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

T. T. Yeh

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.

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
# 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

\[ A_a, A_{ak} \]  \hspace{1cm} Added mass coefficients

\[ A_d, A_{dk} \]  \hspace{1cm} Drag coefficients

\[ \bar{\rho}_k \]  \hspace{1cm} Body force density

\[ C_{1j}, C_{pk}, C_{dk} \]  \hspace{1cm} Generalized coefficients

\[ d_1 \]  \hspace{1cm} Bubble diameter

\[ d_2 \]  \hspace{1cm} Liquid (droplet) diameter

\[ \bar{M}_k \]  \hspace{1cm} Effective interfacial force density

\[ n_a \]  \hspace{1cm} Exponent used for \( w_{ak} \)

\[ n_d \]  \hspace{1cm} Exponent used for \( w_{dk} \)

\[ p \]  \hspace{1cm} Pressure

\[ R \]  \hspace{1cm} Tank radius

\[ Re \]  \hspace{1cm} \( \rho_2 V_s R/\mu_2 \), Reynolds number

\[ R_{j1}, R_{j2} \]  \hspace{1cm} Jet opening, \( R_{j1} < r < R_{j2} \)

\[ R_1 \]  \hspace{1cm} Minimum radius considered in the numerical analysis

\[ r \]  \hspace{1cm} Radial coordinate

\[ t \]  \hspace{1cm} Time

\[ V_j \]  \hspace{1cm} Averaged jet velocity

\[ V_{r1} \]  \hspace{1cm} Gas radial velocity

\[ V_{r2} \]  \hspace{1cm} Liquid radial velocity

\[ V_s \]  \hspace{1cm} Velocity scale

\[ V_{\theta1} \]  \hspace{1cm} Gas tangential velocity
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>$a_1$</td>
<td>Gas volume fraction</td>
</tr>
<tr>
<td>$a_2$</td>
<td>$(1-a_1)$ 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>$\mu_k + \mu_k^t$ total effective viscosity</td>
</tr>
<tr>
<td>$u_k^t$</td>
<td>Turbulence or eddy viscosity</td>
</tr>
<tr>
<td>$u_k$</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>$\langle \rho^2 \rangle$</td>
<td>$a_1 a_2 \rho_1 \rho_2 + A_a (a_1 \rho_1 + a_2 \rho_2)$</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 (Backward Predictor-Forward Corrector Version)</td>
<td>34</td>
</tr>
</tbody>
</table>
| Figure 2.               | Velocity Vector Distributions  
  a) Gas phase, b) Liquid phase  
  The annular region between two dashed circles is the region of injection                                                                                                    | 35   |
| Figure 3.               | Gas Volume Fraction Distributions                                                                                                                                                                            | 36   |
I. Introduction

Mechanical systems have been devised for producing artificial 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 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 (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. \( \frac{\partial}{\partial z} \rho = \frac{\partial}{\partial \theta} \rho = 0 \)) are:

\[
\begin{align*}
\frac{\partial \rho_k}{\partial t} + \frac{\partial \rho_k V_k}{\partial r} &= 0
\end{align*}
\]
\[ \frac{\partial \alpha_{kVrk}}{\partial t} + \frac{\partial \alpha_{kVrk}}{\partial r} - \alpha_{kVrk}^2 - \alpha_{kVpk} \frac{\partial p}{\partial \tau} \]

\[ + \alpha_k^2 \sum_{k=1}^{2} C_{kk} \left( \frac{\partial \alpha_{kVrk}}{\partial \tau} - \alpha_{kVrk} \right) + \alpha_k \sum_{k=1}^{2} C_{dk} (V_{rl} - V_{rk}) \]

\[ \frac{\partial \alpha_{kVrk}}{\partial t} + \frac{\partial \alpha_{kVrk}}{\partial r} + \alpha_{kVrk} \frac{\partial \alpha_{kVrk}}{\partial \tau} - \alpha_k \sum_{k=1}^{2} C_{kk} \left( \frac{\partial \alpha_{kVrk}}{\partial \tau} - \alpha_{kVrk} \right) + \alpha_k \sum_{k=1}^{2} C_{dk} (V_{rl} - V_{rk}) \]

\[ + \alpha_k C_{dk} (V_{rl} - V_{rk}) \]

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

\[ C_{p1} = (a_1 a_2 \rho_2 + A_1) / \langle \rho^2 \rangle \]

\[ C_{p2} = (a_1 a_2 \rho_1 + A_2) / \langle \rho^2 \rangle \]

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

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

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

and

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

The effective stresses are modeled as

\[ \tau_{rrk} = 2 \mu_k^e \frac{\partial V_{rk}}{\partial r} \]
\[ \tau_{\theta rk} = \tau_{\theta kr} = \mu_k^e \frac{\partial}{\partial r} \left( V_{rk} / r \right) \]
\[ \tau_{\theta \theta k} = 2 \mu_k^e \frac{V_{rk}}{r} \]

with

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

and the interfacial forces are modeled in the form of

\[ \bar{F}_1 = A_d (\bar{V}_2 - \bar{V}_1) + A_a \frac{d}{dt} (\bar{V}_2 - \bar{V}_1). \]

\( \bar{F} \) 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 the net momentum gain, \( \frac{\alpha_k \rho_k V_j}{2} (\mathbf{V}_j \mathbf{n} - \mathbf{V}) \) at the nozzle location, where \( V_j \) is the injection speed.

III. Numerical Method

The complete solution of the complicated system of equations can only be obtained through numerical methods. An improved Lax-Wendroff, two-step scheme, (also referred to as MacCormack's method) \([3, 4]\) is adopted for solving this time-dependent problem. This non-centered differencing scheme, using a full step backward prediction and forward correction version, requires no explicit artificial viscosity if a proper stability condition is satisfied. Using this technique for solving fluid flow problems is very efficient and has been in widespread and successful use for some time. It is good both for the time-accurate computation of steady and unsteady flow problems. The general features of the scheme are: 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 \( F, \ P_r, \ gG, \) 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

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

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

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

\[
= \frac{1}{2} (W^p + W^c)
\]

where

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

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, \( W_t^0 \) is the time derivative of \( W \) evaluated at the initial time, and \( W_t^p \) is time derivative of \( W \) evaluated at the predicted time.

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

Numerically, the predicted values are

$$ W_i^p = \frac{1}{2} (W_{i-1/2}^p + W_{i+1/2}^p) $$

where

$$ W_{i-1/2}^p = \frac{1}{2} (W_{i-1}^0 + W_i^0) + \Delta t W_t^0 $$

$$ = \frac{1}{2} (W_{i-1}^0 + W_i^0) + \Delta t \left[ \frac{(g_{i+1}^0 - g_{i-1}^0)}{2} (G_{i+1}^0 - G_{i-1}^0) \right] $$

$$ + \Delta t \left( \frac{S_{i+1}^0 + S_{i-1}^0}{2} + \Delta t \hat{P}_{i-1/2} \right) $$

and the corrected value is evaluated at the predicted time place, that is at $$ W_{i+1/2}^p $$. Thus

$$ W_i^c = W_i^0 + \Delta t W_t^p $$

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

$$ + \frac{\Delta t}{2} \left( S_{i+1/2}^p + S_{i-1/2}^p \right) + \Delta t \hat{P}_{i}^c $$

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 1 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

\[
\Delta t_k = \frac{1}{[C_{dk}] + [V_{rk}] / \Delta r + \frac{2}{\Delta r^2} (a_{1}\mu_1^e c_{k1} + a_{2}\mu_2^e c_{k2})]
\]

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

\[
W_{i}^{1D} = W_{i}^{1} (1-D) + (W_{i-1}^{1} + W_{i+1}^{1} - W_{i}^{1}) D
\]
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}^{\gamma}
- **DS** Density scale $= \rho_2$
- **EVF** $\mu^*/\mu$ effective eddy viscosity factor.
- **GAMMA** Diameter exponent $\gamma$
- **IVTX** Type of simple initial flow: 0 = at rest, 1 = simple rotation, 2 = Hammel-Oseen Vortex, 3 = G.I. Taylor Vortex. This is needed only when there is no data file (NAMR) given for an initial condition.
- **IW** Boundary condition at wall (for tangential component). 0 = free-slip (no-skin friction), 1 = non-slip. Also, when $I_W = 1$, a factor of $(1-r)^{0.1}$ was included on IVTX flow to simulate an initial power law boundary layer.
- **MM** Size of data arrays, $Mf \geq NG$. MM appears in many subroutines.
- **J** $\mu_k^{\gamma_k}$, dynamic viscosity for phase $k$.
- **MLEF** $(1 + EVF) * MU$. Effective viscosity.
- **MU1D2** $\mu_1/\mu_2$, 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, $\omega = \omega_r$, when IVTX > 0.
PS   Pressure scale, $DS \times VS^2$
QJ,VJ Injection flow rate and speed.
RE   Reynolds number, $\rho_2 V_s R / \mu_2$
RJ1,RJ2 Jet opening, RJ1 < r < RJ2
RPEAK Location of the peak speed of the initial vortex, if IVTX > 1
RTANK Tank radius = Length scale LS.
VPEAK Peak speed of the initial vortex, if IVTX > 1.
TJ1,TJ2 Injection time, TJ1 < t < TJ2.
TS   Time scale = LS/VS
VS   Velocity scale.

The format of the data file for the initial condition (if any) is a six column and NG row data file, where NG is the number of grid points. The column sequence is $K$, $\alpha_1(K)$, $V_{V1}(K)$, $V_{V2}(K)$, $V_{L1}(K)$, $V_{L2}(K)$, 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 \((A_a, A_d\text{ and } C_{ij})\). This is the heart of the modeling.

The effective coefficients are modeled as:

\[
\begin{align*}
A_a &= A_{a1} w_1 + A_{a2} w_2 \\
A_d &= A_{d1} d_1 + A_{d2} d_2 \\
A_{a1} &= \frac{a_1 a_2 \beta_2}{(a_1 + 2a_2/(1 + 3a_1))} \\
A_{a2} &= \frac{a_1 a_2 \beta_1}{(a_2 + 2a_1/(1 + 3a_2))} \\
A_{d1} &= 18 \frac{u_1 a_1}{d_1} a_2 \\
A_{d2} &= 18 \frac{u_1 a_2}{(1 - a_2/0.8)^2 d_2^2} \\
w_{a1} &= \frac{a_2^{na}}{(a_1^{na} + a_2^{na})} \\
w_{a2} &= 1 - w_{a1} \\
w_{d1} &= \frac{a_2^{nd}}{(a_1^{nd} + a_2^{nd})} \\
w_{d2} &= 1 - w_{d1}
\end{align*}
\]

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) FNDDT, \&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.
\[ d_k = d_{0k} a_k^\gamma. \]
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

```plaintext
&GLVM T=00004 IS ON CR T4 USING 00126 BLKS R=0080

0001 FTN77
0002 *E3A /DATA/ /WWW/ /C/COEFF/ /SOURCE/ /FANDES/ /TAU/
0003 *FILES 1.2
0004
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 O-C 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 COHESION 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 B50715 TTY GENERALIZED EQUATIONS AND COEFF. C1j
0021 C B51010 TTY IN ANSI 77 STANDARD( WITH A LITTLE EXCEPTION FOR
0022 C TESTING IN HP-1000)
0023 C
0024 C ***** INTERNAL SUBROUTINES *****
0025 C DEFL, DERIV1, DGCOEF, DMPDE, FMDDT, FSFW,
0026 C INIT, JET, SIZES, TAUOFW and UOFW
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),X,K,MM,NPRT,MT
0031 C ,IW,LUP,LUS,NG,NCH1,NCH2
0032 C PARAMETER (MM=101
0033 C REAL=8 BA,MM,2),DDT,DT,DMAX,DMIN,DTPRT,PZERO
0034 C X ,DWI, T,MAX,TPRT,UT,J, VDR(2)
0035 C Y ,RJ1,RJ2,TJ1,TJ2,QJ,VJ
0037 C ,BR, BRH, RHMEUF,VIB,NA,ND
0038 C 5 ,DO,GAMA, D Amp, DR
0039 C 6 ,TRR,TRA,TAH,RTR,RTRA, C,CPA,CD
0040 C
0041 COMMON
0042 Y /JETS/ RJ1,RJ2,TJ1,TJ2,QJ,VJ
0043 Z /CONT/ IW,LUP,LUS,NG,NCH1,DAMP,DR
0044 1 /ALPLMT/ ALM(2)
0045 2 /C/COEFF/ C(MM,2,2),CPA(MM,2),CPA(MM,2)
0046 3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0047 4 /DRA/G1/ RHO(4),MEUF(2),VIB(2),NA,ND
0048 6 /FANDES/ F(MM,5),S(MM,5)
0049 7 /SOURCE/ BR(MM),BRH(MM)
0050 8 /TAU/ TRR(MM,2),TRA(MM,2),TAA(MM,2),RTRR(MM,2),RTRA(MM,2)
0051 9 /WWW/ U(MM,3),WP(MM,3),WN(MM,3),DWM,5,DRP(MM),RH(MM)
0052 C
0053 EQUIVALENCE (IW,BA)
0054 C
0055 C ***** RHO(1) (RHO(2) i.e. PHASE=1=GAS, PHASE=2=LIQUID) *****
0056 C DTPRT TIME STEP FOR PRINTOUT(AND PLOT)
0057 C
0058 C
```

LUP=1   ! LU FOR PRINTING DEBUG DATA(=1 TERMINAL)
LUS=6   ! LU FOR STORING DATA(=6 PRINTER)

0060 FORMAT(2X,A,3(IPE12,4))

0062 WRITE(1,7) 'Enter lu for saving data. D.F.=',LUS
0063 READ(1,7) LUS

0066 C   ! KEEP JOB TIME FOR FUTURE REFERENCE
0067 CALL EXEC(TIME,JTIME(1))

0070 IF(LUS .NE. 1 .AND. LUS .NE. 6) THEN
0071 C   ! *********** Define a file name for string output ***********
0072 WRITE(1,'(2A)') 'Enter FILE NAME for saving data.'
0073 READ(1,'(A)') NAME
0074 LUS=99
0075 OPEN(LUS,FILE=NAME,IODATA=10,STATUS='NEW',ERR=999)
0076 ENDIF

0077 WRITE(1,'(A)') 'Enter NOTES(73 CHAR.) for the job'
0078 READ(1,'(A)') NOTES
0079 WRITE(LUS,'(A41)') JTIME

0080 C   ! To set-up the initial condition.
0081 CALL INIT

0084 NCM2=NC-2
0085 VDR(1)=2.0*MUEF(1)/DR**2 ! For determining time step
0087 VDR(2)=2.0*MUEF(2)/DR**2
0089 T=0.                         ! INITIAL TIME
0090 TMAX=5.
0091 DTPT=0.2
0093 WRITE(1,8) 'Enter INITIAL and FINAL TIMES. D.F.=',T,TMAX
0094 READ(1,8) T,TMAX
0095 WRITE(1,8) 'Enter TIME STEP for output. D.F.=',DTPT
0096 READ(1,8) DTPT
0097 DTMN=1.0D-6                  ! SET MINIMUM TIME STEP
0098 DTMX=DTPT

0101 TPRNT=1
0102 NPT=0
0103 PZERO=0.00
0104 DT=DTMIN

0105 NT=NT+1
0107 CALL UOFW(U,R,4G)

0108 C   ! PRINT OUT AT SELECTED TIME
0110 IF(T .GE. TPRNT OR. T .GT. TMAX) THEN
0111 IF(NT .GT. 1) THEN
0113 NPT=NPT+1
0114 TPRNT=TPRNT+DTPT*(1.+DNINT((T-TPRNT)/DTPT))
0115 WRITE(1H1,2(A5,I4,A5,1PF .3))
0116 WRITE(LUS,21) 'NP=',NPT,'T=',T,'NT=',NT,'DT=',DT
0117 ENDIF
0118 WRITE(LUS,'(A5,A6,5A11)') 'J', 'ALP1', 'U1', 'U2', 'U1', 'V2', 'P'

17
0119      DO 30 J=1,NG
0120  30     WRITE(LUS,'(I3,F6.4,5(IPE1.3))')
0121      J,ALP(J),U(J,1),U(J,2),V(J),P(J)
0122      ENDIF
0123      ENDF
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 DIFFERENCES ON TAU
0129      C FIRST: TO GET THE SPECIAL VARIABLES AND THEIR SPACIAL DERIVATIVES
0130      C CALL DGCOEF(ALP,NG)     ! DRAG AND GENERALIZED COEFF.
0131      C CALL TAUOFW(MUEP,DR,R,NG) ! STRESS
0132      VJT=VT
0133      IF( T .LT. TJ1 OR T .GT. TJ2) VJT=0 ! NO INJECTION
0134      CALL JET(BA,RHO,DR,V,VJT,NG) ! MOMENTUM SOURCE
0135      C CALL FSOFW(w,BA,R,NG)     ! CONVEXTIVE -F AND SOURCE-S
0136      C DETERMINE THE TIME-STEP SIZE
0137      IF( NT .GT. 1) THEN
0138      IF(DT .GE. TMIN) THEN
0139      DT=DT+5.*DTMIN/10.**NT ! INITIAL INPLUSE TREATMENT
0140      I=ILOG10(DT/TMIN)**I
0141      DT=MIN(DT/DDT+0.001)**DDT
0142      IF( DT .LT. TMAX) DT=TMAX
0143      ELSE
0144      WRITE(LUP,'(5X,A,1PE13.3)')
0145      'STOP DUE TO TOO SMALL TIME STEP. DT=',DT
0146      GOTO 9999
0147      ENDIF
0148      ENDIF
0149      C BACKWARD PREDICTOR
0150      C CALL DWPE(DW,DRP,DR,DT,RH,NGM1)    ! INCREMENT
0151      DO 40 J=1,NGM1
0152      DO 40 I=1,5
0153      40 WP(J,I)=0.5*(W(J,I)+W(J+1,I))+DW(J,I) ! BASE+INCREMENT
0154      C PREDICTION DATA COMPLETED, CONTINUE FOR CORRECTION
0155      C CALL UOFW(WP,RH,NGM1)
0156      C CALL DGCOEF(ALP,NGM1)
0157      C CALL TAUOFW(MUEP,DR,RH,NGM1)
0158      C CALL JET(BA,RHO,DRH,V,VJT,NG)
0159      C CALL FSOFW(wp,ba,drh,ngm1)
0160      C FORWARD CORRECTION
0161      C CALL DWPE(DW,P(2),DR,DT,R(2),NGM2)
0162      P(1)=PZERO
0163      DO 50 J=1,NGM2
0164      WRITE(LUP,'(5X,A,1PE13.3)')
0165      'TIME STEP: ',DT
0179 \( P(J+1) = P(J) + (0.25)(RDP(J) + RDP(J+1)) + 0.5P(J+1)/R(J) \)
0180 DO 50 I = 1, 5
0181 50 \( WN(J+1) = 0.25(WP(J+1,I) + WP(J,I)) + 0.5(W(J+1,I) + DW(J,I)) \)
0182 \( P(NG) = P(NGM) \)
0183 0184 C 2ND STEP (PREDICTION+CORRECTION) COMPLETED
0185 IF (NT .EQ. 1) GOTO 10
0186 C Estimation of initial condition completed, return to the initial condition
0187 C and start to advance the program in time.
0188 C
0189 0190 **********************************************
0191 C DATA W AT THE NEW TIME STEP COMPLETED
0192 0193 C IMPOSED B.C. #6.4
0194 DWI = 0.5*(WN(2,1) + WN(2,1)) - 0.
0195 ALP(1,1) = (0.5*(W(1,5) + W(2,5)) - DWI*DT/DR)/RH(1)
0196 IF (ALP(1,1) .LT. ALMT(1)) ALP(1,1) = ALMT(1)
0197 IF (ALP(1,1) .GT. ALMT(2)) ALP(1,1) = ALMT(2)
0198 \( WN(1,5) = ALP(1,1) * R(1) \)
0199 \( DWI = 0. - 0.5*(WN(NGM1,1) + WN(NGM1,1)) \)
0200 ALP(NG,1) = (0.5*(WN(NGM1,1) + WN(NG,1)) - DWI*DT/DR)/RH(NGM1)
0201 IF (ALP(NG,1) .LT. ALMT(1)) ALP(NG,1) = ALMT(1)
0202 IF (ALP(NG,1) .GT. ALMT(2)) ALP(NG,1) = ALMT(2)
0203 \( WN(NG,3) = ALP(NG,1) * R(NG) \)
0204 \( WN(NG,3) = 0. \) 1 NO RADIAL VEL.
0205 \( WN(NG,2) = 0. \)
0206 \( WN(NG,1) = 0. \)
0207 \( WN(NG,1) = 0. \)
0208 \( WN(NG,1) = 0. \)
0209 \( WN(NG,2) = 0. \)
0210 \( WN(NG,3) = 0. \)
0211 \( WN(NG,4) = 0. \)
0212 IF (WJ .EQ. 0) THEN
0213 \( WN(NG,3) = 0. \) NON-SLIP AT WALL
0214 \( WN(NG,4) = 0. \)
0215 ENDIF
0216 C ARTIFICIAL DAMPING
0217 DO 60 I = 1, 5
0218 W(1,1) = (1. - DAMP)*W(1,1) + DAMP*W(2,1)
0219 W(NG,1) = (1. - DAMP)*W(NG,1) + DAMP*W(NG,1)
0220 DO 60 J = 2, NGM1
0221 60 W(J,1) = (1. - DAMP)*W(J,1) + DAMP*W(J-1,1) + WN(J,1) - WN(J,1)
0222 C SOLUTION FOR THIS TIME STEP COMPLETED
0223 0224 0225 C UPDATE TIME AND CONTINUE TO THE NEXT STEP
0226 90 T = T + DT
0227 GOTO 10
0228 0229 0230 999 WRITE(LUP,7) 'OPEN_failed on FILE:'
0231 WRITE(LUP,7) NWHR
0232 WRITE(LUP,7) 'IOSTAT=', IOS
0233 CONTINUE
CALL EXEC(11, JTIME, JTIME(1))
WRITE(1, US, '(JSX, SI)') JTIME
CLOSE(US)
END

C=====================================================================================================
C SUBROUTINE INIT (B60423.1537)
C TO SET-UP THE INITIAL CONDITIONS
INTEGER I, IOS, ITLOG, IVX, J, K, MM
REAL ALMT, W, WP, WM, DW, DWP, RH
1 ,U, V, ALP, P, R, F, S
RHO, MUEF, VIB, MA, ND, D0, GAMMA
7 ,BR, D, DWP, DR
Y ,RJ1, R2, T1, T2, QJ, VJ
9 ,R1, R2, T1, T2, VJ
X ,R1, R2, DEM1D2, D1, D2, PI, RE, RMIN, RTANK
X ,DS, LS, VS, TS, PE
10 X ,ALP10, OMEGA, RPEAK, SJ, VPEAK
X ,EUF(1), UT, MU(2), MU1D2
COMMON
1 /ALPLMT/ ALMT(2)
Y /JETS/ R1, R2, T1, T2, QJ, VJ
Z /CONTPL/ I, WP, WM, DW, DWP, DR
3 /DATA/ W(MM, 2), V(MM, 2), ALP(MM, 2), P(MM), R(MM)
4 /PAG1/ RHO(4), MUEF(2), VIB(2), MA, ND
5 /DSIZE/ D0(2), CWWA(2)
7 /SOURCE/ BR(MM), BRN(MM)
9 /WWW/ W(MM, 5), WP(MM, 5), WM(MM, 5), DW(MM, 5), DWP(MM), R(MM)
DATA PI/3.14159265/

C======= NOTES: PHASE=1=GAS, PHASE=2=LIQUID =======
C RHO DENSITY, RHO(1) = RHO(2)
C ALMT LIMIT VALUES FOR ALP1, ALM(1),ALP1(2)
C DAMP NUMERICAL DAMPING FACTOR, NORMALLY = 0.
C DS, LS, VS, TS DENSITY, LENGTH, VELOCITY AND TIME SCALES
C RE REYNOLDS &s&&, RTANK=RHO(2)/MU(2)
C RTANK TANK RADIUS
C IVX TYPE OF INITIAL FLOW.
C EFV, MUEF VORTEX PARAMETERS
C OMEGA PURE ROTATION. V=OMEGA*
C EVF, MUEF EFFECTIVE VISCOSITY, MUEF=1*EVF*MU
C DO, GAMMA DIA. PARAMETERS: D=DO*ALP**GAMMA
C RTN1, RTN2, T1, T2, TO DEFINE JET SIZE, PUMPING TIME.
C T2, QJ, VJ VOLUME FLOW RATE AND INJECTION MEAN SPEED
C
C FORMAT(2X, A, 2(X, A))
C FORMAT(2X, A, 3(J))
C FORMAT(2X, A, 3(P,E))
C FORMAT(A25, 2(P,E))
C FORMAT(X, 7(1P,E))
DEFINE THE PARAMETERS FOR THE PROBLEM.

RTANK=1.00
RHO(2)=1.00000D+3
MU(2)=1.00000D-3

WRITE(1,6) 'ENTER RTANK(M) OR DEFAULT ',RTANK
READ(1,7) 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 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=1.00D5
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 CHAR. SCALES
i.e. ALL VARIABLES ARE DIMENSIONLESS

DEN12D=1.29300/1.00000D3
MU1D2=1.71000/1.51400

RHO(2)=RHO(2)/DS
MU(2)=MU(2)/(DEN12D*VS)
RHO(1)=DEN12D*RHO(2)
MU(1)=MU1D2*MU(2)

ALHT(1)=0.00100
MIN. OF ALP
ALHT(2)=0.00999
MAX. OF ALP
DO(1)=1.00D-2/LS
GAS DIAMETER at ALP1=1
DO(2)=1.00D-2/LS
LIQUID DIAMETER at ALP2=1
GAMMA(1)=2.0
TURB. PHASE-DISPERSION EFFECTS
GAMMA(2)=2.0
MODEL
EVF(1)=DEN12D/MU1D2*EVF(2)
NUMERICAL DAMPING FACTOR (e.g. =.2)
NAM=4
WEIGHTING EXP. FOR AXM
ND=4
WEIGHTING EXP. FOR DRAG

WRITE(1,8) 'Enter DENSITY and VISCOITY ralse.'
WRITE(1,8) 'D.F.=',DEN12D,MU1D2
READ(1,9) DEN12D,MU1D2

WRITE(1,8) 'Enter BASE PARAMETERS: DO1,DO2'
WRITE(1,8) 'D.F.=',DO
READ(1,9) DO

WRITE(1,8) 'Enter SIZE EXPONENT: GAMMA1,GAMMA2'
WRITE(1,8) 'D.F.=',GAMMA

1
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
READ(1,*) IU
WRITE(1,7) 'Enter wall condition,1=nonslip,0=slip. D.F=\',IU
READ(1,*) IU
WRITE(1,8) 'Enter numerical damping factor. D.F=\',DAMP
READ(1,*) DAMP
DO 10 K=1,2
V10(K)=10.*MU(K)
10 MU=E(K)=MU(K)*(1.+EVF(K))  1 EFFECTIVE VISCOSITY FOR STRESS
RHO(3)=RHO(1)+RHO(2)
RHO(4)=RHO(1)-RHO(2)
RMIN=0.1 1 MINIMUM FLOW RADIUS IN THE TANK
NG=101 1 OF GRID POINTS USED
NGM1=NG-1
DR=1.-RMIN/NGM1
C Initial cleaning-up.
DO 15 J=1,NH
DO 15 K=1,6
W(J,K)=0.00
15 W(J,K)=0.00
C MOMENTUM SOURCE, JET CONDITIONS
R10=0.5D-1
R20=9.5D-1
V11=10. 1 TANGENTIAL INJECTION SPEED
T11=0.
T21=10.
WRITE(1,8) 'Enter JET SIZE defined by R10,R20. D.F=\',R10,R20
READ(1,*) R10,R20
WRITE(1,8) 'Enter INJECTION SPEED AND TIME RANGE, V11,T11,T21'
WRITE(1,8) 'D.F=\',V11,T11,T21
READ(1,*) V11,T11,T21
QJ=(R10-R20)*V11 1 JET VOLUME FLOW RATE
DO 20 J=1,NG
R(J)=RMIN+(J-1)*DR
R(J)=R(J)+0.5D0
DR(J)=DELAM(R(J)),DR,J,RJ1,RJ2)/(2.*PI) 1 JET DISTRIBUTION
20 DR(J)=DELAM(R(J)),DR,J,RJ1,RJ2)/(2.*PI) 1 PER RADIUS
C SETUP INITIAL CONDITIONS
IVTX=0
OMEGA=0.
VPEAK=0.
RPEAK=RMIN.

NAME='Simple vortex'
WRITE(1,5) 'Enter data FILE NAME for initial cond., i^t any';
WRITE(1,5) 'D.F.=',NAMR
READ(1,'(A)') NAMR

IF(NAMR .NE. ',') .AND. NAMR .NE. 'Simple vortex') THEN
   C
   INITIAL CONDITION FROM A GIVEN FILE NAMR.
   OPEN(59,FILE=NAMR,STATUS='OLD',ERR=299)
   DO 25 J=1,NG
      ! INITIAL VALUES FROM FILE NAMR
      25 CONTINUE(99,...,ALP(J,1),U(J,1),U(J,2),V(J,1),V(J,2)
   CLOSE(99)
   ELSE
   TO DEFINE INITIAL CONDITION.
   ALP10=0.001
   IF(IVTX .GT. 0.) THEN
      IF(IVTX .GT. 1.) THEN
         WRITE(1,8) 'Enter type of vortex: 0=At rest,1=pure rotation'
         READ(1,8) ALP10
      ENDIF
      WRITE(1,7) 'Enter spin rate for alp1. D.F.=',ALP10
      READ(1,7) IVTX
      WRITE(1,7) 'Enter initial value of alp1. D.F.=',IVTX
      READ(1,7) IF(IVTX .GT. 0.) THEN
         IF(IVTX .GT. 1.) THEN
            WRITE(1,8) 'Enter peak value and LOCATION for classic vortex'
            READ(1,8) 'D.F.=',VPEAK,PPEAK
         ENDIF
         WRITE(1,8) 'D.F.=',VPEAK,PPEAK
         READ(1,8) IF(PPEAK .LE. 0.) PPEAK=1.
         ELSE
            WRITE(1,8) 'Enter CIRCULAR SPEED(rad./unit time). D.F.=',OMEGA
            READ(i,=) OMEGA
         ENDIF
         ENDIF
         ENDIF
         DO 30 J=1,NG
            ALP(J,1)=ALP10
            VT=OMEGA*R(J)
            IF(IVTX .GT. 0.) THEN
               IF(IVTX .EQ. 1.) THEN
                  H.O. VORTEX
                  UT=UT+1.3*VPEAK/RJB*(1.-DEXP(-1.256*3*RJB**2))
               ELSE
                  G.I.T. VORTEX
                  UT=UT+VPEAK*RJB*DEXP((1.-RJB**2)/2.)
               ENDIF
               ENDIF
               ENDIF
               DO 30 K=1,2
                  IF(IW .EQ. 0.) THEN
                     V(J,K)=UT
                  ELSE
                     V(J,K)=UT*(1.-R(J)**=0.1
                  ENDIF
                  ENDIF
                  ENDIF
                  DO 40 J=1,NG
                     FORM W FOR NUMERICAL CAL.
ALP(J,2)=1.-ALP(J,1)

P(J)=0.D0

W(I,J)=R(J)=ALP(J,1)

DO 40 K=1,2

W(J,K)=ALP(J,K)=U(J,K)=R(J)

C PRINTOUT PARAMETERS

WRITE(LUS,5) 'INITIAL CONDITION FILE:',,NAMR

WRITE(LUS,5) '**DIMENSION UNITS ARE IN MKS**'

WRITE(LUS,9) 'DENSITY SCALE(kg/m3)',DS

WRITE(LUS,9) 'LENGTH SCALE=RTANK,(m)',LS

WRITE(LUS,9) 'VELOCITY SCALE(m/sec)',VS

WRITE(LUS,9) 'TIME SCALE(s)',TS

WRITE(LUS,9) 'PRESSURE SCALE(Pa)',PS

WRITE(LUS,9) 'Reynolds number, Re',RE

WRITE(LUS,9) 'Jet size, RJ1,RJ2',RJ1,RJ2

WRITE(LUS,9) 'Tangential jet, QJ,VJ',QJ,VJ

WRITE(LUS,9) 'Injection time, TI1,TI2',TI1,TI2

WRITE(LUS,'(/3X,"PHASE-1",8X,"PHASE-2")')

WRITE(LUS,9) 'Density',RHO(1),RHO(2)

WRITE(LUS,9) 'Viscosity',MU

WRITE(LUS,9) 'Eddy viscosity factor',EUF

WRITE(LUS,9) 'Base dia.',DO

WRITE(LUS,9) 'Size exp.',GAMMA

WRITE(LUS,9) 'Phase limits',ALMT(1),1.-ALMT(2)

WRITE(LUS,9)

WRITE(LUS,9)

WRITE(LUS,'(" OTHER CONSTANTS: IW,IUTX,NA,ND,DAMP,PPEAK,PPEAK",'

WRITE(LUS,1) ' "OMEGA,DI/D2,MIU/MI2")')

WRITE(LUS,'(/I3:8,F5.2)') IW,IUTX,NA,ND,DAMP

WRITE(LUS,9) 'PPEAK,PPEAK,OMEGA,DI/D2,MIU

WRITE(LUS,9)

WRITE(LUS,'(/110,F10.2)') NG,RMIN

RETURN

STOP

END

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

REAL# FUNCTION DELA(R,DR,RJ1,RJ2),(B44444.1537)

C TO DETERMINE THE EFFECTIVE NOZZLE SIZE AT EACH GRID LOCATE1

C THE SIZE IS IN THE FRACTION OF GRID SIZE DR (1.0 = (DELA(1))

REAL# R,DR,RJ1,RJ2, R1,R2

R1=R-0.5*DR

R2=R1+DR

DELA=0.D0

IF(R1 .GE. RJ2) OR. R2 LE. RJ1) RETURN

IF(R1 .LT. RJ1) R1=RJ1

IF(R2 .GT. RJ2) R2=RJ2

DELA=(R2-R1)/DR

RETURN
0539 END
0540
0541 *********************************************************************
0542 SUBROUTINE DERIV(Y,DY,DX,N2),(860.25.1537)
0543 C GET: 1ST DERIVATIVE, USING CENTERED DIFFERENCE
0544 REAL#8 DX,Y(1),DY(1),C
0545 END Y,DY
0546
0547 C=5.D-1/DX
0548 DO 10 J=2,N2-1
0549 10 DY(J)=C*(Y(J+1)-Y(J-1))
0550 CALL DWSUB(Y(3),1,Y,1,DY(2),1,N2-2)
0551 DO 59 J=2,N2
0552 DY(J)=(Y(2)-Y(1))/DX ! BASED ON 3-END PTS
0553 DO 54 J=N2-1,N2
0554 DY(J)=.5*DY(1)-DY(2)
0555 DO 55 J=2,N2-1
0556 55 CALL DWSMY(5.D-1/DX,Y,1,DY,1,DY,1,N2)
0557 RETURN
0558 STOP
0559 END
0560
0561 *********************************************************************
0562 SUBROUTINE DDCOEF(ALP,N2),(860.25.1537)
0563 C CALCULATE THE DRAG, ADDED MASS AND GENERIZED COEFF.
0564 END ALP
0565
0566 INTEGER J,MM,N2
0567 REAL#8 ALP(MM,2)
0568 EMA ALP
0569
0570 COMMON
0571 1 /COEFF/ (MM,2),CPA(MM,2),CD(MM,2)
0572 4 /DRAG/ (MM,4),RHOUE(2),U18(2),NA,ND
0573 ! COMMON
0574 X ,AA,AA1,AA2,AD,AD1,AD2,AL1,AL2,A12,DA12,DA2,OH,DI,DI2,WT1,WT2,X
0575 DO 50 J=1,1
0576 50 A1=ALP(J,1)
0577 A2=ALP(J,2)
0578 A12=A1*A2
0579
0580 C TO GET DRAG COEFF. AD
0581 CALL SIZEBS(D1,D2,A1)
0582
0583 A18=U18(2)*A1/(A2*D1*D1) ! =AD1 IF A2=.78
0584 IF(AD1=.LT. ,78) THEN
0585 AD2=U18(1)*A2/(1.-A2/.8)*D2**2
0586 IF(AD2=.LT. AD) THEN
0587 WT1=A2*ND
0589 WT2=A1*ND
0590 AD=(AD+WT1+AD2+WT2)/(WT1+WT2)
0591 ENDIF
0592 ENDIF
0593 ENDS
0594
0595 C ADDED MCF COEFF. AA
0596
0597
AA1=A12*RHO(2)/(A1+A2/(-.5+1.5*A1))
AA2=A1*RHO(1)/(A1/(-.5+1.5*A2)+A2)
W1=AA1
W2=AA2
AA=(AA1*W1+AA2*W2)/(W1+W2)

C THE GENERALIZED COEFF. CPA, C, AND CD
DB2=A12*RHO(3)+AA*(RHO(1)+A1*RHO(2)*A2)
CPA(J,1)=A1=(A12*RHO(2)+AA)/DB2
CPA(J,2)=A2=(A12*RHO(1)+AA)/DB2
C(J,1,2)=A6/DB2
C(J,2,1)=C(J,1,2)
C(J,1,1)=A2*RHO(2)/DB2+C(J,1,2)
C(J,2,2)=A1*RHO(1)/DB2+C(J,2,1)
CD(J,1)=A2*RHO(2)+A6/DB2
CD(J,2)=A1*RHO(1)+AD/DB2
50 CONTINUE
RETURN
END

*********************************************
* EMA /DATA/*FANDS/*TAU/*COEFF*/
* SUBROUTINE DWPDE(DW, RDP, DR, DT, RR, N2), (860425.1537)*
C TO GET DW OF THE PDEs
C INTEGER J, JP1, K, XP2, MM, N2
C PARAMETER (MM=101)
C * real = 0, DB(MM,5), RDP(MM), DR, DT, RR(MM)
C * EMA DW, RDP, RR
C NOTES: COEFF. C =ALP WHEN THIS IS CALLED
C real = B, CPA, CD, U, V, ALP, P, R, F, S
C 0, TRR, TRA, TAA, RTRR, RTRA
C 4, RH0, MUEF, V18, NA, ND
C X, ALP1, ALP2, CP1, CP2, DTRD, DW1, DW1DT, G1, G2, HDT, VT, W1, W3, W4
C COMMON
C /COEFF/ C(MM,2,2), CPA(MM,2), CD(MM,2)
C /DATA/ U(MM,2), V(MM,2), ALP(MM,2), P(MM), R(MM)
C /DRAG1/ RH0(4), MUEF(2), V18(2), NA, ND
C /FANDS/ F(MM,5), S(MM,5)
C /TAU/ RH(MM,2), TRA(MM,2), TAA(MM,2), RTRR(MM,2), RTTRR(MM,2)
C DTRD=DT/DR
C HDT=0.5*DT
C DO 10 J=1,N2+1
C CHANGE C TO ALP=C
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 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)
N2
C DO 20 J=1,N2
C JP1=J+1
C DW(J,5)=DTRD*(-F(JP1,5)+F(J,5))*HDT*(S(JP1,5)+S(J,5))
C DO 25 K=1,2
G1 = S*(CJP1,K,1)+C(J,K,1))
G2 = S*(CJP1,K,2)+C(J,K,2))

if (D(J,K)>=CJP1,K,1) +G1*(RTTR(JP1,1)-RTTR(J,1))
+G2*(RTTR(JP1,2)-RTTR(J,2)))
HDT = S(JP1,K)*S(J,K))

KP2 = K+2

if (D(J,K)>=CJP1,K,1) +G1*(RTRA(JP1,1)-RTRA(J,1))
+G2*(RTRA(JP1,2)-RTRA(J,2)))

HDT = S(JP1,K)*S(J,K))

C

DP FOR PRESSURE CORRECT(ON

CP1 = 0.5*C(CPA(J,1)) = CPA(JP1,1))
CP2 = 0.5*C(CPA(J,2)) = CPA(JP1,2))

if (D(J,1)> GT. D(J,2)) = D(J,1) = D(J,1) ! 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)

20 RDP(J)=RDP(J)/DT

RETURN

END

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

SHEMA /COEFF, /DATA, /WWW/

SUBROUTINE FNDTT(DT,DR,UDR,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,WM,DW,RDP,RH, U,V,ALP,P,R

X ,DUM1,DUM2

COMMON

/Vowe/ MM,2), CMAA(MM,2),CD(MM,2)
3 /DATA/ (MM,2),U(MM,2),ALP(MM,2),P(MM,2),R(MM)
4 /DRAG/ (MM,2),MUEF(MM,2),V18(MM,2,2),NA,ND
9 /WWW/ (MM,2),W(MM,5),WP(MM,5),WM(MM,5),DW(MM,5),RDP(MM,2),RH(MM)

DATA LUP/1/

DUM1=0.

DO 10 J=1,NG

10 DUM2=CD(J,1)

1+SAB*(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. DUMZ) DUM1=DUM2

DUM2=CD(J,2)+SAB*(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 DUM
0719 #V CALL DWMAX(I,DW,1,NG)               ! VECTOR OPERATION
0720 #V DUM1=DABS(DW(I,1))
0721 #V CALL DWMAX(J,DW(1,2),1,NG)
0722 #V DUM2=DABS(DW(J,2))
0723 DT=1.0/DUM1
0724 RETURN
0725 END
0726
0728 ################################################################################
0729 *EQA /DATA/,/FANDS/,/TAU/,/COEFF/
0730 SUBROUTINE FSDOF(W,BA,RM,N2),(86025.1537)
0731 C  
0732 C  CALCULATE THE CONVECTIVE-F AND SOURCE-S TERMS
0733 C  
0734 INTEGER J,K,KP2,L,MM,N2
0735 PARAMETER (MM=101)
0736 REAL*8 U,V,ALP,F,R,F,S,RH0,MUEF,V1B,NA,ND
0737     2 TRA,TRA,TAA,RTRA,RTRA, C,CPA,CD
0738     X ALP=2D(2),RDV
0739 REAL=8 W(MM,5),BA(MM,2),RR(11)
0741 EMA W,BA,RR
0742 C  
0743 COMMON
0744 1 /COEFF/ C(MM,2,2),CPA(MM,2),CD(MM,2)
0745 3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0746 4 /DRA/ RH0(4),MUEF2(2),V1B(2),NA,ND
0747 6 /FANDS/ F(MM,5),S(MM,5)
0748 8 /TAU/ TRA(MM,2),TAA(MM,2),RTRA(MM,2),RTRA(MM,2)
0749 C  
0750 #V CALL DWMPY(W,1,U,1,F,1,2*MM)
0751 #V CALL DWMPY(1,1,1,1,F(1,3),1,2*MM)
0752 #V CALL DWMPY(W,1,F(1,5),1,MM)
0753 C  
0754 DO 20 J=1,N2
0755     F(J,5)=U(J,1)
0756     S(J,5)=0.
0757     RDU=R(2,J)*(U(J,1)-U(J,2))
0758     RDU=R(2,J)*(V(J,1)-V(J,2))
0759 DO 20 K=1,2
0760     KP2=K+2
0761     F(J,KP2)=W(J,KP2)*U(J,K)
0762     F(J,KP2)=W(J,KP2)*W(J,K)
0763     S(J,K)~ALP(J,K)*(U(J,K)~2+CD(J,K)*RDU-C(J,K)*TAJ(J,1)
0764     -C(J,K,2)*TAJ(J,2))  
0765     20 S(J,KP2)=ALP(J,K)~1*U(J,K)~CD(J,K)*RDV+C(J,K,1)*BA(J,1)
0766     1 +TRA(J,1)+C(J,K,2)*(BA(J,2)+TRA(J,2))

0770 RETURN
0771 END
0772
0774 ################################################################################
0775 SUBROUTINE JET(BA,RH0,BA,V,VI,NG)
0776 C  
0777 C  INJECTION MOMENTUM SOURCE
0778 C  

28
PARAMETER (MM=181)

REAL*8 BA(MM,2),RHO(1),BM,1,U(MM,2),VJ

END BA,BR,V

DO 10 J=1,3

IF (BR(J) .GT. 0.) THEN

Q=BR(J)*UJ

BA(J,1)=Q*(VJ-U(J,1))=RHO(1)

BA(J,2)=Q*(VJ-U(J,2))=RHO(2)

ELSE

BA(J,1)=0.00

BA(J,2)=0.00

END

10 CONTINUE

RETURN

END

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

SUBROUTINE SIZES(D1,D2,ALP1),(B60425.1537)

TO DETERMINE THE PARTICLE DIAMETERS

REAL*8 D1,D2,ALP1, DO,GAMMA

COMMON /DSIZE/ DO(2),GAMMA(2)

DO=DO1(1)*ALP1=1*GAMMA(1)

D2=DO(2)*(.00-ALP1)*GAMMA(2)

RETURN

END

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

REAL*8 MU(MM,1),DR,RR(1)

END MU,RR

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

PARAMETER (MM=101)

REAL*8 U,V,ALP,P,R

2 ,TRR,TRA,TAA,TRRR,RTRA

X ,TAMU,THU

COMMON

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

8 /T/RR(MM,2),TRA(MM,2),TAA(MM,2),TRRR(MM,2),RTRA(MM,2)

DO 50 K=1,2

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

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

CALL DWSHY2(MU(K),TRR(1,K),1,TRR(1,K),1,N2)

CALL DWDIV(V(1,K),1,RR,1,TAA(1,K),1,N2)/V/R

DO 10 J=1,N2

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

DMS=2*MU(X)

DO 20 J=1,N2

TAMU=MU=ALP(J,K)

TRR(J,K)=TAMU*TRR(J,K)
0039 TRA(J,K)=MU(K)*ALP(J,K)*RR(J)*TRA(J,K)
0040 TAA(J,K)=TAMU(J,K)/RR(J)
0041 RR(J,K)=RR(J)*TRA(J,K)
0042 20  RR(J,K)=RR(J)*TRA(J,K)
0043     CALL DWMY(RR,1,TRA(1,K),1,TRA(1,K),1,N2)  !R
0044     CALL DWMY(ALP(1,K),1,TRA(1,K),1,TRA(1,K),1,N2)  !ALP
0047     CALL DWMY(MU(K),TRA(1,K),1,TRA(1,K),1,N2)  !MU=TRA
0048     CALL DWDIV(U(1,K),1,RR,1,TAA(1,K),1,N2)  !U/P
0049     CALL DWMY(ALP(1,K),1,TAA(1,K),1,TAA(1,K),1,N2)  !AP
0050     CALL DWDMY(2,MU(K),TAA(1,K),1,TAA(1,K),1,N2)  !2#MU=TAA
0052     CALL DWMY(RR,1,TRA(1,K),1,TRA(1,K),1,N2)  !R=TRA
0053     CALL DWMY(RR,1,TRA(1,K),1,TRA(1,K),1.N2)  !R=TRA
0054   50 CONTINUE
0055     RETURN
0056     END
0057     END
0058 ***********************************************************************
0059  $EMA /DATA/
0060  C SUBROUTINE UD/J(M,K,N2),(862425.1537)
0062  C CONVERTS W TO THE INDEPENDENT VARIABLES(U,V,ALP)
0063  PARAMETER (MM=101)
0064  REAL*8 W(MM,5),RR(MM)
0065  C EMA W ,RR
0066  REAL*8 ALMT, U,V,ALP,P,R
0068  X =WM6
0069  COMMON X /ALM/ALMT/ ALMT(2)
0070  REAL*8 ALM/ALM,ALM(2)
0071  /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0072  C CHECK VOLUME FRACTION & FLOW DIRECTIONS
0073  DO 50 J=1,N2
0074  DO 50 J=1,N2
0075  C CHECK VOLUME FRACTION & FLOW DIRECTIONS
0076  DO 50 J=1,N2
0077  ALP(J,1)=W(J,J)/RR(J)
0078  IF(ALP(J,1) LE. ALMT(1) OR. ALP(J,1) GE. ALMT(2)) THEN
0079  IF(ALP(J,1) LE. ALMT(1)) THEN
0080  IF(ALP(J,1) GE. ALMT(2)) THEN
0081  ALP(J,1)=ALMT(1)
0082  ELSE
0083  ALP(J,1)=ALMT(2)
0084  W(J,4)=W(J,3)*(1.0-ALMT(2))/ALMT(2)
0085  ENDIF
0086  W(J,2)=0.0
0087  W(J,2)=0.0
0088  W(J,3)=ALP(J,1)+RR(J)
0089  ENDIF
0091  ENDIF
0092  ALP(J,2)=1.00-ALP(J,1)
0093  IF( W(J,2) .LT. 0.) THEN  ! PHASE-2 DOES NOT MOVE IN
0094  W(J,2)=0
0095  W(J,1)=0;
0097  ENDIF
0390
0899  U(J,1)*W(J,1)/W(J,5)
0900  V(J,1)*W(J,3)/W(J,5)
0901  WJ6=RR(J)*ALP(J,2)
0902  U(J,2)=W(J,2)/WJ6
0903  V(J,2)=W(J,4)/WJ6
0904  50 CONTINUE
0905
0906  *U CALL DWDIV(W(,1),W(1,5),1,U,1,N2)
0907  *U CALL DWDIV(W(1,3),1,W(1,5),1,V,1,N2)
0908  *U CALL DWMPY(ALP(1,2),1,RR,1,W(1,6),1,N2)
0909  *U CALL DWDIV(W(1,2),1,W(1,6),1,U(1,2),1,N2)
0910  *U CALL DWDIV(W(1,4),1,W(1,5),1,V(1,2),1,N2)
0911
0912  RETURN
0913
0914  END
Exhibit A

A Sample Input

G

GLVM
Enter lu for saving data. D.F. = 6
90
Enter FILE NAME for saving data.
TS15S::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. = 1.1438E+02 1.0000E+03

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

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

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

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

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

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

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

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

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

NP= 1 T=0.0000E+00 NT= 2 DT=1.0000E-06
NP= 2 T=1.2000E-02 NT= 5 DT=3.0000E-03
Exhibit B

A Sample Output

<table>
<thead>
<tr>
<th>TS153</th>
<th>T=00004 IS ON OR LB USING 00024 RLKS R=0063</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>SAMPLE RUN OF TEST 153</td>
</tr>
<tr>
<td>0002</td>
<td>INITIAL CONDITION FILE:</td>
</tr>
<tr>
<td>0003</td>
<td><strong>DIMENSION UNITS ARE IN MKS</strong></td>
</tr>
<tr>
<td>0004</td>
<td>DENSITY SCALE (kg/m³) 1.0000E+03</td>
</tr>
<tr>
<td>0005</td>
<td>LENGTH SCALE=RTANK.(m) 1.0000E+00</td>
</tr>
<tr>
<td>0006</td>
<td>VELOCITY SCALE(m/s) 1.5140E-01</td>
</tr>
<tr>
<td>0007</td>
<td>TIME SCALE(s) 6.6050E+00</td>
</tr>
<tr>
<td>0008</td>
<td>PRESSURE SCALE(Pa) 2.2922E+01</td>
</tr>
<tr>
<td>0009</td>
<td>0010</td>
</tr>
<tr>
<td>0011</td>
<td>Jet size, RJ1,RJ2 8.5000E-01 9.5000E-01</td>
</tr>
<tr>
<td>0012</td>
<td>Tangential jet, QJ, VJ 1.0000E-01 1.0000E+00</td>
</tr>
<tr>
<td>0013</td>
<td>Injection time, TJ1,TJ2 0.0000E+00 1.0000E+00</td>
</tr>
<tr>
<td>0014</td>
<td>0015</td>
</tr>
<tr>
<td>0016</td>
<td>Density 1.2930E-03 1.0000E+00</td>
</tr>
<tr>
<td>0017</td>
<td>Viscosity 1.1295E-07 1.0000E-05</td>
</tr>
<tr>
<td>0018</td>
<td>Eddy viscosity factor 1.0000E+03 1.0000E+03</td>
</tr>
<tr>
<td>0019</td>
<td>Base dia. 1.0000E-02 1.0000E-02</td>
</tr>
<tr>
<td>0020</td>
<td>Size exp. 2.0000E-01 2.0000E-01</td>
</tr>
<tr>
<td>0021</td>
<td>Phase limits 1.0000E-04 1.0014E-05</td>
</tr>
<tr>
<td>0022</td>
<td>OTHER CONSTANTS: IW, IVTX, NA, ND, DAMP, VPEAK, RPEAK, OMEGA, D1/D2, MU/MU</td>
</tr>
<tr>
<td>0023</td>
<td>1 0 4 4.00</td>
</tr>
<tr>
<td>0024</td>
<td>0.0000E-04 0.0000E-04 1.2930E-03 1.1295E-02</td>
</tr>
<tr>
<td>0025</td>
<td>0.0000E+00 0.0000E+00 0.0000E+00 3.9730E-05</td>
</tr>
<tr>
<td>0026</td>
<td>OTHER CON..TANS IN THE BETWEEN QM..TED ----</td>
</tr>
<tr>
<td>0027</td>
<td></td>
</tr>
<tr>
<td>0028</td>
<td></td>
</tr>
<tr>
<td>0029</td>
<td>1 NP= 1 T=0.0000E+00 NT= 2 DT=1.0000E-06</td>
</tr>
<tr>
<td>0030</td>
<td>J ALP1 U1 U2 V1 V2 P</td>
</tr>
<tr>
<td>0031</td>
<td>1 2.5000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00</td>
</tr>
<tr>
<td>0032</td>
<td>2 2.5000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00</td>
</tr>
<tr>
<td>0033</td>
<td></td>
</tr>
<tr>
<td>0034</td>
<td></td>
</tr>
<tr>
<td>0035</td>
<td></td>
</tr>
</tbody>
</table>

...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.

D. LIQUID PHASE

E. GAS PHASE