\beta \cdot u_g + S_s$$

where the coefficients  $a_g$ ,  $a_s$  and  $C_\beta$  and the source terms may be defined by referring to Equation 4.5. The velocities are then given by:

$$u_g = (C_\beta \cdot S_s + a_s \cdot S_g) / DEN$$

$$u_s = (C_\beta \cdot S_g + a_g \cdot S_s) / DEN$$

$$DEN = a_s \cdot a_g - C_\beta^2$$

#### 4.3.4 Y-Momentum Equations

The Y-direction momentum equation, Equation 2.3, is integrated over the momentum volume such as that depicted in Figures 4.1, 4.2 and 4.3 to give

$$a_{Py} \cdot (v_\phi)_P = (\overline{\rho'_\phi v_\phi v})_P - \zeta_\phi v_P \cdot \frac{P_n - P_s}{\Delta y_p}$$

$$+ (\beta_y^0 v v_{\phi 0})_P$$

$$+ (\overline{G_\phi v})_P \cdot \frac{(\varepsilon_g)_n - (\varepsilon_g)_s}{\Delta y_p} \quad 4.6$$

where the subscripts indicate

$$P \rightarrow i, j+1/2, k$$

$$n \rightarrow i+1, j, k$$

$$s \rightarrow i, j, k$$

The north (n) and south (s) faces of the X-momentum volume pass through the centers of main control volumes where pressures, viscosities and void fractions are calculated.

$$\begin{aligned}
 v_p &= \text{volume of the momentum volume} \\
 &= (v_m + v_s)/2 \\
 &= (v_{i,j+1,k} + v_{i,j,k})/2 \\
 \Delta y_p &= \text{length of the Y-momentum volume in the Y-direction} \\
 &= (\Delta y_j + \Delta y_{j+1})/2 \\
 \Delta y_j, \Delta y_{j+1} &= \text{lengths of the j and j+1 main control volumes in the Y-direction, an input to the model}
 \end{aligned}$$

Equation 4.6 is similar to the X-momentum equation, Equation 4.5, and the formulation of each term parallels that for the X-momentum equation in the previous section. For example,  $a_{p_y}$  is a velocity coefficient defined like  $a_{p_x}$ ;  $(\overline{\rho'v_yV})_p$  is the tilde term, evaluated at the beginning of the timestep, comparable to  $(\overline{\rho'u_xV})_p$ , and  $\beta_y^o$  is a drag coefficient defined like  $\beta_x^o$ . Because of these similarities, the detailed development of each term is not repeated. Instead, because it is the most complicated, a summary of the tilde term is provided.

- Tilde Term -

This term is evaluated only at the beginning of the timestep. As was done for the X-momentum equation, the designation,  $(\ )^o$ , is not used in the equations below, as all quantities are to be evaluated only at the start of the timestep. The tilde term is written as

$$(\overline{\rho'v_yV})_p = (\overline{\rho'V})_p \cdot (v_y)_p / \Delta t + B_y + C_y + \tau_y$$

Gravity Term:  $B_y$

$$B_y = (\overline{\rho'V})_p \cdot g_y$$

Convection Term:  $C_y$

$$\begin{aligned} C_y &= - \sum Q_q \cdot (\rho' v_\phi)_Q - Q_p \cdot (\rho' v_\phi)_p \\ &= - Q_E \cdot (\rho' v_\phi)_E - Q_W \cdot (\rho' v_\phi)_W \dots Q_B \cdot (\rho' v_\phi)_B \\ &\quad - Q_p (\rho' v_\phi)_p \end{aligned}$$

with

$$\begin{aligned} E &\rightarrow i+1, j+1/2, k \\ W &\rightarrow i-1, j+1/2, k \\ B &\rightarrow i, j+1/2, k-1 \\ e &\rightarrow i+1/2, j+1/2, k \\ w &\rightarrow i-1/2, j+1/2, k \\ b &\rightarrow i, j+1/2, k-1/2 \end{aligned}$$

Convection rates,  $q$ 's, on the faces of the volume are defined as

$$\begin{aligned} (A u_\phi)_e &= [(A_x u_\phi)_{i+1/2, j+1, k} + (A_x u_\phi)_{i+1/2, j, k}] / 2 \\ (A v_\phi)_n &= [(A_y v_\phi)_N + (A_y v_\phi)_P] / 2 \\ (A w_\phi)_b &= [(A_z w_\phi)_{i, j+1, k-1/2} + (A_z w_\phi)_{i, j, k-1}] / 2 \end{aligned}$$

The  $\xi$ -factors are defined according to the expressions given in the previous section with the face velocities such as  $(u_\phi)_e$ ,  $(u_\phi)_w$ , etc., determined by interpolation. The convecting flow rates,  $Q_q$ , are then defined according to the expressions in the previous section.

Diffusion Term:  $\tau_y$

$$\tau_y = \sum D_q \cdot (v_\phi)_Q - D_p \cdot (v_\phi)_p + \tau_c$$

$$D_p = D_E + D_W + D_N + \dots + D_B$$

The diffusion coefficients are defined as

$$D_e = \frac{(A \mu'_\phi)_e}{\Delta x_e}$$

...

$$D_n = \frac{2(A \mu'_\phi)_n}{\Delta y_n}$$

...

$$D_s = \frac{2(A \mu'_\phi)_s}{\Delta y_s}$$

...

$$D_b = \frac{(A \mu'_\phi)_b}{\Delta z_b}$$

with

$$\Delta x_e = (\Delta x_i + \Delta x_{i+1})/2$$

$$\Delta y_n = \Delta y_{j+1}$$

$$\Delta y_s = \Delta y_j$$

$$\Delta z_b = (\Delta z_k + \Delta z_{k-1})/2$$

$$(A \mu'_\phi)_e = [(A_x \mu'_\phi)_{i+1/2, j+1, k} + (A_x \mu'_\phi)_{i+1/2, j, k}] / 2$$

$$(A \mu'_\phi)_n = [(A_y)_N + (A_y)_P] \cdot (\mu'_\phi)_n / 2$$

and similar expressions on the other faces.

The macroscopic viscosities on the faces of main control volumes such as  $(\mu'_\phi)_{i+1/2, j+1, k}$  are determined by interpolation.

The last term,  $\tau_c$ , is made up of the last three terms in Equation 2.3 and is given by:

$$\tau_c = (A \lambda'_\phi \cdot \operatorname{div} V)_n - (A \lambda'_\phi \cdot \operatorname{div} V)_s$$

$$+ (A \mu'_\phi \frac{\partial u_\phi}{\partial y})_e - (A \mu'_\phi \frac{\partial u_\phi}{\partial y})_w$$

$$+ (A \mu'_\phi \frac{\partial w_\phi}{\partial y})_f - (A \mu'_\phi \frac{\partial w_\phi}{\partial y})_b$$

The derivatives are expressed in terms of velocities on the faces of the volume and the macroscopic viscosities on the faces are determined by interpolation.

- Solution for  $v_g$  and  $v_s$  -

The Y-direction momentum equations are solved simultaneously to predict the velocities at the same locations. The two equations,  $\phi=g$  and  $s$  in Equation 4.6, may be written as:

$$a_g \cdot v_g = C_\beta \cdot v_g + S_g$$

$$a_s \cdot v_s = C_\beta \cdot v_g + S_s$$

The solution is similar to that for the X-direction velocities given in the previous section.

#### 4.3.5 Z-Momentum Equations

The Z-direction momentum equation, Equation 2.4, is integrated over the Z-momentum volume such as that depicted in Figures 4.1, 4.2 and 4.3 to give

$$\begin{aligned}
 a_{pz} (v_\phi)_p &= (\overline{\rho' v' v})_p - \tilde{\zeta}_\phi v_p \cdot \frac{p_f - p_b}{\Delta z_p} \\
 &+ (\beta_s^0 \cdot v \cdot v_{\phi_0})_p \\
 &+ (G_1 v)_p \cdot \frac{(\epsilon_g)_f - (\epsilon_g)_b}{\Delta z_p}
 \end{aligned} \tag{4.7}$$

where the subscripts indicate

$p \Rightarrow i, j, k+1/2$

$f \Rightarrow i, j, k+1$

$b \Rightarrow i, j, k$

The front (f) and back(b) faces of the Z-momentum volume pass through the centers of main volumes where pressures, viscosities and void fractions are known.

$v_p$  = volume of the momentum volume

=  $(v_f + v_b)/2$

=  $(v_{i,j,k+1} + v_{i,j,k})/2$

$\Delta z_p$  = length of the Z-momentum volume in the Z-direction

=  $(\Delta z_k + \Delta z_{k+1})/2$

$\Delta z_k, \Delta z_{k+1}$  = lengths of the k and k+1 main control volumes, an input to the model, in the Z-direction

Equation 4.7 is similar to the X- and Y-momentum equations developed earlier. The formulation of Equation 4.7 parallels that for the X-momentum equation described in section 4.3.3. Detailed development of each term is not repeated; most can be derived by inspection. A summary of the tilde term is provided because it is the most complicated.

- Tilde Term -

Like the X- and Y-momentum equations, this term is evaluated only at the beginning of the timestep. The designation ( )<sup>\*</sup> is not used in the equations below because all terms are based on quantities at the start of the timestep. The tilde is given by:

$$(\tilde{\rho'v'V})_p = (\overline{\rho'V})_p \cdot (v')_p / \Delta t + B_z + C_z + \tau_z$$

Gravity Term:  $B_z$

$$B_z = (\overline{\rho'V})_p \cdot g_z$$

Convection Term:  $C_z$

$$\begin{aligned} C_z &= - \sum Q_q (\rho'v')_0 - Q_p (\rho'v')_p \\ &= - Q_e (\rho'v')_E - Q_w (\rho'v')_W - Q_n (\rho'v')_N \dots - Q_b (\rho'v')_B \\ &\quad - Q_p (\rho'v')_p \end{aligned}$$

with

$$E \Rightarrow i+1, j, k+1/2$$

$$W \Rightarrow i-1, j, k+1/2$$

$$N \Rightarrow i, j+1, k+1/2$$

$$B \Rightarrow i, j, k-1/2$$

$$e \Rightarrow i+1/2, j, k+1/2$$

$$w \Rightarrow i-1/2, j, k+1/2$$

$$n \Rightarrow i, j+1/2, k+1/2$$

$$b \Rightarrow i, j, k$$

The convecting rates,  $q$ 's, on the faces of the 2-momentum volume are given by:

$$(A u_\phi)_e = [(A_x u_\phi)_{i+1/2,j,k+1} + (A_x u_\phi)_{i+1/2,j,k}] / 2$$

• • •

$$(A v_\phi)_n = [(A_y v_\phi)_{i,j+1/2,k+1} + (A_y v_\phi)_{i,j+1/2,k}] / 2$$

• • •

$$(A w_\phi)_f = [(A_z w_\phi)_N + (A_z w_\phi)_P] / 2$$

The  $\xi$ -factors are defined as given in section 4.3.3 with the face velocities, such as  $(u_\phi)_e$ ,  $(u_\phi)_w$ ,  $(v_\phi)_n$ , ..., defined by interpolation. The convection flow rates,  $Q_q$ , are then defined as given in section 4.3.3.

Diffusion Term:  $\tau_s$

$$\tau_s = \sum D_q (v_\phi)_Q - D_p (v_\phi)_P + \tau_c$$

$$D_p = D_e + D_v + D_n + \dots D_b$$

The diffusion coefficients,  $D_q$ , are defined as:

$$D_e = \frac{(A u'_\phi)_e}{\Delta x_e}$$

• • •

$$D_n = \frac{(A u'_\phi)_n}{\Delta y_n}$$

• • •

$$D_f = \frac{2(A u'_\phi)_f}{\Delta z_f}$$

$$D_b = \frac{2(A u'_\phi)_b}{\Delta z_b}$$

with

$$\Delta x_e = (\Delta x_j + \Delta x_{j+1})/2$$

$$\Delta y_n = (\Delta y_j + \Delta y_{j+1})/2$$

$$\Delta z_f = \Delta z_{k+1}$$

$$\Delta z_b = \Delta z_k$$

$$(\Lambda \mu'_\phi)_e = [(\Lambda_x \mu'_\phi)_{i+1/2, j, k+1} + (\Lambda_x \mu'_\phi)_{i+1/2, j, k}] / 2$$

$$(\Lambda \mu'_\phi)_f = [(\Lambda_z)_F + (\Lambda_z)_P] \cdot (\mu'_\phi)_f / 2$$

The macroscopic viscosities on the faces of the main volume are defined by interpolation.

The last term,  $\tau_c$ , is made up of the last three terms in Equation 2.4 and is given by:

$$\tau_c = (\Lambda \lambda'_\phi \operatorname{div} V)_f - (\Lambda \lambda'_\phi \operatorname{div} V)_b$$

$$+ (\Lambda \mu'_\phi \frac{\partial u_\phi}{\partial z})_e - (\Lambda_x \mu'_\phi \frac{\partial u_\phi}{\partial z})_w$$

$$+ (\Lambda \mu'_\phi \frac{\partial v_\phi}{\partial z})_n - (\Lambda \mu'_\phi \frac{\partial v_\phi}{\partial z})_s$$

This term may be evaluated like the comparable term in the X- and Y-momentum equations.

- Solutions for  $w_g$  and  $w_s$  -

The gas and solids momentum equations are solved simultaneously like the X- and Y-equations. They may be written in the form:

$$a_g w_g = C_\beta w_s + S_g$$

$$a_s w_s = C_\beta w_g + S_s$$

The solution is similar to that given for the X-direction momentum equation in section 4.3.3.

#### 4.3.6 Pressure Correction Equation

The transient solution scheme involves adjusting pressures throughout the flow field until gas mass is conserved. Two methods are used: i) pressure correction based on the gas mass residue given by Equation 4.4 and ii) a regula-falsi scheme once minimum and maximum nodal pressures are established. The pressure correction equation based on the gas mass residue is developed here. This approach is similar to that given in detail for the steady formulation in section 4.4.6. The pressure correction equation for the transient solution may be derived (Patankar, 1980) by writing the gas continuity equation in terms of a trial velocity and a velocity correction. The trial velocity field, which is determined by solving the gas momentum equations, does not, in general, satisfy continuity and, therefore, results in a mass residue. The momentum equations are then written in terms of the trial velocities, trial pressures and corrections to both. Because the trial pressures are assumed to produce the trial velocities, an approximate, and simple, expression is derived relating the pressure and velocity corrections. These expressions are then used to replace the velocity corrections in the continuity equation resulting in the following:

$$P'_p = - \Omega_p \cdot D G_p / \zeta_g \sum [(A \rho' v) / (a_p \Delta L)]_q \quad 4.8$$

where

$P'_p$  = pressure correction to be applied to the main control volume pressure

$\Omega_p$  = an under-relaxation factor that is used to promote solution convergence:  $0 \leq \Omega_p \leq 1.0$

$D G_p$  = gas mass residue given by Equation 4.4

$\sum$  = indicates that the quantities are summed over each face of the main control volume, i.e.,

$q = e, w, n, s, f$  and  $b$

with

$e \Rightarrow i+1/2, j, k$

$w \Rightarrow i-1/2, j, k$

$n \Rightarrow i, j+1/2, k$

$s \Rightarrow i, j-1/2, k$

$f \Rightarrow i, j, k+1/2$

$b \Rightarrow i, j, k-1/2$

$A$  = flow area normal to the face

$\rho'_q$  =  $(\varepsilon_g \rho_g)$  evaluated on the face,  $q$ , using the upwind approximation

$V$  = volume of the momentum control volume associated with face  $q$ ,  
the momentum volume that connects main volume  $P$  and the neighbor  
that shares face  $q$

$a_p$  = momentum equation coefficient of the face  $q$  velocity. These  
coefficients come from the left-hand side of Equations 4.5, 4.6  
and 4.7

$\Delta L$  = length of the momentum volume normal to face  $q$

The following should be noted:

- o Equation 4.8 is a point solution. In developing Equation 4.8, the pressure corrections in the neighboring volumes have been ignored. This approach is used because the FLUFIX solution scheme is cell-by-cell rather than a coupled method.
- o Equation 4.8 is simpler than the FLUFIX formulation. In both formulations, the objective is to develop a simple and stable method to adj. 't pressure.

#### 4.4 EQUATIONS FOR THE STEADY SIMULATION

##### 4.4.1 Overview

The governing finite difference equations for the steady simulation are developed in this section. The approach is to integrate the governing conservation relations given in section 2.0 over the control volume

arrangement shown in Figures 4.1, 4.2 and 4.3. This results in a set of coupled non-linear equations that may be arranged into a standard form like equation 4.1. Because an iterative solution method will be used, relaxation is next applied to derive the governing finite difference equations.

In this section, equations for the gas void fraction and the phase velocities are derived. The void fraction equation is based on the continuity equations for both phases (Carver 1982). The phase momentum equations are based on a hybrid differencing scheme (Patankar 1980) for the combined convection and diffusion terms. This scheme tends to reduce numerical diffusion and thereby provides a better approximation, compared to the upwind method for example. In addition, it is computationally simple to implement.

#### 4.4.2 Gas Void Fraction Equation

The gas void fraction equation is developed based on Carver (1980). The gas and solids continuity equations are expressed in terms of normalized densities for each phase and then integrated over the Main Control Volume using the upwind approximation for quantities on the volume faces. The two equations are then subtracted and the identity

$$\varepsilon_g + \varepsilon_s = 1$$

applied to yield the following:

$$C_{eu} \varepsilon_p = \sum a_q \varepsilon_Q + S_{eu} \quad 4.9$$

where the subscript  $eu$  indicates unrelaxed parameters and

$$\sum a_q \varepsilon_Q = a_e \cdot \varepsilon_E + a_w \cdot \varepsilon_W + a_n \cdot \varepsilon_N + \dots a_b \cdot \varepsilon_B$$

$$\varepsilon_p = (\varepsilon_g)_{i,j,k}$$

$$\varepsilon_E = (\varepsilon_g)_{i+1,j,k}$$

$$\varepsilon_W = (\varepsilon_g)_{i-1,j,k}$$

$$\varepsilon_N = (\varepsilon_g)_{i,j+1,k}$$

$$\varepsilon_B = (\varepsilon_g)_{i,j,k-1}$$

The coefficient  $C_{eu}$  is given by:

$$C_{eu} = \frac{v_p}{\Delta t} (r_g + r_s)_p + (r_g)_p \cdot \sum a_g + (r_s)_p \cdot \sum a_\phi$$

where  $v_p = v_{i,j,k}$

and  $(r_g)_p$  and  $(r_s)_p$  are the normalized gas and solids densities, respectively, at the center of the main control volume, i.e.,

$$(r_g)_p = (\rho_g)_{i,j,k}/(\rho_g)_R$$

$$(r_s)_p = (\rho_s)_{i,j,k}/(\rho_s)_R$$

where  $(\rho_g)_R$  and  $(\rho_s)_R$  are the gas and solids reference (normalizing) densities, respectively.

The summation terms,  $\sum a_g$  and  $\sum a_\phi$ , are given by:

$$\sum a_\phi = \sum a_g + \sum a_s$$

with

$$\sum a_g = \sum_{q=e,n,f} [A_q \cdot \lambda_{gq} \cdot \xi_{gq}] - \sum_{q=w,s,b} [A_q \cdot \lambda_{gq} \cdot (1-\xi_{gq})]$$

$$\sum a_s = \sum_{q=e,n,f} [A_q \cdot \lambda_{sq} \cdot \xi_{sq}] - \sum_{q=w,s,b} [A_q \cdot \lambda_{sq} \cdot (1-\xi_{sq})]$$

where the subscripts e, w, n, ..., b indicate the east, west, north, ..., back faces of the main control volume,  $\lambda_{\phi q}$  is used to indicate the velocity of phase  $\phi$  ( $\phi = g$  or  $s$ ) on face  $q$ , and  $A_q$  indicates the area of face  $q$ , i.e.,

$$(A_q \cdot \Lambda_{\phi q})_{q=e} = (A_x u_\phi)_{i+1/2, j, k}$$

$$(A_q \cdot \Lambda_{\phi q})_{q=n} = (A_y v_\phi)_{i, j+1/2, k}$$

$$(A_q \cdot \Lambda_{\phi q})_{q=f} = (A_z w_\phi)_{i, j, k+1/2}$$

The weighting factors,  $\xi_{sq}$  and  $\xi_{gq}$ , arise due to the upwind differencing and are given by:

$$\xi_{\phi q} = \begin{cases} 1 & \Lambda_{\phi q} \geq 0 \\ 0 & \Lambda_{\phi q} < 0 \end{cases} \quad \begin{matrix} q = e, w, n, s, f, b \\ \phi = g, s \end{matrix}$$

where

$$(\Lambda_{\phi q})_{q=e} = (u_\phi)_{i+1/2, j, k}$$

$$(\Lambda_{\phi q})_{q=n} = (v_\phi)_{i, j+1/2, k}$$

$$(\Lambda_{\phi q})_{q=f} = (w_\phi)_{i, j, k+1/2}$$

The coefficients,  $a_e, a_w, a_n, \dots, a_b$  involve convection on the faces and are given by:

$$a_{q=e, n, f} = - \sum_{q=e, n, f} (A_q \cdot \Lambda_{gq}) (1 - \xi_{gq}) \cdot (r_g)_0$$

$$- \sum_{q=e, n, f} (A_q \cdot \Lambda_{sq}) (1 - \xi_{sq}) \cdot (r_s)_0$$

where  $Q = E, N, F$

with  $q = e, n$  and  $f$ , respectively.

For the other faces,

$$a_{q=q=w,s,b} = \sum_{q=w,s,b} (A_q \cdot \Lambda_{gq}) \cdot \xi_{gq} \cdot (r_g)_Q$$

$$+ \sum_{q=w,s,b} (A_q \cdot \Lambda_{sq}) \cdot \xi_{sq} \cdot (r_s)_Q$$

where  $Q = W, S$  and  $B$

with  $q = w, s$  and  $b$ , respectively.

The source term,  $S_{\varepsilon u}$ , is given by:

$$S_{\varepsilon u} = \sum_{q=e,n,f} (A_q \Lambda_{sq}) \cdot [\xi_{sq} (r_s)_P + (1 - \xi_{sq}) \cdot (r_s)_Q] - \sum_{q=w,s,b} (A_q \Lambda_{sq}) \cdot [(1 - \xi_{sq}) \cdot (r_s)_P + \xi_{sq} \cdot (r_s)_Q] - \frac{\dot{M}_g}{(\rho_g)_R} - \frac{\dot{M}_s}{(\rho_s)_R} + \frac{V_p}{\Delta t} (\rho'_g + \rho'_s)_P^0$$

where

$Q = E, N$  and  $F$

when

$q = e, n$  and  $f$ , respectively, in the first term and

$Q = W, S$  and  $B$

when

$q = w, s$  and  $b$ , respectively, in the second term.

For completeness, the storage terms have been included. With these terms deleted, i.e.,

$$\frac{V_p}{\Delta t} (r_g + r_s)_p = 0$$

$$\frac{V_p}{\Delta t} (p'_g + p'_s)_p = 0$$

the resulting set becomes the "unrelaxed" finite difference equations for the steady simulation.

Although the coefficient,  $C_{\epsilon_u}$ , in Equation 4.9 will always be positive, an important criteria for numerical stability (Patankar, 1980), Equation 4.9 is modified to further promote solution convergence based on the following two criteria:

i) Eliminate negative source terms.

The gas void fraction must always be in the range

$$(\epsilon_g)_{\text{packed}} < \epsilon_g \leq 1.0$$

where  $(\epsilon_g)_{\text{packed}}$  is the void fraction at a packed condition. During the iterative solution sequence, a "strong" negative source term (i.e., large negative values of  $S_{\epsilon_u}$ ) may drive the void fraction significantly below the packed condition or even negative. To eliminate this behavior, a negative source term is linearized and Equation 4.9 rewritten as

$$(C_{\epsilon_u} - S_p) \epsilon_p = \sum a_q \epsilon_q + S_1 \quad 4.10$$

where  $s_1 = \begin{cases} s_{\epsilon_u} & s_{\epsilon_u} > 0 \\ 0 & s_{\epsilon_u} \leq 0 \end{cases}$

$$s_p = \begin{cases} 0 & s_{\epsilon_u} > 0 \\ s_{\epsilon_u}/(\epsilon_p)^{\text{iter-1}} & s_{\epsilon_u} \leq 0 \end{cases}$$

and

$(\epsilon_p)^{\text{iter-1}}$  = gas void fraction at the end of the previous iteration.

ii) Gradual approach to limiting values.

To promote solution stability near the packed state  $\epsilon_g = (\epsilon_g)_{\text{packed}}$  or near the condition of no solids ( $\epsilon_g = 1.0$ ), Equation 4.9 is relaxed to slow changes in the void fraction. After applying relaxation (Carver, 1982), Equation 4.10 becomes:

$$a_p \cdot \epsilon_p = \sum a_q \epsilon_q + s_\epsilon \quad 4.11$$

$$a_p = (c_{\epsilon_u} - s_p)/\theta_g$$

$$s_\epsilon = s_1 + c_\epsilon (1 - \theta_g) \cdot (\epsilon_p)^{\text{iter-1}}$$

where  $\theta_g$  is the under-relaxation parameter,

$$0 < \theta_g \leq 1.0$$

and is specified as follows:

$$\theta_g = \theta_{\min} \quad , \quad \epsilon'_p \leq \epsilon_{\min}$$

$$\theta_g = \theta_{\text{nom}} \quad , \quad \epsilon_{\min} < \epsilon'_p \leq \epsilon_{\max}$$

$$\theta_g = \theta_{\max} \quad , \quad \epsilon_{\max} < \epsilon'_p$$

where  $\epsilon'_p$  is a point solution of Equation 4.9 based on the void fractions in the neighboring volumes at the end of the previous iteration. The relaxation factors  $\theta_{\min}$ ,  $\theta_{\text{nom}}$  and  $\theta_{\max}$  and the associated void fractions  $\epsilon_{\min}$  and  $\epsilon_{\max}$ , are specified through input to FORCE2.

Equation 4.11 is the finite difference equation that must be solved to predict gas void fraction.

#### 4.4.3 X-Momentum Equation

The X-momentum equation, Equation 2.2, is integrated over the X-momentum volume shown in Figure 4.1 or 4.3. As indicated in section 2, the steady form of the momentum equation does not include the time derivative term nor the last four viscous terms in Equation 2.2. After integration, the convection and diffusion terms are approximated using the hybrid differencing scheme (Patankar 1980). As explained by Patankar (1980), at low Peclet numbers ( $-2 \leq Pe \leq 2$ ), a central difference is used to estimate convected velocities on the volume faces and at high Peclet numbers (outside the noted range), an upwind scheme is used to estimate these velocities. This approach tends to reduce unwanted numerical diffusion and, thereby, improve the modeling.

After integrating and applying the hybrid difference, the X-momentum equation may be written, using the compact notation associated with the control volume arrangement in Figure 4.3, as

$$C_{\phi X} \cdot u_{\phi P} = \sum a_q \cdot u_{\phi Q} + \beta_u \cdot (u_{\phi 0})_P + S_x - S_{\Delta P}, \phi = g \text{ or } s \quad 4.12$$

where the node center (P), east (E), west (W),..., back (B) locations are defined as:

$P \Rightarrow i+1/2, j, k$   
 $E \Rightarrow i+3/2, j, k$   
 $W \Rightarrow i-1/2, j, k$   
 $N \Rightarrow i+1/2, j+1, k$

$B \Rightarrow i+1/2, j, k-1$

and

$u_{\phi P} = (u_\phi)_{i+1/2, j, k}$   
 $u_{\phi E} = (u_\phi)_{i+3/2, j, k}$   
 $u_{\phi W} = (u_\phi)_{i-1/2, j, k}$   
 $u_{\phi N} = (u_\phi)_{i+1/2, j+1, k}$   
 $\cdot$   
 $\cdot$   
 $\cdot$   
 $u_{\phi B} = (u_\phi)_{i+1/2, j, k-1}$   
 $(u_{\phi o})_P = \text{velocity of the other phase at } i+1/2, j, k$   
 $= (u_{\phi o})_{i+1/2, j, k}$

$$\sum a_q \cdot u_{\phi 0} = a_e \cdot u_{\phi E} + a_w \cdot u_{\phi W} + \dots a_b \cdot u_{\phi B}$$

- Face Coefficients,  $a_q$  -

The coefficients,  $a_e, a_w, \dots, a_b$  account for convection and diffusion on the volume faces and are defined for each phase using the hybrid difference as:

$$a_e = \text{MAX} [D_e, |F_e/2|] - F_e/2$$

$$a_w = \text{MAX} [D_w, |F_w/2|] + F_w/2$$

$$a_n = \text{MAX} [D_n, |F_n/2|] + F_n/2$$

$\cdot$

$\cdot$

$\cdot$

$$a_b = \text{MAX} [D_b, |F_b/2|] + F_b/2$$

where MAX indicates the maximum of the two quantities in the brackets. The diffusion conductance,  $D$ , and the convecting strength (or flow rate),  $F$ , are defined on each face of the X-momentum volume, depending on the phase, as follows:

$$D_q = (A \mu'_\phi)_q / \Delta L_q$$

$$F_q = (\rho'_\phi A \Lambda'_\phi)_q$$

with  $A$  and  $\Delta L_q$  denoting the flow area and characteristic length normal to face  $q$ , respectively. For example:

$$\Delta L_e = \Delta x_e = \Delta x_{i+1}$$

$$\Delta L_n = \Delta y_n = (\Delta y_j + \Delta y_{j+1})/2$$

$$\Delta L_f = \Delta z_f = (\Delta z_k + \Delta z_{k+1})/2$$

$$(A \mu'_\phi)_e = [(A_x)_E + (A_x)_P] \cdot (\mu'_\phi)_e/2$$

$$(A \mu'_\phi)_n = [(A_y \mu'_\phi)_{i+1,j+1/2,k} + (A_y \mu'_\phi)_{i,j+1/2,k}]/2$$

$$(A \mu'_\phi)_f = [(A_z \mu'_\phi)_{i+1,j,k+1/2} + (A_z \mu'_\phi)_{i,j,k+1/2}]/2$$

$$(\rho'_\phi A \Lambda'_\phi)_e = [(A_x u'_\phi)_E + (A_x u'_\phi)_P] \cdot (\rho'_\phi)_e/2$$

• • •

$$(\rho'_\phi A \Lambda'_\phi)_n = [(A_y v'_\phi \rho'_\phi)_{i+1,j+1/2,k} + (A_y v'_\phi \rho'_\phi)_{i,j+1/2,k}]/2$$

• • •

$$(\rho'_\phi A \Lambda'_\phi)_f = [(A_z w'_\phi \rho'_\phi)_{i+1,j,k+1/2} + (A_z w'_\phi \rho'_\phi)_{i,j,k+1/2}]/2$$

The macroscopic viscosities on the north, south, front and back faces are evaluated by interpolation such as:

$$(\mu'_\phi)_{i+1,j,k+1/2} = f_3 \cdot (\mu'_\phi)_{i+1,j,k+1} + (1-f_3) (\mu'_\phi)_{i+1,j,k}$$

where

$$f_3 = \Delta z_k / (\Delta z_{k+1} + \Delta z_k)$$

The macroscopic densities on the faces are evaluated based on the upwind approximation, such as:

$$(\rho'_\phi)_{i+1,j,k+1/2} = \xi_f \cdot (\rho'_\phi)_{i+1,j,k} + (1 - \xi_f) \cdot (\rho'_\phi)_{i+1,j,k+1}$$

where

$$\xi_f = \begin{cases} 1 & (v'_\phi)_{i+1,j,k+1/2} \geq 0 \\ 0 & (v'_\phi)_{i+1,j,k+1/2} < 0 \end{cases}$$

Development of the diffusion coefficients and convecting strengths,  $D_q$  and  $F_q$ , on the other faces follows that given above.

- Velocity Coefficient,  $C_{\phi x}$  -

The coefficient,  $C_{\phi x}$ , is defined, depending on the phase, as:

$$C_{\phi x} = \sum a_q + F_p + (\beta_x + k_{\phi x})_p \cdot v_p$$

where

$$\sum a_q = a_e + a_w + \dots + a_b$$

and

$$F_p = \begin{cases} |F_{\text{net}}| & \dot{M}_\phi \geq 0 \\ 0 & \dot{M}_\phi < 0 \end{cases}$$

where

$$F_{\text{net}} = F_e - F_w + F_n - F_s + F_f - F_b$$

The coefficient,  $F_p$ , is defined so that  $C_{\phi x}$  does not become negative, a situation that can lead to an unstable solution. With no mass sources,  $F_p$  will approach zero at solution because it is the net flow out of the  $X$ -momentum volume. With positive mass sources,  $F_p$  should be positive, thereby making  $C_{\phi x}$  positive. However, with mass sinks, i.e.,  $\dot{M}_\phi$  negative,  $F_p$  must be negative if it is retained on the left-hand side of Equation 4.12. For this case,  $F_p$  is included in the source term,  $S_x$ .

The additional terms are defined as:

$(k_{\phi x})_p$  = loss coefficient  $k_{\phi x}$ , evaluated at  $i+1/2$ ,  $j$ ,  $k$  and defined in section 4.3.3

$\beta_x$  = drag coefficient evaluated at  $i+1/2$ ,  $j$ ,  $k$  and defined in section 4.3.3

#### - Drag Term with Other Phase -

The phase momentum equations are explicitly linked through the relaxed drag coefficient,  $\beta_u$ , defined as:

$$(\beta_u)_p = \theta_{\phi\beta} \cdot (\beta_x)_p$$

where

$(\beta_x)_p$  = drag coefficient evaluated at  $i+1/2$ ,  $j$ ,  $k$  as described in section 4.3.3

$\theta_{\phi\beta}$  = drag under-relaxation coefficient in phase  $\phi$ . It is an input parameter to FORCE2 and is defined on the range  $0 \leq \theta_{\phi\beta} \leq 1$

#### - Source Term, $S_x$ -

The source term,  $S_x$ , depends on the phase and is given by:

$$S_x = B_x + (\overline{G_{\phi} \cdot V})_p \cdot \frac{(e_g)_e - (e_g)_w}{\Delta x_p}$$

$$+ (\beta_x - \beta_u)_p \cdot (u_{\phi 0} - u_{\phi})_p^{iter-1} + (|F_{net}| - F_p) (u_{\phi})_p^{iter-1}$$

where

$$(u_{\phi 0} - u_{\phi})_p^{iter-1} = [(u_{\phi 0} - u_{\phi})_{i+1/2, j, k}]^{iter-1}$$

= velocity difference at  $i+1/2$ ,  $j$ ,  $k$  evaluated at the previous iteration

$$(u_\phi)_P^{\text{iter-1}} = \text{velocity at } i+1/2, j, k \text{ evaluated at the end of the previous iteration}$$

$$\Delta x_P = (\Delta x_i + \Delta x_{i+1})/2$$

and the body force term,  $\beta_x$ , and solid stress term,  $(\bar{G}_\phi \cdot \bar{V})_{i+1/2, j, k}$  are as defined in section 4.4.3.

- Pressure Gradient Source Term,  $S_{\Delta P}$  -

The pressure gradient source term is given by:

$$S_{\Delta P} = \zeta_\phi \cdot v_P \cdot \frac{P_e - P_w}{\Delta x_P}$$

- Summary -

Equation 4.12 is the unrelaxed form of the X-direction momentum equation. Inertia and under-relaxation, section 4.7, are applied before this equation is solved.

#### 4.4.4 Y-Momentum Equation

Development of the Y-momentum equations is similar to that given for the X-equation in the previous section. After integrating over the Y-momentum volume shown in Figures 4.1, 4.2 and 4.3 and applying the hybrid difference, the Y-momentum equation is expressed as:

$$C_{\phi y} \cdot v_{\phi P} = \sum a_q v_{\phi Q} + \beta_y \cdot (v_{\phi 0})_P + S_y - S_{\Delta P}, \phi = g \text{ or } s \quad 4.13$$

where locations in the compact notation are related to the  $i, j, k$  locations according to

$P \Rightarrow i, j+1/2, k$   
 $E \Rightarrow i+1, j+1/2, k$   
 $W \Rightarrow i-1, j+1/2, k$   
 $N \Rightarrow i, j+3/2, k$   
 $S \Rightarrow i, j-3/2, k$   
 .  
 .  
 .  
 $B \Rightarrow i, j+1/2, k-1$   
 $e \Rightarrow i+1/2, j+1/2, k$   
 $w \Rightarrow i-1/2, j+1/2, k$   
 $n \Rightarrow i, j+1, k$   
 .  
 .  
 .  
 $b \Rightarrow i, j+1/2, k-1$

with

$$\begin{aligned}
 v_{\phi P} &= (v_{\phi})_{i,j+1/2,k} \\
 v_{\phi E} &= (v_{\phi})_{i+1,j+1/2,k} \\
 &\vdots \\
 v_{\phi B} &= (v_{\phi})_{i,j+1/2,k-1}
 \end{aligned}$$

$$\sum a_q v_{\phi q} = a_e \cdot v_{\phi E} + a_w \cdot v_{\phi W} + \dots + a_b \cdot v_{\phi B}$$

#### - Face Coefficients -

The coefficients are evaluated using the hybrid scheme as outlined in the previous section with the diffusion coefficients,  $D_q$ , and convecting strength,  $F_q$ , defined as follows:

$$\begin{aligned}
 D_q &= (A \mu_{\phi})_q / \Delta L_q \\
 F_q &= (\rho' A v_{\phi})_q
 \end{aligned}$$

with  $A$  and  $\Delta L_q$  defined as flow areas and lengths normal to face  $q$ . For example:

$$\Delta L_e = \Delta x_e = (\Delta x_i + \Delta x_{i+1})/2$$

$$\Delta L_n = \Delta y_n = \Delta y_{j+1}$$

$$\Delta L_f = \Delta z_f = (\Delta z_k + \Delta z_{k+1})/2$$

$$(A\mu'_\phi)_e = [(A_x\mu'_\phi)_{i+1/2,j+1,k} + (A_x\mu'_\phi)_{i+1/2,j,k}]/2$$

$$(A\mu'_\phi)_n = [(A_y)_N + (A_y)_P] \cdot (\mu'_\phi)_n/2$$

$$(A\mu'_\phi)_f = [(A_z\mu'_\phi)_{i,j+1,k+1/2} + (A_z\mu'_\phi)_{i,j,k+1/2}]^2$$

$$(\rho'_A v)_e = [(\rho'_A A_x u_\phi)_{i+1/2,j+1,k} + (\rho'_A A_x u_\phi)_{i+1/2,j,k}]^2$$

$$(\rho'_A v)_n = [(A_y v_\phi)_N + (A_y v_\phi)_P] \cdot (\rho'_\phi)_n/2$$

$$(\rho'_A v)_f = [(\rho'_A A_z v_\phi)_{i,j+1,k+1/2} + (\rho'_A A_z v_\phi)_{i,j,k+1/2}]^2$$

where the macroscopic viscosities on the faces of the main control volumes are evaluated by interpolation and the macroscopic densities by the upwind approximation.

- Velocity Coefficient,  $C_{\phi y}$  -

The coefficient,  $C_{\phi y}$ , is given by:

$$C_{\phi y} = \sum a_q + F_p + (\beta_y + K_{\phi y})_p \cdot v_p$$

with

$$\sum a_q = a_e + a_w + \dots a_b$$

and the flow parameter,  $F_p$ , is as defined in the previous section. The drag coefficient,  $\beta_y$ , and loss coefficients are evaluated at  $i, j+1/2, k$  as outlined in section 4.3.4.

- Drag Term for Other Phase -

The under-relaxed drag coefficient,  $\beta_v$ , is similar to  $\beta_u$ , defined in the previous section, and is defined as:

$$\beta_v = \theta_{\phi\beta} \cdot \beta_y$$

where  $\theta_{\phi\beta}$  is the under-relaxation coefficient defined earlier.

- Source Term,  $S_y$

$$S_y = B_y + (\overline{G_\phi V})_P \cdot \frac{(\varepsilon_g)_n - (\varepsilon_g)_s}{\Delta y_P} + (\beta_y - \beta_v)_P \cdot (v_{\phi 0} - v_\phi)_P^{\text{iter-1}} + (|F_{\text{net}}| - F_P) \cdot (v_\phi)_P^{\text{iter-1}}$$

where  $\beta_y$ , the Y-direction body force term, and  $(\overline{G_\phi V})$ , the solids stress term, are as defined in section 4.3.4 and

$$\Delta y_P = (\Delta y_j + \Delta y_{j+1})/2$$

- Pressure Gradient Term -

The pressure gradient term is given by:

$$S_{\Delta P} = \zeta_\phi \cdot v_P \cdot \frac{P_n - P_s}{\Delta y_P}$$

- Summary -

Equation 4.12 is the unrelaxed form of the Y-momentum equation. Inertia and under-relaxation are applied, section 4.7, before this equation is solved.

#### 4.4.5 Z-Momentum Equation

Development of the Z-momentum equations is similar to that given for the x and y equations in the previous sections. The Z-momentum equation for phase  $\phi$  is expressed as:

$$C_{\phi z} \cdot w_{\phi P} = \sum a_q w_{\phi Q} + \beta_w (w_{\phi 0})_P + S_z - S_{\Delta P}, \phi = g \text{ or } s \quad 4.14$$

where the node center (P), east (E), west (W) and back (B) locations are defined as:

$$P \rightarrow i, j, k+1/2$$

$$E \rightarrow i+1, j, k+1/2$$

$$W \rightarrow i-1, j, k+1/2$$

$$N \rightarrow i, j+1, k+1/2$$

$$S \rightarrow i, j-1, k+1/2$$

$$F \rightarrow i, j, k+3/2$$

$$B \rightarrow i, j, k-1/2$$

$$e \rightarrow i+1/2, j, k+1/2$$

$$w \rightarrow i-1/2, j, k+1/2$$

.

.

.

$$f \rightarrow i, j, k+1$$

$$b \rightarrow i, j, k$$

with

$$w_{\phi P} = (w_{\phi})_{i,j,k+1/2}$$

$$w_{\phi E} = (w_{\phi})_{i+1,j,k+1/2}$$

$$w_{\phi B} = (w_{\phi})_{i,j,k-1/2}$$

$$\sum a_q w_{\phi Q} = a_e w_{\phi E} + a_w \cdot w_{\phi N} + \dots a_b w_{\phi B}$$

The face coefficients,  $a_e, a_w, \dots$ , and  $a_b$  are defined in terms of diffusion conductances and convection strength on the faces of the Z-momentum volume for each phase similar to the face coefficients in the X- and Y-momentum equations. The diffusion coefficients and convection strengths are defined as follows:

$$D_q = (A\mu'_\phi)_q / \Delta L_q$$

$$F_q = (\rho'_\phi A \Lambda_\phi)_q$$

For example,

$$\Delta L_e = \Delta x_e = (\Delta x_i + \Delta x_{i+1})/2$$

$$\Delta L_n = \Delta y_n = (\Delta y_j + \Delta y_{j+1})/2$$

$$\Delta L_f = \Delta z_f = \Delta z_{k+1}$$

$$(A\mu'_\phi)_e = [(A_x\mu'_\phi)_{i+1/2,j,k+1} + (A_x\mu'_\phi)_{i+1/2,j,k}] / 2$$

$$(A\mu'_\phi)_n = [(A_y\mu'_\phi)_{i,j+1/2,k+1} + (A_y\mu'_\phi)_{i,j+1/2,k}] / 2$$

$$(A\mu'_\phi)_f = [(A_z)_F + (A_z)_P] \cdot (\mu'_\phi)_f / 2$$

$$(\rho'_\phi A \Lambda_\phi)_e = [(\rho'_\phi A_x u_\phi)_{i+1/2,j,k+1} + (\rho'_\phi A_x u_\phi)_{i+1/2,j,k}] / 2$$

$$(\rho'_\phi A \Lambda_\phi)_n = [(\rho'_\phi A_y v_\phi)_{i,j+1/2,k+1} + (\rho'_\phi A_y v_\phi)_{i,j+1/2,k}] / 2$$

$$(\rho'_\phi A \Lambda_\phi)_f = [(A_z v_\phi)_F + (A_z v_\phi)_P] (\rho'_\phi)_f / 2$$

As noted in the previous sections, the macroscopic viscosities on the faces are evaluated by interpolation and the macroscopic densities by the upwind method.

- Velocity Coefficient,  $C_{\phi z}$  -

The coefficient,  $C_{\phi z}$ , is defined for each phase according to:

$$C_{\phi z} = \sum a_q + F_p + (\beta_z + k_{\phi z})_p \cdot v_p$$

where all the terms are similar to those defined in the X- and Y-momentum equations.

- Source Term,  $S_z$  -

$$S_z = B_z + (\bar{G}_\phi \bar{V})_P \cdot \frac{(\varepsilon_g)_f - (\varepsilon_g)_b}{\Delta z_P}$$
$$+ (\beta_z - \beta_w)_P \cdot (w_\phi - w_\phi)_P^{\text{iter-1}}$$
$$+ (|F_{\text{net}}| - F_P) \cdot (w_\phi)_P^{\text{iter-1}}$$

with

$$\beta_w = \theta_\phi \beta \cdot \beta_z$$

$$\Delta z_P = (\Delta z_k + \Delta z_{k+1})/2$$

- Pressure Gradient Term,  $S_{\Delta P}$  -

$$S_{\Delta P} = \zeta_\phi \cdot v_P \cdot \frac{P_f - P_b}{\Delta z_P}$$

- Summary -

Equation 4.14 is the unrelaxed form of the Z-momentum equation. Like the X- and Y-equations, inertia and under-relaxation are applied before it is solved.

#### 4.4.6 Pressure Correction Equation

The solution procedure is based on adjusting the pressure field until total mass continuity is achieved. This approach is the same as that for single-phase flow (Patankar, 1980) and is based, in part, on recent work for steam/water flows (Carver, 1982). With this method, the pressure field is modified (or corrected) based on the total (gas + solids) mass residue.

The equation for the pressure correction is developed using the momentum and continuity equations and is given by:

$$C_p \cdot P'_p = \sum a_q P'_Q + S'_p$$

4.15

where

$P'_p$  = correction to the pressure in main control volume P

$$\sum a_q P'_Q = a_e \cdot P'_E + a_w \cdot P'_W + \dots a_b \cdot P'_B$$

$$C_p = \sum a_q$$

$S'_p$  = source term

Because volume P is a main control volume, node and face subscripts are defined as:

$$\begin{aligned} P &\rightarrow i, j, k \\ E &\rightarrow i+1, j, k \\ W &\rightarrow i-1, j, k \\ N &\rightarrow i, j+1, k \\ B &\rightarrow i, j, k-1 \\ e &\rightarrow i+1/2, j, k \\ w &\rightarrow i-1/2, j, k \\ n &\rightarrow i, j+1/2, k \\ b &\rightarrow i, j, k-1/2 \end{aligned}$$

The development of Equation 4.15 follows the formulation of others (Patankar, 1980, Carver, 1982). The momentum equations on each face of the main volume P are written in terms of the desired pressure corrections as:

$$(\Lambda_\phi)_q = (\Lambda_\phi^*)_q - (D\Lambda_\phi)_q \cdot \Delta P'_q \quad 4.16$$

where

$(\Lambda_\phi)_q$  = phase velocities,  $u_\phi$ ,  $v_\phi$  or  $w_\phi$  that will satisfy both mass and momentum conservation

$(\Lambda_\phi^*)$  = trial velocities based on a trial (assumed) pressure field

$\Delta P'_q$  = difference in pressure corrections for cells that share face  $q$

For example, on the north face of the main volume, Equation 4.16 may be written as:

$$(v_\phi)_n = (v_\phi^*)_n - (Dv_\phi)_n \cdot (P_N^* - P_P^*)$$

where

$$(Dv_\phi)_n = \frac{\zeta_\phi \cdot v_n}{(C_\phi)_n \cdot \Delta y_n}$$

and

$$(v_\phi^*)_n = \frac{1}{(C_\phi)_n} \cdot \left[ \sum a_q v_{\phi q} + \beta_v (v_{\phi 0}) + s_v \right]_n - (Dv_\phi)_n (P_N^* - P_P^*)$$

$(C_\phi)_n$  = coefficient of velocity in the momentum equation on the north face of the main control volume (from Equation 4.13)

$v_n$  = volume of the Y-momentum volume on the north face

$P^*, P'$  = trial and associated pressure corrections

$$\Delta y_n = (\Delta y_j + \Delta y_{j+1})/2$$

The velocities on each face, such as that given in Equation 4.16, are now used in the continuity equations to derive Equation 4.15.

The continuity equations for each phase, expressed in terms of normalized densities (section 4.4.2), are added and integrated over the main

volume. Using the velocities on the faces from the momentum equations, such as Equation 4.16, the pressure correction equation (Equation 4.15) is derived with:

$$a_q = \zeta_g \left( \frac{A \cdot V \cdot r'_g}{C_g \cdot \Delta L} \right)_q + \zeta_s \left( \frac{A \cdot V \cdot r'_s}{C_s \cdot \Delta L} \right)_q$$

$$S'_p = - D G_p / (\rho_g)_R - D S_p / (\rho_s)_R$$

where

- $A$  = flow area normal to face  $q$
- $V$  = volume of the momentum volume connecting adjacent main volumes normal to face  $q$
- $r'_\phi, \phi=g \text{ or } s$  = normalized density on the face based on the upwind approximation, i.e., on the north face

$$r'_\phi = \begin{cases} (\rho'_\phi)_{i,j,k} / (\rho_\phi)_R & (v'_\phi)_n \geq 0 \\ (\rho'_\phi)_{i,j+1,k} / (\rho_\phi)_R & (v'_\phi)_n < 0 \end{cases}$$

- $C_g, C_s$  = coefficient of velocity in the gas and solids momentum equations, respectively, for the momentum volume normal to face  $q$

- $\Delta L$  = length of the momentum volume in the direction normal to face  $q$

- $D G_p, D S_p$  = gas, solids mass residuals, respectively

$$D G_p = \sum_{q=e,n,f} (A \rho'_g \Lambda_g^*)_q - \sum_{q=w,s,b} (A \rho'_g \Lambda_g^*) - \dot{M}_g$$

$$D S_p = \sum_{q=e,n,f} (A \rho'_s \Lambda_s^*)_q - \sum_{q=w,s,b} (A \rho'_s \Lambda_s^*) - \dot{M}_s$$

with

$$\sum_{q=e,n,f} (A \rho'_g \Lambda_g^*)_q = (A_x \rho'_g u^*)_e + (A_y \rho'_g v^*)_n + (A_z \rho'_g w^*)_f$$

and similar expressions for both phases on the other faces. The face densities,  $(\rho'_g)_e$ ,  $(\rho'_s)_e$ , etc., are evaluated using the upwind approximation.

- Summary -

Equation 4.15 is the pressure correction equation that is used to adjust the flow field pressures during the iterative solution. It should be noted that when Model B is used ( $\zeta_s = 0$ ), pressure does not appear in the solids momentum equations. And, as a result, the coefficient,  $a_g$ , involves only gas-phase parameters. In this situation, the solids mass residual is not included in the source term,  $S_p$ . Because the solids velocities (and solids continuity) are not directly affected by pressure gradients, it was felt that including the solids residual may cause solution difficulties. Consequently, when Model B is used, the pressure correction equation is based on just the gas continuity equation.

## 4.5 BOUNDARY CONDITIONS

The finite difference equations for mass and momentum conservation are modified to include the effects of obstacles, walls, and mass flow into or out of the flow domain. Boundary conditions for mass inflow and outflow are imposed by extra (also called dummy), non-computational control volumes located around the boundaries of the flow domain.

### 4.5.1 Inflow Boundaries

At inflow boundaries, the gas and solids velocities are prescribed on the faces of the Main Control Volume adjacent to the inflow port. The microscopic density and void fraction of the incoming stream are determined by the upwind scheme: they are determined by the values assigned to the boundary cells that are "upwind" of the interior Main volume. The temperatures, void fractions, and pressures of these boundary cells are changed only through input to FORCE2.

#### 4.5.2 Outflow Boundaries

A continuitive outflow condition is used for the steady simulation and is one option for the transient mode. A constant pressure condition is a second option in the transient mode. At all outflow boundaries, void fraction in the boundary nodes is determined by reflection, i.e., equal to void fraction in the adjacent node that is inside the computational domain.

4.5.2.1 Continuitive Outflow for Transient Mode. In the transient mode, pressure in the boundary node and node geometry are set by reflection. The outflow velocity is then calculated with the appropriate momentum equation. If the calculated velocity is into (instead out of) the flow domain, the velocity is set to zero. This prevents inflow at an outflow port, a situation that can introduce erroneous information into the flow field and indicates the nodalization is inadequate.

The procedure is illustrated below for continuitive outflow in the X-direction. The nodal arrangement at this outflow boundary is illustrated in Figure 4.4. Velocities, voids, pressures, and other quantities at the outlet are determined according to the following:

$$\Delta x_{i+1} = \Delta x_i$$

$$v_{i+1,j,k} = v_{i,j,k}$$

$$(\varepsilon_\phi)_{i+1,j,k} = (\varepsilon_\phi)^o_{i,j,k}$$

$$p_{i+1,j,k} = p^o_{i,j,k}$$

$$(u_\phi)_{i+3/2,j,k} = (\rho_\phi')_{i,j,k} \cdot (u_\phi)_{i+1/2,j,k} / (\rho_\phi')_{i+1,j,k}$$

$$(u_\phi)_{i+1/2,j,k} = \text{calculated with X-momentum equation}$$

$$(v_\phi)_{i+1,j+1/2,k} = (v_\phi)_{i+1,j-1/2,k} = 0$$

$$(w_\phi)_{i+1,j,k+1/2} = (w_\phi)_{i+1,j,k-1/2} = 0$$

4.5.2.2 Continuitive Outflow for Steady Mode. In the steady mode, an outflow velocity distribution is calculated based on global mass conservation

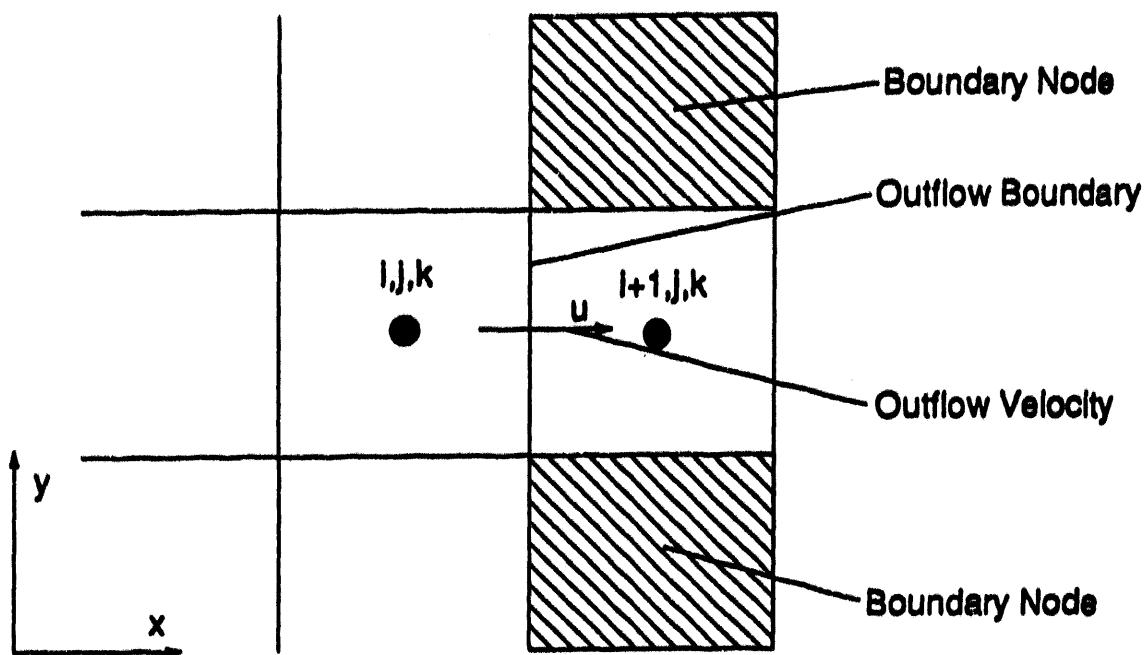


Figure 4.4. Outflow parallel to the X-direction - FORCE2 theory.

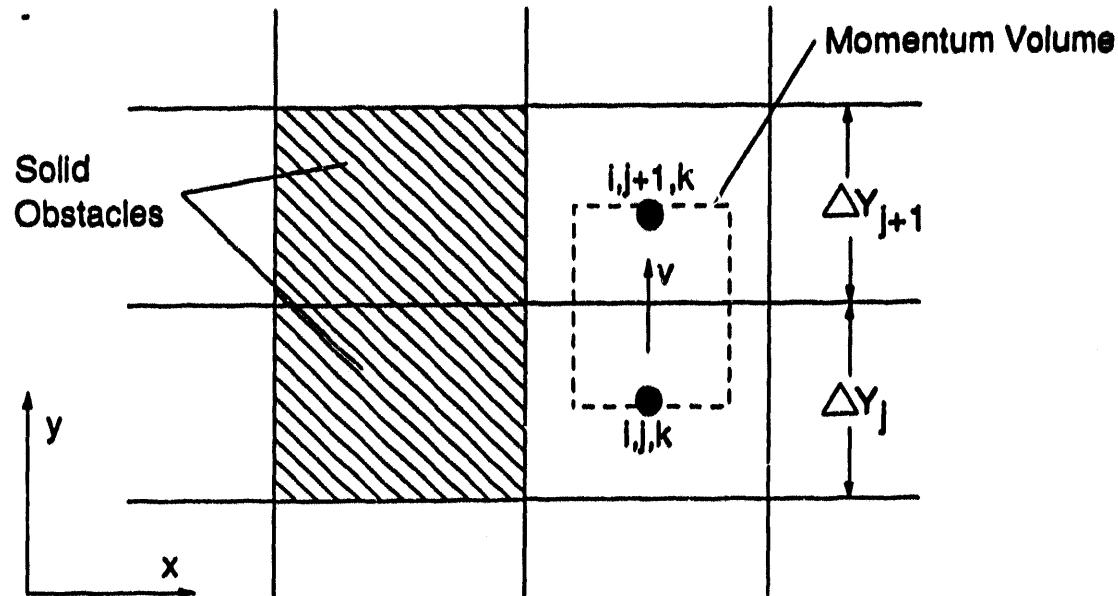


Figure 4.5. Flow in the Y-direction adjacent to a solid boundary - FORCE2 theory.

and the velocity distribution in the cells adjacent to the outflow boundary. Prior to solving the momentum and continuity equations, an initial distribution is calculated or an existing one modified based on mass conservation. The momentum and continuity equations are then solved using this distribution, resulting in a new velocity, void and pressure field. The outflow distribution is next modified based on the new velocities in the cells adjacent to the boundary. This procedure is repeated at each solution iteration and is outlined below for the case of outflow along the upper X boundary shown in Figure 4.4.

i) Calculate Distribution Prior to Momentum & Continuity Solutions Based on Total Mass Conservation

The outflow velocity in the X-direction is calculated according to:

$$(u_\phi)_{i+1/2,j,k}^{\text{iter}} = \bar{\dot{M}}_\phi \cdot (u_\phi)_{i+1/2,j,k}^{\text{iter-1}} \quad 4.17$$

where iter and iter-1 indicate velocities at the current and previous solution iterations, respectively, and

$$\bar{\dot{M}}_\phi = (\dot{M}_T)_\phi / \sum (A_x \rho'_\phi u_\phi)_{i+1/2,j,k}^{\text{iter-1}}$$

with

$(\dot{M}_T)_\phi$  = total outlet mass flow rate for phase  $\phi$  based on the velocity boundary conditions and mass sources for phase  $\phi$

$\sum$  = summation over all j and k nodes at  $i+1/2$ , i.e., over the upper X-face of all main control volumes adjacent to the outflow boundary. The macroscopic density,  $\rho'_\phi$ , is evaluated based on the upwind approximation.

For the first solution iteration (iter=1), if an outlet distribution is not specified, a uniform profile is calculated according to:

$$(u_\phi)_{i+1/2,j,k}^1 = (\dot{M}_T)_\phi / \sum (A_x \rho_\phi^0)_{i+1/2,j,k}^0$$

where

$(\rho_\phi^0)^0$  = macroscopic density based on the initial void fraction and phase density

ii) Revise Distribution Based on Interior Velocities After Solutions to Momentum & Continuity

The distribution is revised according to:

$$(u_\phi)_{i+1/2,j,k}^{\text{iter}} = \bar{\dot{M}}_\phi [(u_\phi)_{i-1/2,j,k} - u_m]^{\text{iter}} \quad 4.18$$

where  $(u_\phi)_{i-1/2,j,k}$  is an interior velocity that has been calculated from mass and momentum conservation and  $u_m$  a minimum velocity defined according to:

$u_m = 0.0$  or  
= the minimum of all negative  $(u_\phi)_{i-1/2,j,k}$  velocities

and

$$\bar{\dot{M}}_\phi = (\dot{M}_T)_\phi / \sum (A_x \rho_\phi^0 u_d)_{i+1/2,j,k}^{\text{iter}}$$

The summation is as defined in step i and the velocity difference,  $u_d$ , is given by:

$$u_d = (u_\phi)_{i-1/2,j,k} - u_m$$

The distribution from this step is then used in step i at the start of the next solution iteration.

With the above formulation, the distribution satisfies global continuity, a constraint that improves solution convergence, and there are no inflow velocities at the outflow boundary. The minimum velocity,  $u_m$ , has been

added to the formulation in step ii to prevent an inflow condition that may arise during the iterative solution of the non-linear momentum and continuity equations. It should be noted that if negative velocities are observed near the outflow boundary during the solution, the nodalization is probably inadequate. Specifically, the flow exit should be located far enough downstream of the problem to avoid flow recirculation in the exit plane.

4.5.2.3 Constant Pressure, Transient Mode Only. The constant pressure condition is identical to continuitive outflow except the pressure in the boundary node, node  $i+1,j,k$  in Figure 4.3, is fixed according to the value that is input to the model.

#### 4.5.3 Velocities Normal to Solid Surfaces

The velocities normal to solid boundaries are initially set to zero. In the transient mode momentum equations for the normal velocities are not solved thereby maintaining this condition. In the steady mode, the velocity coefficients and source terms are modified to maintain a zero velocity during the solution sequence. For example, the Z-Momentum equations, Equation 4.14, are modified as follows for a solid boundary normal to the Z-direction:

$$a_e = a_w = a_n = \dots a_b = 0$$

$$\beta_w = 0$$

$$S_s = S_{\Delta p} = 0$$

$$C_{\phi_s} = 1.0$$

This results in a zero velocity in the Z-direction at the solid boundary. A similar formulation is used for surfaces normal to the X- and Y-directions.

#### 4.5.4 Velocities Parallel to Solid Surfaces

Three boundary conditions have been included to account for solid boundaries such as walls or obstacles in the flow field. These conditions are

applied at blocked cells, along blocked boundaries or along symmetry boundaries. To illustrate the approach, the formulation for the Y-momentum equation is given for flow adjacent to the solid surface shown in Figure 4.5. The formulation is similar for the other directions.

4.5.4.1 No Slip. For this condition, the phase velocity is assumed to be zero at the solid surface. The viscous stress on the face of the Y-momentum volume adjacent to the solid boundary is given by:

$$(A \mu' \frac{\partial v_\phi}{\partial x})_w = \bar{\mu}'_\phi \cdot (v_\phi)_p = \bar{\mu}'_\phi (v_\phi)_{i,j+1/2,k}$$

where

$$\bar{\mu}'_\phi = [(A_x)_{i-1/2,j,k} \cdot (\mu')_{i,j,k} + (A_x)_{i-1/2,j+1,k} \cdot (\mu')_{i,j+1,k}] / \Delta x_i$$

For the transient mode, this boundary condition is implemented by redefining the west face and node center diffusion coefficients in the tilde term as:

$$D_w = 0.0$$

$$D_p = \sum D_q + \bar{\mu}'_\phi$$

For the steady mode, a similar approach is taken to give:

$$D_w = 0$$

$$C_{\phi y} = \sum a_q + F_p + (\beta_y + k_{\phi y})_p \cdot v_p + \bar{\mu}'_\phi$$

4.5.4.2 Partial Slip. The solid particles are assumed to move along the solid surface in this situation. The velocity along the surface is given according to the relation in section 3.5 in terms of the mean free path, L. The viscous term may then be written as above with

$$\overline{u'_s} = [(A_x)_{i-1/2,j,k} \cdot (u'_s)_{i,j,k} + (A_x)_{i-1/2,j+1,k} \cdot (u'_s)_{i,j+1,k}] \cdot \frac{2}{\Delta x_j + 2L}$$

where

$L$  = mean free path defined in section 3.5

This boundary condition is included in the transient and steady modes as noted in section 4.5.4.1 using the above definition for  $\overline{u'_s}$ .

4.5.4.3 Full Slip or Symmetry Boundary. In this situation, the phase velocity at the solid surface or boundary is assumed to be the free stream value. As a result, the above velocity gradient is zero. This condition is implemented in the transient and steady modes with

$$\overline{u'_s} = 0$$

in the formulations given in section 4.5.4.1.

#### 4.5.5 Mass Sources & Sinks

Momentum equations are not solved in momentum volumes connecting adjacent mass sources or sinks. The treatment in both the transient and steady mode is the same that for velocities normal to solid surfaces, section 4.5.3.

#### 4.6 INITIAL CONDITIONS

Initial conditions are provided to the model through input. For the first run, initial velocities and voids must be specified throughout the flow field. A minimum fluidizing condition may also be selected on the first run. In this situation, the solids are assumed at rest, the gas flow field is assumed one dimensional and the pressure field is set to just support the weight of the bed. These conditions assume that the bed is at minimum fluidization.

During the solution, initial conditions for the next timestep in the transient mode or next iteration in the steady mode come from the predicted velocities, voids, pressures, etc., at the end of the previous solution step. At the end of a run, predicted conditions are written to a file and can subsequently be used to start up the solution.

#### 4.7 RESIDUES AND RELAXATION OF STEADY EQUATIONS

Residues for each of the steady equations are calculated and displayed as a means of monitoring solution progress. Convergence is indicated by residues that decrease as the solution progresses. The residues are defined below and used as part of the relaxation schemes.

Because the finite difference equations are very nonlinear, they must be relaxed to promote a stable solution. Two schemes are applied: i) under-relaxation and ii) inertia relaxation. Both are outlined below.

All the steady equations may be written in the form:

$$a_{pu} \phi_p = \sum a_q \phi_q + b_p \psi_p + s_{cu} \quad 4.19$$

where

$$\sum a_q \phi_q = a_e \phi_E + a_w \phi_W + \dots a_b \phi_B$$

$a_{pu}, s_{cu}$  = unrelaxed coefficient and source term, respectively

and the parameters are defined in Table 4.2, depending on the particular steady equation.

##### 4.7.1 Equation Residue

A residue for Equation 4.19, at the beginning of the current iteration (iter), is defined as:

$$R_\phi = \sum a_q^{\text{iter}} \cdot \phi_Q^{\text{iter-1}} + b_p^{\text{iter}} \cdot \psi_p^{\text{iter-1}} + s_{cu}^{\text{iter}} - a_{pu}^{\text{iter}} \cdot \phi_p^{\text{iter-1}}$$

where  $R_\phi$  indicates the residual in the equation for variable  $\phi$ .

Table 4.2  
TERMS IN THE GENERAL FORCE STEADY TRANSPORT EQUATION

<u>Equation</u>	$\phi_p$	$\psi_p$	$a_{pu}$	$b_p$	<u><math>s_{cu}</math></u>
Void fraction, Equation 4.11	$\epsilon_g$	0	$a_p$	0	$s_\epsilon$
X-Momentum, Equation 4.12	$u_\phi$	$u_{\phi_0}$	$C_{\phi_x}$	$\beta_u$	$s_x - s_{\Delta p}$
Y-Momentum, Equation 4.13	$v_\phi$	$v_{\phi_0}$	$C_{\phi_y}$	$\beta_v$	$s_y - s_{\Delta p}$
Z-Momentum, Equation 4.14	$w_\phi$	$w_{\phi_0}$	$C_{\phi_z}$	$\beta_w$	$s_z - s_{\Delta p}$
Pressure Correction, Equation 4.15	$P'$	0	$C_p$	0	$s_p'$

The coefficients and source term at the start of the current integration are based on predicted conditions at the end of the previous solution step, i.e., on  $\phi^{\text{iter-1}}$ , etc. The maximum and average values for  $R_\phi$  are displayed for each solution variable at the start of each solution step.

#### 4.7.2 Relaxation of Steady Equations

Under-relaxation and inertia relaxation are applied to the steady equations before they are solved. Each relaxation method is outlined below.

4.7.2.1 Under-relaxation. The new estimate for  $\phi_p$  is written as:

$$\phi_p^{\text{iter}} = \theta_\phi \cdot \phi_{p,\text{calc}} + (1 - \theta_\phi) \phi_p^{\text{iter-1}} \quad 4.20$$

where  $\phi_p^{\text{iter}}$  = "new" value for  $\phi_p$

$\theta_\phi$  = under-relaxation parameter,  $0 < \theta_\phi \leq 1$

$\phi_{p, \text{calc}} = \phi_p$  calculated with Equation 4.19

$\phi_p^{\text{iter-1}} = \phi_p$  from the previous iteration

The calculated value,  $\phi_{p, \text{calc}}$ , in Equation 4.20 is used in Equation 4.19 along with the residual,  $R_\phi$ , to define  $\phi_p^{\text{iter}}$  as

$$a_{pu} \cdot \phi_p = \sum a_q \phi_0 + b_p \psi_p + s_{cu} - (1 - \theta_\phi) R_\phi$$

where  $\phi_p$  is the new value. The superscript, iter, is not used to simplify the notation.

4.7.2.2 Inertia Relaxation. For inertia relaxation, we assume

$$a_{pu} \cdot \phi_p = a_{pu} \cdot \phi_p \left[ \frac{1}{1 - I} - \frac{I}{1 - I} \right]$$

$$= \frac{a_{pu} \phi_p}{1 - I} - \frac{a_{pu} I \cdot \phi_p^{\text{iter-1}}}{1 - I}$$

where  $I$  is the inertia relaxation parameter defined on the range

$$0 \leq I < 1$$

With inertia relaxation, the governing equation for  $\phi_p$  becomes

$$a_p \cdot \phi_p = \sum a_q \phi_0 + b_p \psi_p + s_c$$

4.21

where

$$a_p = \frac{a_{pu}}{1 - I}$$

$$S_c = S_{cu} - (1 - \theta_\phi)R_\phi + \frac{a_{pu} \cdot I}{1-I} \cdot \phi_p^{iter-1}$$

#### 4.7.3 Summary

Equation 4.21 is the final form for all the steady equations. This equation is solved iteratively, as described in section 6.0, to predict the steady-state operating conditions.

## 5.0 TRANSIENT SOLUTION PROCEDURE

### 5.1 OVERVIEW

The implicit multifield technique (IMF) (Harlow 1975) is the transient solution procedure used in FORCE2 and is based primarily on the IMF scheme as implemented in FLUFIX. The solution procedure is based on adjusting node pressure until gas mass is conserved. Some key features of the method include:

- o Cell-by-Cell solution. The solution for operating conditions at the end of the timestep is developed by solving the governing equations on a cell-by-cell basis. Sweeps over the flow domain are performed until gas mass is conserved in all cells.
- o Simultaneous solution for velocities on the cell faces. The gas and solids momentum equations on the face of a Main Control Volume are solved simultaneously. This improves the modeling and contributes to solution stability.
- o Explicit formulation for convection and diffusion. As noted in Section 4.3, the convection and diffusion terms are evaluated only at the beginning of the timestep. This rough approximation simplifies the formulation. A better approach would be to include these terms in the velocity coefficient on the left side of the momentum equations, similar to the steady formulation.
- o Implicit formulation for the solids stress. The solid stress term is evaluated based on void fractions at the end of the timestep. This formulation is required for a stable solution due to the exponential variation of stress with gas void near the packed state.

The overall transient solution scheme is given below.

## 5.2 OVERALL TRANSIENT SOLUTION METHOD

The solution procedure over a problem timestep consists of the following steps:

### Step 1: Calculate Densities, Viscosities and the Explicit Tilde Terms.

Fluid densities and viscosities are calculated throughout the flow field based on conditions at the beginning of the timestep. The explicit tilde terms that include body forces, convection and diffusion of momentum are also calculated for each momentum volume in the flow field. The initial conditions come from user input or from operating conditions at the end of the previous timestep.

### Step 2: Solve for Velocities Throughout Flow Field.

This is called the explicit stage of the solution (Lyczkowski, 1989). This step results in new velocities throughout the flow field based on the initial pressure field. Velocities are predicted by the simultaneous solution of the gas and solids momentum equations as outlined in sections 4.3.2, 4.3.3, and 4.3.4.

### Step 3: Start the Implicit Phase of the Solution.

The solution is started at the first computational cell located at the Minimum X1-Coordinate, Maximum X2-Coordinate, and Minimum X3-Coordinate in the FORCE2 coordinate system. For the discussion here, the X1-Direction will correspond to X, the X2 to Y and X3 to Z. Main Control Volumes are visited in the following order: at each plane perpendicular to the X1-Axis, visit nodes along X3-lines starting at the minimum X3-coordinate. In a two-dimensional problem, this is equivalent to visiting all j-nodes at each i-location. In the description below, the Main Control Volume will be designated by the indices i,j,k.

Step 4: Check Mass Convergence at Node i,j,k.

The gas mass residue is calculated according to Equation 4.4 and convergence checked. A converged solution at this node is assumed if

$$|DG_p \cdot \Delta t / v_p| < EPSG \cdot (\rho'_g)_p^0$$

where EPSG is the input convergence parameter.

If the convergence criteria is met, go to Step 5, otherwise go to Step 6.

Step 5: Calculate Void Fraction.

The gas void fraction is predicted using Equation 4.3. If an unrealistic void is predicted, warning messages are written to the FORCE2 log file and the solution is stopped. Experience indicates that near the packed state the solid stresses can give rise to very high velocities which, in turn, can result in physically unrealistic voids. Exceeding the material Courant criteria is one condition that can lead to this situation. Otherwise, the solution then continues to Step 11.

Step 6: Adjust Node Pressure.

Two procedures for adjusting the pressures in Main Control Volume i,j,k are used. They are:

i) Apply Pressure Correction Equation, Equation 4.8. This correction is applied until minimum and maximum values are obtained.

ii) Regula-Falsi Using Minimum and Maximum Values. The pressure in the Volume is corrected using

$$P' = P'_{\max} - DG_{\max}/m$$

where

$$m = (P'_{\max} - P'_{\min}) / (DG_{\max} - DG_{\min})$$

$DG_{\max}, P'_{\max}$  = maximum residual and associated pressure correction

$DG_{\min}, P'_{\min}$  = minimum residual and associated pressure correction

The maximum and/or minimum is revised based on the residual and associated pressure correction in the current iteration before the above correction is calculated. The minimum or maximum is determined by:

$$DG_p > 0 \quad P'_{\max} = P'$$

$$DG_p \leq 0 \quad P'_{\min} = P'$$

#### Step 7: Calculate Velocities on Faces of Main Volume i,j,k.

The momentum equations on each face of the control volume, Equations 4.5, 4.6, and 4.7, are solved to predict gas and solids velocities. The coupled equations on each face are solved simultaneously.

#### Step 8: Calculate Void Fraction.

The gas void fraction is predicted using Equation 4.3. As noted in Step 5, if the predicted void is physically unrealistic, warning messages are written to the FORCE2 log file and the solution is stopped.

**Step 9: Check Mass Convergence at Node  $i,j,k$ .**

This step is identical to Step 4 above. If convergence is achieved, the solution continues at Step 11; otherwise, it continues at Step 10.

**Step 10: Check Number of Micro Iterations.**

Steps 6 through 9, which are performed at each Main Control Volume, comprise a Micro-Iteration. The allowable number of these steps is controlled by input to the model (the allowable number of Micro-Iterations is typically 5 to 10). If the allowable number is exceeded, the solution proceeds to Step 11; otherwise it returns to Step 6.

**Step 11: Check for Last Node.**

If node  $i,j,k$  is the last node in the domain, one Macro-Iteration has been completed and the solution continues at Step 12. Otherwise, the node number is incremented according to the scheme outlined in Step 3 and the solution continues at Step 4.

**Step 12: Check for Overall Convergence or Maximum Allowable Macro Iterations.**

Overall convergence is achieved when only steps 4,5, and 11 are performed during a single sweep (Macro-Iteration) over the domain. If this criteria is met, a new timestep is started at Step 1.

If overall convergence is not achieved and the maximum allowable number of Macro-Iterations is not exceeded, a new sweep over the domain is started. The Main Control Volume number is reset to the starting value defined in Step 3 and the solution restarts at Step 4.

If overall convergence is not achieved and the maximum allowable number of Macro-Iterations is exceeded, a new timestep is started at Step 1. This non-converged condition can be detected by examining the FORCE2 log file which displays convergence information.

## 6.0 SOLUTION PROCEDURE FOR STEADY SIMULATION

### 6.1 OVERVIEW

Because the finite difference form of the momentum and continuity equations are very non-linear, an iterative solution procedure is required. The approach is to construct sets of linear equations by evaluating velocity, void fraction, etc., coefficients based on assumed or trial operating conditions. The resulting linear equations are solved using matrix methods and their solutions used to estimate new operating conditions. This sequence is continued until the equation residuals are small indicating that an acceptable solution has been achieved. The non-linear nature of the governing equations also requires that the solution be advanced slowly. Consequently, solutions are developed based on governing equations that have been relaxed according to the methods described in section 4.6. Finally, the implementation of this method in FORCE2 is outlined in section 7.0. Program flow and logic are described there.

A key element of the solution scheme is the adjustment of the flow field pressure to conserve mass in each computational cell. The iterative procedure is based on conserving mass once a solution is achieved. However, during the iterations mass residuals are present indicating that mass has been "added or taken from" the flow domain. The local imbalance is used to adjust pressure which in turn causes the required changes in the phase velocities. This scheme along with inflow boundary conditions results in a solution in which mass is conserved with a certain inventory of the phase in the flow field. For the case of no inflow boundary conditions, such as the solids phase in a bubbling bed, the situation is different. Mass residuals will be present during the solution and can be used to adjust pressure. However, because there are no local constraints on solids mass, such as the storage term in the transient continuity equation, solids mass may not be conserved from start to finish of the solution, that is the solids inventory after several iterations may not be equal to the inventory at the start of the problem. The "loss" or "gain" of solids mass is further "enhanced" by the relaxation of the void equation which is needed to stabilize the solution. Consequently, a constraint of global mass conservation for the solids phase is used to adjust void fraction during the solution. This constraint is applied only for the case of no inflow of the solids phase.

The steady solution procedure is outlined below. It is a modified version of SIMPLER (Patankar 1980) developed by Schnipke (1986).

## 6.2 OVERALL SOLUTION PROCEDURE

The following steps are performed during a single solution iteration:

### Step 1: Evaluate Microscopic Densities and Viscosities

Based on operating conditions from the previous iteration and user input, microscopic densities and viscosities are evaluated throughout the flow domain.

### Step 2: Calculate or Revise Outlet Velocity Distribution

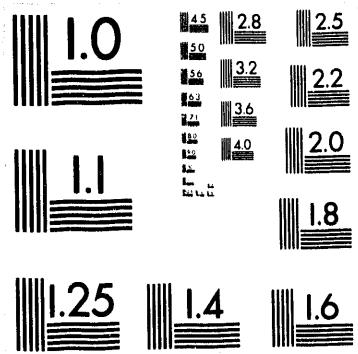
The outlet velocity distribution for each phase is calculated (initial run) or revised (all runs after initial one) based on total mass conservation of the phase. The procedure is outlined in section 4.5.2.2, step ii) .

### Step 3: Solve the X-Momentum Equations for the Trial Velocities, $u_t^i$

The X-Momentum equations are solved to predict trial velocities, denoted  $u_t^i$ , throughout the flow domain based on pressures and void fractions from the previous iteration. The coupled gas and solids equations are solved along lines in the domain using the Coupled Tri-Diagonal Matrix Algorithm (CTDMA, Fortino 1989) described in Appendix A. The solution is iterative in which many sweeps over the domain (the number and direction of sweeps are input parameters to FORCE2) are performed. The solution for the trial velocities consist of two steps:

#### i) Solve the Relaxed Momentum Equations.

The relaxed forms of the X-Momentum Equations, Equation 4.21, are solved over the domain along lines using CTDMA.



2 of 2

ii) Calculate Trial Velocities from Unrelaxed Momentum Equations.

The trial velocities,  $u_g^*$  and  $u_s^*$ , are calculated from the unrelaxed momentum equations, Equation. 4.12, in point form. The neighboring velocities at a node and the other phase velocity in the drag term are based on the velocities from step i. The source terms are based on the trial pressures, void fractions and velocities at the beginning of the iteration.

**Step 4: Solve the Y-Momentum Equations for the Trial Velocities,  $v_\phi^*$**

**Step 5: Solve the Z-Momentum Equations for the Trial Velocities,  $w_\phi^*$**

Steps 4 and 5 (if a 3-D problem) are identical to Step 3. All the momentum equation coefficients are evaluated based on velocities at the start of the iteration, not the trial velocities.

**Step 6: Solve for Pressure Corrections.**

The trial velocities,  $u_\phi^*$ ,  $v_\phi^*$  and  $w_\phi^*$ , are now used to formulate the pressure correction equation, Equation. 4.15. The equation is not relaxed and is solved along lines using a standard tri-diagonal matrix solution algorithm (Patankar, 1980). The solution is iterative with many sweeps over the domain (controlled by input to FORCE2) being performed.

**Step 7: Correct Pressure and Velocities**

The pressure corrections from step 6 are applied to the trial pressure and velocity fields according to

$$P = P^* + P'$$

$$(\Lambda_\phi)_q = (\Lambda_\phi^*)_q - \theta_\Lambda \cdot (D\Lambda_\phi)_q \cdot \Delta P'_q$$

where

$P^*$  = trial pressure field  
 $(\Lambda_\phi^*)_q$  = trial velocity field from steps 3, 4 and 5  
 $(D\Lambda_\phi^*, \Delta P'_q)$  = parameter defined in section 4.4.6  
 $\theta_\Lambda$  = under-relaxation factor for phase velocity,  $\Lambda$ , an input to FORCE2

The resulting pressure and velocity fields,  $P$  and  $(\Lambda_\phi)_q$ , become the starting values for the next solution iteration.

#### Step 8: Solve for Gas Void Fraction

The gas and solids velocities from step 7 are used to formulate the gas void equation, Equation 4.11. The resulting equations for void throughout the domain are relaxed and solved using a tri-diagonal matrix solution algorithm. Like the solution of the momentum and pressure correction equations, many sweeps along lines in the domain are performed to develop the void field.

If there is solids inflow to the field, the predicted voids are used in the next solution iteration and the solution continues to Step 9.

If there is no solids inflow to the field, the predicted voids are adjusted so that global solids mass is conserved according to the following:

$$\epsilon_g^* = \epsilon_g^* - \theta_{es} \cdot f_\epsilon \cdot (1 - \epsilon_g^*)$$

and

$$\epsilon_s = 1 - \epsilon_g$$

where

$\epsilon_g$  = predicted gas void to be used in next solution iteration  
 $\epsilon_g^*$  = predicted gas void from the solution of the void equation  
 $\theta_{\epsilon_s}$  = under-relaxation factor for the solids void fraction, an input to FORCE2

The parameter,  $f_{\epsilon}$ , is determined based on the initial solids inventory as:

$$f_{\epsilon} = M_s / \sum \rho_s \cdot V \cdot \epsilon_s^* - 1.0$$

with

$M_s$  = total solids mass based on the user-specified void distribution and solids density at the start of the problem

$\sum$  = summation over all computational main control volumes in the flow field

$$\epsilon_s^* = 1 - \epsilon_g^*$$

$V$  = volume of the main control volume

The resulting voids are then used in the next solution iteration.

#### Step 9: Adjust Outflow Velocity Distribution

The outflow velocity distribution for each phase is adjusted based on the new interior velocities according to the approach outlined in section 4.5.2.2, step ii). A new iteration is now started at Step 1.

## 7.0 STRUCTURE OF FORCE2

### 7.1 OVERVIEW

This section describes how the hydrodynamic models, solution procedures and complementary input and output routines are implemented in the FORCE2 program. Two major computational models, transient and steady, have been installed in FORCE2. The organization of the code is best explained in terms of two almost distinct modules that implement each of these models. Each module uses common set-up (input, geometry, calculations, properties, data structure) and output (printer output, post-processor data, restart data) routines and do share some calculational routines such as drag coefficient evaluation. The routines in each module, the complementary routines, and the structure of FORCE2 are outlined in the following sections.

### 7.2 PROGRAM ORGANIZATION AND FLOW

The execution flow is divided into three distinct phases: input, computation and output (Figure 7.1). The flow in the first two phases is controlled by three routines: the main program, FORCE2; the computational routine, CNTRL2, which implements the steady simulation; and the computational routine, CNTRLT, which implements the transient simulation. The output phase is controlled by a collection of routines that perform a distinct function and that may be used periodically during the computational phase. The output routines include PRINT, which writes predictions for subsequent printing; RESTRT, which writes a restart file; and POSTA, which writes a file for the FORCE2 post-processor.

During the input phase, data is read from the input file and the restart file. Most of the restart file is also written during the input phase. The input phase is controlled by the main routine, FORCE2, which calls the input routines. The sequence of subroutine calls during the input phase is shown in Figure 7.2.

The fundamental modeling calculations are performed during the calculation phase. Overall execution flow is controlled by routine CNTRL2 for

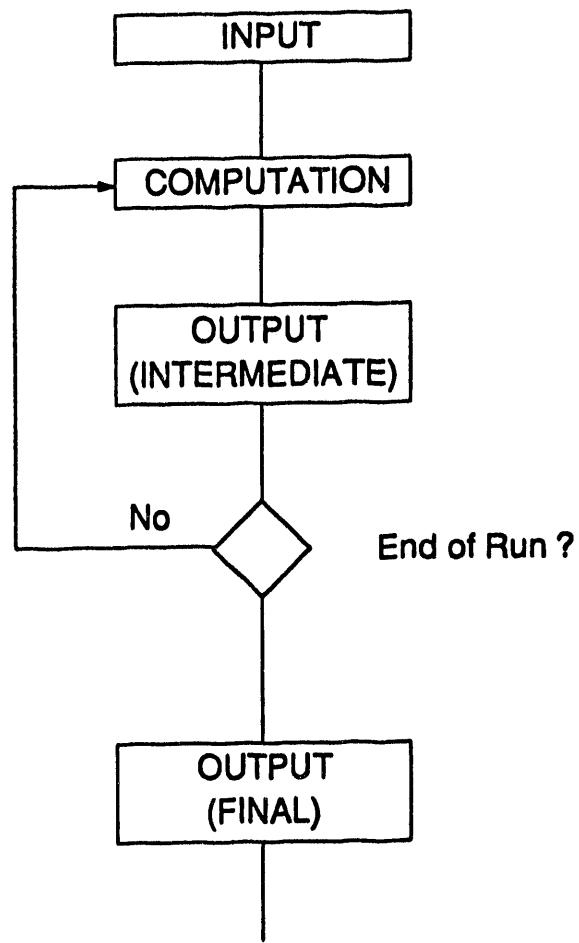


Figure 7.1. Overall FORCE2 execution flow - FORCE2 theory.

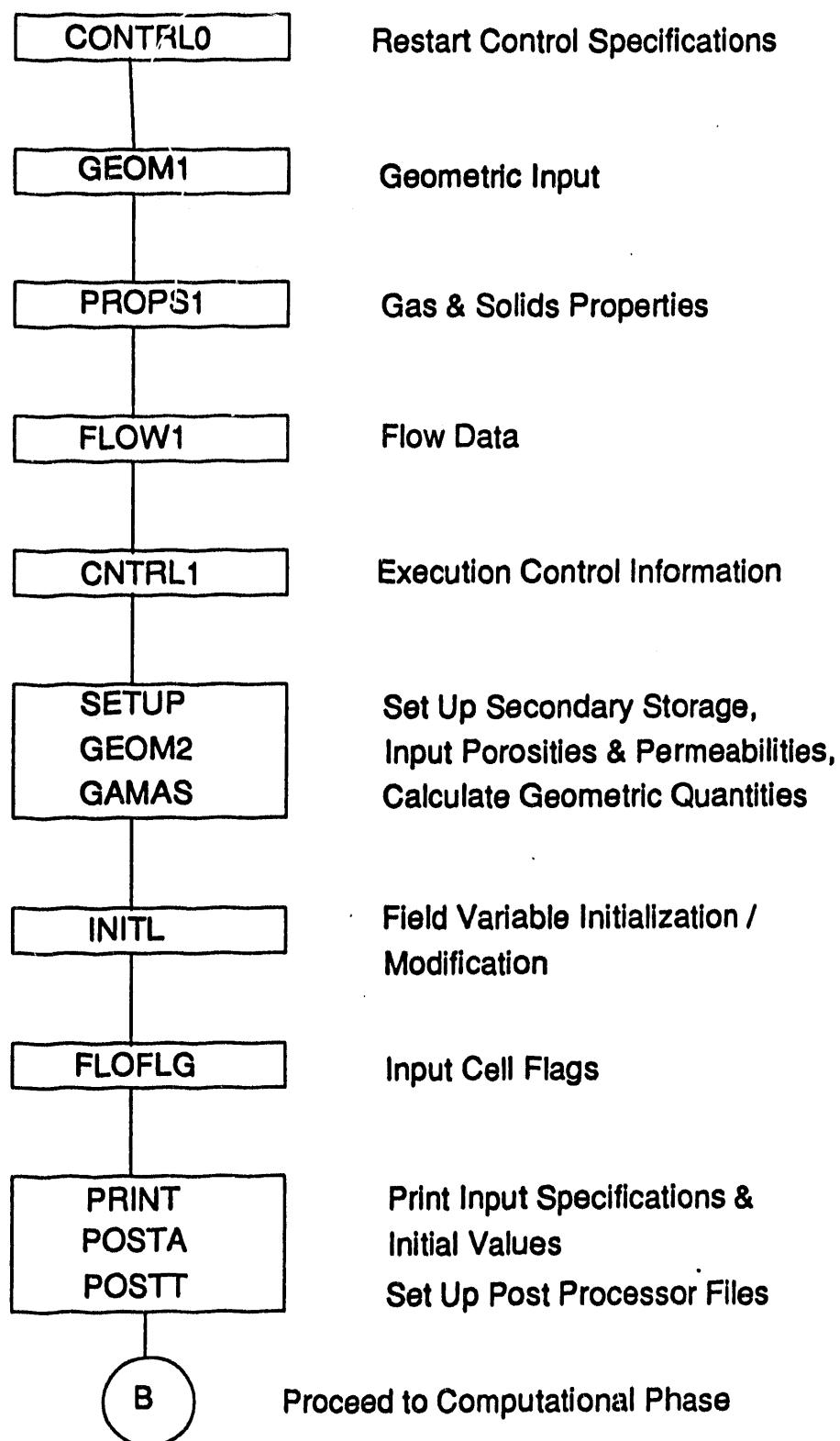


Figure 7.2. FORCE2 input phase - FORCE2 theory.

a steady simulation and by CNTRLT for a transient simulation. The flow in each routine is shown schematically in Figures 7.3 and 7.4.

Results of the modeling calculations are printed intermittently as well as at the end of the calculational phase. Overall execution of the printing is controlled by routine PRINT and is shown schematically in Figure 7.5. Prior to calling print after the calculational phase, the main program calls routine RESTART to write the field variables to the restart file and routines POSTA and POSTT to complete the post-processor files.

### 7.3 GEOMETRIC QUANTITIES

FORCE2 is based on a three-dimensional Cartesian coordinate system ( $X_1, X_2, X_3$ ). The geometric solution domain is a rectangular grid. The number of nodes in each direction and coordinate of cell faces are stored in common block FGEOM with the variable names given in Table 7.1. The rectangular grid system is shown in Figure 7.6. It should be noted that two-dimensional problems use the  $X_1$  and  $X_2$  coordinates. The  $X_3$  coordinate is used only for three-dimensional simulations.

Most geometric quantities vary from cell-to-cell and are, therefore, stored for each cell. The geometric quantities maintained for each cell are listed in Table 7.2.

Each of the geometric quantities is stored as a field variable. Every geometric quantity is associated with a main node and identified in the same way as the main node. Quantities associated with faces of cells (between main nodes) take the identify of the main node in the positive coordinate direction normal to the face. Figure 7.7 shows the relation between the geometric quantities and the main nodes.

The linear interpolation factor,  $FF_n$ , is used to interpolate values at main nodes to faces. The interpolation factor is based on the distance between the main nodes ( $\Delta x$ ) and the distance from the main node to the face ( $\delta x$ ). See Figure 7.8.

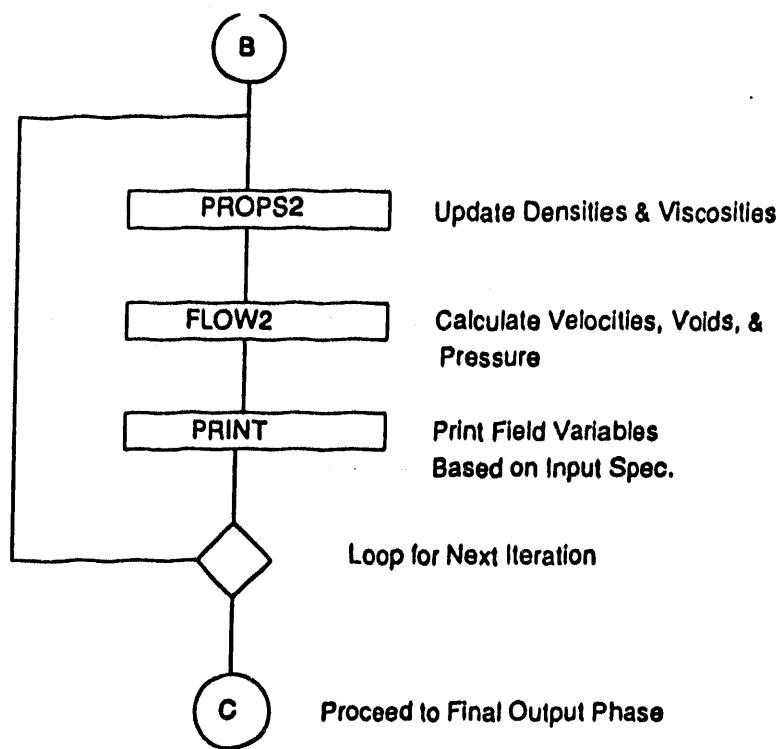


Figure 7.3(a). Execution flow in routine CNTRL2 - FORCE2 theory.

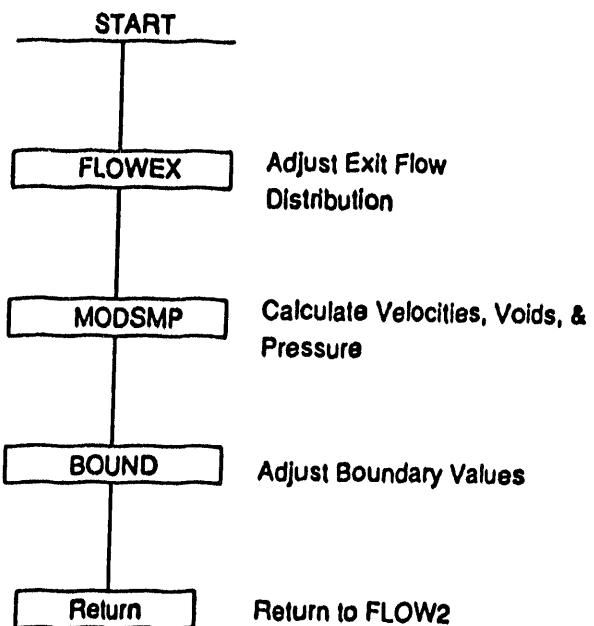


Figure 7.3(b). Execution flow in routine FLOW2 - FORCE2 theory.

Figure 7.3. Calculation phase for a steady simulation in FORCE2 - FORCE2 theory.

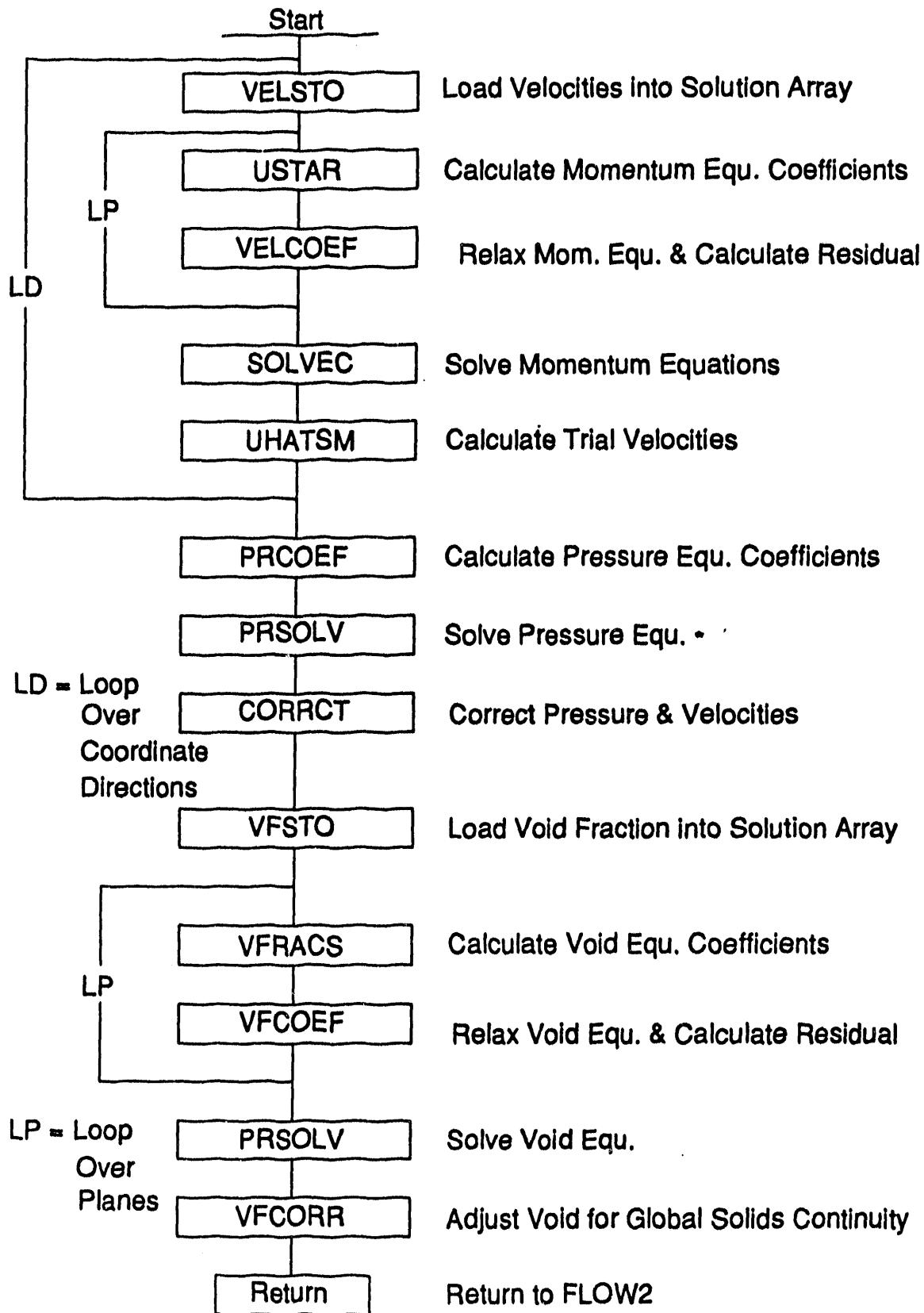


Figure 7.3(c). Execution flow in routine MODSMP - FORCE2 theory.

Figure 7.3. Calculation phase for a steady simulation in FORCE2 - FORCE2 theory. (continued)

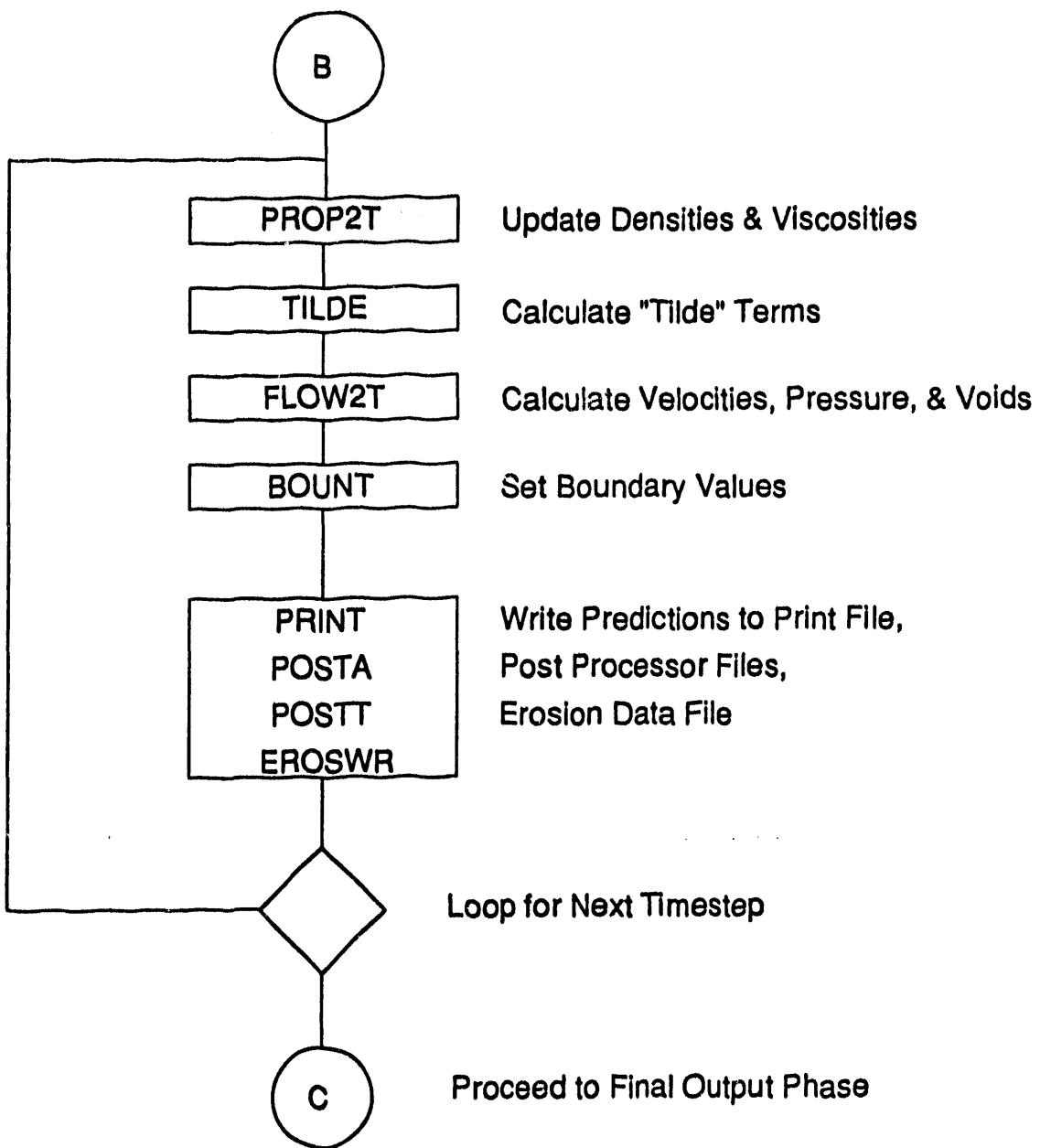
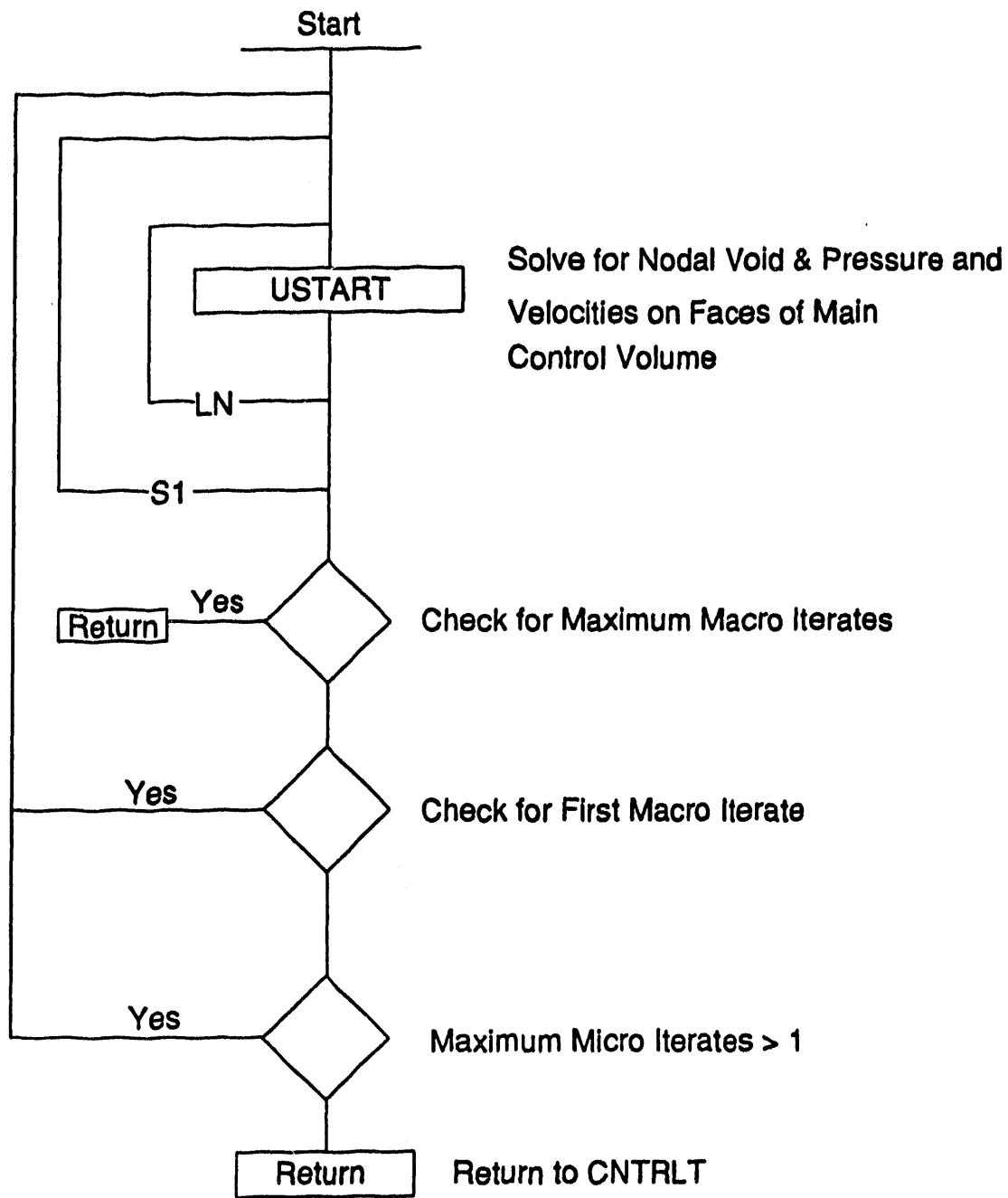


Figure 7.4(a). Execution flow in routine CNTRLT - FORCE2 theory.

Figure 7.4. Calculation phase for a transient simulation in FORCE2 - FORCE2 theory.



S1 = Sweep Along X1-Axis

LN = Loop Over Main Control Volumes in Plane

Figure 7.4(b). Execution flow in routine FLOW2T - FORCE2 theory.

Figure 7.4. Calculation phase for a transient simulation in FORCE2 - FORCE2 theory. (continued)

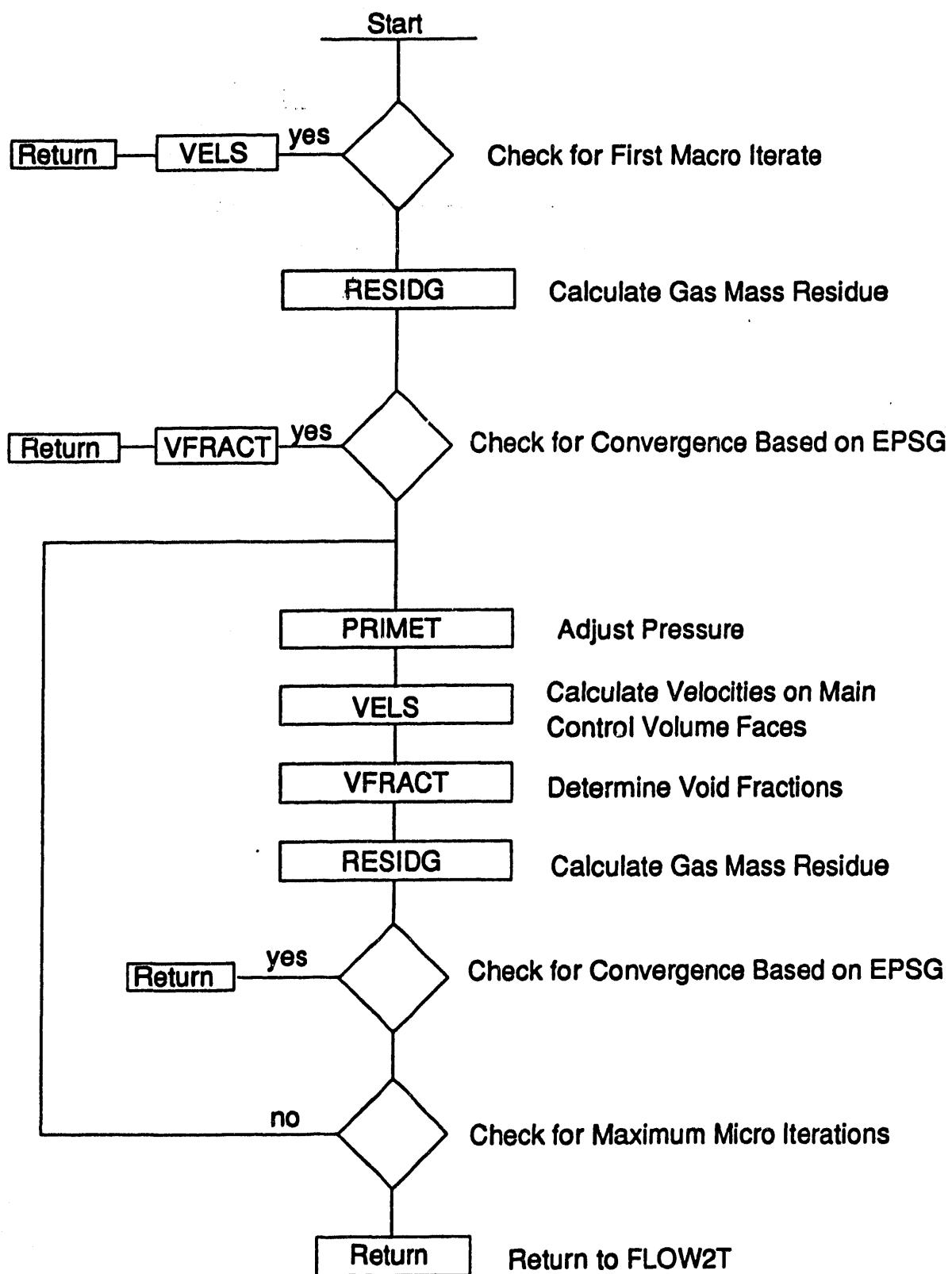


Figure 7.4(c). Execution flow in routine USTART - FORCE2 theory.

**Figure 7.4.** Calculation phase for a transient simulation in FORCE2 - FORCE2 theory. (continued)

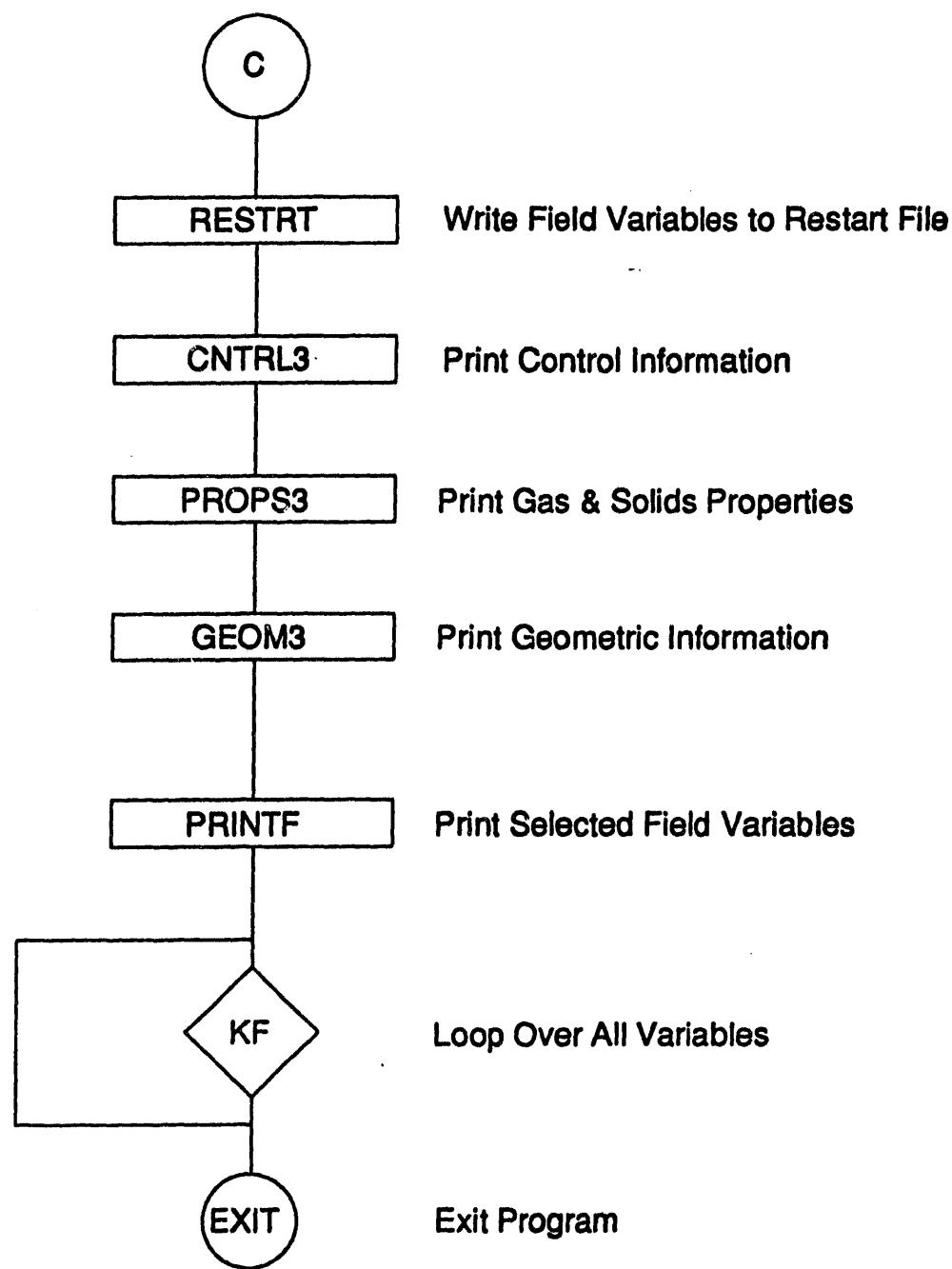


Figure 7.5. Final output phase of FORCE2 - FORCE2 theory.

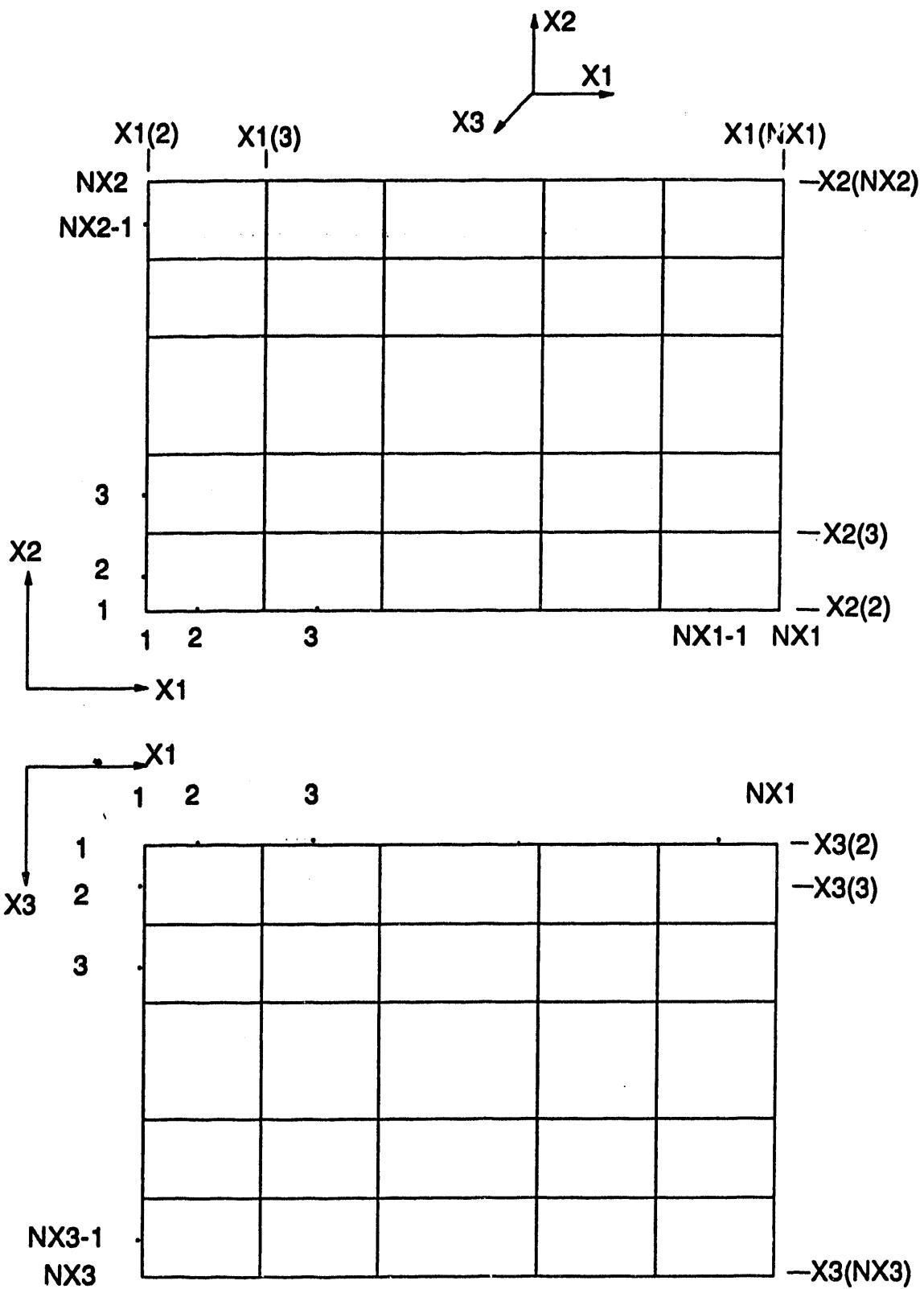


Figure 7.6. FORCE2 grid system - FORCE2 theory.

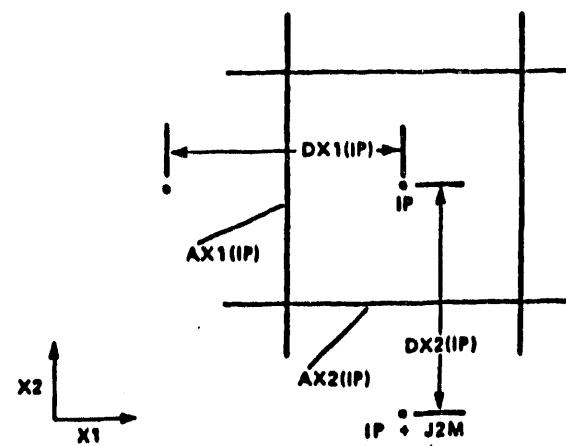


Figure 7.7. Geometric quantities in FORCE2 - FORCE2 theory.

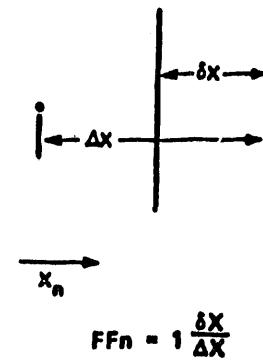


Figure 7.8. Interpolation factors in FORCE2 - FORCE2 theory.

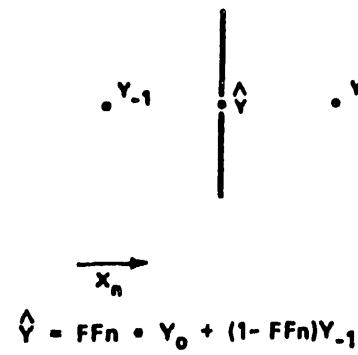


Figure 7.9. Main control volume interpolation in FORCE2 - FORCE2 theory.

$$FF_n = 1 - \frac{\delta x}{\Delta x}$$

Therefore, the distance from a main node to the face below it is calculated:

$$\delta x = D_x * (1 - FF_n)$$

Most quantities, both geometric and dependent variables, are associated with main nodes located at the center of a cell. To interpolate values stored at main nodes to faces, the following expression is used:

$$y = FF_n * Y_0 + (1 - FF_n) Y_{-1}$$

Where  $Y_0$  is the value at the main node above the face and  $Y_{-1}$  is the value at the main node below the face (see Figure 7.9). The interpolation factor is identified with the main node above the face (i.e., the same as  $Y_0$ ).

Finally, the storage scheme for main control volume and momentum control volume quantities should be noted. Quantities on the faces of main control volumes (velocities, areas, etc.) are stored in a "backwards" sense compared to the main volume number. For example, in Figure 7.7, the  $X_1$  and  $X_2$  areas  $AX_1$  (IP) and  $AX_2$  (IP), are on the lower (or "back") faces of main control volume IP. The velocities, which are momentum control volume quantities, are defined in a similar manner. For example,  $U_1$  (IP) and  $U_2$  (IP) denote fluid velocities in the  $X_1$ - and  $X_2$ -directions on the lower  $X_1$  and lower  $X_2$  faces of main volume IP. In contrast, the FLUFIX code uses a "forward" scheme. This difference should be kept in mind when comparing code predictions.

#### 7.4 FIELD VARIABLES

The term "field variable" refers to any geometric quantity or dependent variable which must be maintained for each node of the discrete geometry. In FORCE2, field variables are not stored as ordinary FORTRAN variables, instead they are stored as data on disk.

Each field variable is divided into records corresponding to geometric planes. A record consists of all values of a single-field variable

corresponding to a single position in the X1 direction. Primary memory available for field variables is divided into buffers, each capable of storing a single record. Buffer management and disk I/O are controlled by storage management routines (SMR). When a subroutine needs a field variable, it calls the SMR to locate the buffer containing the record (plane) needed.

A code number is assigned to each field variable and is used to identify the field variable within the program. Code numbers for geometric quantities and the fluid dynamic dependent variables are contained in the common block FIELD. Code numbers for other field variables are assigned by the SMR when the field variable is defined. These codes must be retained by the module defining the variable.

Planes (records) of a field variable are numbered in the positive X1 direction. For field variables associated with X1 faces, planes are numbered from 2 to NX1. For other field variables, planes are numbered from 1 to NX1.

The array FV is used to store field variables in primary memory. It is divided into buffers and managed by the SMR. When a subroutine needs a field variable, it calls the storage management subroutine FLDGET. Arguments identify the field variable and plane desired. FLDGET insures that the record is in memory and returns the offset (index in the array FV) of the buffer containing the record. Subroutine FLDGET is detailed in section 7.8.4.

Example: CALL FLDGET (KRHO, 2, 1, KPLN, JRHO)

This example requests a plane of densities, where KRHO is the code number for density, and KPLN is the number of the plane desired. After the call, JRHO will contain the offset of the buffer. The other arguments are described in section 7.8.4.

Assume that JRHO is the offset of the desired plane of densities. To reference the density at a particular point, IP, in the plane, it would be coded as FV (JRHO+IP).

To improve readability of the code, statement functions are defined so that reference to the field variable is clearer. The statement function for RHO, for example, would be:

RHO (IP) = FV (JRHO+IP)

Therefore, a node of the plane can be referenced by coding RHO(IP) instead of FV (JRHO+IP). (Refer to the FORTRAN language reference manual for a description of statement functions.)

This makes a reference look like an array reference; but actually, it is a statement-function reference. Statement functions are valid only on the right-hand side of an assignment statement. To store a value at a particular node, the notation FV (JRHO+IP) must be used.

By convention, offsets in FV are given the name of the variable with a J prefix. With this convention, FV(JPHI+IP) refers to the same node and variable as PHI(IP), where PHI is some variable name.

FORCE2 maintains additional information for dependent variables:

DFIELD (1,KSV)	Field variable name for input (4 characters)
DFIELD (2,KSV)	Field variable title for output (16 characters)
KPRTF (KSV)	Print control code
RELAX (KSV)	Under-relaxation factor
RELAXI (KSV)	Inertial-relaxation factor

For a field variable with code number KF, the index KSV above is KF-KBASE.

## 7.5 VARIABLE NAMING CONVENTIONS

The first letter of variable names usually indicates its function in the program. Following the naming convention makes the code easier to understand.

Ixxxxx	- General index or counter.
IP	- Index of a point in a plane.
ITER	- Number of current iteration during a run.
Jxxxxx	- Offset (or index) into the secondary storage array, FV. In general, JXYZ is the name of the offset for field variable XYZ.
JCP	- Offset for field variable CP.
JU1	- Offset for the current plane of axial velocities (U1).
J2P	- Offset of the next node in the positive X2 direction. In other words, IP + J2P is the index of the next node in the positive X2 direction from the node with index IP.
Nxxxxx	
NX1	- Number of nodes in the X1 direction.
NITER	- Total number of iterations.
Mxxxxx	- Maximum number of items which the code can handle with current dimensions or limiting value of a count.
MX1	- Maximum number of nodes in the X1 direction. No problem can be run with more than MX1 nodes.
MITER	- Maximum number of iterations for this run.
Kxxxxx	- Miscellaneous integer codes and parameters.
KU1	- Secondary storage code for variable U1.
KPHASE or KPH	- Phase number, 1 (gas) or 2 (solids)

A complete list of common variable names is provided in section 7.9.

## 7.6 MODULE INTERFACING

A module is a collection of subroutines which implements a physical model. Each module contains four subroutines which interface the module and the main code. These subroutines are called by the main code for module specific processing of: input, calculation, output and computation of source

terms for other modules. By convention, these subroutines are numbered 1 through 4. For example, the flow module contains of the following subroutines:

<u>Subroutine</u>	<u>Function</u>
FLOW1	Flow input processing
FLOW2	Flow calculations
FLOW3	Flow output
FLOW4	Source-term calculation for other variables

Additional subroutines are called by the four interface routines.

#### 7.6.1 Input Routine

The input routine is responsible for:

- o Reading data from the input file.
- o Reading and writing the restart file.
- o Performing set-up processing.

Overall flow of processing in the input routine is shown in Figure 7.10.

Input must be read using the subroutine INPUT or INPUTD (described in section 7.8).

Data which must be retained for a restart is written on the restart file. The restart file is an unformatted file. Each record begins with two words: the record number and the number of data words in the record. Records must be written to the file in sequence by record number. Each module is assigned a restart code as  $n \times 1000$  (e.g., 1000, 2000, 3000, etc.). The restart code is passed to the module when it is called. The record numbers consist of the restart code plus a three-digit number to uniquely identify the record.

Field variables are not available when the input routine is called and cannot be accessed from the input routine. New field variables necessary for

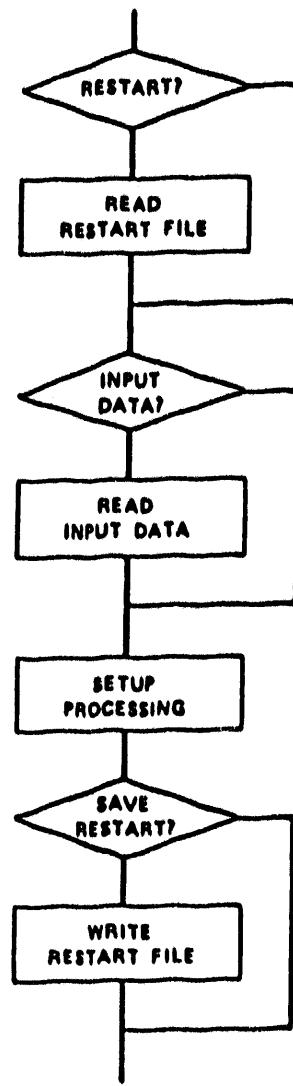


Figure 7.10. Input processing in FORCE2 - FORCE2 theory.

this module are allocated by using subroutine FLDDEF. All field variables are automatically restarted.

#### 7.6.2 Calculation Routine

The calculation routine is called either once per iteration or plane by plane depending on the module. It calculates all field variables for which it is responsible.

#### 7.6.3 Output Routine

The output routines are responsible for module output, which is written to Unit 6 for printing or to Units 69, 70 and 71 for the graphical post-processor and the ANL erosion code. The frequency at which output is written is specified by the user in the control input paragraph.

Output routines for printing are called from subroutine PRINT. PRINT is called once after all input is processed, several times during steady or transient execution depending on user specifications and finally after all steady iterations or transient timesteps are complete. A single argument is passed, indicating which call this is: 0 or 1 after input, and 9 after iterations.

The output routine is responsible for updating the global line counter, LINCNT. It must be decremented for each line printed. When LINCNT is zero or less, subroutine HEADER is called to skip to a new page and reset the line count.

#### 7.6.4 Source-Term Calculation Routine

The source-term calculation routine is responsible for calculating source terms for field variables. It is called by subroutine SOURCE.

The module source term routine (e.g., FLOW4) is passed a single argument, KF, the field variable code number. The routine calculates source terms from this module for the current plane (KPLN) of variable KF.

For standard field variables, the codes are stored in the FIELD common block. For example, if KF is equal to KU2, then this is the momentum equation for the U2 velocity. Nonstandard field variables (defined through FLDDEF) can be tested by comparing DFIELD (1, KF) with the field variable name.

The standard form of the finite-difference equation for some variable  $\phi$  is:

$$a_p \phi_p = a_n \phi_n + a_s \phi_s + a_w \phi_w + a_e \phi_e + a_z \phi_z + a_b \phi_b + S$$

Where the  $a_i$  are the finite-difference coefficients, and  $S$  is the source term. The provision is made for the source term to be expressed as a linear function of the variable.

$$S = S_c + S_p \phi_p$$

Two temporary field variables, SC and SP, are used for storing the source term, where the source term for a particular cell is:

$$SC(IP) + SP(IP)*PHI(IP)$$

The components of the source term must be calculated for each cell and added to SC and SP, respectively.

Source-Term Linearization for Steady Simulation. The recommended linearization for a source term is based on a Taylor series for the source term:

$$S(\phi) = S(\phi^n) + \frac{\delta S}{\delta \phi} (\phi - \phi^n) + \dots$$

$$\approx S(\phi^n) - \frac{\delta S}{\delta \phi} \cdot \phi^n + \frac{\delta S}{\delta \phi} \cdot \phi$$

Therefore,

$$S_c = S(\phi^n) - \frac{\delta S}{\delta \phi} \phi^n$$

$$Sp = \frac{\delta S}{\delta \phi} \phi$$

n = iter-1

However, because the solution techniques currently used in the code do not use pivoting, the diagonal terms must dominate for numerical accuracy. This implies that Sp should be negative since it contributes to the diagonal term. If Sp is positive, numerical problems may result.

## 7.7 FORCE2 SOURCE CODE FILES

Each subroutine is stored as a separate file. Common blocks are stored in separate files and included in the subroutines via the FORTRAN INCLUDE statement. This guarantees that the common blocks are identical in each subroutine and facilitates modification of common block definitions. Default values for variables in common are defined by data statements in another file. For example, the basic FORCE2 common blocks are stored in FORCECOM, and default specifications for the variables are stored in FORCEDEF.

Module specific common blocks are stored in a separate file and included in the module subroutines (more than one common file may be used). Default values for data in common are specified as DATA statements in an additional file and are included in the input module.

Single precision (adequate for steady simulations) or double precision (recommended for transient simulations) versions of FORCE2 can be created by modifying the DOUBLE and FORCECOM files. The default is a double precision version. By modifying the declarations in DOUBLE and replacing FORCECOM (the default) with FORCECOM.sgl, a single precision version of the code can be created.

## 7.8 UTILITY ROUTINES

### 7.8.1 Transport Coefficient Calculation

Subroutine COEF calculates the finite difference coefficients for the gas and solids phase transport equations for the steady simulation.

Subroutine COEFT performs a similar task for the transient simulation. The transport coefficients are calculated for a single plane and stored in temporary field variables. The offsets of these coefficient planes are stored in common block FCOEF.

Both COEF and COEFT require a single argument, KF, the code number of the variable. In addition, variable KPLN (in common block FCNTRL) must be set to the number of the geometric plane being considered.

#### 7.8.2 Tilde Term Calculation

For the transient simulation, subroutine TILDE calculates the "tilde" terms for each phase in each coordinate direction. A sweep over all X1-planes is performed by TILDE. At each plane, subroutines TILDX1, TILDX2 and TILDX3 are called to calculate the X1-, X2- and X3-direction tilde terms, respectively. These routines are passed an argument, KF, that identifies the phase velocity in the direction of interest.

#### 7.8.3 Error Subroutine

An error subroutine is available for error handling; all error messages should be printed by the error subroutine.

Subroutine ERROR has 3 arguments:

KLEV

Error level

1 -- Warning only

2 -- Noncritical error, continue

3 -- Critical error; may not produce a usable run.

But corrective action has been taken, and  
procession can continue

4 Fatal error, run cannot be made

**NCHAR** Number of characters in error message

## XMSG Error message

By convention, error messages have the following format:

↳ program ↳↳ message

The first space is for carriage control, "program" is the subroutine name and "message" is the error message text. For example:

CALL ERROR (2, 32, 32H PROPS1 - Numeric value expected)

#### 7.8.4 Field-Variable Definition

Subroutine **FLDDEF** is used to define new field variables.

**Subroutine FLDDEF has three arguments:**

**VNAME** A unique four-character name for the variable. This name is used for input related to the variable.

**VTITLE(4)** A print title, four words of four characters each. This title is used to identify the variable in the printed output.

**KF** Output, code number for the variable.

### 7.8.5 Field-Variable Access

Secondary storage is organized into planes of variables, i.e., each record in secondary storage is a single plane of a single variable. Access to secondary storage is by the routine FLDGET. FLDGET reads and writes secondary storage records into a main storage buffer (the array FV). In response to requests FLDGET returns offsets into the buffer (indexes in FV). The arguments to FLDGET are:

KF(NV)	Code numbers of the variables to be accessed.
KA(NV)	Access codes: 1=temporary, 2=read only, 3=update and 4=write only.
NV	Number of variables requested.
KPLN	Number of the plane requested for all variables.
JV(NV)	Offsets of planes in main storage buffer (e.g., node IP is specified by FV(JV(N) + IP)).

Figure 7.11 shows the processing flow for storage management.

Access read or update. When the access code specifies read or update, FLDGET is requested to obtain plane KPLN for variable KF and return the offset as JV. If the values are not changed, read-only access (2) is specified, indicating that the data need not be rewritten on secondary storage after it is used. If the values are changed, the update access is specified to insure that the data is rewritten.

Access temporary. Occasionally a routine needs some array space to temporarily store data when it is running. Since the need is temporary, it is useful to have storage allocated only when it is needed. Therefore FLDGET provides the ability to allocate a plane of storage for temporary use. In this case, KF is ignored by FLDGET since no variable on secondary storage is associated with the request. When the program is done with temporary storage, it is freed by calling RFREE.

Access write only. At the beginning of the program, there are no values for the field variables on secondary storage. They must first be initialized. Write-only access is used to specify that the variable is being initialized. Space is allocated for the variable in the buffer but nothing is read into it. The variable is then written to secondary storage after it is used.

Internal allocation of buffer space. When buffer space is needed to satisfy a request, the storage management routine uses free space in the

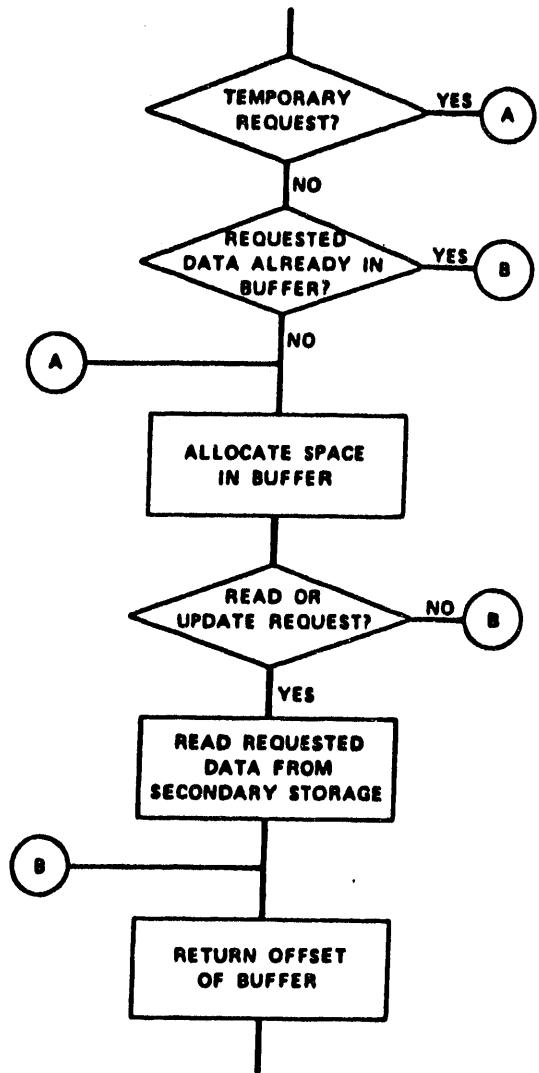


Figure 7.11. Storage management in FORCE2.- FORCE2 theory.

buffer to satisfy the request. If there is no free space, the least recently requested record is used. If it was allocated with a WRITE or UPDATE request, the data is written to secondary storage before it is used.

With this scheme of operation, a record cannot be expected to remain in memory for a very long time. As a general rule, data needed by a routine is requested each time that the routine is entered.

There is another potential problem with this scheme. If a particular calculation requires more data than can be held in the buffer, it will not run. The data management routines cannot guarantee that sufficient space is available. However, when the buffer is initialized, a check is made to insure that a reasonable number of records can be stored in memory.

#### 7.8.6 Page Heading

Subroutine HEADER causes a page eject on the printer and prints a standard heading. It also resets the line counter, LINCNT.

#### 7.8.7 Input Processing

The basic input routine is INPUT. It reads an input statement, breaks it into elements (fields) and stores the data in an array.

Subroutine INPUT. The subroutine INPUT stores its data in the arrays XVALUE and XTYPE. In addition it sets two other values, XFLAG and NVALUE.

<u>Variable</u>	<u>Contents</u>
XFLAG	An integer code, a value of 99 indicates an end of file for the current input section.
NVALUE	Number of elements (fields) specified on the input statement.

**XVALUE(I)** The value of the *i*th element of the input statement. A numeric field is stored as a real number. A character field is stored as an A4 containing the first four characters.

**XTYPE(I)** An integer code indicating the type of data stored in XVALUE(I).

0 no data (XVALUE(I) = 0.0)  
1 numeric data  
-1 character data

**CODE** Equivalent to XVALUE(1)

**CODEX** Equivalent to XVALUE(2)

Subroutine INPUT has no arguments. After calling INPUT, XFLAG is checked for 99 and treated as an end of file; otherwise, the input statement is processed.

## 7.9 VARIABLE NAME DICTIONARY

The commonly used variables are listed in Table 7.3. A variable name containing a lower case "n" stands for three variables, one for each coordinate direction. The actual names have the "n" replaced by a digit (1, 2, or 3) for the coordinate direction. Unless local is specified under the type column, the variable is in common. For arrays, the dimensions are specified under type.

Since statement functions are used to reference field variables, the statement function names are included in this list. The notation SF appears in the type column for statement functions.

## 7.10 SUBROUTINE LIST

The subroutines which make up FORCE2 are listed in Table 7.4, which includes a brief description of each subroutine. All subroutines are

classified into four groups, as indicated under the function column. The four functions groups are: input, output, calculation and utility. FORCE2 contains some routines not listed in Table 7.4. Such routines are inactive yet have been retained to maintain program logic. FORCE2 was derived from a more general model that included heat transfer and combustion.

Table 7.1  
FORCE2 GEOMETRIC VARIABLES

<u>Variable</u>	<u>Contents</u>
NX1	Number of nodes in the X1-direction
NX2	Number of nodes in the X2-direction
NX3	Number of nodes in the X3-direction
X1(NX1)	X1-face coordinates
X2(NX2)	X2-face coordinates
X3(NX3)	X3-face coordinates

Table 7.2  
GEOMETRIC VARIABLES FOR EACH CELL IN THE FORCE2 GRID SYSTEM

<u>Quantity</u>	<u>Name</u>
Area of cell face normal to direction n	AXn
Distance (arc length) between main nodes along direction n	DXn
Linear interpolation factor in direction n (based on distance)	FFn
Momentum interpolation factor for velocity in direction n (based on areas)	FXn
Area	APn
Area x density	ARn
Area x effective viscosity	AVn
Volume of cell	VOL

Table 7.3  
VARIABLE NAME DICTIONARY FOR FORCE2

<u>Name</u>	<u>Type</u>	<u>Description</u>
APn	SF	Area x Permeability at lower face in direction n.
APnP	SF	Area x Permeability at lower face in direction n on the next plane.
ARn	SF	Area x Permeability x Density at lower face in direction n.
ARnP	SF	Area x Permeability x Density at lower face in direction n on the next plane.
AVn	SF	Area x Permeability x Viscosity at lower face in direction n.
AVnP	SF	Area x Permeability x Viscosity at lower face in direction n on the next plane.
CODE		First field of card input (equivalence with XVALUE(1)).
CODEX		Second field of card input (equivalence with XVALUE(2)).
CON	SF	Constant term in finite-difference equations.
CP	SF	Coefficient of the main node in the finite-difference equations (the diagonal coefficient).
CnM	SF	Coefficient of the lower node in direction n for the finite-difference equations.
CnP	SF	Coefficient for the higher node in direction n for the finite-difference equations.
CVOL	SF	Volume x Porosity of control volume for current variable.
DFIELD	(5,MFV)	Dictionary for solution field variables.
DPROP	(5,MPROP)	Dictionary for properties.
DXn	SF	Distance between main node and lower node in direction n.
DXnP	SF	Distance between main node and lower node in direction 1 on the next plane.

Table 7.3  
VARIABLE NAME DICTIONARY FOR FORCE2

EPSG		Gas residual solution tolerance for transient solution.
FFn	SF	Interpolation factor for computing main node values at faces. The ratio of the distance from the high node to the face and the distance from node to node.
FFnP	SF	Same as FFn but in the next plane.
FV	(MWORD)	Field variable storage in primary memory, only the values currently in use are stored in FV.
FXn	SF	Interpolation factor for momentum control volume.
FXnP	SF	Same as FXn but in the next plane.
IP	Local	Index of a point in a plane.
IPO	Local	First value for IP in a DO loop.
IP1	Local	Last value for IP in a DO loop.
IPREF		Node within plane KREF associated with the reference pressure.
ITER		Current iteration counter. During a run, ITER goes from 1 to MITER.
ITRYX		Micro iteration counter for transient solution.
ITRYMX		Maximum micro iterations at a node for transient solution.
Jxxx		Offset in FV for xxx.
JCON		Offset in FV for CON.
JCP		Offset in FV for CP.
JCnM		Offset in FV for CnM.
JCnP		Offset in FV for CnP.
JCVOL		Offset in FV for CVOL
JRES		Offset in FV for RES.
JSC		Offset in FV for SC.
JSP		Offset in FV for SP.

**Table 7.3**  
**VARIABLE NAME DICTIONARY FOR FORCE2**

JV	Local	Offsets returned by FLDGET for planes retrieved.
JnM		Offset in a plane of the lower node in direction n (e.g., for node IP the lower node in direction 2 is IP+J2M).
JnP		Offset in a plane of the higher node in direction n (see JnM).
KAPn		Code number for APn.
KARn		Code number for ARn.
KAVn		Code number for AVn.
KAXn		Code number for AXn.
KAn	Local	Array of access codes passed to FLDGET for allocating planes.
KBASE		Base for solution variable code numbers (e.g., RHO is solution variable KRHO-KBASE).
KDXn		Code number for DXn.
KFn	Local	Array of code numbers passed to FLDGET for allocating planes.
KFFn		Code number for FFn.
KFV	(MFV)	Array containing I/O record numbers for each field variable (e.g., the I/O number for RHO is KFV(KRHO) ).
KFXn		Code number for FXn.
KP		Code number for pressure, P.
KPHASE		Phase identifier, 1 (gas), 2 (solids).
KPH	Local	Same as KPHASE.
KPLN		Number of current plane.
KPREF		Number of the plane containing the reference pressure node.
KPREST		Restart parameter.
KPRTD		Debug print parameter.
KPRTF	(MSV)	Solution variable print parameters.

Table 7.3  
VARIABLE NAME DICTIONARY FOR FORCE2

KPRTG		Geometry print parameter.
KPRTP		Properties print parameter.
KPRTW		Print width, number of values per line.
KPRTX		Filler parameter, unused.
KPSAVE		Restart save parameter.
KPSTAT		Print control for subroutine calls and file I/O statistics.
KRHO	(3)	Code number for density, RHO.
KUn	(2)	Code number for velocity, Un.
KVISC	(2)	Code number for viscosity, VISC.
KVOL		Code number for volume, VOL.
KX		Code number for cartesian X coordinate.
KY		Code number for cartesian Y coordinate.
KZ		Code number for cartesian Z coordinate.
LINCNT		Print line counter.
LINMAX		Maximum number of lines per page.
MFV		Maximum number of field variables.
MITER		Number of iterations to be performed for this run.
MPROP		Maximum number of properties.
MVALUE		Maximum number of fields per input card.
MWORD		Number of words of buffer storage in array FV.
MXn		Maximum number of nodes in direction n.
NFV		Number of field variable stored on secondary storage.
NGEOM		Code number for type of geometry.
NIT		Macro iteration counter for transient solution.
NITER		Total number of iterations.

Table 7.3  
VARIABLE NAME DICTIONARY FOR FORCE2

NLINE		Input line number.
NITMAX		Maximum macro iterations at a timestep for the transient solution.
NP		Number of nodes per plane.
NPRI0	(3)	Rank of flow directions as to predominance of flow in that direction.
NPROP		Number of properties.
NSTEPS		Total number of timesteps for this run.
NSTEPT		Total number of timesteps for this problem.
NSV		Number of solution variables.
NVALUE		Number of fields on previous input card.
NXn		Number of nodes in direction n.
P	SF	Local pressure.
PDELT		The local pressure at the reference pressure node.
PM	SF	Local pressure on next lower plane.
PP	SF	Local pressure on next higher plane.
PREF		Reference pressure level (the pressure at any node is the sum of the reference pressure and the local pressure).
PRHO		Nominal density.
PTIME		Problem time.
PVISC		Nominal viscosity.
RELAX	(MSV)	Under-relaxation factors for solution variables.
RELAXI	(MSV)	Inertial-relaxation factor for solution variables.
RES	SF	Cell residuals, updated by solve each time it is called.
RHO	SF	Density.
RNDATE	(3)	Current date, used in print heading.
RSAVG	(MSV)	Average absolute residual for each solution variable.

Table 7.3  
VARIABLE NAME DICTIONARY FOR FORCE2

RSMAX	(MSV)	Maximum absolute residual for each solution variable.
SC	SF	External source-term constant.
SP	SF	External source-term coefficient of the current variable.
TILnG	SF	Tilde term for gas momentum equation in direction n.
TILnS	SF	Tilde term for solids momentum equation in direction n.
TSTEP		Problem timestep.
TITLE	(20)	Problem title from input.
Un	SF	Velocity in direction n.
UnG	SF	gas velocity in direction n.
UnM	SF	Velocity in direction n on the next lower plane.
UnP	SF	Velocity in direction n on the next higher plane.
UnS	SF	Solids velocity in direction n.
VISC	SF	Viscosity.
VOL	SF	Volume of main control volume.
WORK	(500)	Work array used by utility routines.
XFLAG		Code for end of paragraph and end of file form routine input.
XTYPE	(MVALUE)	Type of input filed in corresponding element in XVALUE: 1-number, -1-alpha, 0=unspecified.
XVALUE	(MVALUE)	Value of input field, either numeric or alphabetical, depending on content.
Xn	(MXn)	Face coordinates in direction n.

Table 7.4  
SUBROUTINES IN FORCE2

<u>Routine</u>	<u>Function</u>	<u>Description</u>
ANLRES	Calculation	Implements ANL distributed resistance model for a tube bundle.
BANNER	Output	Prints output banner.
BLKSET	Utility	Initializes a specified range of a field variable.
BOUN2T	Calculation	Adjusts tilde terms to account for wall boundary conditions.
BOUND	Calculation	Calculates exit-plane boundary conditions.
BOUND1	Calculation	Performs setup calculations for blockage logic.
BOUND2	Calculation	Adjusts finite-difference coefficients to account wall boundary conditions.
BOUNT	Calculation	Sets boundary conditions for transient solution.
CNDIFT	Calculation	Calculates convection and diffusion in a single direction for the tilde term.
CNTRL0	Input	Restart input paragraph interpretation routine.
CNTRL1	Input	Control input paragraph interpretation routine.
CNTRL2	Control	Controls overall calculation sequence.
CNTRL3	Output	Control output routine.
COEF	Calculation	Finite difference equation coefficient calculation.
CONDIF	Calculation	Calculates convection and diffusion in a single coordinate direction.
COEFT	Calculation	Finite difference equation coefficient calculation for the tilde term. COEFT calls CNDIFT.
COEFV	Calculation	Relaxes the steady finite difference equation for void fraction as described in section 4.4.2.
CONFAC	Calculation	Calculates upwind factor based on input velocity
CORRCT	Calculation	Corrects velocity and pressure after solution of pressure equation in the steady solution method. CORRCT performs step No. 7, section 6.2.
CTDMA2	Calculation	Coupled tridiagonal matrix solver

Table 7.4  
SUBROUTINES IN FORCE2

<u>Routine</u>	<u>Function</u>	<u>Description</u>
CTIMER	Utility	Accumulates number of calls and cpu time for each subroutine which calls it.
DENSO	Calculation	User routine to calculate phase microscopic density.
DRAG	Calculation	Calculates drag coefficient for gas-solids flow based on Main Control volume parameters.
DRAGSP	Calculation	Calculates drag coefficients for momentum equations. DRAGSP is used for both transient and steady solutions and calls DRAG extensively.
EROSWR	Output	Writes hydrodynamic predictions to Unit 71 for subsequent input to the ANL erosion model. EROSWR is called only during a transient solution.
ERROR	Utility	Error message output routine.
FIND	Utility	General linear interpolation routine.
FINIT	Utility	Random access file I/O initialization routine (system dependent).
FLDDEF	Utility	Field variable creation routine.
FLDGET	Utility	Field variable allocation routine.
FLDSET	Utility	Field variable initialization routine.
FLDEV1	Input	Distributed resistance input routine.
FLDEV2	Calculation	Performs setup calculations for the distributed resistance modelling.
FLDEV3	Output	Output routine for distributed resistance modelling.
FLDEV4	Calculation	Calculates distributed resistance parameter for finite difference equations.
FLOFLG	Input	Input routine for cell flow flags.
FLOW1	Input	Input routine for flow solution.
FLOW2	Control	Manages overall flow solution for the steady simulation.
FLOW2T	Control	Manages overall flow solution for the transient simulation.

Table 7.4  
SUBROUTINES IN FORCE2

<u>Routine</u>	<u>Function</u>	<u>Description</u>
FLOW3	Output	Output routine for flow solution parameters.
FLOW4	Calculation	Calculates flow source terms for steady solution.
FLOW5	Calculation	Adjusts finite difference equations for flow blockages in steady solution
FLOW5T	Calculation	Adjusts tilde terms for flow blockages.
FLOWEX	Calculation	Adjusts exit velocity profile for steady solution. FLOWEX performs step No. 9 of the steady solution sequence, section 6.2.
FORCE2	Control	Main program, controls all processing.
FREAD	Utility	Random access file I/O input routine (system dependent).
FWRITE	Utility	Random access file I/O output routine (system dependent).
GAMAS	Input	Input routine for permeabilities and porosities.
GEOM1	Input	Geometry paragraph interpretation routine.
GEOM2	Input	Calculates geometric quantities (AXn, DXn, FXn, FFn, VOL, etc.).
GEOM3	Output	Geometric output.
GEOM5	Calculation	Adjusts finite difference equations for geometric boundaries.
GEOM5T	Calculation	Adjusts tilde terms for geometric boundaries.
GETFUN	Control	Initializes commons with field variable numbers that are frequently used in various calculational routines.
HEADER	Utility	Produces output page break and resets line counter.
INITL1	Calculation	Initializes pressure distribution based on minimum fluidization condition.
INITL	Input	Initialization paragraph interpretation routine.
INPUT	Utility	Reads and input card and converts fields to array of values for interpretation by calling routine.
INITLS	Calculation	Calculates solids mass for initial run of problem.

Table 7.4  
SUBROUTINES IN FORCE2

<u>Routine</u>	<u>Function</u>	<u>Description</u>
INTFL1	Control	Manages temporal integration of certain field variables.
INTFL2	Calculation	Performs temporal integration and stores result.
MODSMP	Control	Manages flow and void solution in for steady simulation. This routine performs steps 3-8 of the steady solution sequence, section 6.2.
POSTA	Output	Writes all field variables to Unit 69 for subsequent use with the FORCE2 graphical post processor.
POSTT	Output	Writes timewise data to Unit 70 for subsequent use with the FORCE2 graphical post processor.
PRCOEF	Calculation	Calculates finite difference coefficients for the steady pressure correction equation, performed in Step 6 of the steady solution sequence, section 6.2.
PRESS1	Input	Input routine for steady pressure solution.
PRIMET	Calculation	Pressure correction routine for transient solution.
PRINT	Control	Printed output driver
PRINTD	Utility	Prints coefficients, residuals and variables for debug purposes.
PRINTF	Utility	Prints a single field variable.
PRINTR	Utility	Print maximum and average residuals for each field variable.
PROPS1	Input	Properties input paragraph interpretation routine.
PROPS2	Calculation	Controls phase properties calculations.
PROPS3	Output	Properties output other than field variables.
PROP2T	Calculation	Calculates macroscopic phase densities and viscosities for the transient solution.
PROPCT	Calculation	Calculates microscopic phase densities and viscosities.
PROPS2U	Calculation	Calculates Area X Density and Area X Viscosity parameters on faces of Main Control Volume based on upwind values for properties.

Table 7.4  
SUBROUTINES IN FORCE2

<u>Routine</u>	<u>Function</u>	<u>Description</u>
PRSOLV	Calculation	Solves finite difference equations over flow domain using tridiagonal matrix approach. The pressure correction and gas void equations are solved in PRSOLV.
RESIDG	Calculation	Calculates gas mass residual at a cell for the transient solution.
RESTART	Input	Field variable restart input and output.
RFREE	Utility	Secondary storage - free a record.
RGET	Utility	Secondary storage - get a record.
RINIT	Utility	Secondary storage initialization.
RPRINT	Output	Radiation module special print routine.
RSTAT	Utility	Secondary storage statistics output.
SETUP	Input	Calculates geometric control parameters and initializes secondary storage.
SOURCE	Calculation	External source term calculation via module source term routines.
SMPINI	Calculation	Sets up arrays for direct solution method used for steady simulation.
SOLSTR	Calculation	Calculates the solid stress source terms for the steady solution.
SOURCT	Calculation	Calculates source terms and distributed loss coefficients for the transient solution.
TDMA	Calculation	Tridiagonal matrix solver.
TILDE	Control	Manages calculation of tilde terms.
TILDX1	Calculation	Calculates tilde terms in the X1-Direction.
TILDX2	Calculation	Calculates tilde terms in the X2-Direction.
TILDX3	Calculation	Calculates tilde terms in the X3-Direction.
TRNINI	Calculation	Sets up arrays to store operating conditions during a transient simulation. Arrays are used to restore conditions at an earlier time if convergence problems arise with transient method.

Table 7.4  
SUBROUTINES IN FORCE2

<u>Routine</u>	<u>Function</u>	<u>Description</u>
TRNSTR	Calculation	Periodically stores operating conditions during a transient simulation.
UEST	Calculation	Performs point solution of steady momentum equations. Check for solids flow out of a Main Control volume that may not contain any solids.
UHATSM	Calculation	Calculates trial velocities for steady momentum equations, parts of Steps 3, 4, and 5 of the steady solution sequence, section 6.2.
USRRES	Calculation	Routine to be developed by the User to calculate distributed loss coefficients.
USTAR	Calculation	Momentum equation setup for steady solution.
USTART	Control	Manages the solution of the momentum and gas void equations at a single node for the transient solution method.
VELCOEF	Calculation	Calculates momentum equation residuals, relaxes the equations, and calculates the final finite difference equation coefficients prior to their solution for a steady simulation.
VELS	Control	Manages the solution for velocities on the faces of a Main Control volume during the transient solution.
VELSX1	Calculation	Calculates phase velocities on the X1-faces of a Main Control volume during the transient solution.
VELSX2	Calculation	Calculates phase velocities on the X2-faces of a Main Control volume during the transient solution.
VELSX3	Calculation	Calculates phase velocities on the X3-faces of a Main Control volume during the transient solution.
VFCOEF	Calculation	Calculates the void equation residual, relaxes the equation, and calculates the final finite difference equation coefficients prior to their solution for a steady simulation
VFMOD	Calculation	Modifies steady void based on global solids continuity, Step 8 of the steady solution sequence, section 6.2.
VFRACS	Calculation	Calculates finite difference coefficients for the steady void equations.

Table 7.4  
SUBROUTINES IN FORCE2

<u>Routine</u>	<u>Function</u>	<u>Description</u>
VFRAC	Calculation	Calculates the gas void fraction in a Main Control volume during the transient solution.
VISCO	Calculation	Routine to be developed by the User to calculate phase microscopic viscosity.

## 8.0 REFERENCES

Carver, M.B., "Numerical Computation of Phase Separation in Two Phase Flow," ASME Paper 82-FE-2, 1982.

Davidson, J.F., Symposium on Fluidization - Discussion, Trans. Institution of Chemical Engineers, 39, 230-232, 1961.

Ergun, S., "Fluid Flow through Packed Columns," Chemical Engineering Progress, 48, 89-94, 1952.

Ettehadieh, B., et al., "Hydrodynamics of Fluidization in a Semicircular Bed with a Jet," J. American Institute of Chemical Engineers, 30, 529-536, 1984.

Fortino, R.T., Private Communication, 1989.

FLUFIX Workshop Notes, Argonne National Laboratory, October 1989.

Gidaspow, D., "Hydrodynamics of Fluidization and Heat Transfer: Supercomputer Modeling," Twenty-Third National Heat Transfer Conf., Applied Mechanics Review, 39, 1-23, 1986.

Gidaspow, D. and Ettehadieh B., "Fluidization in Two-Dimensional Beds with a Jet: 2. Hydrodynamic Modeling," Industrial and Engineering Chemistry Fundamentals, 22, 193-201, 1983.

Gidaspow, D. and Syamlal, "Solid-Gas Critical Flow," presented at 1985 Annual American Institute of Chemical Engineers Meeting, Chicago, Paper 74e, Nov. 10-15, 1985.

Harlow, F.H. and Amsden, A.A., "Numerical Calculation of Multiphase Fluid Flow," J. Comp. Phys., 17, 1975.

Lyczkowski, R.W., et. al., "Characteristics and Stability Analyses of Transient One-Dimensional Two-Phase Flow Equations and Their Finite Difference Approximations," Nuclear Science and Engineering, 66:378-396, 1978.

Lyczkowski, R.W., et. al, "Computer Modeling of Erosion in Fluidized Beds," ANL/ESD/TM-1, Argonne National Laboratory, January, 1990.

Lyczkowski, R.W. and Bouillard, J.X., "Interim User's Manual for FLUFIX/MOD1: A Computer Program for Fluid-Solids Hydrodynamics," ANL/EES-TM-361, Argonne National Laboratory, February 1989.

Patankar, S.V., Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing, McGraw-Hill Company, New York, 1980.

Rivard, W.C. and Torrey, M.D., "K-FIX: A Computer Program for Transient, Two-Dimensional, Two-Fluid Flow," LA-NUREG-6623, Los Alamos Scientific Laboratory, April 1977.

Smoot, L., "Modeling of Coal-Combustion Processes, Progress in Energy and Combustion Science," 10, 229, 272, 1984.

Schnipke, R.J., "A Streamline Upwind Finite Element Method for Laminar and Turbulent Flow," Ph.D. Dissertation, University of Virginia, 1986.

Syamlal, M., "Multiphase Hydrodynamics of Gas-Solids Flow," Ph.D. Thesis, Illinois Institute of Technology, 1985.

Wen, C.Y. and Yu, C.H., Mechanics of Fluidization, B.S. Lee, ed., American Institute of Chemical Engineers Series, 62, 100-112, New York, 1966.

Appendix A  
COUPLED TRI-DIAGONAL MATRIX SOLVER

The gas and solids momentum equations are solved simultaneously along lines using a coupled tri-diagonal matrix algorithm (CTDMA, Fortino 1989). The formulation of this method is given in this appendix.

The coupled momentum equations may be expressed, along a  $j$ -line, as:

$$P1_j \phi_j = A1_j \cdot \phi_{j+1} + B1_j \cdot \phi_{j-1} + C1_j \cdot \psi_j + S1_j \quad (A-1)$$

$$P2_j \cdot \psi_j = A2_j \cdot \psi_{j+1} + B2_j \cdot \psi_{j-1} + C2_j \phi_j + S2_j \quad (A-2)$$

where  $j$  indicates a node along the line and the variables  $\phi$  and  $\psi$  represent velocities in the current application.

We assume a solution of the form:

$$\phi_j = G1_j + H1_j \phi_{j+1} + F1_j \psi_j \quad (A-3)$$

$$\psi_j = G2_j + H2_j \psi_{j+1} + F2_j \phi_j \quad (A-4)$$

and will develop recursive relations for the  $F$ 's,  $G$ 's and  $H$ 's.

Equations A-3 and A-4 are combined to give:

$$\phi_j = G1_j^* + H1_j^* \cdot \phi_{j+1} + F1_j^* \cdot \psi_{j+1} \quad (A-5)$$

$$\psi_j = G2_j^* + H2_j^* \cdot \psi_{j+1} + F2_j^* + \phi_{j+1} \quad (A-6)$$

where

$$G1_j^* = (G1_j + F1_j \cdot G2_j) / D_j$$

$$H1_j^* = H1_j / D_j$$

$$F1_j^* = F1_j \cdot H2_j/D_j$$

$$G2_j^* = (G2_j + F2_j \cdot G1_j)/D_j$$

$$H2_j^* = H2_j/D_j$$

$$F2_j^* = F2_j \cdot A1_j/D_j$$

$$D_j = 1.0 - F1_j \cdot F2_j$$

Also from Equations A-5 and A-6, we get

$$\phi_{j-1} = G1_{j-1}^* + H1_{j-1}^* \cdot \phi_j + F1_{j-1}^* \cdot \psi_j \quad (A-7)$$

$$\psi_{j-1} = G2_{j-1}^* + H2_{j-1}^* \cdot \psi_j + F2_{j-1}^* \cdot \phi_j \quad (A-8)$$

Equations A-7 and A-8 are used to eliminate  $\phi_{j-1}$  and  $\psi_{j-1}$  from Equation A-1. By comparing this equation to Equation A-3, the following expressions for F1, G1 and H1 may be derived:

$$F1_j = (C1_j + B1_j \cdot F1_{j-1} \cdot H2_{j-1}^*)/D1_j$$

$$G1_j = (S1_j + B1_j \cdot G1_{j-1} + B1_j \cdot F1_{j-1} \cdot G2_{j-1}^*)/D1_j$$

$$H1_j = P1_j/D1_j$$

$$D1_j = P1_j - B1_j \cdot (H1_{j-1} + F1_{j-1} \cdot F2_{j-1}^*)$$

The same is done using Equation A-2 to define F2, G2 and H2 as follows:

$$F2_j = (C2_j + B2_j \cdot F2_{j-1} \cdot H1_{j-1}^*)/D2_j$$

$$G2_j = (S2_j + B2_j \cdot G2_{j-1} + B2_j \cdot F2_{j-1} \cdot G1_{j-1}^*)/D2_j$$

$$H2_j = P2_j/D2_j$$

$$D2_j = P2_j - B2_j(H2_{j-1} + F2_{j-1} \cdot F1_{j-1}^*)$$

The F's, G's and H's at node j are now defined in terms of known coefficients in Equations A-1 and A-2 and in terms of the F's, G's and H's at node j-1. The solution is then developed as follows:

Step 1: Define F's, G's and H's at j=1

At the first node, Equations A-1 and A-2 are used to define the F's, G's and H's as follows:

$$G1_1 = S1_1/P1_1$$

$$H1_1 = A1_1/P1_1$$

$$F1_1 = C1_1/P1_1$$

$$G2_1 = S2_1/P2_1$$

$$H2_1 = A2_1/P2_1$$

$$F2_1 = C2_1/P2_1$$

Step 2: Define F's, G's and H's from j=2 to j=N

Using the recursive relation for the F's, G's and H's at node j in terms of their values at node j-1, the F's, G's and H's are defined along the line to node N, the last node.

Step 3: Solve for  $\phi_N$  and  $\psi_N$

At the last node, Equations A-3 and A-4 become:

$$\phi_N = G1_N + F1_N \cdot \psi_N$$

$$\psi_N = G2_N + F2_N \cdot \phi_N$$

These are solved simultaneously to give

$$\phi_N = (G1_N + F1_N \cdot G2_N)/D_N$$

$$\psi_N = (G2_N + F2_N \cdot G1_N)/D_N$$

Step 4: Solve for  $\phi_j$  and  $\psi_j$  from  $j=N-1$  to  $j=1$

Equations A-3 and A-4 are now used to solve for  $\phi_j$  and  $\psi_j$ .

## **Notice**

**The FORCE2 computer program may be obtained from:**

**Energy Science and Technology Software Center  
P. O. Box 1020  
Oak Ridge, TN 37833**

**Telephone (615) 576-2606  
FAX (615) 576-2865  
E-Mail:ESTSC@ADONIS.OSTI.GOV**

**DATE  
FILMED**

**6/2/94**

**END**

