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
NBSIR 86-3414

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

T. T Yeh

Inputs

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 code for gas-liquid two-phase vortex motions: GLYN

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 any proper names; and separate key words by semicolons)
computer code; gas-liquid separation; numerical modeling; two-phase vortex motions

Availability
☑ Unlimited
☐ For Official Distribution. Do not release to NTIS
☐ Order from National Technical Information Service (NTIS), Springfield, VA. 22161
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
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
\[ C_{1j}, C_{pk}, C_{dk} \] Generalized coefficients
\[ d_1 \] Bubble diameter
\[ d_2 \] Liquid (droplet) diameter
\[ \bar{\kappa}_k \] Effective interfacial force density
\[ n_a \] Exponent used for \( w_{ak} \)
\[ n_d \] Exponent used for \( w_{dk} \)
\[ p \] Pressure
\[ R \] Tank radius
\[ Re \] \( \rho_2 V_0 R/\mu_2 \), 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
\( V_{02} \)  
Liquid tangential velocity

\( \bar{V}_1 \)  
Gas velocity

\( \bar{V}_2 \)  
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

\( e \)  
\( \mu_k + \mu_k^t \) total effective viscosity

\( \mu_k^t \)  
Turbulence or eddy viscosity

\( \nu_k \)  
\( \mu_k/\rho_k \), 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
### 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 (Backward Predictor-Forward Corrector Version)</td>
<td>34</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 which 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:

\[ a_1 + a_2 = 1 \]

\[ \frac{\partial \rho_k}{\partial t} + \frac{\partial \rho_k v_k}{\partial r} = 0 \]
\[
\frac{\partial \alpha_1 V_{rk}}{\partial t} + \frac{\partial \alpha_2 V_{rk}}{\partial r} - \alpha_1 V_{rk}^2 = -\alpha_2 C_{pk} \frac{\partial \rho}{\partial r} \\
+ \alpha_1 \sum_{l=1}^{2} C_{kl} \left( \frac{\partial \alpha_{l r \theta \theta}}{\partial r} - \alpha_2 r \theta \theta \theta \theta + \alpha_2 \rho \rho B_{rl} r \right) \\
+ \alpha_1 C_{dk} \left( V_{rl} - V_{r2} \right)
\]

\[
\frac{\partial \alpha_2 V_{8k}}{\partial t} + \frac{\partial \alpha_1 V_{rk}}{\partial r} + \alpha_2 V_{rk} V_{8k} = \alpha_1 \sum_{l=1}^{2} C_{kl} \left( \frac{\partial \alpha_{l r \theta \theta}}{\partial r} - \alpha_2 r \theta \theta \theta \theta + \alpha_2 \rho \rho B_{8l} r \right) \\
+ \alpha_1 C_{dk} \left( V_{91} - V_{92} \right)
\]

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

\[
C_{p1} = (a_1 a_2 \rho_2 + A_a)/\langle \rho^2 \rangle
\]

\[
C_{p2} = (a_1 a_2 \rho_1 + A_a)/\langle \rho^2 \rangle
\]

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

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

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

\[ C_{d2} = a_1 \rho_1 A_d / \langle \rho^2 \rangle \]

and

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

The effective stresses are modeled as

\[ \tau_{rrk} = 2 \mu_k^e \partial V_{rk} / \partial r \]

\[ \tau_{\theta rk} = \tau_{\theta kr} = \mu_k^e r \partial (V_{\theta k} / r) / \partial r \]

\[ \tau_{\theta \theta k} = 2 \mu_k^e V_{rk} / r \]

with

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

and the interfacial forces are modeled in the form of

\[ \overline{M}_1 = A_d (\overline{V}_2 - \overline{V}_1) + A_\alpha \frac{d}{dt} (\overline{V}_2 - \overline{V}_1) . \]

\( \overline{M}_1 \) is the force density acting on the phase 1 by the phase 2. \( A_\alpha \) 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 $a_k \rho_k \bar{B}_k$ will be replaced by
the net momentum gain, $\frac{a_k \rho_k V_j}{2\pi} (V_j \bar{n} - \bar{V})$ 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: i) its explicitly conservative form, ii) it is a two-step
predictor-correction type, iii) 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
iv) 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

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

\[
= W^{0} + \frac{\Delta t}{2} W_t^0 + \frac{\Delta t}{2} (W_t^0 + \Delta t W_{tt}^0)
\]

\[
= \frac{1}{2} (W^0 + \Delta t W_t^0) + \frac{1}{2} (W^0 + \Delta t W_t^0)
\]

\[
= \frac{1}{2} (W^P + W^C)
\]

where

\[ W^P = W^0 + \Delta t W_t^0 \] is the predicted value,

and

\[ W^C = W^0 + \Delta t W_t^P \] 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

\[ \hat{w}_i^p = \frac{1}{2} (\hat{w}_{i-1/2}^p + \hat{w}_{i+1/2}^p) \]

where

\[ \hat{w}_{i-1/2}^p = \frac{1}{2} (\hat{w}_{i-1}^o + \hat{w}_i^o) + \Delta t \hat{w}_t^o \]

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

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

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

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

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

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

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 $R_1$ 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

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

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

$$w_{i-1}^{1D} = w_i^1 (1-D) + (w_{i-1}^1 + w_{i+1}^1 - 2w_i^1) D$$
where $w_1^1$ is the value obtained based on the two-step scheme, and $w_1^D$ 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., $d_k = d_{ok} \alpha_k^Y$
- **DS**: Density scale = $\rho_2$
- **EVF**: $u^t/\mu$ 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, $\text{Mf} \geq \text{NG}$. MM appears in many subroutines.
- **\(\mu_k\)**: Dynamic viscosity for phase k.
- **MLEF**: $(1 + \text{EVF}) \times \text{MJ}$. 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, \( \rho_2 V_3 R/\mu_2 \)
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.
TJJ,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 \( K \), \( \alpha_1(K) \), \( V_{1r}(K) \), \( V_{2r}(K) \), \( V_{1t}(K) \), \( V_{2t}(K) \), 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.

h) 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:

\[
\begin{align*}
A_a &= A_{al} w_{al} + A_{a2} w_{a2} \\
A_d &= A_{d1} w_{d1} + A_{d2} w_{d2} \\
A_{al} &= \frac{a_1 a_2 \rho_2}{(a_1 + 2 a_2/(1 + 3 a_1))} \\
A_{a2} &= \frac{a_1 a_2 \rho_1}{(a_2 + 2 a_1/(1 + 3 a_2))} \\
A_{d1} &= 18 \frac{u_2 a_1}{d_1^2 a_2} \\
A_{d2} &= 18 \frac{u_1 a_2}{(1 - a_2/0.8^2 d_2^2)} \\
w_{al} &= \frac{a_2}{(a_1 + a_2)} \\
w_{a2} &= 1 - w_{al} \\
w_{d1} &= \frac{a_2}{(a_1 + a_2)} \\
w_{d2} &= 1 - w_{d1}
\end{align*}
\]

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) FND?T, \( \Delta t \)

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 \( \alpha, V_r, V_\theta \).

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

```fortran
&GLVM T=00004 IS ON CR T4 USING 00126 BLKS R=0000

0001 FTN77
0002 $EMA /DATA/ , /WNN/ , /COEFF/ , /SOURCE/ , /FANDS/ , /TAU/
0003 $FILES 1.2
0004 $PROGRAM GLVM(99) , (860425.1537)
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 0502XX TTY ZI 0-G FUEL TRANSFER, START-UP STAGE.
0014 C LAM. WENDROFF 2-STEP SCHEME (FULL STEP PREDICTION+CORRECTION)
0015 C WITH NUMERICAL DAMPING FACTOR (NORMALLY SET TO ZERO)
0016 C CONSERVATION FORM, VARIABLE(AUTO) TIME STEP
0017 C REAL#8
0018 C INTERFACIAL FORCES: DRAG, ADDED MASS
0019 C PUMP CONDITION: MOMENTUM SOURCE BUT NO MASS SOURCE
0020 C 050715 TTY GENERALIZED EQUATIONS AND COEFF. C1j
0021 C 051018 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, DMPDE, FMDDT, 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, IOS, J, JTIME(3), K, MM, NPT, NT
0031 C , IU, LUP, LUS, NG, NKM1, NKM2
0032 C PARAMETER (MM=101)
0033 C REAL# BA, MM, 2, DDDT, DT, DMAX, DTMIN, DTPRT, PZERO
0034 C X , DWI, T, TMAX, TPTRT, VIJ, VDR(2)
0035 C Y , RJ1, RJ2, TJ1, TJ2, QJ, VJ
0037 C BR, BRM, RH, MUEF, VIJ, NA, ND
0038 C DO, GAMMA, DMP, DR
0039 C TR, TRA, TAA, TR,TRA, RTRA, C, CPA, CD
0040 C
0041 COMMON
0042 Y /JETS/ RJ1, RJ2, TJ1, TJ2, QJ, VJ
0043 Z /CONT/ IU, LUP, LUS, NG, NKM1, DAMP, DR
0044 1 /ALPLNT, ALM(2)
0045 2 /CCOEFF/ C(MM, 2, 2), CPA(MM, 2), CP(MM, 2)
0046 3 /DATA/ U(MM, 2), V(MM, 2), ALP(MM, 2), P(MM), R(MM)
0047 4 /DRAI/ RHO(4), MUEF(2), VIJ(2), NA, ND
0048 6 /FANDS/ F(MM, 5), B(MM, 5)
0049 7 /SOURCE/ BR(MM), BRM(MM)
0050 8 /TAU/ TRA(MM, 2), TRA(MM, 2), TAA(MM, 2), RTRA(MM, 2), RTRA(MM, 2)
0051 9 /WWW/ W(MM, 3), WP(MM, 3), WN(MM, 3), DW(MM, 3), RD(MM), RM(MM)
0052 EQUVALENCE (: WN, BA)
0053 C
0054 C ***** RHO(1) ( RHO(2) ( i.e. PHASE-1=GAS, PHASE-2=LIQUID) *****
0055 C DTPRT TIME STEP FOR PRINTOUT (AND PLOT)
0056 C
0057 C
0058 C
```
LUP=1  ! LU FOR PRINTING DEBUG DATA(#1 TERMINAL)
LUS=6  ! LU FOR STORING DATA(#6 PRINTER)

FORMAT(2X,A,3J15)
FORMAT(2X,A,3(1PE12.4))
WRITE(1,7) 'Enter lu for saving data. D.F.=',LUS
READ(1,*) LUS

CALL EXEC(11,'TIME,JTIME(1))

IF(LUS .NE. 1 .AND. LUS .NE. 6) THEN
C **** Define a file name for string output ***************
WRITE(1,(2A)) 'Enter FILE NAME for saving data.'
READ(1,'(A') NAMN
LUS=99
OPEN(LUS,FILE=NAMN,IOSTA=I0S,STATUS='NEW',ERR=999)
ENDIF

WRITE(1,'(A)') ' Enter NOTES((73 CHAR.) for the job'
READ(1,'(A)') NOTES
WRITE(LUS,'(3H1 ,A)') NOTES
WRITE(LUS,'(5I4)') JTIME

C To set-up the initial condition.
CALL INIT

NGM2=NG-2
VDR(1)=2.*MUEF(1)/DRT2  ! For determining time step
VDR(2)=2.*MUEF(2)/DRT2
T=0.  ! INITIAL TIME
TMAX=5.
DTPRD=0.2
WRITE(1,8) 'Enter INITIAL and FINAL TIMES. D.F.=',T,TMAX
READ(1,*) T,TMAX
WRITE(1,8) 'Enter TIME STEP for output. D.F.=',DTPRD
READ(1,*) DTPRD

DTMIN=1.0D-6  ! SET MINIMUM TIME STEP
DTMAX=DTPRD

NT=0  ! Time step number
TPRT=T
NPRT=0
PZERO=0.00  ! Pressure at center core
DT=DTMIN

10 NT=NT+1
CALL UOFM(W,R,4G)

C PRINT OUT AT SELECTED TIME
IF(T .GE. TPRT OR. T .GT. TMAX) THEN
IF(NT .GE. T) THEN
NPRT=NPRT+1
TPRT=TPRT+DT*(1.+DNINT((T-TPRT)/DTPRD))
FORMAT(1HI,2(A5,I4,A5,IPF .3))
WRITE(LUS,21) 'NP=',NPRT,'T=',T,'NT=',NT,'DT=',DT
WRITE(LUS,21) 'NP=',NPRT,'T=',T,'NT=',NT,'DT=',DT
WRITE(LUS,'(A5,A6,5A11)') 'I','ALP1','U1','U2','U1','U2','P'

17
DO 30 J=1,NG
   WRITE(LUS,'(I3,F6.4,5(1PE11.3))')
   ENDIF
   ENDIF
   IF(T.GT.TMAX) GOTO 9999
C TO SOLVE THE DIFFERENTIAL EQN.
C USING 2 STEP LAX-WENDROFF SCHEME
C MacCormack's method.  BACKWARD PREDICTOR, FORWARD CORRECT
C CENTER DIFFERENCED ON TAU
C FIRST TO GET THE SPECIAL VARIABLES AND THEIR SPACIAL DERIVATIVE
C CALL DGCOF(ALP,NG)   ! DRAG AND GENERALIZED COEFF.
C CALL TAUOFW(MUEF,DR,R,NG)   ! STRESS
C
VJT=VJ
   IF(T.LT.TJ1 OR T.GT.TJ2) VJT=0 ! NO INJECTION
   CALL JET(BA,RHO,DR,V,VJT,NG) ! MOMENTUM SOURCE
   CALL FSOFW(W,BA,R,NG)     ! CONVECTIVE -F AND SOURCE-S
C DETERMINE THE TIME-STEP SIZE
   IF(N.T.GT.1) THEN
      CALL FNDT(DT,DR,NG)
      DT=DT*.5*DTMIN/10.*NT
   ENDIF
   IF(DT.GE. DTMIN) THEN
      I=DLOG10(DT/DTMIN)
      DDT=DTMIN*10.**I
      DT=DTMIN*(DT/DDT+0.001)*DDT
      IF(DT.GT. DTMAX) DT=DTMAX
   ELSE
      WRITE(LUP,'(5X,A,1PE13.3)')
      GOTO 9999
   ENDIF
   GOTO 9999
C STOP DUE TO TOO SMALL TIME STEP.  DT=',DT
C BACKWARD PREDICTOR
   CALL DWPEDE(DW,RDP, DR,DT,RH,NGM1)   ! INCREMENT
   DO 40 J=1,NGM1
      DO 40 I=1,5
         WP(J,I)=0.5*(W(J+1,I)+W(J,I))+DW(J,I)  ! BASE+INCREM
   40
   PREDICTION DATA COMPLETED, CONTINUE FOR CORRECTION
   CALL UOFW(WP,RH,NGM1)
   CALL DGCOF(ALP,NGM1)
   CALL TAUOFW(MUEF,DR,RH,NGM1)
   CALL JET(BA,RHO,DR,RH,V,VJT,NG)
   CALL FSOFW(WP,BA,RH,NGM1)
C FORWARD CORRECTION
   CALL DWPEDE(DW,P(2), DR,DT,R(2),NGM2)
   P(1)=PZERO
   DO 50 J=1,NGM2
P(J+1) = P(I) + 0.25*(RDP(J) + RDP(J+1)) + 0.5*P(J+1)/R(J)

DO 50 I=1,5

WN(J+1) = 0.25*(W(J+1,1) + W(J,1)) + 0.5*(W(J+1,1) + DW(J,1))

P(NG) = P(NGM1)

2ND STE. (PREDICTION+CORRECTION) COMPLETED

IF(N.T. .EQ. 1) GOTO 10

C Estimation of initial P completed, return to the initial condition

C and start to advance the program in time.

DATA W AT THE NEW TIME STEP COMPLETED

C IMPOSED B.C.

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

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

ALP(1,1) = 0.5*(WN(1,5) + W(1,5)) - DW(1,5)/DT/DR/RH(1)

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

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

DO 200 I = 1,5

WN(I,1) = 0.

DO 209 I = 1,5

WN(I,2) = 0.

DO 209 I = 1,5

WN(I,3) = 0.

DO 210 I = 1,5

WN(I,4) = 0.

DO 215 I = 1,5

WN(I,5) = 0.

C NO RADIAL VEL.

C NON-SLIP AT WALL

ENDIF

C ARTIFICIAL TAMING

DO 220 I=1,5

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

W(I,2) = (1 - DAMP)*W(I,2) + DAMP*W(I,3)

W(I,3) = (1 - DAMP)*W(I,3) + DAMP*W(I,4)

W(I,4) = (1 - DAMP)*W(I,4) + DAMP*W(I,5)

W(I,5) = (1 - DAMP)*W(I,5)

C SOLUTION FOR THIS TIME STEP COMPLETED

C UPDATE TIME AND CONTINUE TO THE NEXT STEP

GOTO 10

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

7 WRITE(LUP,7) NAMR

7 WRITE(LUP,7) 'IOSTAT=',I0

9999 CONTINUE
CALL EXEC(11, JTIME, JTIME(1))
WRITE(1,US,'(3SX,3I4)') JTIME
CLOSE(LUS)
END

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

TO SET-UP THE INITIAL CONDITIONS
INTEGER I,IOS,ITLOG,JIT,J,K,MM
Z,JTIME,IM,LUP,LUS,NG,NGM1
PARAMETER (MM=10)
CHARACTER NAME(*)
REAL=R,ALMT, W,MP,WN,DM,RDP,RH
1 = U,V,ALP,P,R, F,S
4 = RIO, MUEF, U18, NA, ND, DO,GAMMA
7 = DM, BRH, DAMP, DR
RJ1,RJ2,TJ1,TJ2,QJ,VI
X = A2, SEDA, SEMID2, D1, D2, PI, RE, RMIN, RTANK
X = DS, LS, VS, TS, PE
X = ALP10, OMEGA, RPEAK, SJB, VPEAK
X = EIVF(2), VT, MU(2), MU1D2
COMMON
/ALPLMT/ ALMT(2)
/JOETS/ RJ1, RJ2, TJ1, TJ2, QJ, VI
/CONT/ IW, LUP, LUS, NG, NGM1, D<BR, DR
/DATA/ U(MM,2), V(MM,2), ALP(MM,2), P(MM), R(MM)
/PA/GI / RHO(4), MUEF(2), U18(2), NA, ND
/DSIZE/ DO(2), QHMM(2)
/SOURCE/ BR(MM), BRH(MM)

DATA PI/3.141596D0/

****** NOTES: PHASE-1=GAS, PHASE-2=LIQUID******
RHO DENSITY, RHO(1) = RHO(2)
ALMT LIMIT VALUES FOR ALP1, ALM(1)(ALP1(ALM(2)
DAMP NUMERICAL DAMPING FACTOR, NORMALLY = 0.
DS, LS, VS, TS DENSITY, LENGTH, VELOCITY AND TIME SCALES
RE REYNOLDS = U*RTANK*RHO(2)/MU(2)
RTANK TANK RADIUS
IVTX TYPE OF INITIAL FLOW.
(0=gas, 1=Pure Rotation, 2=H.O.3=CIT Vertex)
RPEAK, VPEAK VORTEX PARAMETERS
OMEGA PURE ROTATION. W=OMEGA R
EIVF, MUEF EFFECTIVE VISCOSITY, MUEF=(1+EIVF)*MU
DO, GAMMA DIA. PARAMETERS: D=DO ALP1=GAMMA
RJ1, RJ2, TJ1, TO DEFINE JET SIZE, PUMPING TIME
TJ2, QJ, VI VOLUME FLOW RATE AND INJECTION MEAN SPEED

5 FORMAT(2X,A,2X,A))
7 FORMAT(2X,A,3I5)
8 FORMAT(2X,A,3(FPE12.4))
9 FORMAT(A25,2(FPE15.4))
92 FORMAT(X,7(FPE11.4))
DEFINE THE PARAMETERS FOR THE PROBLEM.

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

WRITE(1,8) 'ENTER RTANK(M) OR DEFAULT ',RTANK
READ(1,9) 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  LS#2/(MU#TS)=LS#VS/MU

WRITE(1,8) 'enter Reynolds no., RE, D.F.=',RE
READ(1,9) RE

LS= 'TANK  LENGTH SCALE (M)
DS=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#2  PRESSURE SCALE

AFTER THIS POINT ALL VARIABLES ARE BASED ON THE CHARA. SCALES
1.0 ALL VARIABLES ARE DIMENSIONLESS

DEN12=1.293D9/1.000D3  D1/D2
MU12=1.710D-5/1.514D-7  MU1/MU2

RHO(2)=RHO(2)/DS
MU(2)=MU(2)/(DEN12#LS#VS)
RHO(1)=DEN12#RHO(2)
MU(1)=MU12#MU(2)

ALMT(1)=8.0818  MIN. OF ALP;
ALMT(2)=8.9999  MAX. OF ALP

D0(1)=1.0-2/LS  GAS DIAMETER at ALP1=1
D0(2)=1.0-2/LS  LIQUID DIAMETER at ALP2=1

GAMMA(1)=2.0-1
GAMMA(2)=2.0-1

EVF(2)=1.0  TURB.*PHASE-DISPERSION EFFECTS

DAMP=.0  NUMERICAL DAMPING FACTOR(a.g =.2)
NA=4.  WEIGHTING EXP. FOR ABM
ND=4.  WEIGHTING EXP. FOR DRAG

WRITE(1,8) 'Enter DENSITY and VISCOITY ratios.'
WRITE(1,8) 'D.F.=',DEN12,MU12
READ(1,9) DEN12,MU12

WRITE(1,8) 'Enter BASE DIAMETERS: D01, D02'
WRITE(1,8) 'D.F.=',D0
READ(1,9) D0

WRITE(1,8) 'Enter SIZE EXPONENT: GAMMA1,GAMMA2'
WRITE(1,8) '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.',
WRITE(1,8) 'D.F.=',ALMT
READ(1,*) ALMT
WRITE(1,8) 'Enter eddy viscosity factor. D.F.=',EVF
READ(1,*) EVF
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
V1B(K)=10.*MU(K)
10 MU(K)=10.*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  ; # OF GRID POINTS USED
NGM1=NG-1
DR=(1.-RMIN)/NGM1
C Initial cleaning-up.
DO 15 J=1,NI
DO 15 K=1,6
W(J,K)=0.00
15   W(J,K)=0.00
C MOMENTUM SOURCE, JET CONDITIONS
RI=0.5D-1
RJ2=9.5D-1
VJ=10.  ; TANGENTIAL INJECTION SPEED
T1=8.  ; T2=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'
WRITE(1,8) 'D.F.=',VJ,T1,T2
READ(1,*) VJ,T1,T2
QJ=(RJ2-RJ1)*VJ  ; JET VOLUME FLOW RATE
C TANGENTIAL INJECTION SPEED
DO 20 J=1,NC
R(J)=RMIN+(J-1)*DR
R(H(J))=R(J)+0.5*DR
DR(J)=DELA((R(J)),DR,RJ1,RJ2)/(2.*PI)  ; JET DISTRIBUTION
20   DR(H(J))=DELA((R(H(J))),DR,RJ1,RJ2)/(2.*PI)  ; PER RADIAN
C SETUP (INITIAL CONDITIONS
IVTX=1

22
OMEGA=0.
VPEAK=0.
RPEAK=RMIN

NAME='Simple vortex'
WRITE(1,5) 'Enter data FILE NAME for initial cond., i" any,'
WRITE(1,5) 'DF.',NAMR
READ(1,'(A)') NAMR

IF( NAMR .NE. '' .AND. NAMR .NE. 'Simple vortex') THEN
C INITIAL CONDITION FROM A GIVEN FILE NAMR.
OPEN(99,FILE=NAMR,STATUS='OLD',ERR=299)
DO 25 J=1,NG ! INITIAL VALUES FROM FILE NAMR
:END
CLOSE(99)
ELSE
C TO DEFINE INITIAL CONDITION.
ALP10=2.00-1  ! INITIAL GAS VOL. FRACTION
WRITE(1,8) 'Enter initial -value of alp1. D.F.=',ALP10
READ(1,#) ALP10
WRITE(1,7) 'Enter type of vortex, 0=At rest, 1=pure rotation'
READ(1,#) IVTX
IF(IUTX .GT. 0) THEN
IF(IUTX .GT. 1) THEN
WRITE(1,8) 'Enter PEAK and LOCATION for classic vortex'
READ(1,#) UPEAK,RPEAK
WRITE(1,8) 'DF.',UPEAK,RPEAK
READ(1,#) IVTX
IF(RPEAK .LE. 0.) RPEAK=1. ! SINGULAR AT ZERO
ELSE
WRITE(1,8) 'Enter CIRCULAR SPEED(rad./unit time). D.F. = '
READ(1,8) OMEGA
ENDIF
ENDIF
DO 30 J=1,NG
ALP(J,1)=ALP10
VT=OMEGA*R(J)
IF(IUTX .GT. 0) THEN
RJB=R(J)/RPEAK
IF(IUTX .EQ. 1) THEN
H.O. VORTEX
UT=UT+1.398*UPEAK/RJB*(1.-DEXP(-1.25643*RJB**2))
ELSE
G.I.T. VORTEX
UT=UT+UPEAK*RJB*DEXP((1.-RJB**2)/2.)
ENDIF
ENDIF
DO 30 K=1,2
IF(IW .EQ. 0) THEN
V(J,K)=UT
NO WALL
ELSE
V(J,K)=UT*(1.-R(J))**2.1
BOUNDARY LAYER
ENDIF
30
U(J,K)=0.
NO RADIAL VEL.
ENDIF
DO 40 J=1,NG
FORM W FOR NUMERICAL CAL.
C PRINTOUT PARAMETERS

0488 WRITE(LUS,5) 'INITIAL CONDITION FILE:', NAMR
0489 WRITE(LUS,5) 'DIMENSION UNITS ARE IN MKS**'
0490 WRITE(LUS,9) 'DENSITY SCALE(kg/m3)', DS
0491 WRITE(LUS,9) 'LENGTH SCALE=wall,(m)', LS
0492 WRITE(LUS,9) 'VELOCITY SCALE(m/s)', VS
0493 WRITE(LUS,9) 'TIME SCALE(s)', TB
0494 WRITE(LUS,9) 'PRESSURE SCALE(Pa)', PS
0495 WRITE(LUS,9) 'Reynolds number, Re', RE
0496 WRITE(LUS,9) 'Jet size, RJ1,RJ2', RJ1,RJ2
0497 WRITE(LUS,9) 'Tangential jet, QJ,VJ', QJ,VJ
0498 WRITE(LUS,9) 'Injection time, TJ1,TJ2', TJ1,TJ2

0500 WRITE(LUS,'(/33X,"PHASE-1",8X,"PHASE-2")')
0502 WRITE(LUS,9) 'Density', RHO(1), RHO(2)
0503 WRITE(LUS,9) 'Viscosity', MU
0504 WRITE(LUS,9) 'Eddy viscosity factor', EVF
0505 WRITE(LUS,9) 'Base dia.', DO
0506 WRITE(LUS,9) 'Size exp.', GAMMA
0507 WRITE(LUS,9) 'Phase limits', ALMT(1),1.-ALMT(2)
0508 WRITE(LUS,9)
0509 WRITE(LUS,9)
0510 WRITE(LUS,'(" OTHER CONSTANTS: IV,IVX,NA,ND,DAMP,VPEAK,REPEAK",'
0511 1
0512 WRITE(LUS,'("2J: 0,F5.2")') IV,IVX,NA,ND,DAMP
0513 WRITE(LUS,9) 'VPEAK,REPEAK,OMEGA,DEN1D2,MU1D2
0514 WRITE(LUS,9)
0515 WRITE(LUS,'("NM,F10.2")') NG,RMIN

0516 RETURN

0518 399 WRITE(LUP,5) 'OPEN FILE FAILED ON INPUT FILE:', NAMR
0520 WRITE(LUP,7) 'IODETACH', IOD

0521 STOP 111

0523 END

0524*******************************************************************************

0525 REAL*8 FUNCTION DELA(R,DR,RJ1,RJ2),(0.00423/3.37)

0526 C TO DETERMINE THE EFFECTIVE NOZZLE SIZE AT EACH GRID LOCATION

0527 C THE SIZE IS IN THE FRACTION OF GRID SIZE DR (1.* #DEL/(1*1))

0528 REAL*8 R,DR,RJ1,RJ2, R1,R2

0529 R1=R-0.5*DR

0530 R2=R1+DR

0533 DELA=0.8

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)**

**C**

GET: 1ST DERIVATIVE, USING CENTERED DIFFERENCE

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

**EMA Y,DY**

**C=5.D-1/DX**

**DO 10 J=2,N2-1**

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

**DY(1)=(Y(2)-Y(1))/DX**

**DY(2)=2.0*DY(1)-DY(2)**

**DY(N2)=2.0*DY(N2)-DY(N2-1)**

**RETURN**

**END**

**SUBROUTINE DSDCOEF(ALP,N2)**

**C**

CALCULATE THE DRAG, ADDED MASS AND GENERIZED COEFF.

**INTEGER J,MM,N2**

**PARAMETER (MM=101)**

**REAL#8 ALP(MM,2)**

**C**

REAL#8 C,CPA,CD, RHO,MUEF,VI8,NA,ND

**COMMON**

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

**/DRAG/ RHO(4),MUEF(2),VI8(2),NA,ND**

**DO 50 J=1,K2**

**A1=ALP(J,1)**

**A2=ALP(J,2)**

**A12=A1*A2**

**TO GET DRAG COEFF. AD**

**CALL SIZES(D1,D2,A1)**

**AD=VI8(2)*A1/(A2*D1*D1)**

**IF (A2 .LT. 78) THEN**

**AD2=VI8(1)*A2/(1.-A2/8)*D2)**

**IF (A2 > 200 .LT. AD) THEN**

**WT1=A2**

**WT2=A1**

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

**ENDIF**

**END**

**C**

ADDED MASS COEFF. AA
0599  AA1=AA1+RHO(2)/(AA1+2*(.5+1.5*AA1))
0600  AA2=AA2+RHO(1)/(AA1/(.5+1.5*AA2)+AA2)
0601  WT1=AA2
0602  WT2=AA1
0603  AA=(AA1*WT1+AA2*WT2)/(WT1+WT2)
0604
0605  THE GENERIZED COEFF. CPA, C, AND CD
0606  DB2=AA1+RHO(3)+AA2*(RHO(1)+AA2+RHO(2))
0607  CPA(J,1)=AA1*(AA1+RHO(2)+AA2)+DB2
0608  CPA(J,2)=AA2*(AA1+RHO(1)+AA2)+DB2
0609  C(J,1,2)=AA1/DB2
0610  C(J,2,1)=C(J,1,2)
0611  C(J,1,1)=AA1/RHO(2)/DB2+C(J,1,2)
0612  C(J,2,2)=AA1/RHO(1)/DB2+C(J,2,1)
0613  CD(J,11)=AA1/RHO(2)*AD/DB2
0614  CD(J,2)=AA1/RHO(1)*AD/DB2
0615
0616  50 CONTINUE
0617
0618  END
0619
0620  ************************************************************
0621  $MA /DATA/, /FANDS/, /TAU/, /COEFF/
0622  SUBROUTINE DWDEW(DW,RDP, DR,DT,RR,N2), (860425.1537)
0623
0624  TO GET DW OF THE PDEs
0625
0626  INTEGER J, JP1, K, XP2, MM, N2
0627  PARAMETER (MM=101)
0628  ..* 8  DDW(MM,5), RDP(MM), DR, DT, RR(MM)
0629  EMA DW,RDP,RR
0630
0631  NOTES: COEFF. C =CP+ALP WHEN THIS IS CALLED
0632  REAL B, CPA, CD, U, V, ALP, P, R, F, S
0633  8  TRR, TRA, TAA, RTRR, RTRA
0634  4  NH, MU, V, F1, V18, NA, ND
0635  X  ALP1, ALP2, CP1, CP2, DTD, DW1, DW1T, G1, G2, HDT, WT, WJ1, WJ3, WJ4
0636  37
0637  COMMON
0638  1 /COEFF/ C(MM,2,2), CPA(MM,2), CD(MM,2)
0639  3 /DATA/ U(MM,2), V(MM,2), ALP(MM,2), F(MM), R(MM)
0640  4 /DRAG/ RHO(4), MU, F1, V18(2), NA, ND
0641  6 /FANDS/ F(MM,5), S(MM,5)
0642  8 /TAU/ NH, TRR, TRA, TAA, RTRR, RTRA, MM, 2
0643  45
0644  DTD=DT/DR
0645  HDT=0.5*DT
0646
0647  DO 10 J=1, N2
0648  CHANGE C TO ALP=C
0649  C(J,1,1)=ALP(J,1)*C(J,1,1)
0650  C(J,2,1)=ALP(J,2)*C(J,2,1)
0651  C(J,1,2)=ALP(J,1)*C(J,1,2)
0652  C(J,2,2)=ALP(J,2)*C(J,2,2)
0653  10
0654
0655  DO 20 J=1, N2
0656  JP1=J+1
0657  DW(J,5)=DTD*(-F(JP1,5)+F(J,5)) + HDT*(S(JP1,5)+S(J,5))
0658  20
0659  25 K=1, 2
0660

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

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

KP2 = K+2

D(J,K) = DTDR(-F(JP1,KP2)+F(J,KP2)
+G1*(RTRA(JP1,1)-RTRA(J,1))
+G2*(RTRA(JP1,2)-RTRA(J,2))
+HDT*(S(JP1,KP2)+S(J,KP2))

C  DP FOR PRESSURE CORRECT(ON
CP1 = 0.5*(CPh(J),1=CPPh(JP1,1))
CP2 = 0.5*(CPh(J),2=CPPh(JP1,2))

IF(-D(J,1)=GT. D(J,2)) D(J,1)= -D(J,2) ! DP>0
RDP(J) = (D(J,1)+D(J,2))/(CP1+CP2)

D(J,1)=D(J,1)-CP1*RDP(J)
D(J,2)=-D(J,1)

RDP(J)=RDP(J)/DT

RETURN
END

****************************************************
SMA /CUFF/,/DATA/,/WWW/
SUBROUTINE FNDT(DT,DR,VDR,NG),(860425.1537)

C  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
9 ,W,WP,WN,DW,RDP,PH, 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,5),WN(MM,5),DW(MM,5),RDP(MM),PH(MM)
3 DATA LUP/1/

DUM1=0.

DO 10 J=1,NG

10 DUM2=-CD(J,1)

1 +DABS(U(J,1))/DR
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,2)*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 D W
CALL DWMAX(I,DW,1,NG)  !VECTOR OPERATION
CALL DUM1=DABS(DW(I,1))
CALL DWMAX(J,DW(1,2),1,NG)
DUM2=DABS(DW(J,2))
DT=1.0/DUM1
RETURN
END

*************************************************************************
** EMA /DATA//FANDS//,TAU//,COEFF/                                860425.1537**
SUBROUTINE FSOF(W,BA,RH,N2), (SUBROUTINE)
*************************************************************************
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,MUEF,V18,NA,ND
2     TRA,TR,A,TAAR,TRA,T, C, CPA, CD
X     ALPD(2),RDU, RDV
REAL*8 W(MM,5),BA(MM,2),RR(1)
EMA W,BA,RR

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     /DRAG/ RHO(4),MUEF(2),V18(2,NA,ND
6     /FANDS/ F(MM,5),S(MM,5)
8     /TAU/ TRA(MM,1,2),TAAR(MM,1,2),RTRA(MM,1,2)
*************************************************************************
*   CALL DUMPY(W,1,1,1,F,1,2*MM)
*   CALL DUMPY(W(1,3),1,1,F(1,3),1,2*MM)
*   CALL DUMDV(W,1,F(1,5),1,MM)
*************************************************************************
DO 20 J=1,N2
F(J,5)=W(J,1)
S(J,5)=0.

20   RDU=RR(J)*(U(J,1)-U(J,2))
RDV=RTR(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)*F(J,1)+TA(A(J,1))
1     -C(J,(2)*TA(A(J,2))
20   S(J,KP2)=ALP(J,K)*(-U(J,K)+CD(J,K)*RDV-C(J,K)*F(J,1)+TR(J,1)+C(J,K,2)*(BA(J,2)+TR(J,2)))
RETURN
END

*************************************************************************
** SUBROUTINE JET(BA,RHO,BR,V,WJ,NG)                                **
*************************************************************************
C   INJECTION MOMENTUM SOURCE
C
28
PARAMETER (MM=161)
REAL*8 BA(MM,2),RHO(1),B(1),V(MM,2),U
EMA BA,DR,V
DO 10 J=1,..2
IF( BA(J,1) .GT. 0. ) THEN
  Q=BA(J,1)*U
  BA(J,1)=Q*(VJ-U(J,1))**RHO(1)
  BA(J,2)=Q*(VJ-U(J,2))**RHO(2)
ELSE
  BA(J,1)=0.0
  BA(J,2)=0.0
ENDIF
10 CONTINUE
RETURN
END

******************************************************************************
SUBROUTINE SIZES(D1,D2,ALP1),(960425.1537)
TO DETERMINE THE PARTICLE DIAMETERS
******************************************************************************
REAL*8 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

******************************************************************************
*COMMON /DATA/, /TAU/ *
SUBROUTINE TAUDFW(MU,DR,RR,N2),(960425.1537)
STRESSES AND THEIR DERIVATIVES
******************************************************************************
REAL*8 MU(N2),DR,RR(1)
EMA RR
******************************************************************************
PARAMETER (MM=101)
REAL*8 U,V,ALP,P,R
******************************************************************************
DO 10 J=1,N2
******************************************************************************
DO 20 K=1,2
******************************************************************************
CALL DERIV(U(1,K),TRR(1,K),DR,N2)
CALL DUMPY(ALP(1,K),1,TRR(1,K),1,N2)
CALL DSMDY(2,MU(1),TRR(1,K),1,TRR(1,K),1,N2)
CALL DWDIV(V(1,K),1,RR,1,TAA(1,K),1,N2)
DO 10 J=1,N2
10 TAA(J,K)=V(J,K)/RR(J)
CALL DERIV(TAA(1,K),TRR(1,K),DR,N2)
THM=2.*MU(X)
DO 20 J=1,N2
THM=THM+ALP(J,K)
TRR(J,K)=THM+TRR(J,K)
0839   TRA(J, K) = MU(K) * ALP(J, K) * RR(J) * TRA(J, K)
0840   TAA(J, K) = TAMUU(J, K) / RR(J)
0841   RTRR(J, K) = RR(J) * TRA(J, K)
0842   20 RTRR(J, K) = RR(J) * TRA(J, K)
0843 #V CALL DUMPY(RR, 1, TRA(1, K), 1, TRA(1, K), 1, N2)  # R
0845 #V CALL DUMPY(ALP(1, K), 1, TRA(1, K), 1, TRA(1, K), 1, N2)  # ALP
0847 #V CALL DUMPY(MU(K), TRA(1, K), 1, TRA(1, K), 1, N2)  # MU = TRA
0848 #V CALL DUMDIV(U(1, K), 1, RR, 1, TAA(1, K), 1, N2) U/P
0849 #V CALL DUMPY(ALP(1, K), 1, TAA(1, K), 1, TAA(1, K), 1, N2)  # ALP
0850 #V CALL DUMPY(2. * MU(K), TAA(1, K), 1, TAA(1, K), 1, N2)  # 2 * MU = TAA
0852 #V CALL DUMPY(RR, 1, TRA(1, K), 1, RTRR(1, K), 1, N2)  # R = TRR
0853 #V CALL DUMPY(RR, 1, TRA(1, K), 1, RTRR(1, K), 1, N2)  # R = TRR
0854 50 CONTINUE
0855 RETURN
0856 END
0857
0858 ***************************************************************
0859 $EMA /DATA/
0860 6 SUBROUTINE UDFW(W, RR, N2), (06425.153)
0861 6 C CONVERTS W TO THE INDEPENDENT VARIABLES(U, V, ALP)
0862 6
0863 6 REAL = 8(W(HM, 5), RR(MM))
0865 6 EMA W, RR
0866 6
0867 6 REAL*8 ALMT, U, V, ALP, P, R
0868 6 X, WJ6
0869 6 COMMON
0870 6 ALPLMT/ ALMT(2)
0871 6 /DATA/ U(HM, 2), U(MM, 2), ALP(MM, 2), P(MM), R(MM)
0872 6
0873 #V CALL DUMDIV(W(1, 5), 1, RR, 1, ALP, 1, N2)
0874 6
0875 6 C CHECK VOLUME FRACTION & FLOW DIRECTIONS
0876 6 DO 50 J = 1, N2
0877 6 ALP(J, 1) = W(J, 1) / RR(J)
0878 6 IF(ALP(J, 1) .LE. ALMT(1) .OR. ALP(J, 1) .GE. ALMT(2)) THEN
0879 6 IF(ALP(J, 1) .LE. ALMT(1)) THEN
0880 6 ALP(J, 1) = ALMT(1)
0881 6 W(J, 3) = W(J, 4) * ALMT(1) / (1.0 - ALMT(1))
0882 6 ELSE
0883 6 ALP(J, 1) = ALMT(2)
0884 6 W(J, 4) = W(J, 3) * (1.0 - ALMT(2)) / ALMT(2)
0885 6 ENDIF
0886 6 W(J, 1) = 0.0
0887 6 W(J, 2) = 0.0
0888 6 W(J, 5) = ALP(J, 1) * RR(J)
0889 6 ENDIF
0890 6
0891 6 ALP(J, 2) = 1.0 - ALP(J, 1)
0892 6
0893 IF(W(J, 2) .LT. 0.) THEN  ! PHASE-2 DOES NOT MOVE IN
0895 6 W(J, 2) = 0
0896 6 W(J, 1) = 0.
0897 6 ENDIF
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 DDIV(W(I,1),W(I,5),1,U,1,N2)
0907  *U  CALL DDIV(W(I,3),1,W(I,5),1,V,1,N2)
0908  *U  CALL DWMPY(ALP(I,1,2),1,RR,1,W(I,6),1,N2)
0909  *U  CALL DDIV(W(I,2),1,W(I,6),1,U(1,2),1,N2)
0910  *U  CALL DDIV(W(I,4),1,W(I,5),1,V(1,2),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: DO1,DO2
  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. = 1.448E+02 1.0000E+03
1000,1000
  Enter wall condition.i=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
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
0.01
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

T5153 I=00004 IS ON OR LB USING 00024 RLS ES R=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 Reynolds number, Re 1.0000E+05
0010 Jet size, RJ1,RJ2 8.5000E-01 9.5000E-01
0011 Tangential jet, QJ,VJ 1.0000E-01 1.0000E+00
0012 Injection time, TJ1,TJ2 0.0000E+00 1.0000E+00
0013
0014 PHASE-1 PHASE-2
0015 Density 1.2930E-03 1.0000E+00
0016 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 1.0014E-05
0022
0023 OTHER CONSTANS: IW,IIVTX,NA,ND,DAMP,VPEAK,RPEAK,OMEGA,D1/D2,MU1/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 0029 1 NP= 1 T=0.000E+00 NT= 2 DT=1.000E-06
0030 J ALP1 U1 U2 V1 V2 P
0031 1 .2500 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0032 2 .2500 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0033 ------- OUTPUT IN THE BETWEEN OMITTED -------
0034 0230 97 .2500 -8.685E-10 2.895E-10 3.513E-04 3.474E-04 3.969E-05
0035 0231 98 .2500 -1.775E-10 5.918E-11 1.268E-04 1.250E-04 3.972E-05
0033 100 .2500 -5.225E-12 1.742E-12 1.553E-05 1.531E-05 3.973E-05
0234 101 .2500 0.000E+00 0.000E+00 0.000E+00 0.000E+00 3.973E-05
0033 ------- THE REST OF THE OUTPUT IS OMITTED -------

33
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