Computer Code for Gas-Liquid Two-Phase Vortex Motions: GLVM

(U.S.) National Bureau of Standards (NEL)
Boulder, CO

Prepared for
National Aeronautics and Space Administration
Cocoa Beach, FL

Jul 86
A Computer Code for Gas-Liquid Two-Phase Vortex Motions: GLVM

T. T. Yeh

July 1986

Prepared for
National Aeronautics and Space Administration
John F. Kennedy Space Center
Kennedy Space Center, FL 32899

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Chemical Engineering
Gaithersburg, MD 20899

July 1986

Prepared for
National Aeronautics and Space Administration
John F. Kennedy Space Center
Kennedy Space Center, FL 32899

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Chemical Engineering
Gaithersburg, MD 20899

July 1986

Prepared for
National Aeronautics and Space Administration
John F. Kennedy Space Center
Kennedy Space Center, FL 32899

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Chemical Engineering
Gaithersburg, MD 20899

July 1986

Prepared for
National Aeronautics and Space Administration
John F. Kennedy Space Center
Kennedy Space Center, FL 32899

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Chemical Engineering
Gaithersburg, MD 20899

July 1986

Prepared for
National Aeronautics and Space Administration
John F. Kennedy Space Center
Kennedy Space Center, FL 32899

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Chemical Engineering
Gaithersburg, MD 20899

July 1986

Prepared for
National Aeronautics and Space Administration
John F. Kennedy Space Center
Kennedy Space Center, FL 32899
A computer program aimed at the phase separation between gas and liquid at zero gravity, induced by vortex motion, is developed. It utilizes an explicit solution method for a set of equations describing rotating gas-liquid flows. The vortex motion is established by a tangential fluid injection. A Lax-Wendroff two-step (McCormack's) numerical scheme is used. This program can be used to study the fluid dynamical behavior of the rotational two-phase fluids in a cylindrical tank. It provides a quick/easy sensitivity test on various parameters and thus provides the guidance for the design and use of actual physical systems for handling two-phase fluids.

KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

computer code; gas-liquid separation; numerical modeling; two-phase vortex motions
A COMPUTER CODE FOR GAS-LIQUID
TWO-PHASE VORTEX MOTIONS:
GLVM

T T Yeh

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Chemical Engineering
Gaithersburg, MD 20899

July 1986

Prepared for:
National Aeronautics and Space Administration
John F. Kennedy Space Center
Kennedy Space Center, FL 32899

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, Secretary
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director
Table of Contents

Abstract ........................................ ii
List of Notations ................................ iii
List of Figures .................................... v
  I. Introduction .................................. 1
  II. Model Equations ............................. 2
III. Numerical Method ............................ 5
IV. Program Details .............................. 10
  V. Summary ..................................... 14
  VI. Acknowledgments ............................ 15
VII. References ................................. 15
Appendix A. Code Listing ........................ 16
Exhibit A. A Sample Input .................... 32
Exhibit B. A Sample Output .................... 33
Abstract

A computer program aimed at the phase separation between gas and liquid at zero gravity, induced by vortex motion, is developed. It utilizes an explicit solution method for a set of equations describing rotating gas-liquid flows. The vortex motion is established by a tangential fluid injection. A Lax-Wendroff two-step (McCormack's) numerical scheme is used. This program can be used to study the fluid dynamical behavior of the rotational two-phase fluids in a cylindrical tank. It provides a quick/easy sensitivity test on various parameters and thus provides the guidance for the design and use of actual physical systems for handling two-phase fluids.

Key Words: computer code; gas-liquid separation; numerical modeling; two-phase vortex motions
List of Notations

- $A_a, A_{ak}$: Added mass coefficients
- $A_d, A_{dk}$: Drag coefficients
- $\bar{\rho}_k$: Body force density
- $C_{ij}, C_{pk}, C_{dk}$: Generalized coefficients
- $d_1$: Bubble diameter
- $d_2$: Liquid (droplet) diameter
- $\bar{M}_k$: Effective interfacial force density
- $n_a$: Exponent used for $w_{ak}$
- $n_d$: Exponent used for $w_{dk}$
- $p$: Pressure
- $R$: Tank radius
- $Re$: $\rho \bar{v} R / \nu$, Reynolds number
- $R_{j1}, R_{j2}$: Jet opening, $R_{j1} < r < R_{j2}$
- $R_1$: Minimum radius considered in the numerical analysis
- $r$: Radial coordinate
- $t$: Time
- $V_j$: Averaged jet velocity
- $V_{r1}$: Gas radial velocity
- $V_{r2}$: Liquid radial velocity
- $V_s$: Velocity scale
- $V_{\theta 1}$: Gas tangential velocity
\( V_{\theta 2} \) Liquid tangential velocity
\( \overline{V}_1 \) Gas velocity
\( \overline{V}_2 \) Liquid velocity
\( w_{ak} \) Weighting function for added mass coefficients
\( w_{dk} \) Weighting function for drag coefficients
\( \alpha_1 \) Gas volume fraction
\( \alpha_2 \) (1-\( \alpha_1 \)), liquid volume fraction
\( \gamma \) Exponent for diameter variation
\( \theta \) Circumferential coordinate
\( \mu_1 \) Gas dynamic viscosity
\( \mu_2 \) Liquid dynamic viscosity
\( \mu_k \) Total effective viscosity
\( \mu_k^t \) Turbulence or eddy viscosity
\( \nu_k \) Kinematic viscosity
\( \rho_1 \) Gas density
\( \rho_2 \) Liquid density
\( \rho_k \) Averaged density of k-phase
\( \langle \rho^2 \rangle \) \( \alpha_1 \alpha_2 \rho_1 \rho_2 + A_a (\alpha_1 \rho_1 + \alpha_2 \rho_2) \)
\( \omega \) Angular velocity
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.</td>
<td>Two Step Difference Scheme (Backward Predictor-Forward Corrector Version)</td>
<td>34</td>
</tr>
</tbody>
</table>
| Figure 2. | Velocity Vector Distributions  
a) Gas phase, b) Liquid phase  
The annular region between two dashed circles is the region of injection | 35   |
| Figure 3. | Gas Volume Fraction Distributions                                           | 36   |
I. Introduction

Mechanical systems have been devised for producing artificial gravit fields to spin-up liquids in containers. These involve rotating mechani are cumbersome and, more importantly, have moving parts that can wear out. Here, liquid rotation created by fluid injection is considered. The detail analysis of the two-phase vortex model can be found elsewhere [1]. In this report, the details of the computer code are described.

The computer program was developed to study the fluid dynamical behavior of two-phase fluids in a tank at zero gravity. The phase separation between gas and liquid, induced by vortex motions, is of primary interest. The program utilizes an explicit solution method for a set of equations describing rotating gas-liquid flows. The vortex motion is established by a tangential fluid injection. A Lax-Wendroff two-step (McCormack's) numerical scheme is used in the computer program. This scheme uses a conservation form of a system of equations together with an auto time step feature.

The program was developed and tested on an HP-1000 minicomputer. The HP-1000's FORTRAN 77 is based on the American National Standards Institute (ANSI) 77 standard programming language FORTRAN (ANSI X3.9-1978). The HP FORTRAN 77 has extensions to provide a more structured approach to program development and more flexibility in computing for scientific applications. It fully implements the Military Standard Definition (MIL-STD-1753) of extensions to the ANSI 77 standard. In order to make the computer code more useful for other computer systems, modifications have been made so that the code is closer to the ANSI 77 standard and thus less system dependent. Some limited extensions are still kept in order to produce the code in the HP-1000. Since the graphic routines are system dependent and must be modified with their equivalents at each computing facility, the original graphic code has not been included in this report. All
lines preceded by "*V" are originally adopted to use the vector operation package supplied by Hewlett Packard. The speed of the code can be increased by replacing many "do-loop" operations in the code with high speed vector operations. Little effort is required to incorporate the vector operation into the code if the vector operation package is in the system.

Thus with limited effort, the program can be adapted easily to most systems accepting the ANSI 77 standard FORTRAN. For example, the EMA (Extended Memory Area) statements may have to be removed from the code for some computers. Also double precision real numbers (Real * 8) could be replaced by single precision real numbers.

II. Model Equations

The vortex induced model is based on a two-phase, two-fluid continuum [2]. It incorporates several interactions between phases; namely fluid drag and virtual mass effects and it can be modified to include additional interaction effects. Detailed analysis of the model has been reported in Ref. 1. A brief summary of the system of equations is given below.

The equations for the conservation of mass and momentum for the two fluid two-phase model in an one-dimensional, axisymmetric case (i.e. $\frac{\partial}{\partial z} + \frac{\partial}{\partial \theta} = 0$) are:

$$a_1 + a_2 = 1$$

$$\frac{3 \rho_k c_k}{\partial t} + \frac{3 \rho_k V r_k}{\partial r} = 0$$
\[
\frac{\partial \alpha_{rk}}{\partial t} + \frac{\partial \alpha_{rk} \alpha_{rk}^2}{\partial r} = -\alpha_{rk} \theta_{rk} \frac{\partial p}{\partial r}
\]

\[+ \alpha_{k} \sum_{l=1}^{2} C_{kl} \left( \frac{\partial \alpha_{rl \theta_{rl}}}{\partial r} - \alpha_{l \theta_{rl}} + \alpha_{l \rho_{rl} B_{rl}} \right) \]

\[+ \alpha_{k} C_{dk} \left( V_{r1} - V_{r2} \right) \]

\[
\frac{\partial \alpha_{rk rk}}{\partial t} + \frac{\partial \alpha_{rk} \alpha_{rk} \alpha_{rk}}{\partial r} + \alpha_{k} \alpha_{rk} \alpha_{rk} - \alpha_{rk} \frac{\partial p}{\partial r} = \alpha_{k} \sum_{l=1}^{2} C_{kl} \left( \frac{\partial \alpha_{rl r\theta_{rl}}}{\partial r} + \alpha_{l r \theta_{rl}} + \alpha_{l \rho_{rl} B_{rl}} \right) \]

\[+ \alpha_{k} C_{dk} \left( V_{g1} - V_{g2} \right) \]

for \( k = 1 \) and 2 and with

\[C_{p1} = \frac{a_{1} a_{2} R_{2} + a_{1}}{\langle \rho^2 \rangle} \]

\[C_{p2} = \frac{a_{1} a_{2} R_{1} + a_{1}}{\langle \rho^2 \rangle} \]

\[C_{l1} = \frac{a_{2} \rho_{2} + a_{2}}{\langle \rho^2 \rangle} \]

\[C_{l2} = C_{21} = \frac{a_{1}}{\langle \rho^2 \rangle} \]

\[C_{22} = \frac{a_{1} \rho_{1} + a_{1}}{\langle \rho^2 \rangle} \]
\[ C_{d1} = -a_2 \rho_2 A_d / \langle \rho^2 \rangle \]
\[ C_{d2} = a_1 \rho_1 A_d / \langle \rho^2 \rangle \]

and

\[ \langle \rho^2 \rangle = a_1 \rho_1 \rho_2 + A_a (a_1 \rho_1 + a_2 \rho_2) \]

The effective stresses are modeled as

\[ \tau_{\rho r k} = 2 \mu_k e \partial V_{rl} / \partial r \]
\[ \tau_{\rho \theta k} = \mu_k e \partial (V_{rl} / r) / \partial r \]
\[ \tau_{\rho \phi k} = 2 \mu_k e \partial V_{rl} / \partial r \]

with

\[ \mu_k^e = \mu_k^t + \mu_k^\tau \]

and the interfacial forces are modeled in the form of

\[ \bar{M}_1 = A_d (V_2 - V_1) + A_a d \partial (V_2 - V_1) / dt. \]

\[ \bar{M}_1 \]

is the force density acting on the phase 1 by the phase 2. \( A_a \) and \( A_d \) are the added mass and drag coefficients, respectively.

The incompressibility condition is reduced to \( a_1 V_{r1} + a_2 V_{r2} = Q_r / r \), where \( Q_r \) is the net radial outflow.

In the program \( Q_r = 0 \) is assumed, since the mixture pumped out is injected immediately back into the tank at the nearby location. The net volume or mass in the system is effectively unchanged except for the net change on the angular momentum. Thus, the pump system (withdrawal and injection) acts as a body force.
on the mixture at the nozzle location. The net momentum gain is thus the momentum introduced into the system minus the local momentum pumped out. Therefore, we will model this pumping dynamic by body forces without considering the mass transfer. That is, the body force density $\alpha_k \rho_j \mathbf{B}_k$ will be replaced by the net momentum gain, $\frac{\alpha_k \rho_j V_j}{2\pi} (V_j \mathbf{n} - \mathbf{v})$ at the nozzle location, where $V_j$ is the injection speed.

III. Numerical Method

The complete solution of the complicated system of equations can only be obtained through numerical methods. An improved Lax-Wendroff, two-step scheme, (also referred to as MacCormack's method) [3, 4] is adopted for solving this time-dependent problem. This non-centered differencing scheme, using a full step backward prediction and forward correction version, requires no explicit artificial viscosity if a proper stability condition is satisfied. Using this technique for solving fluid flow problems is very efficient and has been in widespread and successful use for some time. It is good both for the time-accurate computation of steady and unsteady flow problems. The general features of the scheme are: 1) its explicitly conservative form, 2) it is a two-step predictor-correction type, 3) it is three point, two level - that is, the solution of $f_i^{n+1}$ at level $n+1$ depends only on three values of $f_i^n$ at level $n$, and 4) it is second-order accurate in time and in space.

For using the MacCormack's numerical technique, the system of equations can be expressed in the conservative form as:

$$W_t = F_r + P_r + gG_r + S$$

Here the subscripts (t and r) denote partial differentiation with respect to $t$ and $r$, respectively, and $W, F$, $P_r$, $gG_r$ and $S$ are column matrices with five
elements. All the components of $F$, $P_r$, $qG_r$, and $S$ can be regarded as functions of the components of $W$ which are the independent variables. The fundamental theory of the MacCormack's scheme is briefly given below.

For second order accuracy, the solution could be written as

$$W^1 = W^0 + \Delta t \cdot W_t^0 + \frac{(\Delta t)^2}{2} \cdot W_{tt}^0$$

$$= W^0 + \Delta t \cdot W_t^0 + \frac{\Delta t}{2} \left( W_t^o + \Delta t \cdot W_{tt}^o \right)$$

$$= \frac{1}{2} \left( W^0 + \Delta t \cdot W_t^0 \right) + \frac{1}{2} \left( W^0 + \Delta t \cdot W_t^p \right)$$

$$= \frac{1}{2} \left( W^p + W^c \right)$$

where

$W^0 = W^0 + \Delta t \cdot W_t^0$ is the predicted value,

and

$W^c = W^0 + \Delta t \cdot W_t^p$ is the corrected value.

The superscripts denote the time-level of the information and subscripts denote the partial derivative with respect to either time $t$ or space $r$. Specifically, superscripts 0 and 1 are the initial and the completely advanced time (here two steps) plane; $p$ and $c$ are the predicted (1st step) and corrected (2nd step) time plane. Thus, $W_t^0$ is the time derivative of $W$ evaluated at the initial time, and $W_t^p$ is time derivative of $W$ evaluated at the predicted time.

Fig. 1 shows the diagram of the two step difference scheme used in the computer program. Due to the difference scheme, the spatial location after each step in time is a half grid off from the original one. Thus, the spatial offset
which resulted from a backward predicting step will cancel with those of the forward correcting step.

Numerically, the predicted values are

\[ w_i^p = \frac{1}{2} (w_{i-1/2}^p + w_{i+1/2}^p) \]

where

\[ w_{i-1/2}^p = \frac{1}{2} (w_{i-1}^o + w_i^o) + \Delta t w_t^o \]

\[ = \frac{1}{2} (w_{i-1}^o + w_i^o) + \frac{\Delta t}{\Delta r} [(f_i^o - f_{i-1}^o) + \frac{(g_i^o + g_{i-1}^o)}{2} (g_i^o - g_{i-1}^o)] \]

\[ + \frac{\Delta t}{2} (s_i^o - s_{i-1}^o) + \Delta t \hat{p}_{i-1/2}^c \]

and the corrected value is evaluated at the predicted time place, that is at \( w_{i+1/2}^p \). Thus

\[ w_i^c = w_i^o + \Delta t w_t^p \]

\[ = w_i^o + \frac{\Delta t}{\Delta r} [(f_i^{p+1/2} - f_{i-1/2}^p) + \frac{(g_i^{p+1/2} + g_{i-1/2}^p)}{2} (g_i^{p+1/2} - g_{i-1/2}^p)] \]

\[ + \frac{\Delta t}{2} (s_i^{p+1/2} + s_{i-1/2}^o) + \Delta t \hat{p}_i^c \]

Here \( \hat{p}_{i-1/2}^c \) and \( \hat{p}_i^c \) are the pressure correction terms at half and full time steps respectively. Thus, for each time step, the advance is carried out in two
steps: a full step backward predictor, and then a forward corrector. As indicated in the diagram, the subscript i is the regular mesh spatial location at which solution is to be advanced, i + 1 is the spatial location of regular mesh points immediately to the right and left of the location i, i + 1/2 is the location midway between i and i + 1 or between i - 1 and i at the predictor plane. Thus, for each time step as the procedure advanced, the outermost data points at the boundary are not updated through the numerical scheme. The values at the boundary are to be given through some suitable boundary conditions. The numerical procedure utilizes a uniformly preselected spatial mesh and variable time increment. To avoid a singularity at the center of the core region, a finite radius $R_i$ is used for the inner boundary. The tank radius $R$ is the outer boundary. The time step is determined at each time step to ensure numerical stability [5]. For a finite grid size $\Delta r$, the maximum time step $\Delta t$ is given by

$$\Delta t_k = \frac{1}{|C_{dk}| + |V_{rk}|/\Delta r + \frac{2}{\Delta r^2} (a_1 u_1 c_{kl} + a_2 u_2 c_{k2})}$$

where $k = 1$ and 2. The minimum $\Delta t_k$ (with some rounding off) is used for the time step. Normally, the technique with the time step condition gives fairly good numerical stability. However, in critical conditions numerical damping can be added either for damping oscillations due to large gradients or for accelerating the calculation by increasing the time step. A damping factor, $D$ thus was added in the program as

$$W_{i}^{1D} = W_{i}^{1} (1-D) + (W_{i-1}^{1} + W_{i+1}^{1} - 2W_{i}^{1}) D$$
where \( \hat{w}_i^1 \) is the value obtained based on the two-step scheme, and \( \hat{w}_i^D \) is the value after the damping factor \( D \) is added. A typical value of \( D = 0.2 \) can be used for debugging the program. If no damping factor is desired, \( D = 0 \) should be used.

The computer program was written in a Fortran 77 based computer code. The code will permit evaluation of the effects of various parameters which control the fluid dynamical behavior. These include tank size, fluid properties, such as density and viscosity, etc., characteristic gas bubble and liquid drops sizes, and relative location of injection nozzles.

A sample input and its output are shown on Exhibits A and B, respectively. The initial conditions for the gas and liquid volume fractions are taken to be 25% gas and 75% liquid. These fractions are uniformly distributed over the circular cross-section of the cylindrical tank. Initially both fluids are at rest. Other parameters can be found in the sample input in Exhibit A. The resulting velocity distributions and gas volume fraction as function of time for the sample run are given in Figs. (2) and (3) respectively. The velocity distributions are displayed along equally spaced rays at different times to enable clear observation. These velocity vector fields indicate all flows are primarily in angular rotation with gas phase tending to move inward and liquid phase trying to move outward, as expected. As the result of these radial movements, the volume fraction distribution is also changed with time. And as expected, the gas volume fraction is increasing at the inner region and decreasing at the outer region as shown in Figure 3. More detailed results have been reported in Ref. 1.
IV. Program Details

The complete computer code is listed in Appendix A. The code consists of a main program, GLVM and several subroutines. It is written in subroutine form such that each subroutine performs an individual task. Each logical part is clearly isolated and it can be easily modified to reflect different modelings for the interfacial forces. The interactive input mode with self-instruction is used for easy parameter insertion. Many instructive internal documentations are included in the program. In the following, each subroutine is listed with a brief description of its major function.

1) GLVM, the Main Program.

*To initialize data and start the program: Logical unit to save data (LUS), (Logical Unit is 1 for terminal, and 6 for printer), job identification notes (NOTES), data file name for saving data (NAMR), initial time (TU), final time (TMAX), time interval for data output (DTPRT), etc.

*To control the calling sequences to the other subroutines.

*To check the time step.

*To save, print (and plot) the output data.

*To obtain the predicted and corrected values in the two step, numerical scheme.

*To impose boundary condition.

*To update the data, time, and step number for the new time step.

*To provide a shutdown procedure either in normal (e.g., \( t > t_{\text{max}} \)) or abnormal (e.g., \( \Delta t \) is too small) conditions.

2) INIT, Initialization.

*To input the test parameters, initial conditions and set-up the initial column matrix W.
Default values are provided for most of the parameters. The default values are listed at each interactive input step. If the default value is acceptable, a comma "," is inputted.

Some of the relevant symbols used are listed below:

- **ALMT**  Limit values of \( a_1 \), \( \text{ALMT}(1) < a_1 < \text{ALMT}(2) \).
- **DAMP**  Numerical damping factor. Normally set to 0.
- **DEN1D2**  Density ratio, \( \rho_1/\rho_2 \)
- **DO**  Base diameter, i.e., \( d_k = d_{ok}^\gamma \)
- **DS**  Density scale = \( \rho_2 \)
- **EVF**  \( \mu^t/\mu \) effective eddy viscosity factor.
- **GAMMA**  Diameter exponent \( \gamma \)
- **IVTX**  Type of simple initial flow: 0 = at rest, 1 = simple rotation, 2 = Hammel-Oseen Vortex, 3 = G.I. Taylor Vortex. This is needed only when there is no data file (NAMR) given for an initial condition.
- **IW**  Boundary condition at wall (for tangential component). 0 = free-slip (no-skin friction), 1 = non-slip. Also, when \( I_W = 1 \), a factor of \((1-r)^{0.1}\) was included on IVTX flow to simulate an initial power law boundary layer.
- **MM**  Size of data arrays, \( Mf \geq NG \). MM appears in many subroutines.
- **\( \mu_k \)**  Dynamic viscosity for phase \( k \).
- **MLEF**  \((1 + \text{EVF}) * \muJ\). Effective viscosity.
- **MULD2**  \( \mu_1/\mu_2 \), Viscosity ratio.
- **NA, ND**  \( n_a, n_b \), Exponents for weighting function for drag and added mass coefficients, \( A_a \) and \( A_d \).
NAMR: Data file name for initial condition, if any.
NG: Number of grid points used.
OMEGA: \( \omega \), initial rotation speed, \( V_\theta = \omega r \), when IVTX > 0.
PS: Pressure scale, DS \* VS^2
QJ, VJ: Injection flow rate and speed.
RE: Reynolds number, \( \rho_2 V \sqrt{\frac{R}{\mu_2}} \)
RJ1, RJ2: Jet opening, RJ1 < r < RJ2
RPEAK: Location of the peak speed of the initial vortex, if IVTX > 1
RTANK: Tank radius = Length scale LS.
VPEAK: Peak speed of the initial vortex, if IVTX > 1.
TJ1, TJ2: Injection time, TJ1 < t < TJ2.
TS: Time scale = LS/VS
VS: Velocity scale.

The format of the data file for the initial condition (if any) is a six column and NG row data file, where NG is the number of grid points. The column sequence is K, \( \alpha_1(K) \), \( V_{r1}(K) \), \( V_{r2}(K) \), \( V_{s1}(K) \), \( V_{s2}(K) \), where K is the grid point number, and the rest of the terms are the gas volume fraction, radial gas velocity, radial liquid velocity, tangential gas velocity and tangential liquid velocity respectively. The data format is free.

3) DELA, \( \Delta A \)

To determine the fraction of grid size in which the injection is made.

0 \( \leq \Delta A \leq 1 \). This is used to define the location of jet. The region of injection could cover several full or fractions of grid sizes.

4) DERIV1, Derivative

To get the first derivative of a data array using a center difference scheme except the two end points in which three points near the boundary are used.
5) DGCOEF, Generalized Coefficients

To calculate the added mass, drag and all the generalized coefficients ($A_a$, $A_d$ and $C_{ij}$). This is the heart of the modeling.

The effective coefficients are modeled as:

$$A_a = A_{a1}w_1 + A_{a2}w_2$$
$$A_d = A_{d1}w_1 + A_{d2}w_2$$
$$A_{a1} = \alpha_1\alpha_2\beta_2/(\alpha_1 + 2\alpha_2/(1 + 3\alpha_1))$$
$$A_{a2} = \alpha_1\alpha_2\beta_1/(\alpha_2 + 2\alpha_1/(1 + 3\alpha_2))$$
$$A_{d1} = 18 \mu_2\alpha_1/d_1^2\alpha_2$$
$$A_{d2} = 18 \mu_1\alpha_2/((1 - \alpha_2/0.8^2d_2^2)$$
$$w_{a1} = \alpha_2\alpha_2/(\alpha_1\alpha_1 + \alpha_2\alpha_2)$$
$$w_{a2} = 1 - w_{a1}$$
$$w_{d1} = \alpha_2\alpha_2/(\alpha_1\alpha_1 + \alpha_2\alpha_2)$$
$$w_{d2} = 1 - w_{d1}$$

6) DWPDE, Partial Differential Equations.

To evaluate the values of the increments on the column matrix $W$ from the partial differential equations. This is the major part of the McCormack’s scheme. In each complete time step this routine will have to be called twice.

7) FNDST, At

To determine the suitable time-step size.

8) FSOFW, Column matrices $F$ and $S$.

To determine the convective matrix $F$ and the source matrix $S$. 
9) JET, Injection.

To determine the momentum source due to the jet injection.

10) SIZES

To determine the gas bubble and liquid droplet sizes. In the model the sizes were modeled to be functions only of the volume fraction, i.e.

\[ d_k = d_{0k} \alpha_k^\gamma. \]

Different models for size distributions could be easily adopted here.

11) TauOfw

To determine the stress tensor \( \mathbf{\tau} \) and its derivative.

12) UOFW

To convert the column matrix \( \mathbf{w} \) into the physical independent variables, such as \( \alpha, V_r, V_g \).

V. Summary

A computer program aimed at the phase separation between gas and liquid at zero gravity, induced by vortex motion, is developed. The vortex motion is created by fluid injections. The computer program uses a FORTRAN 77 based code and HP-1000 minicomputer. It is flexible and accepts various input parameters for different flow conditions. Other interaction effects can also be added or modified easily. This program can be used to study the fluid dynamical behavior of the rotational two-phase fluids in a cylindrical tank. It provides a quick/easy sensitivity test on various parameters and thus provides the guidance for the design and use of actual physical systems for handling two-phase fluids.
VI. Acknowledgments

We would gratefully acknowledge the support received from NASA's Kennedy Space Center to carry out the model-development work described herein. Specific thanks go to Mr. Frank Howard whose involvements and inputs on this project have been most helpful.

VII. References

Appendix A

Code Listing

:GLVM T=00004 IS ON CR T4 USING 00126 BLKS R=0000

0001 FTN77
0002 $EMA /DATA/, /WWW/, /COEFF/, /SOURCE/, /FANDS/, /TAU/
0003 $FILES 1.2
0004 PROGRAM GLVM('99), (B6D425.1537)
0005 C
0006 C THIS PROGRAM WAS DEVELOPED TO STUDY THE FLUID DYNAMICAL BEHAVIOR
0007 C OF A ROTATIONAL TWO-PHASE FLUIDS (GAS/LIQUID) IN A CYLINDRICAL TANK.
0008 C THE VORTEX MOTIONS ARE ESTABLISHED BY TANGENTIAL FLUID INJECTION.
0009 C THE PROGRAM WAS DEVELOPED ORIGINALLY BY T.T. YEH OF NBS.
0010 C IT WAS BASED ON NP'S FORTRAN 77 (ANSI 77 + MIL-STD-1753)
0011 C
0012 C WHEN WHO WHAT
0013 C 5062XX TTY 21 O-G FUEL TRANSFER, START-UP STAGE.
0014 C 50144 C LA. WENDROFF 2-STEP SCHEME (FULL STEP PREDICTION+CORRECTION)
0015 C WITH NUMERICAL DAMPING FACTOR (NORMALLY SET TO ZERO)
0016 C COHESITION FORM, VARIABLE (AUTO) TIME STEP
0017 C REAL=
0018 C INTERFACIAL FORCES: DRAG, ADDED MASS
0019 C PUMP CONDITION: MOMENTUM SOURCE BUT NO MASS SOURCE
0020 C 50715 TTY GENERALIZED EQUATIONS AND COEFF. Cij
0021 C 501015 TTY IN ANSI 77 STANDARD (WITH A LITTLE EXCEPTION FOR
0022 C TESTING IN HP-1000)
0023 C
0024 C ***** INTERNAL SUBROUTINES *****
0025 C DELA, DERIV1, DDCOEFF, DNPDE, FNDDT, FSDFW,
0026 C INIT, JET, SIZES, TAOUFW, AND UDFW
0027 C *** MOST OF THE LIST OF NOTATIONS ARE GIVEN IN SUBROUTINE..INIT ***
0028 C
0029 C CHARACTER NAMR=16, NOTES=72
0030 C INTEGER I, JOS, J, JTIME(3), K, MM, NPRT, NT
0031 C , IW, LUP, LUS, NG, NGM1, NGM2
0032 C PARAMETER (HM=101)
0033 C REAL= 6 BA, MM, 2, DDT, DT, DMAX, DMIN, DTPRT, PZERO
0034 C X , DVI, T, TMAX, TPTRT, UJT, VDR(2)
0035 C Y , R1, R2, T1, T2, Q, VJ
0036 C 1 , ALNT, U, V, ALF, P, R, W, WP, WN, DW, RDP, RH, FS
0037 C 4 , BR, BRH, RHO, MUEF, V1B, NA, ND
0038 C 5 , DO, GAMMA, DAMP, DR
0039 C 6 , TR, TRA, TAA, RTRA, RTRA, C, CPA, CD
0040 C COMMON
0041 C Y /JETS/ R1, R2, T1, T2, Q, VJ
0042 C Z /CONT/ IW, LUP, LUS, NG, NGM1, DAMP, DR
0043 C 1 /ALPLMT/ ALMT(2)
0044 C 2 /COEFF/ C(MM, 2, 2), CPA(MM, 2, 2), CD(MM, 2)
0045 C 3 /DATA/ U(MM, 2), V(MM, 2), ALF(MM, 2), P(MM, 2), R(MM)
0046 C 4 /DRA/ RHO(4), MUEF(2), V1B(2), NA, ND
0047 C 6 /FANDS/ F(MM, 5), S(MM, 5)
0048 C 7 /SOURCE/ BR(MM), BRH(MM)
0049 C 8 /TAU/ TR, TRA(MM, 2), TAA, RTRA(MM, 2), RTRA(MM, 2)
0050 C 9 /WWW/ W(MM, 3), WP(MM, 3), WN(MM, 3), DW(MM, 3), RDP(MM, RH(MM)
0051 C 0052 C EQUIVALENCE (UW, Va)
0053 C
0054 C ***** RHO(1) RHO(2) (I.E. PHASE=1=GAS, PHASE=2=LIOUID) *****
0055 C DTPRT TIME STEP FOR PRINTOUT (AND PLOT)
0056 C
0057 C
LUP=1  ! LU FOR PRINTING DEBUG DATA(=1 TERMINAL)
LUS=6  ! LU FOR STORING DATA(=6 PRINTER)

7 FORMAT(2X,A,3(JPE12.4))
6 FORMAT(2X,A,3(JPE12.4))

WRITE(1,7) 'Enter lu for saving data. D.F.=',LUS
READ(1,*1) LUS

C KEEP JOB TIME FOR FUTURE REFERENCE
CALL EXEC(11,'TIME,JTIME(1))

IF(LUS .NE. 1 .AND. LUS .NE. 6) THEN
  WRITE(1,'(A)') 'Enter FILE NAME for saving data.'
  READ(1,'(A)') NAME
  LUS=99
  OPEN(LUS,FILE=NAME,IOSTA)=105,STATUS='NEW',ERR=999,
  ENDIF

WRITE(1,'(A)') 'Enter NOTES((73 CHAR.) for the job'
READ(1,'(A)') NOTES
WRITE(LUS,('(9I4)')) JTIME

C To set-up the initial condition.
CALL INIT

'NGM2=NG-2
VD(1)=2.*MUEF(1)/DR**2  ! For determining time step
VD(2)=2.*MUEF(2)/DR**2

T=0.  ! INITIAL TIME
TM=5.
DPP=0.2

WRITE(1,8) 'Enter INITIAL and FINAL TIMES. D.F.=',T,TM
READ(1,*1) T,TM

WRITE(1,8) 'Enter TIME STEP for output. D.F.=',DPP
READ(1,*1) DPP

DTPM=1.0D-6  ! SET MINIMUM TIME STEP
DTPX=DPP

NT=0  ! Time step number
NPR=0
PZ=0.0

DT=DTPM

NT=NT+1

CALL UOFM(W,R,NG)

IF(T .GE. TPR .OR. T .GT. TM) THEN
  IF(NP .GE. 1) THEN
    NPP=NPR-1
  ELSE
    TPR=TPR+DPP*(1.+DNINT((T-TPR)/DPP))
  ENDIF
ENDIF
ENDIF

WRITE(LUS,21) 'MP=',NPR,'=0','NT=',NT,'=0','DT=',DT
WRITE(LUS,21) 'MP=',NPR,'=0','NT=',NT,'=0','DT=',DT
WRITE(LUS,21) 'MP=',NPR,'=0','NT=',NT,'=0','DT=',DT
DO 30 J=1,NG
  WRITE(LUS,'(I3,F6.4,5(IPE(1.3)))')
  CALL DGLDEF(ALP,NG)  ! DRAG AND GENERALIZED COFFFF.
  CALL TAUFW(MUE,W,DR,R,NG)  ! STRESS
  VJT=VJ
  IF( T .LT. TJ1 .OR. T .GT. TJ2) VJT=0  ! NO INJECTION
  CALL JET(BA,RMO,DR,V,VJT,NG)  ! MOMENTUM SOURCE
  CALL FSOFW(W,BA,R,NG)  ! CONVECTIVE -F AND SOURCE-S
  C DETERMINE THE TIME-STEP SIZE
  IF(NT .GT. 1) THEN
    CALL FNDT(DT,DR,NG)
    DT=DT+5.*DTMIN/10.**NT  ! INITIAL INPLUSE TREMENT
  ENDIF
  IF(DT .GE. DTMIN) THEN
    I=LOG10(DT/DTMIN)
  ENDIF
  DT=DTMIN*10.**I
  IF(DT .GT. DTMAX) DT=DTMAX
  ELSE
    WRITE(LUP,'(SX,A1PEI3.3)')
  ENDIF
  STOP DUE TO TOO SMALL TIME STEP. DT=',DT
  GOTO 9999
  ENDDIF
  ENDDIF
  C BACKWARD PREDICTOR
  CALL DWPE(DW,RDP,DR,DT,RH,NGM1)  ! INCREMENT
  DO 40 J=1,NGM1
    DO 40 I=1,5
    WP(J,I)=0.5*(W(J+I)+W(J,I)+DW(J,I))  ! BASE+INCREMEMT
  40   CONTINUE
  C PREDICTION DATA COMPLETED, CONTINUE FOR CORRECTION
  CALL UDIF(WP,RH,NGM1)
  CALL DGLDEF(ALP,NGM1)
  CALL TAUFW(MUEF,DR,RH,NGM1)
  CALL JET(BA,RMO,DRH,V,VJT,NG)
  CALL FSOFW(W,BA,RH,NGM1)
  C FORWARD CORRECTION
  CALL DWPE(DW,P(2),DR,DT,R(2),NGM2)
  P(1)=PZERO
  DO 50 J=1,NGM2
  50  CONTINUE
C 2ND STEP (PREDICTION+CORRECTION) COMPLETED

C Estimation of initial P completed, return to the initial condition
C and start to advance the program in time.

C DATA W AI THE NEW TIME STEP COMPLETED

C IMPOSED B.C. #6.4

DFl=0.5*(W(2,1)+W(2,2))-0.

ALP(1,1)=.5*(W(1,5)+W(2,5))-DFl*DT/DR/RH(1)

IF(ALP(1,1) .LT. ALMT(1)) ALP(1,1)=ALMT(1)

IF(ALP(1,1) .GT. ALMT(2)) ALP(1,1)=ALMT(2)

WN(1,5)=ALP(1,1)*R(1)

WN(1,1)=0.  'NO RADIAL VEI.

WN(1,2)=0.

WN(NG,1)=0.

WN(NG,2)=0.

WN(NG,3)=0.

WN(NG,4)=0.

ENDIF

C ARTIFICIAL TAMING

DO 60 I=1,5

W(1,1)=W(1,1)-DAMP*WN(1,1)+DAMP*WN(2,1)

W(NG,I)=W(NG,I)+DAMP*WN(NG,I)

DO 60 NG=2,NGM

W(j,j)=W(j,j)+DAMP*(WN(j-1,j)+WN(j+1,j)-2*WN(j,j))

C SOLUTION FOR THIS TIME STEP COMPLETED

90 T=T+DT  'UPDATE TIME AND CONTINUE TO THE NEXT STEP

GOTO 10

999 WRITE(LUP,7) 'OPEN FILE FAILED ON FILE:

999 WRITE(LUP,7) NMHR

999 WRITE(LUP,7) 'IOSTAT=',I0S

999 CONTINUE
CALL EXEC(11, JTIME, JTIME(1))
WRITE(*, '(JSX, SI4)') JTIME
CLOSE(LUS)
END

==============================================
REMA /DATA/ /WWW/ /SOURCE/
SUBROUTINE INIT(, (300000, 1537)

C TO SET-UP THE INITIAL CONDITIONS
INTEGER I, IOS, ITLOG, JTX, J, K, MM
Z, JTIME, JN, LUP, LUS, NG, NGM
PARAMETER (MM=101)
REAL= B AMT, W, WP, WM, DW, DDP, RH
1 U, V, ALP, P, R, F, S
4 RIO, MUEF, VIB, MA, ND, DO, GAMMA
7 BM, BHN, DSTR, DR
Y RX, RJ2, TIJ, TIJ, QJ, VJ
X A2, SELA, DEMID2, DI, DJ, PI, RE, RMIN, RTANK
X, DS, LS, VS, TS, PE
X ALP10, OMEGA, RPEAK, SJ, B, VPEAK
X, EDF(2), VT, MU(2), MU1D2

COMMON
L /ALPLMT/ ALMT(2)
V /JETS/ RJ1, RJ2, TIJ, TIJ2, QJ, VJ
7 /CONTP/ IW, LUP, LUS, NG, NGM, D=WP, DR
3 /DATA/ U(MM, 2), V(MM, 2), ALP(MM, 2), P(MM), R(MM)
4 /PAGI/ RH(4), MUEF(2), VIB(2), MA, ND
5 /DSIZE/ DO(2), GWMA(2)
7 =/SOURCE/ BR(MM), BRH(MM)
9 =/WWW/ W(MM, 5), WP(MM, 5), WM(MM, 5), DW(MM, 5), DDP(MM), RH(MM)

DATA PI/3.14159628/

C ===== NOTES: PHASE=1=GAS, PHASE=2=LIQUID =====
C RHO DENSITY, RHO(1) ( RHO(2)
C ALMT LIMIT VALUES FOR ALP1, ALM(1)(ALP1)(ALM(2)
C DAMP NUMERICAL DAMPING FACTOR, NORMALLY =0.
C DS, LS, VS, TS DENSITY, LENGTH, VELOCITY AND TIME SCALES
C RE REYNOLDS =US*RTANK*RHO(2)/MU(2)
C RTANK TANK RADIUS
C IVXT TYPE OF INITIAL FLOW.
C =/GAS, 1=PURE EROSION, 2=M.3=GIT Vortex
C RPEAK, VPEAK VORTEX PARAMETERS
C OMEGA PURE ROTATION. W=OMEGA^R
C EVF, MUEF EFFECTIVE VISCOITY, MUEF(1+EVF)*MU
C DO, GAMMA DIA. PARAMETERS: D=DO*ALP**GAMMA
C RX, RJ2, TIJ, TO DEFINE JET SIZE, PUMPING TIME
C TJ2, QJ, VJ VOLUME FLOW RATE AND INJECTION MEAN SPEED

FORMAT(2X, A, 2X, A))
FORMAT(2X, A, 3J15)
FORMAT(2X, A, 3(1PE12.4))
FORMAT(2X, A, 3(1PE15.4))
FORMAT(2X, A, 2(1PE11.4))
DEFINE THE PARAMETERS FOR THE PROBLEM.

RTANK=1.0D0
RHO(2)=1.000D+3
MU(2)=1.514D-3

WRITE(1,0) 'ENTER RTANK(M) OR DEFAULT',RTANK
READ(1,0) RTANK

The values of RTANK,RHO(2), and MU(2) could all be set to 1, since the length and density scales are based on RTANK, and RHO(2) and the value of the viscosity MU(2) can be combined into and specified by the Reynolds number RE. Thus all characteristic scales, LS, VS, TS, and DS are fixed after RE is given.

RE=1.0DS
WRITE(1,0) 'enter Reynolds no., RE. D.F.=',RE
READ(1,0) RE
LS='TANK
LENGTH SCALE (M)

D1=RHO(2)
DENSITY SCALE (KG/M^3)

VS=RE/MU(2)/RHO(2)/LS VELOCITY SCALE (M/S)

TS=LS/VS
TIME SCALE (S)

PS=DS/VS
PRESSURE SCALE

AFTER THIS POINT ALL VARIABLES ARE BASED ON THE CHAR. SCALES
i.e. ALL VARIABLES ARE DIMENSIONLESS

DEN12=1.293D0/1.000D3
DI/D2
MUID2=1.710D-5/1.514D-7
MUI/MU2

RHO(2)=RHO(2)/DS

MU(2)=MU(2)/(DS*LS*VS)

RHO(1)=DEN12=RHO(2)

MU(1)=MUID2=MU(2)

ALMT(1)=0.0818
MIN. OF ALP1

ALMT(2)=0.9999
MAX. OF ALP1

DO(1)=1.0-2/LS GAS DIAMETER at ALP1=1

DO(2)=1.0-2/LS LIQUID DIAMETER at ALP1=1

GAMMA(1)=2.0-1

GAMMA(2)=2.0-1

EVF(2)=1.03 TURB. + PHASE-DISPERSION EFFECTS

EVF(1)=DEN12/MUID2*EVF(2)

MODEL

DAMP=0.0 NUMERICAL DAMPING FACTOR (e.g. = .2)

NA=4
WEIGHTING EXP. FOR ABM

ND=4
WEIGHTING EXP. FOR DRAG

WRITE(1,0) 'Enter DENSITY and VISCOSITY ratios.'
WRITE(1,0) 'D.F.=',DEN12,MUID2
READ(1,0) DEN12,MUID2

WRITE(1,0) 'Enter BASE DIAMETERS: DO1,DO2'
WRITE(1,0) 'D.F.=',DO
READ(1,0) DO

WRITE(1,0) 'Enter SIZE EXPONENT: GAMMA1,GAMMA2'
WRITE(1,0) 'D.F.=',GAMMA
READ(1,*) GAMMA

WRITE(1,0) 'Enter weighting exponent: NA,ND, D.F.=',NA,ND
READ(1,0) NA,ND
WRITE(1,0) 'Enter GAS VOLUME FRACTION limits:ALMT1,ALMT2.'
WRITE(1,0) 'D.F.=',ALMT
READ(1,0) ALMT
WRITE(1,0) 'Enter eddy viscosity factor. D.F.=',EVF
READ(1,0) EVF

IW=1
WRITE(1,7) 'Enter wall condition,1=nonslip,0=slip. D.F.=',IW
READ(1,7) IW
WRITE(1,0) 'Enter numerical damping factor. D.F.=',DAMP
READ(1,0) DAMP

DO 10 K=1,2
VIB(K)=18.*MU(K)
10 MIXE(K)=MU(K)*(1.-EVF(K))  ; EFFECTIVE VISCOSITY FOR STRESS

RHO(3)=RHO(1)*RHO(2)
RHO(4)=RHO(1)-RHO(2)
RMIN=0.1  ; MINIMUM FLOW RADIUS IN THE TANK
NG=101
NGM=NG-1
DR=(1.-RMIN)/NGM

C Initial cleaning-up.
DO 15 J=1,AM
   DO 15 K=1,6
10   W(J,K)=.0.D0
15   U(J,K)=.0.D0
C
C MOMENTUM SOURCE, JET CONDITIONS
RJ1=3.5D-1
RJ2=9.5D-1
VJ=10.;  ; TANGENTIAL INJECTION SPEED
TJ1=8.
TJ2=10.

WRITE(1,0) 'Enter JET SIZE defined by RJ1,RJ2. D.F.=',RJ1,RJ2
READ(1,0) RJ1,RJ2
WRITE(1,0) 'Enter INJECTION SPEED AND TIME RANGE, VJ,T1,T2'
WRITE(1,0) 'D.F.=',VJ,TJ1,TJ2
READ(1,0) VJ,TJ1,TJ2
READ(1,0) QJ=(RJ2-RJ1)*VJ  ; JET VOLUME FLOW RATE

DO 20 J=1,NG
   RJ=RMN+(J-1)*DR
   RH(J)=RJ*(J)+0.5*DR
   D(RJ)=DELAR(R(J)),DR,RJ1,RJ2)/(2.*PI)  ; JET DISTRIBUTION
   DRH(J)=DELAR(RH(J)),DR,RJ1,RJ2)/(2.*PI)  ; PER RADIAN
20   CONTINUE

C SETUP INITIAL CONDITIONS
IVX=4
0419 OMEGA=0.
0420 UPEAK=0.
0421 RPEAK=RMIN
0422
0423 NAME='Simple vortex.'
0424 WRITE(1,5) 'Enter data FILE NAME for initial cond., 1" any';
0425 WRITE(1,5) 'D.F.=',NAMR
0426 READ(1,'(A)') NAMR
0427
0428 IF( NAMR .NE. ',' .AND. NAMR .NE. 'Simple vortex') THEN
0429 C
0430 INITIAL CONDITION FROM A GIVEN FILE NAMR.
0431 OPEN(99,FILE=NAMR,1,STAT=IDS,STATUS='OLD',ERR=299)
0432 DO 25 J=1,NG
0433 25 cnd(99,...,ALP(J,1),U(J,1),U(J,2),V(J,1),V(J,2)
0434 CLOSE(99)
0435 ELSE
0436 C
0437 TO DEFINE INITIAL CONDITION.
0438 ALP10=2.00-1
0439 WRITE(1,8) 'Enter initial value of alpi. D.F.=',ALP10
0440 READ(1,#) ALP10
0441 WRITE(1,7) 'Enter type of vortex: 0=At rest, 1=Pure rotation'
0442 WRITE(1,7) '2=H.O.,3=GIT. D.F.=',IUTX
0443 READ(1,#) IUTX
0444 IF(IUTX GT 0) THEN
0445 IF(IUTX GT 1) THEN
0446 WRITE(1,8) 'Enter PEAK and LOCATION for classic vortex'
0447 WRITE(1,8) 'D.F.=',UPEAK,PPEAK
0448 READ(1,#) PPEAK,PPEAK
0449 IF(PPEAK LE. 0.) PPEAK=1. SINGULAR AT ZERO
0450 ELSE
0451 WRITE(1,8) 'Enter CIRCULAR SPEED(rad./unit time). D.F.='
0452 1
0453 READ(i,#) OMEGA
0454 ENDIF
0455 ENDIF
0456 DO 30 J=1,NG
0457 ALP(J,1)=ALP10
0458 VT=OMEGA*R(J)
0459 IF(IUTX GT 0) THEN
0460 RJB=R(J)/RPEAK
0461 IF(IUTX EQ. 0) THEN
0462 RJB=-H.O. VORTEX
0463 UT=UT+3.384*UPEAK/RJB*(1.-DEXP(-1.256*3*RJB**2))
0464 ELSE
0465 UT=UT+VPEAK*RJB*DEXP((1.-RJB*2)/2.)
0466 ENDIF
0467 ENDIF
0468 DO 30 X=1,2
0469 IF(IW .EQ. 0) THEN
0470 V(J,X)=UT
0471 ELSE
0472 V(J,X)=UT*(1.-R(J))**0.1
0473 ENDIF
0474 30 U(J,X)=0.
0475 ENDIF
0476 ENDIF
0477 DO 40 J=1,NG
0478 END
ALP(J,2)=1.-ALP(J,1)
P(J)=0.DO
W(J,5)=R(J)*ALP(J,1)
DO 40 K=1,2
W(J,K)=ALP(J,K)*W(J,K)*R(J)
40 W(J,K+2)=ALP(J,K)*W(J,K)*R(J)
C PRINTOUT PARAMETERS
WRITE(LUS,5) 'INITIAL CONDITION FILE:',NAMR
WRITE(LUS,5) 'DIMENSION UNITS ARE IN MKS**'
WRITE(LUS,9) 'DENSITY SCALE(kg/m3)',DS
WRITE(LUS,9) 'LENGTH SCALE=RTANK,(m)',LS
WRITE(LUS,9) 'VELOCITY SCALE(m/s)',VS
WRITE(LUS,9) 'TIME SCALE(s)',TS
WRITE(LUS,9) 'PRESSURE SCALE(Pa)',PS
WRITE(LUS,9) 'Reynolds number',Re',Re'
WRITE(LUS,9) 'Jet size, RJ1,RJ2',RJ1,RJ2
WRITE(LUS,9) 'Injection time',TI1,TI2',TI1,TI2
WRITE(LUS,9) 'Phase limits',ALMT(1),1.-ALMT(2)
WRITE(LUS,9) 'Other constants',IU,IUTX,NA,ND,DAMP,UPEAK,RPEAK',
WRITE(LUS,9) 'OMEGA',DI/2,2,1
WRITE(LUS,9) 'INJ',F1,F2,'INJ',IU,IUTX,NA,ND,DAMP
WRITE(LUS,9) 'VEPE',RPEAK,OMEGA,DEHN2,2,1
WRITE(LUS,9) 'DEHN2,2',1
WRITE(LUS,9) 'NG,RMIN'
RETURN
RETURN
399 WRITE(LUS,5) 'OPEN FILE FAILED ON INPUT FILE:',NAMR
WRITE(LUS,7) 'IDSTATE=',IDS
STOP 111
END

REAL*8 FUNCTION DELA(R,DR,RI,RI2).(044423.1537)
REAL*8 R,DR,RI,RI2, R1,R2
R1=R-0.5*DR
R2=R1+DR
DELA=0.DO
IF(R1.LE. RI) RETURN
IF(R1.LT. RI) R1=RI
IF(R1.GT. RI2) R2=RI2
DELA=(R2-R1)/DR
RETURN
0539         END
0540
0541 *********** ****************************
0542     SUBROUTINE DERIV(Y,DY,DX,N2),(06425.1537)
0543 C     GET 1ST DERIVATIVE, USING CENTERED DIFFERENCE
0544     REAL#8 DX,Y(1),DY(1),C
0545     EMA Y,DY
0546
0547 C=5.D-1/DX
0548 DO 10 J=2,N2-1
0549     10    DY(J)=C*(Y(J+1)-Y(J-1))
0550     #V CALL DWSUB(Y(3),1,Y,1,DY(2),1,N2-2)
0551
0552     DY(1)=(Y(2)-Y(1))/DX  ! BASED ON 3-END PTS
0553     DY(N2)=(Y(N2)-Y(N2-1))/DX
0554     DY(1)=2.*DY(1)-DY(2)
0555     DY(N2)=2.*DY(N2)-DY(N2-1)
0556
0557     #V CALL DWSMY(5.D-1/DX,DY,1,DY,1,N2)
0558     RETURN
0559     END
0560
0561 *********** ****************************
0562     SUBROUTINE DDCOF(F,ALP,N2),(06425.1537)
0563 C     CALCULATE THE DRAG, ADDED MASS AND GENERALIZED COEFF.
0564
0565     INTEGER J,MM,N2
0566     PARAMETER (MM=101)
0567     REAL#8 ALP(MM,2)
0568     EMA ALP
0569
0570     REAL#8 C,CPA,CD, RHO,MUEF,V18,NA,ND
0571     COMMON
0572     /DDCF/ C(MM,2),CPA(MM,2),CD(MM,2)
0573     /DRAG/ RHO(4),MUEF(2),V18(2),NA,ND
0574
0575     DO 20 J=1,12
0576     A1=ALP(J,1)
0577     A2=ALP(J,2)
0578     A12=A1*A2
0579     20
0580
0581     C TO GET DRAG COEFF. AD
0582     CALL SIZEB(D1,D2,A1)
0583     AD=V18(2)*A1/((A2*D1*D1))  ! =AD1 IF A2).78
0584     0585     IF(A2 .LT. .78) THEN
0586     AD2=V18(1)*A2/((1.-A2/.8)*D2)**2
0587     IF(A2 .LT. AD) THEN
0588     WT1=A2*ND
0589     WT2=A1*ND
0590     AD=(AD+WT1+AD2+WT2)/(WT1+WT2)
0591     ENDIF
0592     ENDIF
0593     0594     C ADDED MASS COEFF. AA
0599 4A1=12#RHO(2)/(A1+A2/(.5+1.5*A1))
0600 4A2=12#RHO(1)/(A1/(.5+1.5*A2)+A2)
0601 4WT1=A2##NA
0602 4WT2=A1##NA
0603 4AA=(AA+WT1+AA+WT2)/(WT1+WT2)
0604 4
0605 4C
0606 C THE GENERALIZED COEFF. CPA, C, AND CD
0607 4DB2=12#RHO(3)+AA*(RHO(1)+A1#RHO(2)-A2)
0608 4CPA(J,1)=A1*(A12#RHO(2)+AA)/DB2
0609 4CPA(J,2)=A2*(A12#RHO(1)+AA)/DB2
0610 4C(J,1,2)=A6/DB2
0611 4C(J,2,1)=C(J,1,2)
0612 4C(J,1,1)=A2#RHO(2)/DB2+C(J,1,2)
0613 4C(J,2,2)=A1#RHO(1)/DB2+C(J,2,1)
0614 4CD(J,1)=-A2#RHO(2)*AD/DB2
0615 4CD(J,2)=A1#RHO(1)*AD/DB2
0616 4
0617 50 CONTINUE
0618 4
0619 4RETURN
0620 4END
0621 4
0622 4************************************************************
0623 4*EMA /DATA/, /FANDS/, /TAU/, /COEFF/
0624 4SUBROUTINE DWPDF(DW,RDP, DR,DT,RR,N2),(060425.1537)
0625 C
0626 C TO GET DW OF THE PDE
0627 C
0628 C INTEGER J,JP1,K,KP2,MM,N2
0629 C PARAMETER (MM=101)
0630 C 
0631 C Notes: COEFF. C =ALP WHEN THIS IS CALLED
0632 C REAL*8 C,CPA,CD, U, V, ALP, P, R, F, S
0633 C
0634 C 8 TRR,TRA,TAA,RTTR,RTAA
0635 C 4 RHO,HUEF,V18,NA,ND
0636 C X ALP1,ALP2,CP1,CP2,DTDR,DW1,DW1T,G1,G2,HDT,UT,WJ1,WJ3,WJ4
0637 C
0638 C COMMON
0639 C
0640 C 1 /COEFF/ C(MM,2,2),CPA(MM,2),CD(MM,2)
0641 C 3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0642 C 4 /DRAG1/ RHO(4),HUEF(2),V18(2),NA,ND
0643 C 6 /FANDS/ F(MM,5),S(MM,5)
0644 C 8 /TAU/ TRA(MM,2),TAA(MM,2),RTTR(MM,2),RTAA(MM,2)
0645 C
0646 C DTDR=DT/DR
0647 C HDT=0.5*DT
0648 C
0649 C DO 10 J=1,N2+1               CHANGE C TO ALP=C
0650 C C(J,1,1)=ALP(J,1)*C(J,1,1)
0651 C C(J,2,1)=ALP(J,2)*C(J,2,1)
0652 C C(J,1,2)=ALP(J,1)*C(J,1,2)
0653 C C(J,2,2)=ALP(J,2)*C(J,2,2)
0654 C
0655 C DO 20 J=1,N2
0656 JP1=J+1
0657 DW(J,5)=DTDR*(-F(JP1,5)+F(J,5))*HDT*(S(JP1,5)+S(J,5))
0658 C DO 25 K=1,2

26
G1 = 5*(C(JP1, 1) + C(J, K))
G2 = 5*(C(JP1, 2) + C(J, K))

DU(J, K) = DTDR(-F(JP1, K) + F(J, K) + G1*(TRR(JP1, 1) - TRR(J, 1))
+ G2*(TRR(JP1, 2) - TRR(J, 2))
+ HDT*(S(JP1, K) - S(J, K))

KP2 = K + 2

1
+ G1*(TRA(JP1, 1) - TRA(J, 1))
+ G2*(TRA(JP1, 2) - TRA(J, 2))
+ HDT*(S(JP1, KP2) + S(J, KP2))

C
DP FOR PRESSURE CORRECTION ON
CP1 = 0.5*(CPA(J, 1) + CPA(JP1, 1))
CP2 = 0.5*(CPA(J, 2) + CPA(JP1, 2))

IF(-DW(J, 1) > 0.DP(J)) DU(J, 1) = -DW(J, 1) ! DP = 0

RDP(J) = (DW(J, 1) + DW(J, 2))/(CP1 + CP2)

DW(J, 1) + DW(J, 2) - CP1*RDP(J)

RETURN
END

******************************************************************************
**EMA */COEFF/ */DATA/ */WWW/

** C SUBROUTINE FNDDT(DT, DR, VDR, NG), (860425.1537)
** DETERMINE THE TIME-STEP SIZE

** INTEGER I, J, LUP, MM, NG
** PARAMETER (MM = 101)
** REAL*8 DT, DR, VDR(2)

** REAL*8 C, CPA, CD, RHO, MUEF, V18, NA, ND
** U, W, WP, WN, DW, RDP, RH, U, V, ALP, P, R
** X, DUM1, DUM2

** COMMON
** 2 */COEFF/ C(MM, 2), CPA(MM, 2), CD(MM, 2)
** 3 */DATA/ U(MM, 2), V(MM, 2), ALP(MM, 2), P(MM), R(MM)
** 4 */DRAG/ RHO(4), MUEF(2), V18(2), NA, ND
** 9 */WWW/ W(MM, 5), WP(MM, 2), WN(MM, 5), DW(MM, 5), RDP(MM), R(MM)
** DATA LUP/ 1/

** DUM1 = 0.
** DO 10 J = 1, NG
**   DUM2 = CD(J, 1)
** 1
** 1 + DABS(U(J, 1)) / DR : CONVECTIVE
** 2 + VDR(1) + ALP(J, 1) * C(J, 1, 1)
** 3 + VDR(2) + ALP(J, 2) * C(J, 1, 2)
** IF(DUM1 .LT. DUM2) DUM1 = DUM2
** DUM2 = CD(J, 2) + DABS(U(J, 2)) / DR
** 1 + VDR(1) + ALP(J, 1) * C(J, 2, 1)
** 2 + VDR(2) + ALP(J, 2) * C(J, 2, 2)
** IF(DUM1 .LT. DUM2) DUM1 = DUM2
** CONTINUE

** C FIND THE MAXIMUM OF DUM
0719 *U    CALL DWMAX(I,DW,1,NG)            ! VECTOR OPERATION
0720 *U    DWIM=DABS(DW(I,1))
0721 *U    CALL DWMAX(J,DW(1,2),1,NG)
0722 *U    DW2=DABS(DW(J,2))
0723 DT=1.0/DU
0724 RETURN
0725 END
0726
0728 ******************************************************************************
0729 *EMA /DATA/,/FANDS/,/TAU/,/COEFF/
0730 SUBROUTINE FSOFU(W,BA,RH,N2),((860425.1537)
0731 C       CALCULATE THE CONVECTIVE-F AND SOURCE-S TERMS
0732 C
0733 C       INTEGER J,K,KP2,L,MM,N2
0734 C       PARAMETER (MM=101)
0735 C       REAL*8 U,V,ALP,P,R,F,S,RH0,MUEF,V1B,NA,ND
0736 C       2 X ,TRA,TRA,TAJ,TRR,TRA, C,CPA,CD
0737 C
0738 C       REAL*8 W(MM,5),BA(MM,2),RR(1)
0739 EMA W,BA,RR
0740 C
0741 C       COMMON
0742 C       1 /COEFF/ C(MM,2,2),CPA(MM,2),CD(MM,2)
0743 C       3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0744 C       4 /DRAG/ RH0(4),MUEF(2),V1B(2),NA,ND
0745 C       6 /FANDS/ F(MM,5),S(MM,5)
0746 C       8 /TAU/ TRR(MM),TRA(MM),TAJ(MM,2),TRR(MM,2),TRA(MM,2)
0747 C
0748 #U    CALL DUMPY(W,1,1,F,1,2*MM)
0749 #U    CALL DUMPY(W(1,3),1,1,F(1,3),1,2*MM)
0750 #U    CALL DHODV(W,1,F(1,5),1,MM)
0751 #U
0752 DO 20 J=1,N2
0753 F(J,5)=W(J,1)
0754 S(J,5)=0.
0755 RDU=RR(J)*(U(J,1)-U(J,2))
0756 RDV=RR(J)*(V(J,1)-V(J,2))
0757 DO 20 K=1,2
0758 KP2=K+2
0759 F(J,K)=W(J,K)#U(J,K)
0760 F(J,KP2)=W(J,KP2)#U(J,K)
0761 S(J,K)=ALP(J,K)*(U(J,K)*#2+CD(J,K)#RDU-C(J,K,1)#TAJ(J,1)
0762 1 -C(J,(J,2)#TAJ(J,2))
0763 S(J,KP2)=ALP(J,K)*(U(J,K)+CD(J,K)#RDV#C(J,K,1)#(BA(J,1)
0764 1 +TRA(J,1))#C(J,K,2)#(BA(J,2)+TRA(J,2)))
0765 RETURN
0766 20
0767 RETURN
0768 END
0769
0770******************************************************************************
0771 SUBROUTINE JET(BA,RH0,BA,V,VI,NG)
0772 C
0773 C       INJECTION MOMENTUM SOURCE
PARAMETER (MM=181)
REAL= B, BA(MM,2), RHO(1), BK(1), V(MM,2), VJ
C
DO 10 J=1,15
IF( BA(J) .GT. 0.) THEN
  Q=BA(J)*VJ
  BA(J,1)=Q*(VJ-V(J,1))*RHO(1)
  BA(J,2)=Q*(VJ-V(J,2))*RHO(2)
  ELSE
  BA(J,1)=0.00
  BA(J,2)=0.00
  ENDIF
10 CONTINUE
RETURN
END

******************************************************************************

SUBROUTINE SIZES(D1,D2,ALP1),(860425.1537)
C TO DETERMINE THE PARTICLE DIAMETERS

REAL= D1,D2,ALP1, DO,GAMMA
COMMON /DSIZE/ DO(2), GAMMA(2)

D1=DO(1)*ALP1+GAMMA(1)
D2=DO(2)*1.0D0-ALP1+GAMMA(2)

RETURN
END

******************************************************************************

REAL= MU(2), DR, RR(1)
C
REAL= U, V, ALP, P, R

PARAMETER (MM=101)

REAL= U, V, ALP, P, R

C
3 /DATA/ (MM,2), J(MM,2), ALP(MM,2), P(MM), R(MM)

8 /TAU/ TRR(MM,2), TRA(MM,2), TAA(MM,2), RTRR(MM,2), RTRA(MM,2)

DO 50 K=1,2

CALL DERIV(U(1,K), TRR(1,K), DR, N2)

CALL DUMPY(ALP(1,K), 1, TRR(1,K), 1, TRR(1,K), 1, N2)

CALL DUMPY(U(1,K), 1, RR, 1, TAA(1,K), 1, N2)

DO 10 J=1,N2

10 TAA(J,K)=U(J,K)/RR(J)

CALL DERIV(TAA(1,K), TRA(1,K), DR, N2)

THU=U(K)

DO 20 J=1,N2

THU=THU+ALP(J,K)

TRR(J,K)=THU*TRR(J,K)

DO 30 K=1,2

CALL DERIV(U(1,K), TRR(1,K), DR, N2)

CALL DUMPY(ALP(1,K), 1, TRR(1,K), 1, TRR(1,K), 1, N2)

CALL DUMPY(U(1,K), 1, RR, 1, TAA(1,K), 1, N2)

DO 10 J=1,N2

10 TAA(J,K)=U(J,K)/RR(J)

CALL DERIV(TAA(1,K), TRA(1,K), DR, N2)

THU=U(K)

DO 20 J=1,N2

THU=THU+ALP(J,K)

TRR(J,K)=THU*TRR(J,K)

DO 30 K=1,2

CALL DERIV(U(1,K), TRR(1,K), DR, N2)

CALL DUMPY(ALP(1,K), 1, TRR(1,K), 1, TRR(1,K), 1, N2)

CALL DUMPY(U(1,K), 1, RR, 1, TAA(1,K), 1, N2)

DO 10 J=1,N2

10 TAA(J,K)=U(J,K)/RR(J)

CALL DERIV(TAA(1,K), TRA(1,K), DR, N2)

THU=U(K)

DO 20 J=1,N2

THU=THU+ALP(J,K)

TRR(J,K)=THU*TRR(J,K)
0839      TRA(J,K)=MU(K)*ALP(J,K)*RR(J)*TRA(J,K)
0840      TAA(J,K)=TAMU(J,K)/RR(J)
0841      RTRR(J,K)=RR(J)*TRR(J,K)
0842      20      RTRA(J,K)=RR(J)*RTA(J,K)
0844  1V      CALL DWMPY(RR,1,TRA(1,K),1,TRA(1,K),1,N2)  ; R
0845  1V      CALL DWMPY(ALP(1,K),1,TRA(1,K),1,TRA(1,K),1,N2) ; ALP = TRA
0847  1V      CALL DWMPY(MU(K),TRA(1,K),1,TRA(1,K),1,N2)  ; #MU = TRA
0848  1V      CALL DWDIV(U(1,K),1,RR,1,TAA(1,K),1,N2)  ; U/P
0849  1V      CALL DWMPY(ALP(1,K),1,TAA(1,K),1,TAA(1,K),1,N2) ; #ALP = TAA
0850  1V      CALL DWMPY(2,MU(K),TAA(1,K),1,TAA(1,K),1,N2)  ; #2*MU = TAA
0852  1V      CALL DWMPY(RR,1,TRR(1,K),1,TRR(1,K),1,N2)  ; R = TRR
0853  1V      CALL DWMPY(RR,1,TRA(1,K),1,TRA(1,K),1,N2)  ; R = TRA
0854  50      CONTINUE
0855      RETURN
0856      END
0857
0858  ****************************
0859  1V      SUBROUTINE UDFM(W,RR,N2), (O6425.1537)
0860  1C      CONVERTS W TO THE INDEPENDENT VARIABLES (U,V,ALP)
0862
0863      PARAMETER (MM=101)
0864      REAL*8 W(MM,5),RR(MM)
0865      EMHM W,RR
0866      REAL*8 ALMT, U,V,ALP,P,R
0868      X       ,WJ6
0869  3       COMMON
0870  3       X /ALPLMT/ ALMT(2)
0871  3       J /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0872
0873  1V      CALL DWDIV(W(1,5),1,RR,1,ALP,1,N2)
0874
0875  1C      CHECK VOLUME FRACTION & FLOW DIRECTIONS
0876  50      DO 50 J=1,N2
0877  50      ALP(J,1)=W(J,J)/RR(J)
0878      IF(ALP(J,1) .LE. ALMT(1) .OR. ALP(J,1) .GE. ALMT(2)) THEN
0879      IF(ALP(J,1) .LE. ALMT(1)) THEN
0880      ALP(J,1)=ALMT(1)
0881      W(J,3)=W(J,4)*ALMT(1)/(1.D0-ALMT(1))
0882      ELSE
0883      ALP(J,1)=ALMT(2)
0884      W(J,4)=W(J,3)*((1.D0-ALMT(2))/ALMT(2))
0885      ENDIF
0886      W(J,1)=0.D0
0887      W(J,2)=0.D0
0888      W(J,3)=ALP(J,1)*RR(J)
0889      ENDIF
0890      ENDIF
0892      ALP(J,2)=1.D0-ALP(J,1)
0893      IF( W(J,2) .LT. 0. ) THEN  ; PHASE-2 DOES NOT MOVE IN
0894      W(J,2)=0.
0895      W(J,1)=0.
0896      ENDIF
0897
0898
0899  U(J,1)=W(J,1)/W(J,5)
0900  V(J,1)=W(J,3)/W(J,5)
0901  WJ6=RR(J)*ALP(J,2)
0902  U(J,2)=W(J,2)/WJ6
0903  V(J,2)=W(J,4)/WJ6
0904  50 CONTINUE
0905
0906  *U CALL DWDIV(W(1,1),W(1,5),1,1,U,1,N2)
0907  *U CALL DWDIV(W(1,3),1,W(1,5),1,1,V,1,N2)
0908  *U CALL DWMPY(ALP(1,2),1,RR,1,W(1,6),1,1,N2)
0909  *U CALL DWDIV(W(1,2),1,W(1,6),1,U(1,2),1,1,N2)
0910  *U CALL DWDIV(W(1,4),1,W(1,5),1,U(1,2),1,1,N2)
0911
0912  RETURN
0913  END
0914
A Sample Input

GLVM

Enter lu for saving data. D.F. = 6
90

Enter FILE NAME for saving data.
TS153::LB

Enter NOTES(<73 CHAR.) for the job
SAMPLE RUN OF TEST :53

enter Reynolds no., RE. D.F. = 1.0000E+05

Enter DENSITY and VISCOSITY ratios.
D.F. = 1.2930E-03 1.1295E-02

Enter BASE DIAMETERS: D01,D02
D.F. = 1.0000E-02 1.0000E-02

Enter SIZE EXPONENT: GAMMA1,GAMMA2
D.F. = 2.0000E-01 2.0000E-01

Enter weighting exponent: NA,ND. D.F. = 4.0000E+00 4.0000E+00

Enter GAS VOLUME FRACTION limits:ALMT1,ALMT2.
D.F. = 1.0000E-04 9.9999E-01

Enter eddy viscosity factor. D.F. = 1.443E+02 1.0000E+03

Enter wall condition. 1=nonslip, 0=slip. D.F. =

Enter numerical damping factor. D.F. = 0.0000E+00

Enter JET SIZE defined by RJ1,RJ2. D.F. = 8.5000E-01 9.5000E-01

Enter INJECTION SPEED AND TIME RANGE, VJ,T1,T2
D.F. = 1.0000E+01 0.0000E+00 1.0000E+01

Enter data FILE NAME for initial cond. if any:
D.F. = Simple vortex

Enter initial value of alpl. D.F. = 2.5000E-01

Enter type of vortex: 0=At rest, 1=pure rotation
2=H.O., 3=GIT. D.F. = 0

Enter INITIAL and FINAL TIMES. D.F. = 0.0000E+00 5.0000E+00

Enter TIME STEP for output. D.F. = 2.0000E-01
Exhibit B

A Sample Output

10153 T=0.0004 IS ON CR LB USING 0024 RLKS R=0.0000

0001 1 SAMPLE RUN OF TEST 153
0002 INITIAL CONDITION FILE:
0003 **DIMENSION UNITS ARE IN MKS**
0004 DENSITY SCALE(kg/m^3) 1.0000E+03
0005 LENGTH SCALE=RTANK,(m) 1.0000E+00
0006 VELOCITY SCALE(m/s) 1.5140E-01
0007 TIME SCALE(s) 6.6050E+00
0008 PRESSURE SCALE(Pa) 2.2922E+01
0009
0010 Reynolds number, Re 1.0000E+05
0011 Jet size, RJ1,RJ2 8.5000E-01 9.5000E-01
0012 Tangential jet, QJ1,VJ 1.0000E-01 1.0000E+00
0013 Injection time,TJ1,TJ2 0.0000E+00 1.0000E+00
0014
0015 PHASE-1 PHASE-2
0016 Density 1.2930E-03 1.0000E+00
0017 Viscosity 1.1295E-07 1.0000E-05
0018 Eddy viscosity factor 1.0000E+03 1.0000E+03
0019 Base dia. 1.0000E-02 1.0000E-02
0020 Size exp. 2.0000E-01 2.0000E-01
0021 Phase limits 1.0000E-04 1.0000E-04
0022 OTHER CONSTANTS: Iw,IVTX,AN,ND,DAMP,VPEAK,RPEAK,OMEGA,D1/D2,ML/ML
0023 1 0.4 4.0 0.00
0024 0.0000E+00 1.0000E-01 0.0000E+00 1.2930E-03 1.1295E-02
0025
0026 1 NP= 1 T=0.000E+00 NT= 2 DT=1.0000E-06
0027
0028 J ALP1 U1 U2 V1 V2 P
0029 1 2.500 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0030 2 2.500 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0031 ------ OUTPUT IN THE BETWEEN OMITTED ------
0032 97 2.500 -8.685E-10 2.895E-10 3.513E-04 3.474E-04 3.969E-05
0033 98 2.500 -1.775E-10 5.918E-11 1.268E-04 1.250E-04 3.972E-05
0035 100 2.500 -5.225E-12 1.742E-12 1.533E-05 1.531E-05 3.973E-05
0036 101 2.500 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0037 ------ THE REST OF THE OUTPUT IS OMITTED ------
Figure 1. Two Step Difference Scheme (Backward Predictor - Forward Corrector Version)
Figure 2. Velocity Vector Distributions

The annular region between two dashed circles is the region of injection.

LIQUID PHASE

GAS PHASE
Figure 3. Gas Volume Fraction Distributions