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

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.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize any 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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_a$, $A_{ak}$</td>
<td>Added mass coefficients</td>
</tr>
<tr>
<td>$A_d$, $A_{dk}$</td>
<td>Drag coefficients</td>
</tr>
<tr>
<td>$\bar{\rho}_k$</td>
<td>Body force density</td>
</tr>
<tr>
<td>$C_{1j}$, $C_{pk}$, $C_{dk}$</td>
<td>Generalized coefficients</td>
</tr>
<tr>
<td>$d_1$</td>
<td>Bubble diameter</td>
</tr>
<tr>
<td>$d_2$</td>
<td>Liquid (droplet) diameter</td>
</tr>
<tr>
<td>$\bar{\mu}_k$</td>
<td>Effective interfacial force density</td>
</tr>
<tr>
<td>$n_a$</td>
<td>Exponent used for $w_{ak}$</td>
</tr>
<tr>
<td>$n_d$</td>
<td>Exponent used for $w_{dk}$</td>
</tr>
<tr>
<td>$p$</td>
<td>Pressure</td>
</tr>
<tr>
<td>$R$</td>
<td>Tank radius</td>
</tr>
<tr>
<td>$Re$</td>
<td>$\rho_2 V_3 R / \mu_2$, Reynolds number</td>
</tr>
<tr>
<td>$R_{j1}$, $R_{j2}$</td>
<td>Jet opening, $R_{j1} &lt; r &lt; R_{j2}$</td>
</tr>
<tr>
<td>$R_1$</td>
<td>Minimum radius considered in the numerical analysis</td>
</tr>
<tr>
<td>$r$</td>
<td>Radial coordinate</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$V_J$</td>
<td>Averaged jet velocity</td>
</tr>
<tr>
<td>$V_{r1}$</td>
<td>Gas radial velocity</td>
</tr>
<tr>
<td>$V_{r2}$</td>
<td>Liquid radial velocity</td>
</tr>
<tr>
<td>$V_s$</td>
<td>Velocity scale</td>
</tr>
<tr>
<td>$V_{\theta 1}$</td>
<td>Gas tangential velocity</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$V_{02}$</td>
<td>Liquid tangential velocity</td>
</tr>
<tr>
<td>$V_1$</td>
<td>Gas velocity</td>
</tr>
<tr>
<td>$V_2$</td>
<td>Liquid velocity</td>
</tr>
<tr>
<td>$w_{ak}$</td>
<td>Weighting function for added mass coefficients</td>
</tr>
<tr>
<td>$w_{dk}$</td>
<td>Weighting function for drag coefficients</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>Gas volume fraction</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>$(1-\alpha_1)$, liquid volume fraction</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Exponent for diameter variation</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Circumferential coordinate</td>
</tr>
<tr>
<td>$\mu_1$</td>
<td>Gas dynamic viscosity</td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>Liquid dynamic viscosity</td>
</tr>
<tr>
<td>$\mu_k$</td>
<td>$u_k + u_k^t$ total effective viscosity</td>
</tr>
<tr>
<td>$u_k^t$</td>
<td>Turbulence or eddy viscosity</td>
</tr>
<tr>
<td>$\nu_k$</td>
<td>$\mu_k/\rho_k$, kinematic viscosity</td>
</tr>
<tr>
<td>$\rho_1$</td>
<td>Gas density</td>
</tr>
<tr>
<td>$\rho_2$</td>
<td>Liquid density</td>
</tr>
<tr>
<td>$\rho_k$</td>
<td>Averaged density of k-phase</td>
</tr>
<tr>
<td>$&lt;\rho^2&gt;$</td>
<td>$\alpha_1 \alpha_2 \rho_1 \rho_2 + A_a (\alpha_1 \rho_1 + \alpha_2 \rho_2)$</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Angular velocity</td>
</tr>
</tbody>
</table>
List of Figures

<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</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>(Backward Predictor-Forward Corrector Version)</td>
<td></td>
</tr>
<tr>
<td>Figure 2.</td>
<td>Velocity Vector Distributions</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>a) Gas phase, b) Liquid phase</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The annular region between two dashed circles is the region of injection</td>
<td></td>
</tr>
<tr>
<td>Figure 3.</td>
<td>Gas Volume Fraction Distributions</td>
<td>36</td>
</tr>
</tbody>
</table>
I. Introduction

Mechanical systems have been devised for producing artificial gravity fields to spin-up liquids in containers. These involve rotating mechanisms, are cumbersome and, more importantly, have moving parts that can wear out. Here, liquid rotation created by fluid injection is considered. The detailed 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 (Mc Cormack'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:

$$q_1 + q_2 = 1$$

$$\frac{\partial q_k}{\partial t} + q_k \frac{\partial V_k}{\partial r} = 0$$
\[
\frac{\partial r_k V_{rk}}{\partial t} + \frac{\partial r_k V_{rk}^2}{\partial r} - a_k V_{rk}^2 = -a_k C_{pk} r \frac{\partial p}{\partial r}
\]

\[
+ a_k \sum_{l=1}^{2} C_{kl} \left( \frac{\partial r_{lr}}{\partial r} - a \frac{\partial r_{lr}}{\partial r} + a \frac{\partial B_{rl}}{\partial r} \right)
\]

\[
+ a_k C_{dk} r (V_{r1} - V_{r2})
\]

\[
\frac{\partial a_{rk} V_{rk}}{\partial t} + \frac{\partial a_{rk} V_{rk} V_{rk}}{\partial r} + a_k V_{rk} V_{rk} = a_k \sum_{l=1}^{2} C_{kl} \left( \frac{\partial a_{lr}}{\partial r} + a \frac{\partial a_{rl}}{\partial r} + a \frac{\partial B_{gl}}{\partial r} \right)
\]

\[
+ a_k C_{dk} r (V_{gl} - V_{g2})
\]

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

\[
C_{p1} = (a_1a_2\rho_2 + A_a) / \langle \rho^2 \rangle
\]

\[
C_{p2} = (a_1a_2\rho_1 + A_a) / \langle \rho^2 \rangle
\]

\[
C_{11} = (a_2\rho_2 + A_a) / \langle \rho^2 \rangle
\]

\[
C_{12} = C_{21} = A_a / \langle \rho^2 \rangle
\]

\[
C_{22} = (a_1\rho_1 + A_a) / \langle \rho^2 \rangle
\]
\[ C_{d1} = -a_2 \rho_2 A_d / \langle \rho^2 \rangle \]
\[ C_{d2} = a_1 \rho_1 A_d / \langle \rho^2 \rangle \]

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

The effective stresses are modeled as
\[ \tau_{rrk} = 2 \mu_k^e \partial \nu_{rk} / \partial r \]
\[ \tau_{\theta rk} = \tau_{\theta k} = \mu_k^e \partial (V_{8k} / r) / \partial r \]
\[ \tau_{\theta \theta k} = 2 \mu_k^e V_{rk} / r \]

with
\[ \mu_k^e = \mu_k + \mu_k^t \]
and the interfacial forces are modeled in the form of
\[ \overline{M}_1 = A_d (\overline{V}_2 - \overline{V}_1) + A_a \frac{d}{dt} (\overline{V}_2 - \overline{V}_1). \]

\( \overline{M}_1 \) 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 $\alpha_k \rho_k \mathbf{B}_k$ will be replaced by

$$\frac{\alpha_k \rho_k V_j}{2\pi} (V_j \mathbf{n} - \mathbf{v})$$

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) \cite{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

\[
\begin{align*}
W^1 &= W^0 + \Delta t W_t^0 + \frac{(\Delta t)^2}{2} W_{tt}^0 \\
&= W^0 + \frac{\Delta t}{2} W_t^0 + \frac{\Delta t}{2} (W_t^O + \Delta t W_{tt}^G) \\
&= \frac{1}{2} (W^O + \Delta t W_t^O) + \frac{1}{2} (W^O + \Delta t W_t^P) \\
&= \frac{1}{2} (W^P + W^C)
\end{align*}
\]

where

\[
W^O = W^0 + \Delta t W_t^0 \text{ is the predicted value,}
\]

and

\[
W^C = W^0 + \Delta t W_t^P \text{ is the corrected value.}
\]

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, \( W_t^O \) 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

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

\[ = \frac{1}{2} (w_{i-1}^o + w_i^o) + \frac{\Delta t}{\Delta r} \left[ (F_1^o - F_{i-1}^o) + \frac{(g_1^o + g_{i-1}^o)}{2} (G_1^o - G_{i-1}^o) \right] \]

\[ + \frac{\Delta t}{2} (S_1^o + S_{i-1}^o) + \Delta t \hat{p}_{i-1/2}^p \]

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

\[ \hat{w}_i^c = w_i^o + \Delta t \, w_t^p \]

\[ = w_i^o + \frac{\Delta t}{\Delta r} \left[ (F_{i+1/2}^p - F_{i-1/2}^p) + \frac{(g_{i+1/2}^p + g_{i-1/2}^p)}{2} (G_{i+1/2}^p - G_{i-1/2}^p) \right] \]

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

Here \( \hat{p}_{i-1/2}^p \) 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

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

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 \( \hat{W}_1 \) is the value obtained based on the two-step scheme, and \( \hat{W}^D \) 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 $a_1$, $ALMT(1) < a_1 < ALMT(2)$.

**DAMP** Numerical damping factor. Normally set to 0.

**DEN1D2** Density ratio, $\rho_1/\rho_2$

**DO** Base diameter, i.e., $d_k = d_{ok}a_k^Y$

**DS** Density scale = $\rho_2$

**EVF** $u^t/\mu$ effective eddy viscosity factor.

**GAMMA** Diameter exponent $Y$

**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, $M1 > N1$. MM appears in many subroutines.

**$\mu_k$** Dynamic viscosity for phase $k$.

**MLEF** $(1 + EVF) * MU$. 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 ω, 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, \( \rho_2 V_a R/u_2 \)
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 ≤ \( \Delta A \) ≤ 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:

\[
A_a = A_{a1}w_1 + A_{a2}w_2 \\
A_d = A_{d1}w_1 + A_{d2}w_2 \\
A_{a1} = a_1a_2\rho_2/(a_1 + 2a_2/(1 + 3a_1)) \\
A_{a2} = a_1a_2\rho_1/(a_2 + 2a_1/(1 + 3a_2)) \\
A_{d1} = 18 \mu_2a_1/d_1^2a_2 \\
A_{d2} = 18 \mu_1a_2/(1 - a_2/0.8^2d_2^2) \\
w_{al} = a_2\rho_a/(a_1\rho_a + a_2\rho_a) \\
w_{a2} = 1 - w_{al} \\
w_{d1} = a_2\rho_d/(a_1\rho_d + a_2\rho_d) \\
w_{d2} = 1 - w_{d1}
\]

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) FNDTT, 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.

$$d_k = d_{ok} \alpha_k^\gamma$$

Different models for size distributions could be easily adopted here.

11) TAUOFW

To determine the stress tensor $\tau$ 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

GLEVM T=08054 IS ON CR T4 USING 00126 BLKS R=0000

0001 FTM77
0002 $E=MA /DATA/.,/WWW/.,/COEFF/.,/SOURCE/.,/FANDS/.,/TAU/
0003 $FILES 1.2
0004 PROGRAM GLEVM(99), (B6O425.1537)
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 HP'S FORTRAN 77 (ANSI 77+MIL-STD-1753)
0011 C
0012 C WHEN WHO WHAT
0013 C A502XX TTY ZI 0-G FUEL TRANSFER, START-UP STAGE.
0014 C LAX WENDROFF 2-STEP SCHEME (FULL STEP PREDICTION+CORRECTION)
0015 C WITH NUMERICAL DAMPING FACTOR (NORMAL SET TO ZERO)
0016 C CONVECTION 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 B50715 TTY GENERALIZED EQUATIONS AND COEFF. C1
0021 C B51018 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, DDCOEF, DNPDE, FMDT, FSDFW,
0026 C INIT, JET, SIZES, TAUDFW and UDFW
0027 C *** MOST OF THE LIST OF NOTATIONS ARE GIVEN IN SUBROUTINE..INIT
0028 C
0029 C CHARACTER NMNR=16, NOTES=72
0030 INTEGER I, IOS, J, JTIME(3), K, M, MPRT, NT
0031 Z
0032 , IW, LUP, LUS, NG, NGM1, NGM2
0033 M PARAMETER (MM=101)
0034 REAL*8 BA, MM, 2, DDT, DT, DTMX, DTMN, DTRT, PZERO
0035 X
0036 , DW, T, DMAX, TRT, UJT, VDR(2)
0037 Y
0038 , RJ1, RJ2, TJ1, TJ2, QJ, VJ
0039 1
0040 , ALNT, U, V, ALP, P, R, W, WP, WN, DW, RDP, RH, FS
0041 4
0042 , BR, BRH, RHO, MUEF, VIB, NA, ND
0043 5
0044 , DO, GAMMA, DAMP, DR
0045 6
0046 , TRR, TRA, TAA, RTRA, RTRA, C, CPA, CD
0047 0
0048 COMMON
0049 Y
0050 , JETS/ RJ1, RJ2, TJ1, TJ2, QJ, VJ
0051 Z
0052 , /CONTR/ IW, LUP, LUS, NG, NGM1, DAMP, DR
0053 1
0054 /ALPNT, ALMT(2)
0055 2
0056 /COEFF/ C(MM, 2), CPA(MM, 2), CM(MM, 2)
0057 3
0058 /DATA/ U(MM, 2), V(MM, 2), ALP(MM, 2), P(MM), R(MM)
0059 4
0060 /DRAG/ RHO(4), MUEF(4), VIB(2), NA, ND
0061 6
0062 /FANDS/ F(MM, 5), S(MM, 5)
0063 7
0064 /SOURCE/ BR(MM), BRH(MM)
0065 8
0066 /TAU/ TRR(MM, 2), TRA(MM, 2), TAA(MM, 2), RTRA(MM, 2), RTRA(MM, 2)
0067 9
0068 /WWW/ W(MM, 3), WP(MM, 3), DW(MM, 3), RDP(MM), RH(MM)
0069 0
0070 EQUIVALENCE (MN, BA)
0071
0072 C RHOD(1) RHOD(2) (I.E. PHASE-1=GAS, PHASE-2=LIQUID) *****
0073 C DTRT TIME STEP FOR PRINTOUT (AND PLOT)
0074 C
0075 C
0076 C
0077 C
0078 C
LUP=1  ! Lu for printing debug data(1 terminal)
LUS=6  ! Lu for storing data(6 printer)

0062 7  FORMAT(2X,A,3(JPE12.4))
0063 B  FORMAT(2X,A,3(JPE12.4))
0064 WRITE(1,7) 'Enter lu for saving data. D.F.=',LUS
0065 READ(1,7) LUS
0066 C  KEEP JOB TIME FOR FUTURE REFERENCE
0067 C  CALL EXEC(I,J/TIMEP,ITIME(1))
0068 C  IF(LUS .NE. 1 .AND. LUS .NE. 6) THEN
0069 C  ***Define a file name for string output:*************
0070 C  WRITE(1,(2A)) 'Enter FILE NAME for saving data.'
0071 C  READ(1,'(A*)') NAMN
0072 U  LUS=99
0073 C  OPEN(LUS,FILE=NAMN,IOSTA=IOS,STATUS='NEW',ERR=999)
0074 ENDIF
0075 WRITE(1,'(A*)') 'Enter NOTES((73 CHAR.) for the job'
0076 C  READ(1,'(A*)') NOTES
0077 C  WRITE(LUS,'(3H1,A*)') NOTES
0078 C  WRITE(LUS,'(5H4)') ITIME
0079 C  To set-up the initial condition.
0080 C  CALL INIT
0081 C  UMG2=MG-2
0082 UDR(1)=2.*MUEF(1)/DR**2  ! For determining time step
0083 UDR(2)=2.*MUEF(2)/DR**2
0084 T=0.  ! INITIAL TIME
0085 TMAX=5.
0086 DT=0.2
0087 WRITE(1,8) 'Enter INITIAL and FINAL TIMEs. D.F.=',T,TMAX
0088 READ(1,'(A*)') T,TMAX
0089 WRITE(1,8) 'Enter TIME STEP for output. D.F.=',DT
0090 READ(1,'(A*)') DT
0091 DTM=1.0D-6  ! SET MINIMUM TIME STEP
0092 DTMAX=DT
0093 GY9
0094 NT=0
0095 TPRT=1
0096 MPRT=0.0
0097 PZERO=0.0
0098 DT=DTMIN
0099 10 NT=NT+1
0100 CALL UGF(W,R,G)
0101 C  PRINT OUT AT SELECTED TIME
0102 C  IF(T .GE. TPRT .OR. T .GT. TMAX) THEN
0103 IF(NT .GE. 1) THEN
0104 READ(NT,MPRT)+1
0105 TPRT=TPRT+DT
0106 FORMAT(1H1,2(A5,4,A5,1P,F.3))
0107 WRITE(LUS,21) 'NP=',MPRT,'T=',T,'NT=',NT,'DT=',DT
0108 WRITE(LUS,21) 'NP=',MPRT,'T=',T,'NT=',NT,'DT=',DT
0109 WRITE(LUS,21) 'NP=',MPRT,'T=',T,'NT=',NT,'DT=',DT
0110 WRITE(LUS,'(A5,A6,5A11)') 'I',ALP','U1','U2','U1','U2','P'

17
0119 DO 30 J=1,NG
0120 30 WRITE(LUS, ' (13,F6.4,5(1PE1.3))')
0121 1 J,ALP(J,1),U(J,1),U(J,2),V(J,1),V(J,2),P(J)
0122 ENDIF
0123 ENDIF
0124 IF ( T .GT. TMAX) GOTO 9999
0125 C TO SOLVE THE DIFFERENTIAL EQUATIONS
0126 C USING 2 STEP LAX-WENDROFF SCHEME
0127 C MacCormack's method: BACKWARD PREDICTOR, FORWARD CORRECT
0128 C CENTER DIFFERENCED ON TAU
0129 C FIRST: TO GET THE SPECIAL VARIABLES AND THEIR SPATIAL DERIVATIVES
0130 C DETERMINE THE TIME-STEP SIZE
0131 C IF ( NT .GT. 1) THEN
0132 C CALL FNODT(DT,DR,NG)
0133 C IF ( DT .GE. DTMIN/10.0*N'T )
0134 C I=DLOG10(DT/DTMIN)
0135 C DD'T'=DTMIN/10.0*N'T
0136 C DD'T'=DTMIN/10.0*N'T
0137 C IF ( I .GT. DT . GT. DTMAX) DT=DTMAX
0138 ELSE
0139 C WRITE(LUS, ' (5X,A,1PE13.3)')
0140 C 'STOP DUE TO TOO SMALL TIME STEP. DT=', DT
0141 C GOTO 9999
0142 ENDIF
0143 C BACKWARD PREDICTOR
0144 C CALL DWPDE(DW,RDP,DR,DT,RH,NGM1)
0145 C INCREMENT
0146 C DO 40 J=1,NGM1
0147 C DO 40 I=1,5
0148 40 WP(I,J)=0.5*(U(J+1,I)+U(J,I))*(U(J+1,I)+U(J,I))
0149 C BASE+INCREMENT
0150 C PREDICTION DATA COMPLETED, CONTINUE FOR CORRECTION
0151 C CALL UDFW(WP,RA,NGM1)
0152 C CALL DGCOEF(ALP,NGM1)
0153 C CALL TAOFW(MUEF,DR,RH,NGM1)
0154 C CALL JET(BA,RH,BR,H,VJ,T,NG)
0155 C CALL FSDFW(WP,BA,RH,NGM1)
0156 C FORWARD CORRECTION
0157 C CALL DWPDE(DW,P(2),DR,DT,R(2),NGM2)
0158 C P(1)=ZERO
0159 C DO 50 J=1,NGM2
0160
0179  P(I+1)=P(I)+(0.25*(RDP(J)+RDP(J+1))+0.5*P(J+1)/R(J)
0180  DO 50 I=1,5
0181  50  W(N+1,J)=P(N,J)*0.5*(W(J+1,I)+W(J,I))+0.5*(W(J+1,I)+W(J,I))
0182  P(N+1)=P(N+1)
0183  C  2ND STEP (PREDICTION+CORRECTION) COMPLETED
0184  IF(NT.EQ.1) GOTO 10
0185  C  Estimation of initial condition completed, return to the initial condition
0186  C  and start to advance the program in time.
0187  C  **************DATA AT THE NEW TIME STEP COMPLETED**************
0188  C  IMPOSED B.C. #6.4
0189  DNI=-5*(W(2,1)+W(2,1))
0190  ALP(1,1)+.5*(W(1,5)+W(2,5))-DNI*DT/DR/RH(I)
0191  IF(ALP(1,1).LT.ALMT(I)) ALP(I,1)=ALMT(I)
0192  IF(ALP(1,1).GT. ALMT(2)) ALP(I,1)=ALMT(2)
0193  WN(I+1,5)=ALP(I,1)*R(I)
0194  WN(I+1,5)=0.05*(WN(NGM1,1)+WN(NGM1,1))
0195  ALP(NG,1)+.5*(WN(NGM1,1)+WN(NGM1,1))-DWN*DT/DR/RH(NGM1)
0196  IF(ALP(NG,1).LT. ALMT(I)) ALP(I,1)=ALMT(I)
0197  IF(ALP(NG,1).GT. ALMT(2)) ALP(I,1)=ALMT(2)
0198  WN(NG,3)=ALP(NG,1)*R(NG)
0199  WN(NG,3)=0
0200  WN(NG,3)=0
0201  WN(NG,3)=0
0202  WN(NG,3)=0
0203  WN(NG,3)=0
0204  WN(NG,3)=0
0205  WN(NG,3)=0
0206  WN(NG,3)=0
0207  WN(NG,3)=0
0208  WN(NG,3)=0
0209  WN(NG,3)=0
0210  WN(NG,3)=0
0211  WN(NG,3)=0
0212  IF(IW.EQ.0) THEN
0213  WN(NG,3)=WN(NGM1,3)+WN(NGM1,3)+WN(NGM1,3)+WN(NGM1,3)
0214  WN(NG,3)=WN(NGM1,3)+WN(NGM1,3)+WN(NGM1,3)+WN(NGM1,3)
0215  1
0216  ELSE
0217  WN(NG,3)=0
0218  WN(NG,3)=0
0219  ENDIF
0220  C  ARTIFICIAL TAMING
0221  DO 60 I=1,5
0222  W(I,1)=(1.-DAMP)*W(I,1)+DAMP*W(2,1)
0223  W(NG,1)=(1.-DAMP)*W(I,1)+DAMP*W(NGM1,1)
0224  DO 60 J=2,NGM1
0225  W(I,J)=(1.-DAMP)*W(I,J)+DAMP*W(I,J)+W(I,J)
0226  60  W(I,J)=(1.-DAMP)*W(I,J)+DAMP*W(I,J)+W(I,J)
0227  C  SOLUTION FOR THIS TIME STEP COMPLETED
0228  C  **************DATA AT THE NEW TIME STEP COMPLETED**************
0229  T=T+DT
0230  GO TO 10
0231  WRITE(LUP,7) 'OPEN FILE FAILED ON FILE:
0232  WRITE(LUP,7) NAME
0233  WRITE(LUP,7) 'IOSTAT=',I0S
0234  WRITE(LUP,7) 'CONTINUE'
CALL EXEC(I1, JTIME, JTIME(I1))
WRITE(IUS, '(3S5, 314)') JTIME
CLOSE(IUS)

END

========================================
*EMA /DATA/ /WWW/ /SOURCE/
SUBROUTINE INIT((860425.1537)

C TO SET-UP THE INITIAL CONDITIONS
INTEGER I, IOS, ITLOG, IVTX, J, K, MM
Zioned
PARAMETER (NM=101)
REAL*8 ALMR, WM, WP, U, WD, RDP, RH
1 ,U, V, ALP, P, R, F, S
4 ,RI0, MUEF, VB, NA, ND, DO, GAMMA
7 ,DM, BRH, DAMP, DR
Y, RJJ, RJ2, TJ1, TJ2, QJ, VJ
X, AZ, SELA, DEN12, D1, D2, PI, RE, RMH, RTANK
X ,DS, LS, VS, TS, PE
X ,ALP10, OMEGA, RPEAK, SJR, VPEAK
X ,EUF(2), VT, MU(2), MUID2

COMMON

C ALPLMT/ ALMT(2)
Y /JETS/ RJJ, RJ2, TJ1, TJ2, QJ, VJ
Z /CONT/ IW, LUP, LUS, NG, NGM1, D, MP, DR
C DATA/ V(MM, 2), V(MM, 2), P(MM), R(MM)
C DPAG/ RHO(4), MUEF(2), VB(2), NA, ND
C DSIZE/ DO(2), GDOMA(2)
C SOURCE/ BR(MM), BRH(MM)
C V(MM, 5), WP(MM, 5), WN(MM, 5), DW(MM, 5), RDP(MM), RMH(MM)
C DATA PI/3.141596D0/

**** NOTES: PHASE-1=GAS, PHASE-2=Llfuid ****
C RHO DENSITY, RHO(1) = RHO(2)
C LIMIT VALUES FOR ALP1, ALM1 (ALP1, ALM1)
C DAMP NUMERICAL DAMPING FACTOR, NORMALLY 0.
C DS, LS, VS, TS DENSITY, LENGTH, VELOCITY AND TIME SCALES
C RE REYNOLDS 0=SMALL, RMH*1000>D=NU(2)/MU(2)
C RTANK TANK RADIUS
C IVTX TYPE OF INITIAL FLOW.
C RPEAK, VPEAK VORTEX PARAMETERS
C OMEGA PURE ROTATION. V=OMEGA*
C EVF, MUEF EFFECTIVE VISCOSITY, MUEF=1+EVF*D
C DO, GAMMA DIA. PARAMETERS: D=DO=ALP**GAMMA
C RJJ, RJ2, TJ1, TO DEFINE JET SIZE, PUMPING TIME:
C TJ2, QJ, VJ VOLUME FLOW RATE AND INJECTION MEAN SPEED

C FORMAT(2X, A, 2X, A))
C FORMAT(2X, A, 315)
C FORMAT(2X, A, 3(1PE12.4))
C FORMAT(A25, 2(1PE13.4))
C FORMAT(X, 7(1PE11.4))
DEFINING THE PARAMETERS FOR THE PROBLEM.

RTANK=1.00  
RHO(2)=1.00003  
MU(2)=1.0143-3  

WRITE(1,9) 'ENTER RTANK(M) OR DEFAULT 'RTANK
READ(1,4) 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
data 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=1.0D5  
WRITE(1,9) 'enter Reynolds n. RE. D.F.=',RE
READ(1,4) RE

LS=TANK  
LENGTH SCALE (M)

D=V=1.0D5  
DENSITY SCALE (KG/M=3)

V3=RE*MU(2)/RHO(2)/LS  
VELOCITY SCALE (M/S)

TS=LS/VS  
TIME SCALE (S)

PS=DS/VS=2  
PRESSURE SCALE

AFTER THIS POINT ALL VARIABLES ARE BASED ON THE CHAR. SCALES
I.E. ALL VARIABLES ARE DIMENSIONLESS

DEN2=1.293D0/1.000D3  
DI/D2

MU2=1.710D5/1.514D-7  
MU/MU2

RHO(2)=RHO(2)/DS  

MU(2)=MU(2)/(DEN2/LS)*VS  

RHO(1)=DEN2/RHO(2)  

MU(2)=MU(2)/MU(2)

ALMT(1)=0.0518  
MIN. OF ALP:

ALMT(2)=0.9999  
MAX. OF ALP

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

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

GAMMA(1)=2.0-1  

GAMMA(2)=2.0-1  

EVF(1)=DEN2/MU2=EVF(2)  
TURB. PHASE-DISPERSION EFFECTS

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

NW=4  
WEIGHTING EXP. FOR ABN

MD=4  
WEIGHTING EXP. FOR DRAG

WRITE(1,9) 'Enter DENSITY and VISCOITY ratios.'
WRITE(1,9) 'D.F.=',DENID2,MU2
READ(1,4) DENID2,MU2

WRITE(1,9) 'Enter BASE PARAMETERS: DO1,DO2'
WRITE(1,9) 'D.F. ',DO
READ(1,4) DO

WRITE(1,9) 'Enter SIZE EXPONENT: GAMMA1,GAMMA2'
WRITE(1,9) 'D.F.=',GAMMA
READ(1,*) GAMMA

WRITE(1,8) 'Enter weighting exponent: NA,ND. D.F.=',NA,ND
READ(1,*) NA,ND
WRITE(1,8) 'Enter GAS VOLUME FRACTION limits: ALMT1, ALMT2.',ALMT
WRITE(1,8) 'D.F.=',ALMT
READ(1,*) ALMT
WRITE(1,8) 'Enter eddy viscosity factor. D.F.=',EUF
READ(1,*) EUF

IW=1
WRITE(1,7) 'Enter wall condition,1=nonslip,0=slip. D.F.',IW
READ(1,*) IW
WRITE(1,8) 'Enter numerical damping factor. D.F.=',DAMP
READ(1,*) DAMP
DO 10 K=1,2
V18(K)=18.*MU(K)
10 MUEF(K)=MU(K)*(1.*EUF(K))  ; EFFECTIVE VISCOSITY FOR STRESS

RHO(3)=RHO(1)*RHO(2)
RHO(4)=RHO(1)-RHO(2)
RMN=.1  ; MINIMUM FLOW RADIUS IN THE TANK
NG=101  ; # OF GRID POINTS USED
NGM=NG-1
DR=(1.-RMN)/NGM

C Initial cleaning-up.
DO 15 J=1,NH
DO 15 K=1,6
W(J,K)=0.D0
15 W(J,K)=0.D0

C MOMENTUM SOURCE, JET CONDITIONS
RJ1=.5D-1
RJ2=.5D-1
VJ=10.  ; TANGENTIAL INJECTION SPEED
TJ1=0.
TJ2=10.

WRITE(1,8) 'Enter JET SIZE defined by RJ1,RJ2. D.F.=',RJ1,RJ2
READ(1,*) RJ1,RJ2
WRITE(1,8) 'Enter INJECTION SPEED AND TIME RANGE, VJ,T1,T2'
READ(1,8) 'D.F.=',VJ,TJ1,TJ2
READ(1,8) 'D.F.=',QJ=RJ2-RJ1)*VJ  ; JET VOLUME FLOW RATE

DO 20 J=1,NG
R(J)=RMN+(J-1)*DR
RH(J)=R(J)+.15*DR
BR(J)=DEL((R(J)),DR,RJ1,RJ2)/(2.*PI)  ; JET DISTRIBUTION
20 BRH(J)=DEL((RH(J)),DR,RJ1,RJ2)/(2.*PI)  ; PER RADIUS

C SETUP INITIAL CONDITIONS
IVTX=0
0419  OMEGA=0.
0420  VPEAK=0.
0421  RPEAK=RMIN
0422
0423  NAME='Simple vortex'
0424  WRITE(1,5) 'Enter data FILE NAME for initial cond., i.e. 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,STAT=IDS,STATUS='OLD',ERR=299)
0432      DO 25 J=1,NG
0433        25  READ(99,..,ALP(J,1),U(J,1),U(J,2),V(J,1),V(J,2)
0434      CLOSE(99)
0435
0436  ELSE
0437    C
0438      TO DEFINE INITIAL CONDITION.
0439    ALP10=2.00-1
0440    WRITE(1,8) 'Enter initial value of alp1. D.F.=',ALP10
0441    READ(1,9) ALP10
0442
0443    IF(IUTX .GT. 0) THEN
0444      IF(IUTX .GT. 1) THEN
0445        WRITE(1,8) 'Enter type of vortex: 0=At rest,1=pure rotation'
0446        WRITE(1,8) '2=H.O.,3=GIT. D.F.=',IUTX
0447        READ(1,9) IUTX
0448        IF(RPEAK .LE. 0.) RPEAK=1.  SINGULAR AT ZERO
0449      ELSE
0450        WRITE(1,8) 'Enter CIRCULAR SPEED(rad./unit time). D.F.='
0451          1
0452        READ(i,9) OMEGA
0453      ENDIF
0454    ENDIF
0455
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. 1 ) THEN
0462        UT=UT+1.398*VPEAK/RJB/(1.-DEXP(-1.25643*RJB**2))
0463      ELSE
0464        UT=UT+VPEAK/RJB*DEXP((1.-RJB**2)/2.)
0465      ENDIF
0466    ENDIF
0467    ENDIF
0468    DO 30 K=1,2
0469    IF(IW .EQ. 0) THEN
0470      V(J,K)=UT
0471      ELSE
0472        V(J,K)=UT*(1.-R(J))/R(J)
0473      ENDIF
0474    30  U(J,K)=0.
0475    ENDIF
0476    ENDIF
0477
0478    DO 40 J=1,NG
0479    FORM W FOR NUMERICAL CAL.
C PRINTOUT PARAMETERS
0479 ALP(J,2)=1.-ALP(J,1)
0480 P(J)=0.D0
0481 W(J,5)=R(J)*ALP(J,1)
0482 DO 40 K=1,2
0483 W(J,K)=ALP(J,K)*U(J,K)*R(J)
0484 DO 40 K=1,2
0485
0486 C...
0488 WRITE(US,5) 'INITIAL CONDITION FILE:',NAMR
0489 WRITE(US,5) 'DIMENSION UNITS ARE IN MKS**'
0490 WRITE(US,9) 'DENsITY SCALE(kg/m3)',DS
0492 WRITE(US,9) 'LENGTH SCALE= Tank,(m)',LS
0493 WRITE(US,9) 'VELOCITY SCALE(m/s)',V6
0494 WRITE(US,9) 'TIME SCALE(s)',TS
0495 WRITE(US,9) 'PRESSURE SCALE(Pa)',PS
0496 WRITE(US,9) 'Reynolds number, Re',RE
0497 WRITE(US,9) 'Jet size, RJ1,RJ2',RJ1,RJ2
0498 WRITE(US,9) 'Tangential jet, QJ, VJ',QJ,VJ
0499 WRITE(US,9) 'Injection time, TJ1,TJ2',TJ1,TJ2
0500 WRITE(US,9) '3/33,' PHASE-1',8X,'PHASE-2')
0502 WRITE(US,9) 'DensiTy',RHO(1),RHO(2)
0503 WRITE(US,9) 'Viscosity',MU
0504 WRITE(US,9) 'Eddy viscosity factor',EUF
0505 WRITE(US,9) 'Dense dia.',DO
0506 WRITE(US,9) 'Size exp. ',GAMMA
0507 WRITE(US,9) 'Phase limits',ALMT(1),1.-ALMT(2)
0508 WRITE(US,9)
0509 WRITE(US,9)
0510 WRITE(US,9) 'OTHER CONSTANS: Iw,Jvtx,NA,ND,DAMP,VPEAK,RPEAK',
0511 I_1
0512 WRITE(US,9) 'OMEGA',D1/D2,MU1/MU2')
0513 WRITE(US,9) 'Iw,Jvtx,NA,ND,DAMP
0514 WRITE(US,9) 'VPEAK,RPEAK,OMEGA,DEN12,MU1D2
0515 WRITE(US,9)
0516 WRITE(US,9) '(I10,F10.2)') NG,RMIN
0517 RETURN
0518 399 WRITE(LUP,5) 'OPEN FILE FAILED ON INPUT FILE:',NAMR
0520 WRITE(LUP,7) 'IODESTATE',IDS
0521 STOP 111
0522 END
0523
0524 **********************************************************************
0525 REAL*8 FUNCTION DELA(R,DR,RJ1,RJ2),(0.48423.1537)
0526 C TO DETERMINE THE EFFECTIVE NOZZLE SIZE AT EACH GRID LOCATI
0527 C THE SIZE IS IN THE FRACTION OF GRID SIZE DR (1.0, # (DELA(1)
0528 C REAL*8 R,DR,RJ1,RJ2, R1,R2
0529 REAL*8 R,DR,RJ1,RJ2, R1,R2
0530 R1=R-0.5*DR
0532 R2=R1+DR
0533 DELA=0.D0
0534 IF(R1 .GE. RJ2 .OR. R2 .LE. RJ1) RETURN
0535 IF(R1 .LT. RJ1) R1=RJ1
0536 IF(R2 .GT. RJ2) R2=RJ2
0537 DELA=(R2-R1)/DR
0538 RETURN
END

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

SUBROUTINE DERIV(Y,DY,DX,N2),(860425.1537)

C GET: 1ST DERIVATIVE, USING CENTERED DIFFERENCE

REAL#8 DX,Y(1),DY(1),C

EMA Y,DY

C=5.0/DX

DO 10 J=2,N2-1

10  DY(J)=CY(Y(J)-Y(J-1))

CALL DWSUB(Y(3),1,Y,1,DY(2),1,N2-2)

DY(1)=(Y(2)-Y(1))/DX  ! BASED ON 3-END PTS

DY(N2)=(Y(N2)-Y(N2-1))/DX

DY(1)=2.*DY(1);-DY(2)

DY(N2)=2.*DY(N2);-DY(N2-1)

CALL DWVY(5.0/DX,DY,1,DY,1,N2)

RETURN

EMA END

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

EMA /COEFF/

SUBROUTINE DDCOEF(ALP,N2),(860425.1537)

C CALCULATE THE DRAG, ADDED MASS AND GENERIZED COEFF.

INTEGER J,MM,N2

PARAMETER (MM=101)

REAL#8 ALP(NM,2)

EMA ALP

REAL#8 C,CPA,CD, RHO,MUEF,V18,NA,ND

COMMON X

/C0EFF/ (C(MM,2),CPA(MM,2),CD(MM,2)

/DRAG/ RHO(4),MUEF(2),V18(2),NA,ND

DO 30 J=1,2

USE ALP(J,1)

A2=ALP(J,2)

A2=A1#A2

TO GET DRAG COEFF. AD

CALL SIZEB(D1,D2,A1)

AD=V18(2)*A1/(A2#D1*D1)  ! AD IF A2)

IF( A2 .LT. .78) THEN

AD2=V18(1)*A2/(1.-A2/.8)*D2)**2

IF(AD2 .LT. AD) THEN

WT1=A2**ND

WT2=A1**ND

AD=(AD+WT1+AD2#WT2)/(WT1+WT2)

ENDIF

ENDIF

ADDED MASS COEFF. AA

***********************************************************************
AA1=A12*RH0(2)/(A1+2/A1) *(.5+1.5*A1))
AA2=A12*RH0(3)/(A1/5+1.5*A2)+A2)
WT1=A2*NA
WT2=A1*NA
AA=(AA1*WT1+AA2*WT2)/(WT1+WT2)

C THE GENERALIZED COEFF. CPA, C, AND CD
DB2=A12*RH0(3)+AA*(RH0(1)+A1)*RH0(2)*A2)
CPA(J,1)=A1*(A12*RH0(2)+AA)/DB2
CPA(J,2)=A2*(A12*RH0(1)+AA)/DB2
C(J,1,2)=A1/DB2
C(J,2,1)=C(J,1,2)
C(J,1,1)=A2*RH0(2)/DB2+C(J,1,2)
C(J,2,2)=A1*RH0(1)/DB2+C(J,2,1)
CD(J,1)=-A2*RH0(2)*AD/DB2
CD(J,2)=A1*RH0(1)*AD/DB2

50 CONTINUE

RETURN
END

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

$EMA /DATA//FANDS//TAU//COEFF/

SUBROUTINE DWPDE(DW, D, DT, RR, N2), (864025.1537)

C TO GET DW OF THE PDEs

C INTEGER JP1, X, KP2, MM, N2

C PARAMETER (MM=101)

C C =A B DW(MM,5), DMPDE(DM, DMP, C, D, GT, EN)

C C =A B EMA DW, DMP, RR

C C NOTES: COEFF. C =C=ALP WHEN THIS IS CALLED

REAL B, CPA, CD, U, V, ALP, P, R, F, S

8 TRR, TRA, TAA, RTR, RTRA

4 RH0, MUEF, V18, NA, ND

3 ALP1, ALP2, CP1, CP2, DTR, DW1, DW1T, G1, G2, HDT, VT, W1, W3, W4

COMMON

1 /COEFF/ C(MM,2,2), CPA(MM,2), CD(MM,2)

3 /DATA/ U(MM,2), V(MM,2), ALP(MM,2), P(MM), R(MM)

4 /DRAG1/ RH0(4), MUEF(2), V18(2), NA, ND

6 /FANDS/ F(MM,5), S(MM,5)

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

DTR=D/DT

HDT=0.5*DT

DO 10 J=1,N2+1

C(J,1,1)=ALP(J,1)*C(J,1,1)

C(J,2,1)=ALP(J,2)*C(J,2,1)

C(J,1,2)=ALP(J,1)*C(J,1,2)

C(J,2,2)=ALP(J,2)*C(J,2,2)

10

DO 20 J=1,N2

JP1=J+1

DW(J,5)=DTR*(-F(JP1,5)+F(J,5))/HDT*(S(JP1,5)-S(J,5))

1

DO 25 K=1,2

26
C  DP FOR PRESSURE CORRECTION
CP1=0.5*(CPA(J,1)+CPA(JP1,1))
CP2=0.5*(CPA(J,2)+CPA(JP1,2))
IF(-DW(J,1).GT.DW(J,2)) DW(J,1)=-DW(J,2) ! DP=0
RDP(J)=(DW(J,1)+DW(J,2))/(CP1+CP2)
DW(J,1)=-CP1*RDP(J)
RETURN
END

********************************************************************
SERA /CUEFF/.DATA/.WWW/
SUBROUTINE FDNT(DT,DR,VDR,NG),(86035.1537)

C DETERMINE THE TIME-STEP SIZE
INTEGER I,J,LUP,MM,NG
PARAMETER (MM=10)
REAL*8 DT,DR,VDR(2)
REAL*8 C,CPA,CD, RHO,MUEF,V18,NA,ND
9 ,W,WP,WM,DW,RDP,RH, U,V,ALP,P,R
X ,DUM1,DUM2
COMMON
2 /CUEFF/, CM,2,CPA,CD, RHO,MUEF,V18,NA,ND
4 /DRAV/ RHO(4),MUEF(2),V18(2),NA,ND
DATA LUP/1/
DUM1=0.
DO 10 J=1,NG
10 DUM2=-CD(J,1)
   1/DJ
   : CONVECTIVE
   1 VDR=2*MUEF/DR**2
   1 CROSS VISCOITY
   IF(DUM1.LT. DUM2) DUM1=DUM2
   DUM2=CD(J,2)+DABS(U(J,2))/DR
   : 1/VDR
   : 1 ALP/J,2)*C(J,2,2)
   IF(DUM1.LT. DUM2) DUM1=DUM2
   10 CONTINUE
C FIND THE MAXIMUM OF DW
27
CALL DWMAX(I,DW,1,NG)  VECTOR OPERATION
CALL DUM1=DBS(DW(I,1))
CALL DWMAX(J,DW(1,2),1,NG)
DUM2=DBS(DW(J,2))
D1=1.0/DUM1
RETURN
END

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

Cm /DATA/,/FANDS/,/TAU/,/COEFF/
SUBROUTINE FSOF(W,BA,RH,N2),(B68429,1837)
C CALCULATE THE CONVECTIVE-F AND SOURCE-S TERMS
C
INTEGER J,K,KP2,L,MM,N2
PARAMETER (MM=101)
REAL*8 U,V,ALP,P,R,F,S,RHO,HUEF,VB,NA,ND
2 !TR,TRA,TAA,RTRR,RTRA, CPA,CD
ALP(2),RDU,RDV
REAL*8 W(MM,5),BA(MM,2),RR(1)
Cm W,BA,RR
CMOM
C
DO 20 J=1,N2
F(J,5)=W(J,1)
S(J,5)=0.
RDU=RR(J)*(U(J,1)-U(J,2))
RDV=RR(J)*(V(J,1)-V(J,2))
DO 20 K=1,2
KP2=K+2
F(J,K)=W(J,K)#U(J,K)
F(J,KP2)=W(J,KP2)#U(J,K)
S(J,K)=ALP(J,K)*(U(J,K)#2+CD(J,K)RDU-C(J,K)TAA(J,K)
1 -C(J,J,1)#TAA(J,1)
20 S(J,KP2)=ALP(J,K)#{-U(J,K)#U(J,K)+CD(J,K)RDV+C(J,K,1)#BA(J,1)
1 +TRA(J,1)+C(J,K,2)#{BA(J,2)+TRA(J,2)})
RETURN
END

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

SUBROUTINE JET(BA,RH,BA,VB,VJ,NG)
C INJECTION MOMENTUM SOURCE
0779  PARAMETER (MM=181)
0780  REAL*8 BA(MM,2),RHO(1),BK(1),V(MM,2),U
0781  END
0782
0783  DO 10 J=1,MM
0784  IF( BK(J) .GT. 0. ) THEN
0785  Q=BA(J,1)*U(J)
0786  BA(J,1)=Q*(V(J-1,1)-RHO(1))
0787  BA(J,2)=Q*(V(J,1)-RHO(2))
0788  ELSE
0789  BA(J,1)=0.0
0790  BA(J,2)=0.0
0791  ENDIF
0792  10 CONTINUE
0793  RETURN
0794  END
0795
0796  SUBROUTINE SIZE(D1,D2,ALP1),(860425.1537)
0797  C TO DETERMINE THE PARTICLE DIAMETERS
0798  C
0799  REAL*8 D1,D2,ALP1, DO,GAMMA
0800  COMMON /DSIZE/ DO(2),GAMMA(2)
0801
0802  D1=DO(1)#ALP1#GAMMA(1)
0803  D2=DO(2)#(1.0-ALP1)#GAMMA(2)
0804
0805  RETURN
0806  END
0807
0808  !SUBROUTINE TAUDWF(MU,DR,RR,N2),(860425.1537)
0809  !STRESSES AND THEIR DERIVATIVES
0810  C
0811  REAL*8 MU(2),DR,RR(1)
0812  END
0813
0814  PARAMETER (MM=101)
0815  REAL*8 U,V,ALP,P,R
0816
0817  !
0818  REAL*8 X
0819  2 ,TRR,TRA,TAA,RTRR,RTRA
0820  X
0821  ,TAMU,THU
0822
0823  COMMON
0824  3 /DATA/ (U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0825  !
0826  8 /TAU/ TRR(MM,2),TRA(MM,2),TAA(MM,2),RTRR(MM,2),RTRA(MM,2)
0827
0828  DO 50 K=1,2
0829  CALL DERIV(U(1,K),TRR(1,K),DR,N2)
0830  CALL DUMPY(ALP(1,K),1,TRR(1,K),1,N2)
0831  DO 10 J=1,N2
0832  CALL DUMPY(2,#MU(K),TRR(1,K),1,N2)
0833  CALL DUMPY(V(1,K),1,RR,1,TAA(1,K),1,N2)
0834  DO 10 J=1,N2
0835  CALL DERIV(TAA(1,K),TRR(1,K),DR,N2)
0836  10 CONTINUE
0837  DO 20 J=1,N2
0838  THU=MU(2,#MU(K)
0839  DO 20 J=1,N2
0840  THU=ALP(J,K)
0841  20 CONTINUE
0842
0843  RETURN
0844  END
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)*TRA(J,K)
0843 #U CALL DWHPY(RR,1,TRA(1,K),1,TRA(1,K),1,N2) ! R
0845 #U CALL DWHPY(ALP(1,K),1,TRA(1,K),1,TRA(1,K),1,N2) ! ALP
0847 #U CALL DWHPY(MU(K),TRA(1,K),1,TRA(1,K),1,N2) ! MU=TRA
0848 #U CALL DWDIV(U(1,K),1,RR,1,TAA(1,K),1,N2) ! U/P
0849 #U CALL DWHPY(ALP(1,K),1,TAA(1,K),1,TAA(1,K),1,N2) ! ALP
0850 #U CALL DWHPY(ZU(1,K),TAA(1,K),1,TAA(1,K),1,N2) ! ZU=TRA
0852 #U CALL DWHPY(RR,1,TRR(1,K),1,TRR(1,K),1,N2) ! R=TRR
0853 #U CALL DWHPY(RR,1,TRA(1,K),1,TRA(1,K),1,N2) ! R=TRA
0854 50 CONTINUE
0855 RETURN
0856 END
0857 END

0858 ***********************************************************************************************
0859 $EMA /DATA/
0860 C SUBROUTINE UDFW(W,RR,N2),(860425.1537)
0862 C CONVERTS W TO THE INDEPENDENT VARIABLES(U,V,ALP)
0863 PARAMETER (MM=101)
0864 REAL*8 W(MM,5),RR(MM)
0865 COMMON W,RR
0866 REAL*8 ALMT, U,V,ALP,P,R
0868 X
0869 X
0870 X
0871 X
0872 X
0873 #U CALL DWDIV(W(1,5),1,RR,1,ALP,1,N2)
0874 #U CALL DWDIV(W(1,5),1,RR,1,ALP,1,N2)
0875 C CHECK VOLUME FRACTION & FLOW DIRECTIONS
0876 DO 30 J=1,N2
0877 IF(ALP(J,1).LE.ALMT(1)),OR.,ALP(J,1).GE.ALMT(2)) THEN
0878 IF(ALP(J,1).LE.ALMT(1)) THEN
0879 ALP(J,1)=ALMT(1)
0880 W(J,3)=W(J,4)*ALMT(1)/(1.0D0-ALMT(1))
0881 ELSE
0883 ALP(J,1)=ALMT(2)
0884 W(J,4)=W(J,3)/(1.0D0-ALMT(2))/ALMT(2)
0885 ENDIF
0886 W(J,1)=0.0D0
0887 W(J,2)=0.0D0
0888 W(J,3)=ALP(J,1)*RR(J)
0889 ENDIF
0891 ENDIF
0892 ALP(J,2)=1.0D0-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,3)=W(J,4)=0.
0897 ENDIF
0898 END
U(J,1)=W(J,1)/W(J,5)
V(J,1)=W(J,3)/W(J,5)
W(J)=RR(J)*ALP(J)

50 CONTINUE

*CALL DWDIV(W(I_5),I_5,W(I),I,ALP(I_2),I,U(1,2),I,N2)
*CALL DWDIV(W(I_3),I_3,W(I),I,ALP(I_2),I,U(1,2),I,N2)
*CALL DWDIV(W(I_2),I_2,W(I),I,ALP(I_2),I,U(1,2),I,N2)
*CALL DWDIV(W(I_4),I_4,W(I),I,ALP(I_2),I,U(1,2),I,N2)

RETURN
EMD
Exhibit A

A Sample Input

G

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. = 1.1442E+02 1.0000E+03
1000,1000
Enter wall condition: i=nonslip, o=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
1.0,1
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
0.01
Enter TIME STEP for output. D.F. = 2.0000E-01
.01
1 NP= 1 T=0.0000E+00 NT= 2 DT=1.0000E-06
1 NP= 2 T=1.2000E-02 NT= 5 DT=3.0000E-03
Exhibit B

A Sample Output

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 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-05
0022
0023 OTHER CONSTANTS: IW,IVTX,NA,ND,DAMP,VPEAK,RPEAK,OMEGA,D1/D2,MU/MU
0025 1 0 4 4 0.00
0026 0.0000E+00 1.0000E-01 0.0000E+00 1.2930E-03 1.1295E-02
0027
0028 1 NP= 1 T=0.0000E+00 NT= 2 DT=1.0000E-06
0030 J ALP1 U1 U2 V1 V2 P
0031 1 2.5000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0032 2 2.5000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

-------- OUTPUT IN THE BETWEEN OMITTED ------
0230 97 2.5000 -8.685E-10 2.895E-10 3.513E-04 3.474E-04 3.969E-05
0231 98 2.5000 -1.775E-10 5.918E-11 1.268E-04 1.250E-04 3.972E-05
0232 99 2.5000 -2.783E-11 4.204E-12 4.319E-05 4.256E-05 3.973E-05
0233 100 2.5000 -5.225E-12 1.742E-12 1.553E-05 1.531E-05 3.973E-05
0234 101 2.5000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 3.973E-05

-------- 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.
Figure 3. Gas Volume Fraction Distributions