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

REPRODUCED BY
U.S. DEPARTMENT OF COMMERCE
NATIONAL TECHNICAL
INFORMATION SERVICE
SPRINGFIELD, VA 22161
A computer code for gas-liquid two-phase vortex motions: GLVM

T. T. Yeh

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

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 low 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

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, Secretary
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>ii</td>
</tr>
<tr>
<td>List of Notations</td>
<td>iii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>v</td>
</tr>
<tr>
<td>I. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>II. Model Equations</td>
<td>2</td>
</tr>
<tr>
<td>III. Numerical Method</td>
<td>5</td>
</tr>
<tr>
<td>IV. Program Details</td>
<td>10</td>
</tr>
<tr>
<td>V. Summary</td>
<td>14</td>
</tr>
<tr>
<td>VI. Acknowledgments</td>
<td>15</td>
</tr>
<tr>
<td>VII. References</td>
<td>15</td>
</tr>
<tr>
<td>Appendix A. Code Listing</td>
<td>16</td>
</tr>
<tr>
<td>Exhibit A. A Sample Input</td>
<td>32</td>
</tr>
<tr>
<td>Exhibit B. A Sample Output</td>
<td>33</td>
</tr>
</tbody>
</table>
Abstract

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

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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_a, A_{ak} )</td>
<td>Added mass coefficients</td>
</tr>
<tr>
<td>( A_d, A_{dk} )</td>
<td>Drag coefficients</td>
</tr>
<tr>
<td>( \bar{\rho}_k )</td>
<td>Body force density</td>
</tr>
<tr>
<td>( C_{1j}, C_{pk}, C_{dk} )</td>
<td>Generalized coefficients</td>
</tr>
<tr>
<td>( d_1 )</td>
<td>Bubble diameter</td>
</tr>
<tr>
<td>( d_2 )</td>
<td>Liquid (droplet) diameter</td>
</tr>
<tr>
<td>( \bar{\omega}_k )</td>
<td>Effective interfacial force density</td>
</tr>
<tr>
<td>( n_a )</td>
<td>Exponent used for ( \omega_{ak} )</td>
</tr>
<tr>
<td>( n_d )</td>
<td>Exponent used for ( \omega_{dk} )</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Pressure</td>
</tr>
<tr>
<td>( R )</td>
<td>Tank radius</td>
</tr>
<tr>
<td>( Re )</td>
<td>( \rho \alpha^2 V_y R/\mu_2 ), Reynolds number</td>
</tr>
<tr>
<td>( R_{j1}, R_{j2} )</td>
<td>Jet opening, ( R_{j1} &lt; r &lt; R_{j2} )</td>
</tr>
<tr>
<td>( R_1 )</td>
<td>Minimum radius considered in the numerical analysis</td>
</tr>
<tr>
<td>( r )</td>
<td>Radial coordinate</td>
</tr>
<tr>
<td>( t )</td>
<td>Time</td>
</tr>
<tr>
<td>( V_j )</td>
<td>Averaged jet velocity</td>
</tr>
<tr>
<td>( V_{r1} )</td>
<td>Gas radial velocity</td>
</tr>
<tr>
<td>( V_{r2} )</td>
<td>Liquid radial velocity</td>
</tr>
<tr>
<td>( V_g )</td>
<td>Velocity scale</td>
</tr>
<tr>
<td>( V_{\theta 1} )</td>
<td>Gas tangential velocity</td>
</tr>
</tbody>
</table>
\( V_{v2} \)
Liquid tangential velocity

\( V_1 \)
Gas velocity

\( V_2 \)
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

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

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

\( v_k \)
\( u_k / \rho_k \), 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

Figure 1. Two Step Difference Scheme
(Backward Predictor-Forward Corrector Version) 34

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 that 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. $\frac{\partial}{\partial z} = \frac{\partial}{\partial \theta} = 0$) are:

$$a_1 + a_2 = 1$$

$$\frac{\partial r a_k}{\partial t} + \frac{\partial r a_k V r_k}{\partial r} = 0$$
\[
\frac{\partial \rho \rho_{rk}}{\partial t} + \frac{\partial \rho \rho_{rk}}{\partial r} - \alpha_k \rho_{rk}^2 = -\alpha_k C_{pk} \frac{\partial \rho}{\partial r} \\
+ a_k \sum_{l=1}^{2} C_{kl} \left( \frac{\partial \rho \rho_{rl}}{\partial r} - \alpha_l \rho \rho_{gl} + \alpha_2 \rho_{l} \rho_{rl}^2 \right) \\
+ a_k C_{dk} \rho (\psi_{r1} - \psi_{r2})
\]

\[
\frac{\partial \rho \rho_{rk}}{\partial t} + \frac{\partial \rho \rho_{rk}}{\partial r} - \alpha_k \rho_{rk}^2 = a_k \sum_{l=1}^{2} C_{kl} \left( \frac{\partial \rho \rho_{rl}}{\partial r} - \alpha_l \rho \rho_{gl} + \alpha_2 \rho_{l} \rho_{rl}^2 \right) \\
+ a_k C_{dk} \rho (\psi_{g1} - \psi_{g2})
\]

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

\[
C_{p1} = \frac{(\alpha_1 \alpha_2 \rho_{2} + A_a)}{\langle \rho^2 \rangle}
\]

\[
C_{p2} = \frac{(\alpha_1 \alpha_2 \rho_{1} + A_a)}{\langle \rho^2 \rangle}
\]

\[
C_{11} = \frac{(\alpha_2 \rho_{2} + A_a)}{\langle \rho^2 \rangle}
\]

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

\[
C_{22} = \frac{(\alpha_1 \rho_{1} + A_a)}{\langle \rho^2 \rangle}
\]
\[ C_{d1} = -a_2 \rho_2 A_d / \langle \rho^2 \rangle \]

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

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

\[ \tau_{rrk} = 2 \mu_e \frac{\partial V_{rk}}{\partial r} \]

\[ \tau_{\theta rk} = \tau_{\theta kr} = \mu_e r \frac{\partial (V_{rk} / r)}{\partial r} \]

\[ \tau_{\theta \theta k} = 2 \mu_e \frac{V_{rk}}{r} \]

with

\[ \mu_e = \mu_k + \mu^t_k \]

and the interfacial forces are modeled in the form of

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

\[ \bar{M}_1 \] is the force density acting on the phase 1 by the phase 2. \( A_a \) and \( A_d \) are the added mass and drag coefficients, respectively.

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

In the program \( Q_r = 0 \) is assumed, since the mixture pumped out is injected immediately back into the tank at the nearby location. The net volume or mass in the system is effectively unchanged except for the net change on the angular momentum. Thus, the pump system (withdrawal and injection) acts as a body force.
on the mixture at the nozzle location. The net momentum gain is thus the
momentum introduced into the system minus the local momentum pumped out.
Therefore, we will model this pumping dynamic by body forces without considering
the mass transfer. That is, the body force density $\alpha_k \rho_b B_r$ will be replaced by
the net momentum gain, $\frac{\alpha_k \rho_b V_j}{2\pi} (V_j \hat{n} - \vec{V})$ 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-corrector type, iii) it is three point, two level - that is, the
solution of $f^{n+1}_i$ at level n+1 depends only on three values of $f^n_i$ 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

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

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

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

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

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

where

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

and

\[ W^c = W^0 + \Delta t W_{tp} \] 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 \( W_{tp} \) is the time derivative of \( W \) evaluated at the corrected time.

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

Numerically, the predicted values are

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

where

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

\[ = \frac{1}{2} (W_{i-1}^{o} + W_{i}^{o}) + \frac{\Delta t}{\Delta r} [(F_{i}^{o} - F_{i-1}^{o}) + \frac{(g_{i}^{o} + g_{i-1}^{o})}{2} (G_{i}^{o} - G_{i-1}^{o})] \]

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

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

\[ W_{i}^{c} = W_{i}^{o} + \Delta t W_{t}^{p} \]

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

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

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

\[
\Delta t_k = \frac{1}{|C_{dk}| + |V_{rk}|/\Delta r + \frac{2}{\Delta r^2} (a_1 u_1 c_{k1} + a_2 u_2 c_{k2})}
\]

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

\[
W_{1D} = W_1 (1-D) + (W_{i-1} + W_{i+1} - W_i) D
\]
where \( \hat{w}_1 \) is the value obtained based on the two-step scheme, and \( \hat{w}^{1D} \) 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_max) or abnormal (e.g., Δt 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 $\alpha_i$, $\text{ALMT}(1) < \alpha_i < \text{ALMT}(2)$.
- **DAMP**: Numerical damping factor. Normally set to 0.
- **DENID2**: Density ratio, $\rho_1/\rho_2$
- **DO**: Base diameter, i.e., $d_k = d_{ik} \alpha_k$
- **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, $M_k \geq NG$. MM appears in many subroutines.
- **MULD2**: $\mu_1/\mu_2$, Viscosity ratio.
- **NA, ND**: Exponents for weighting function for drag and added mass coefficients, $A_a$ and $A_d$. 

**11**
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, \( \rho_2 V_\theta R/\mu_2 \)
RJ1,RJ2  Jet opening, RJ1 < r < RJ2
RPEAK  Location of the peak speed of the initial vortex, if IVTX > 1
RTANK  Tank radius = Length scale LS.
VPEAK  Peak speed of the initial vortex, if IVTX > 1.
TJ1,TJ2  Injection time, TJ1 < t < TJ2.
TS  Time scale = LS/VS
VS  Velocity scale.

The format of the data file for the initial condition (if any) is a six column and NG row data file, where NG is the number of grid points. The column sequence is K, \( \alpha_1(K) \), \( V_{r1}(K) \), \( V_{r2}(K) \), \( V_{\theta1}(K) \), \( V_{\theta2}(K) \), where K is the grid point number, and the rest of the terms are the gas volume fraction, radial gas velocity, radial liquid velocity, tangential gas velocity and tangential liquid velocity respectively. The data format is free.

3) DELA, \( \Delta A \)

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

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

4) DERIV1, Derivative

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

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

The effective coefficients are modeled as:

\[
A_a = A_{al}w_{al} + A_{a2}w_{a2}
\]

\[
A_d = A_{d1}w_{d1} + A_{d2}w_{d2}
\]

\[
A_{al} = \frac{a_1a_2\rho_2}{(a_1 + 2a_2/(1 + 3a_1))}
\]

\[
A_{a2} = \frac{a_1a_2\rho_1}{(a_2 + 2a_1/(1 + 3a_2))}
\]

\[
A_{d1} = 18 \frac{u_2a_1}{d_1^2a_2}
\]

\[
A_{d2} = 18 \frac{u_1a_2}{(1 - a_2/0.8d_2^2)}
\]

\[
w_{al} = \frac{na}{(a_1 + a_2)^2}
\]

\[
w_{al} = 1 - w_{al}
\]

\[
w_{dl} = \frac{a_{nd}}{(a_1 + a_2)^2}
\]

\[
w_{d2} = 1 - w_{dl}
\]

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) FNDT, \( \Delta t \)

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.
   \[ d_k = d_{ok} \alpha_k^Y \]
   Different models for size distributions could be easily adopted here.

11) TAUOFW
   To determine the stress tensor \( \tau \) and its derivative.

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

V. **Summary**

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

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

VII. References


Appendix A

Code Listing

```plaintext
&GLVM T=00004 IS ON CR T4 USING 00126 BLKS R=0000
0001 FTM77
0002 &EMA /DATA/,/WWW/,/COEFF/,/SOURCE/,/FANDS/,/TAU/
0003 &FILES 1.2
0004 PROGRAM GLVM,(99),(B60425.1537)
0005 C
0006 C THIS PROGRAM WAS DEVELOPED TO STUDY THE FLUID DYNAMICAL BEHAVIOR
0007 C OF A ROTATIONAL TWO-PHASE FLUIDS(GAS/LIQUID) IN A CYLINDRICAL TANK.
0008 C THE VORTEX MOTIONS ARE ESTABLISHED BY TANGENTIAL FLUID INJECTION.
0009 C THE PROGRAM WAS DEVELOPED ORIGINALLY BY T.T. YEH OF NBS
0010 C IT WAS BASED ON HP'S FORTRAN 77 ANSI 77+MIL-STD-1753
0011 C
0012 C WHEN WHO WHAT
0013 C A502XX TTY ZI O-G FUEL TRANSFER, START-UP STAGE.
0014 C LAM WENDROFF 2-STEP SCHEME (FULL STEP PREDICTION+CORRECTION)
0015 C WITH NUMERICAL DAMPING FACTOR (USUALLY SET TO ZERO)
0016 C COHERENT FORM, VARIABLE (AUTO) TIME STEP
0017 C REAL="
0018 C INTERFACIAL FORCES: DRAG,ADDED MASS
0019 C PUMP CONDITION: MOMENTUM SOURCE BUT NO MASS SOURCE
0020 C B50715 TTY GENERALIZED EQUATIONS AND COEFF. Cj
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 DELA, DERIV1, GCCOF, DWDE,FNDDT,FSDFW,
0026 C INIT, JET, SIZES, TAUOFW and UFOFW
0027 C *** MOST OF THE LIST OF NOTATIONS ARE GIVEN IN SUBROUTINE..INIT
0028 C
0029 C CHARACTER NAMR=16, NOTES=72
0030 C INTEGR I, JOS,J, JTIME(3), KM, NPRT, NT
0031 Z ,IW,LUP, LUS, NG, NGM1, NGM2
0032 C PARAMETER (HM=10)
0033 C REAL=8 BA, KM,2, DDT, DI, DTHM, DTMH, DTPRT, PZERO
0034 X ,DW, T, TMHX, TPRT, UJT, VDR(2)
0035 Y ,RJ1, RJJ, TJ1, TJ2, QJ, VJ
0036 1, ALN, U, V, ALP, P, R, W, WP, WN, DW, RDP, RH, FS
0037 4, BR, BRH, RHO, HUEF, VIB, NA, ND
0038 5, DO, GAMMA, DMAP, DR
0039 6, TRR, TRIA, TAA, RRR, RTRA, C, CPA, CD
0040 C
0041 COMMON
0042 Y /JETS/ RJJ1, RJJ2, TJJ, TJ2, QJ, VJ
0043 Z /CONT/ IW, LUP, LUS, NG, NGM, DAMP, DR
0044 1 /ALPLM/ ALM(2)
0045 2 /CEOF/ C(MM, 2, 2), CPA(MM, 2), CM(MM, 2)
0046 3 /DATA/ U(MM, 2), V(MM, 2), ALP(MM, 2), P(MM), R(MM)
0047 4 /DRAG/ RHO(4), HUEF(2), VIB(2), NA, ND
0048 6 /FANDS/ F(MM, 5), S(MM, 5)
0049 7 /SOURCE/ BR(MM), BRH(MM)
0050 8 /TAU/ TRR(MM, 2), TAA(MM, 2), RRR(MM, 2), RTRA(MM, 2)
0051 9 /WWW/ W(MM, 2), WP(MM, 2), WN(MM, 2), DW(MM, 2), RDP(MM), RH(MM)
0052 C
0053 EQUIVALENCE (W, BA)
0054 C
0055 C ****** RHO(1) ( RHO(2) i.e., PHASE=1=GAS, PHASE=2=LIQUID) ******
0056 C DTPRT TIME STEP FOR PRINTOUT(AFTER PLOT)
0057 C
0058 C
```
LUF=1            ! LU FOR PRINTING DEBUG DATA(#1 TERMINAL)
LUS=6            ! LU FOR STORING DATA(#6 PRINTER)

FORMAT(2X,A,3(IPE12.4))
FORMAT(2X,A,3(IPE12.4))
WRITE(1,7) 'Enter lu for saving data. D.F.=',LUS
READ(1,*) LUS

C       ! KEEP JOB TIME FOR FUTURE REFERENCE
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)') NAME
LUS=99
OPEN(LUS,FILE=NAME,IODTA=10S,STATUS='NEW',ERR=999)
END IF

WRITE(1,'(A)') 'Enter NOTES((F3 CHAR.) for the job'
READ(1,'(A)') NOTES
WRITE(LUS,'(5I4)') JTIME

C       ! To set-up the initial condition.
CALL INIT
'NGM2=NG-2
UDR(1)=2.*MUEF(1)/DR**2   ! For determining time step
UDR(2)=2.*MUEF(2)/DR**2

T=0.            ! INITIAL TIME
TMAX=5.
DTPRT=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.=',DTPRT
READ(1,*) DTPRT

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

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

10 NT=NT+1
C       ! PRINT OUT AT SELECTED TIME
CALL UOFW(W,R,4G)

IF(T .GE. TPRT OR. T .GT. TMAX) THEN
   IF(NP .GT. 1) THEN
      NPRT=NPRT+1
   ELSE
      TPRT=TPRT+DTPRT*(1.0-DNINT((T-TPRT)/DTPRT))
   END IF
   FORMAT(1HI,2(A5,I4,A5,1PF .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)') 'J','ALP1','U1','U2','V1','V2','P'

0059 0060 0061 0062 0063 0064 0065 0066 0067 0068 0069 0070 0071 0072 0073 0074 0075 0076 0077 0078 0079 0080 0081 0082 0083 0084 0085 0086 0087 0088 0089 0090 0091 0092 0093 0094 0095 0096 0097 0098 0099 0100 0101 0102 0103 0104 0105 0106 0107 0108 0109 0110 0111 0112 0113 0114 0115 0116 0117 0118
DO 30 J=1,NG
WRITE(*,13,F6.4,5(1PE1.3))
13 1 J,ALP(J),U(J,1),U(J,2),V(J,1),V(J,2),P(J)
ENDDO
ENDDO
IF( T .GT. TMAX) GOTO 9999
C TO SOLVE THE DIFFERENTIAL EQUATIONS
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 DERIVATIVES
C CALL DGLDEF(ALP,NG)          ! DRAG AND GENERALIZED CDTFF.
C CALL TAUOFW(MUEP,DR,R,NG)    ! STRESS

137 UJT=UT
138 IF( T .LT. TJL OR .T .GT. TJ2) UJT=0     ! NO INJECTION
139 DO 141 I=1,NG
140 CALL JET(BA,RHO,DR,V,UJT,NG)          ! MOMENTUM SOURCE
141 CALL FSOFW(W,BA,R,NG)              ! CONVECTIVE -F AND SOURCE-S
ENDDO
C DETERMINE THE TIME-STEP SIZE
C IF NT .GE. 1 THEN
C CALL FNDT(DT,DR,NG)
C DT=DT+D.TMIN/10.*NT                  ! INITIAL INPLUSE TREATMENT
C IF(DT .GE. DTMIN) THEN
C I=LOG10(DT/DTMIN)                    ! ROUND OFF TIME STEP
C DO =D.TMIN*10.^I
C DT=D.TMIN(DT/DT+.001)*DDT
C IF( DT .GT. DTMAX) DT=DTMAX
C ELSE
C WRITE(*,13,F6.4,5(1PE1.3))
C 154 'STOP DUE TO TOO SMALL TIME STEP. DT=',DT
C GOTO 9999
155 ENDIF
156 D GOTO 9999
157 ENDIF
158 ENDDO
159 C BACKWARD PREDICTOR
160 CALL DWPDF(DW,RP,DR,DT,RH,NGM1)      ! INCREMENT
161 DO 40 J=1,NGM1
162 DO 40 I=1,5
163 40 WP(J,I)=0.5*(W(J+1,I)+W(J,I))+DW(J,I)    ! BASE+INCREAM
40 CONTINUE
164 ENDIF
165 ENDIF
166 C PREDICTION DATA COMPLETED, CONTINUE FOR CORRECTION
167 C CALL UOFW(WP,RP,NGM1)
168 CALL DGDEF(ALP,NGM1)
169 CALL TAUOFW(MUEP,DR,RH,NGM1)
170 CALL JET(BA,RHO,RBH,V,UJT,NG)
171 CALL FSOFW(WP,BA,RH,NGM1)
172 C FOREWARD CORRECTION
173 CALL DWPDF(DW,P(2),DR,DT,R(2),NGM2)
174 P(1)=PZERO
175 DO 50 J=1,NGM2
C 2ND STEP (PREDICTION+CORRECTION) COMPLETED
C
IF( NT .EQ. 1) GOTO 10
C Estimation of initial condition completed, return to the initial condition
C and start to advance the program in time.
C
C DATA W AT THE NEW TIME STEP COMPLETED

C IMPOSED B.C. #6.4
DNI=0.5*(WN(2,1)+W(2,1))
ALP(1,1)=(.5*(W(1,5)+W(2,5))=DNI*DT/DR/RH(1))
ALP(1,1).LT. ALMT(1) ALP(1,1)=ALMT(1)
IF(ALP(1,1).GT. ALMT(2)) ALP(1,1)=ALMT(2)

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

C
WN(1,1)=0. 
NO RADIAL VEL.

WN(1,2)=0.

WN(NG,1)=0.

WN(NG,2)=0.

C
WN(1,3)=WN(2,3)/WN(1,5)/WN(1,5)*R(1)/R(2) 1 OMEGA=COND.

WN(1,4)=WN(2,4)/(R(1)-WN(1,5))/(R(2)-WN(2,5))*R(1)/R(2)

C IF(W. EQ. 0) THEN

W(1,1)=W(1,1) W(1,3)

W(NG,3)=WN(NG,3)+W(NG,5)*R(NG)/R(NMNG)

C

W(NG,4)=WN(NG,4)+W(NG,5)-W(NG,5)/(R(NMNG)-WN(NG,5))

C

1 *R(NG)/R(NMNG)

C ELSE

W(NG,3)=0. 
NON-SLIP AT WALL

W(NG,4)=0.

ENDIF

C ARTIFICIAL TAMPING

DO 60 I=1,5
W(1,1)=(1.-DAMP)*W(1,1)+DAMP*W(2,1)
W(NG,1)=(1.-DAMP)*W(NG,1)+DAMP*W(NG,1)
60 DO 60 J=2,NMNG

C SOLUTION FOR THIS TIME STEP COMPLETED
C
C
C
T=T+DT 1 UPDATE TIME AND CONTINUE TO THE NEXT STEP
C
GOTO 10
C
C
C WRITE(LUP,7) 'OPEN FILE FAILED ON FILE:'
C WRITE(LUP,7) NMHR
C WRITE(LUP,7) 'IOSTAT=',I0S
C
C
C CONTINUE
CALL EXEC(11, JTIME, JTIME(1))
WRITE((1,US,'(3S5,SI4)') JTIME
CLOSE(LUS)
END

*---------------------------------------------------------------*
* SUBROUTINE INIT , (B60425.1537)                             *
*---------------------------------------------------------------*
TO SET-UP THE INITIAL CONDITIONS

INTEGER I, IJS, ITLOG, IUTX, J, K, MM
Z ,JTIME, IM, LUP, LUS, NG, NCM1
PARAMETER (MM=181)
CHARACTER NAME(16)
REAL ALMT, W, WP, WM, DM, RDP, RH
IU, V, ALP, P, R, S
RI0, MUF, V18, MA, ND, DO, GAMMA
BM, BRH, DAMP, DR
RJ1, RJ2, TJ1, TJ2, Q1, VJ
AX, A2, SELA, DEM1D2, D1, D2, PI, RE, RMIN, RTANK
DS, LS, VS, TS, PE
ALP10, OMEGA, RPEAK, SJ8, VPEAK
EUF(2), VT, MU(2), MU1D2

COMMON
ALMT(2)
JETS, RJ1, RJ2, TJ1, TJ2, Q1, VJ
CONDZ, W, LUP, LUS, NG, NCM1, DO, DR
DATA/ VM(2), VM(2), ALP(2), P(2), R(2), MU, ND
DSIZE/ DO(2), GWMM(2)
SOURC/ BR(MM), BRH(MM)
DATA PI/3.14159629/

***** 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 =V*SRTANK/RHO(2)/MU(2)
RTANK TANK RADIUS
IUTX TYPE OF INITIAL FLOW.

RPEAK, VPEAK VORTEX PARAMETERS
OMEGA PURE ROTATION. V=OMEGA*

EUF, MUF EFFECTIVE VISCOITY, MUF=(1+EVF)*MU
DO, GAMMA DIA. PARAMETERS: D=DO*ALP**GAMMA

RJ1, RJ2, TJ1, TO DEFINE JET SIZE, PUMPING TIME

TJ2, Q1, VJ VOLUME FLOW RATE AND INJECTION MEAN SPEED

FORMAT(2X,A,2X,A)
FORMAT(2X,A,3I5)
FORMAT(2X,A,3(IPE12.4))
FORMAT(A25,2(IPE15.4))
FORMAT(X,7(IPE11.4))
DEFINE THE PARAMETERS FOR THE PROBLEM.

RTANK=1.0     ; COULD BE SET TO 1 (M)
RHO(2)=1.000D+3 ; " (KG/M**3)
MU(2)=1.014D-3  ; " (KG/M-S)

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. Thru all characteristic scales (LS,VS,TS, and DS) are fixed after RE is given.

RE=1.0DS
WRITE(1,8) 'enter Reynolds no. RE: D.F. = ',RE
READ(1,9) RE

LS='TANK
LENGTH SCALE (M)

RHO(2)=RHO(2)/DS
DENSITY SCALE (KG/M**3)

MU(2)=MU(2)/(DS*LS*VS)
VELOCITY SCALE(M/S)

RHO(1)=DEN12/RHO(2)

MU(1)=MU12*MU(2)

ALMT(1)=0.0813
MIN. OF ALP1

ALMT(2)=0.9999
MAX. OF ALP1

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

DO(2)=1.0-2/LS
LIQUID DIAMETER at ALP2=1

GAMMA(1)=2.0-1

GAMMA(2)=2.0-1

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

EVF(1)=DEN12/MU12*EVF(2)
MODEL

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

W=4.0
WEIGHTING EXP. FOR ADM

W=4.0
WEIGHTING EXP. FOR DRAG

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

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
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.=',EUF
READ(1,*) EUF

IW=1
WRITE(1,7) 'Enter wall condition, 1=nonslip, 0=slip. D.F.=',IW
READ(1,*) IW

WRITE(1,8) 'Enter numerical damping factor. D.F.=',DAMP
READ(1,*) DAMP

DO 10 K=1,2

V0(K)=10.*MU(K)

10 MU(EF(K))=MU(K)*1.+EUF(K) ; EFFECTIVE VISCOITY FOR STRESS

RHO(3)=RHO(1)*RHO(2)
RHO(4)=RHO(1)-RHO(2)

RMIN=.1

NG=101
NGM1=NG-1
DR=(1.-RMIN)/NGM1

C Initial cleaning-up.

DO 15 J=1,NM

DO 15 K=1,6

W(J,K)=0.D0

15 W(J,K)=0.D0

C MOMENTUM SOURCE, JET CONDITIONS

RJ1=0.5D-1
RJ2=9.5D-1
VJ=10.D1

10 TJ1=0.D1

TJ2=10.D1

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,TJ1,TJ2
READ(1,*) VJ,TJ1,TJ2

QJ=(RJ2-RJ1)/VJ

DO 20 J=1,NG

R(J)=RMIN+(J-1)*DR
RN(J)=R(J)+0.5D0

20 RN(J)=DELA(R(J)),DR,RJ1,RJ2)/(2.*PI) ; JET DISTRIBUTION

20 RN(J)=DELA(R(J)),DR,RJ1,RJ2)/(2.*PI) ; PER RADIUS

C SETUP INITIAL CONDITIONS

IVT=8
0419  OMEGA=0.
0420  VPEAK=0.
0421  RPEAK=RMIN
0422  NAME='Simple vortex'
0423  WRITE(1,5) 'Enter data FILE NAME for initial cond., i.e. any,'
0424  WRITE(1,5) 'D.F.=',NAMR
0425  READ(1,','(A)') NAMR
0426  READ(1,','(A)') NAMR
0427  IF( NAMR .NE. ' ', .AND. NAMR .NE. 'Simple vortex') THEN
0428      C
0429   INITIAL CONDITION FROM A GIVEN FILE NAMR.
0430      OPEN(99,FILE=NAMR,STAT=IOB,STATUS='OLD',ERR=299)
0431      DO 25 J=1,NG
0432          READ(99,..,ALP(J,1),U(J,1),U(J,2),V(J,1),V(J,2)
0433      25   CLOSE(99)
0434  ELSE
0435   C
0436   TO DEFINE INITIAL CONDITION.
0437      ALP10=2.00-1  ! INITIAL GAS VOL. FRACTION
0438      WRITE(1,8) 'Enter initial value of alpl. D.F.=?',ALP10
0439      READ(1,','(A)') ALP10
0440      WRITE(1,7) 'Enter type of vortex: 0=At rest, 1=pure rotation'
0441      WRITE(1,7) '2=H.O., 3=GIT. D.F.=?',IVTX
0442      READ(1,','(A)') IVTX
0443      IF(IVTX .GT. 0) THEN
0444          IF(IVTX .GT. 1) THEN
0445              WRITE(1,8) 'Enter PEAK ALP and LOCATION for classic vortex'
0446          ELSE
0447              WRITE(1,8) 'D.F.=',VPEAK,RPEAK
0448          ENDIF
0449          IF(RPEAK LE. 0.) RPEAK=1.  ! SINGULAR AT ZERO
0450      ELSE
0451      WRITE(1,8) 'Enter CIRCULAR SPEED(rad./unit time). D.F.=','
0452      1
0453          READ(1,','(A)') OMEGA
0454      ENDF
0455  ENDF
0456  DO 30 J=1,NG
0457      ALP(J,1)=ALP10
0458      VT=OMEGA*R(J)
0459      IF(IVTX .GT. 0) THEN
0460          RJB=R(J)/RPEAK
0461      ELSE
0462          IF(IVTX .EQ. 1) THEN
0463              UT=UT+1.256*VPEAK/RJB*(1.-DEXP(-1.256*VPEAK/RJB))
0464          ELSE
0465              UT=UT+VPEAK*RJB*DEXP((1.-RJB**2)/2.)
0466          ENDIF
0467      ENDF
0468  DO 30 X=1,2
0469      IF(IW .EQ. 0) THEN
0470          V(J,X)=UT
0471      ELSE
0472          V(J,X)=UT*(1.-R(J)**0.5)
0473      ENDF
0474      EndIF
0475      30
0476      ENDF
0477      DO 40 J=1,NG
0478      FORM W FOR NUMERICAL CAL.
ALP(J,2)=1.-ALP(J,1)

P(J)=0.DO

W(J,5)=R(J)=ALP(J,1)

DO 40 K=1,2

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

40

C PRINTOUT PARAMETERS

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

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

WRITE(LUS,9) 'DENSITY SCALE(kg/m^3)',D5

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

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

WRITE(LUS,9) 'TIME SACLE(s)',TB

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', TJ1,TJ2',TJ1,TJ2

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

WRITE(LUS,5) 'Other constants: IW,IVA,NA,ND,DAMP,VPEAK,ABPEAK',

WRITE(LUS,5) 'OMEGA,DI/D2,MU1/MU2')

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

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

WRITE(LUS,5) 'NMe',RMIN

RETURN

---

399 WRITE(LUP,7) 'OPEN FILE FAILED ON INPUT FILE:',NAMR

STOP 111

END

---

REAL# FUNCTION DELA(R,DR,RJ1,RJ2),(040423.1337)

C TO DETERMINE THE EFFECTIVE NOZZLE SIZE AT EACH GRID LOCATI

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.DR

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

IF(R1.LE.RJ1) R1=RJ1

IF(R2.GT.RJ2) R2=RJ2

DELA=(R2-R1)/DR

RETURN

24
0539      END
0540
0541      SUBROUTINE DERIV4(Y,DY,DX,N2),(BBV4.1537)
0542      C      GE: 1ST DERIVATIVE, USING CENTERED DIFFERENCE
0544      REAL*8 DX,Y(1),DY(1),C
0545      EMA Y,DY
0546
0547      C=5.D-1/DX
0548      DO 10 J=2,N2-1
0549      10      DY(J)=C*(Y(J+1)-Y(J-1))
0550      CALL DWSUB(Y(3),1,Y,1,DY(2),1,N2-2)
0551      DY(1)=(Y(2)-Y(1))/DX    ! BASED ON 3-END PTS
0552      DY(N2)=(Y(N2)-Y(N2-1))/DX
0554      DY(1)=2.*DY(1)-DY(2)
0555      DY(N2)=2.*DY(N2)-DY(N2-1)
0557      CALL DWSMY(5.D-1/DX,DY,1,DY,1,N2)
0558      RETURN
0559      END
0560
0562
0563      SUBROUTINE DDCOEF(ALP,N2),(86425.1537)
0566      C      CALCULATE THE DRAG, ADDED MASS AND GENERIZED COEFF.
0567
0568      INTEGER J,MM,N2
0569      PARAMETER (MM=101)
0570      REAL*8 ALP(MM,2)
0571      EMA ALP
0572
0573      REAL*8 C,CPA,CD, RHO,HUEF,V18,NA,ND
0574      X ,AA,AA1,AA2,AD,AD1,AD2,A1,A2,A12,AL2,DL2, D1,D2,W1,W2,X
0575      COMMON
0576      1 /COEFF/ (MM,2),CPA(MM,2),CD(MM,2)
0577      2 /DRAG1/ RHO(4),HUEF(2),V18(2),NA,ND
0578      DO 50 J=1,2
0579      50      ALP(J)\=1
0580      A1=ALP(J,1)
0581      A2=ALP(J,2)
0582      A12=A1*A2
0583
0584      TO GET DRAG COEFF. AD
0585      CALL SIZES(D1,D2,A1)
0586      A1=V18(2)*A1/(A2*D1*D1)     ! =AD1 IF A2>0.78
0587      IF(AD .LT. .78) THEN
0588          AD=V18(1)*A2/(1.-A2*9.8)*D2**2
0589          WT1=A2*ND
0590          WT2=A1*ND
0591          AD=(AD*WT1+AD2*WT2)/(WT1+WT2)
0592          ENDIF
0593
0594      EMA ALP
0595      END
0597
0598      C      ADDED MASS COEFF. AA
0599
0600
25
0599  AA1=A12*RHOD(2)/(A1*A2/(1.5+1.5*A1))
0600  AA2=A12*RHOD(1)/(A1/(1.5+1.5*A2)+A2)
0601  WT1=A2*NA
0602  WT2=A1*NA
0603  AA=(AA1*WT1+AA2*WT2)/(WT1+WT2)
0604  C THE GENERALIZED COEFF. CPA, C, AND CD
0605  DB2=A12*RHOD(3)+AA*(RHOD(1)+A1*RHOD(2)*A2)
0606  CPA(J,1)=A1*(A12*RHOD(2)+AA)/DB2
0607  CPA(J,2)=A2*(A12*RH0(1)+AA)/DB2
0608  C(J,1,2)=AA/DB2
0609  C(J,2,1)=C(J,1,2)
0610  C(J,1,1)=A2*RHOD(2)/DB2+C(J,1,2)
0611  C(J,2,2)=A1*RHOD(1)/DB2+C(J,2,1)
0612  CD(J,1)=-A2*RHOD(2)*AD/DB2
0613  CD(J,2)=A1*RHOD(1)*AD/DB2
0614  50 CONTINUE
0615  RETURN
0616  END
0617  ******************************************************************************
0618  0619  SUBROUTINE DWPDE(DW,RDP, DR,DT,RR,N2) (060425.1537)
0620  ******************************************************************************
0621  C TO GET DW OF THE PDEs
0622  ******************************************************************************
0623  0624  C NOTE: COEFF. C =C=ALP WHEN THIS IS CALLED
0627  0628  INTEGER J,JP1,K,XP2,MM,N2
0629  0630  PARAMETER (MM=101)
0631  0632  COMMON
0633  0634  1 /COEFF/ C(MM,2,2),CPA(MM,2),CD(MM,2)
0635  0636  3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0637  0638  4 /DRA1/ RHOD(4),MUEF(2),V18(2),NA,ND
0639  0640  6 /FANDS/ F(MM,5),S(MM,5)
0641  0642  8 /TAU/ :RH(MM,2),TRA(MM,2),TAA(MM,2),RTRR(MM,2),RAINT(MM,2)
0643  0644  0645  DTR=DTR/DR
0646  0647  HDT=0.5*DT
0648  0649  DO 10 J=1,N2+1  \ CHANGE C TO ALP=C
0650  0651  C(J,1,1)=ALP(J,1)*C(J,1,1)
0652  0653  10 C(J,2,2)=ALP(J,2)*C(J,2,2)
0654  0655  DO 20 J=1,N2
0656  0657  JP1=J+1
0658  0659  DW(J,5)=DTR*(-F(JP1,5)+F(J,5))+HDT*(S(JP1,5)+S(J,5))
0660  0661  DO 25 K=1,2

26
C SUBROUTINE FNDT(DT)
C
KP2=K+2
C
DO 25 DW(J,KP2)=DTDR(-F(JP1,KP2)+F(J,KP2))
C
G1=5*(C(JP1,K,1)+C(J,K,1))
G2=5*(C(JP1,K,2)+C(J,K,2))
DW(J,K)=DTDR(-F(JP1,J)+F(J,K)+G1*RTRR(JP1,1)-RTRR(J,1))
C
+G2*RTRR(JP1,2)-RTRR(J,2))
C
+HDT*(S(JP1,K)+S(J,K))
C
C
IF(-DW(J,1).GT.DW(J,2)) DW(J,1)=-DW(J,2) ! DP=0
C
RDP(J)=(DW(J,1)+DW(J,2))/(CP1+CP2)
C
DW(J,1)=DW(J,1)-CP1*RDP(J)
C
C
25 DW(J,2)=DW(J,1)
C
C
DO 10 RDP(J)=RDP(J)/DT
C
C
RETURN
C
END
C
---------------------------------------------------------------------
C
C SUBROUTINE FNDT(DT,DR,VDR,NG),(E80425.1537)
C
C DETERMINE THE TIME-STEP SIZE
C
C
INTEGER I,J,LUP,MM,NG
C
PARAMETER (MM=101)
C
REAL*8 DT,DR,VDR(2)
C
REAL*8 C,CPA,CD, RHO,HUEF,V18,NA,ND
C
9 ,U,W,WN,DW,RDP,RH,U,Y,ALP,P,R
C
X ,DUM1,DUM2
C
COMMON
C
C
2 /COEFF/ C(MM,2),CPA(MM,2),CD(MM,2)
C
3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
C
4 /DRAG/ RHO(4),HUEF(2),V18(2),NA,ND
C
9 /WWW/ W(MM,5),WP(MM,5),WN(MM,5),DW(MM,5),RDP(MM),RH(MM)
C
DATA LUP/1/
C
C
DUM1=0.
C
DO 10 J=1,NG
C
10 DUM2=CD(J,1)
C
+DABS(U(J,1))/DR
C
1/ST : CONV
C
+VDR(1)*ALP(J,1)*C(J,1,1)
C
VDR=2*HUEF/DR**2
C
+VDR(2)*ALP(J,2)*C(J,1,2)
C
CROSS VISCO\SY
C
IF(DUM1 .LT. DUM2) DUM1=DUM2
C
DUM2=CD(J,2)+DABS(U(J,2))/DR
C
1/ST
C
+VDR(1)*ALP(J,2)*C(J,2,2)
C
+VDR(2)*ALP(J,1)*C(J,2,1)
C
CROSS VISCO\SY
C
IF(DUM1 .LT. DUM2) DUM1=DUM2
C
CONTINUE
C
C FIND THE MAXIMUM OF DW
C
C
27
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))

RETURN
END

******************************************************************************
**EHA /DATA/,/FANDS/,/TAU/,/COEFF/                             (860425.1537)
******************************************************************************
SUBROUTINE FSOFU(W,BA,RH,N2), (860425.1537)

CALCULATE THE CONVECTIVE-F AND SOURCE-S TERMS
INTEGER J,K,KP2,L,MM,N2
PARAMETER (MM=101)
REAL*8 U,V,ALP,P,R,F,S,RHO,MUEF,VI1,N1,ND
2 ,TRA,TRA,TAAD,TRA,T,CPA,CD
ALP(2),RDV

REAL*8 W(MM,5),BA(MM,2),RR(1)

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 /DRA/ RHO(4),MUEF(2),VI1(2),NA,ND
6 /FANDS/ F(MM,5),S(MM,5)
8 /TAU/ TRA(MM,2),TRA(MM,2),TAA(MM,2),RTR(MM,2),RTRA(MM,2)

CALL DWMPY(W,1,U,1,F,1,2*MM)
CALL DWMPY(W(1,3),1,U,1,F(1,3),1,2*MM)
CALL DWHDV(W,1,F(1,5),1,MM)

DO 20 J=1,N2
20 F(J,5)=W(J,1)
S(J,5)=0.

RDV=RR(J)*(U(J,1)-U(J,2))
RDV=RR(J)*(V(J,1)-V(J,2))
DO 20 K=1,2
KP2=K+2
F(J,K)=W(J,K)+U(J,K)
F(J,KP2)=W(J,KP2)+U(J,K)

S(J,K)=ALP(J,K)*(U(J,K)**2+CD(J,K)*RDV-C(J,K)*TAA(J,1)
1 -C(J,2)*TAA(J,2)
20 S(J,KP2)=ALP(J,K)*(U(J,K)-U(J,K)+CD(J,K)*RDV+C(J,K,1)*(BA(J,1)
1 +TRA(J,1))+C(J,K,2)*(BA(J,2)+TRA(J,2)))

RETURN
END

******************************************************************************
**SUBROUTINE JET(BA,RH,BA,VJ,NG)                                  
******************************************************************************
INJECTION MOMENTUM SOURCE
PARAMETER (MM=181)

REAL*8 BA(MM,2),RHO(1),Bk(1),V(MM,2),VJ

END BA,BR,V

DO 10 J=1,;5

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

Q=BR(J)*VJ

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

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

ELSE

BA(J,1)=0.D0

BA(J,2)=0.D0

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/ DO2,GAMMA(2)

D1=DO(1)#ALP1=#GAMMA(1)

D2=DO(2)#(1.0D0-ALP1)#GAMMA(2)

RETURN

END

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

REAL*8 MU(2),DR,RR(1)

END MU,RR

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

PARAMETER (MM=101)

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

2 TRR,TRA,TAA,RRR,RTA

X TAMU,THU

COMMON

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

8 /TAU/ TRR(MM,2),TRA(MM,2),TAA(MM,2),RRR(MM,2),RTA(MM,2)

DO 50 K=1,2

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

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

CALL DUMPY2(MU(K),TRR(1,K),I,TRR(1,K),I,N2)

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

DO 10 J=1,N2

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

THU=2.0#MU(X)

DO 20 J=1,N2

TAMU=THU#ALP(J,K)

TRR(J,K)=TAMU#TRR(J,K)

DO 20 J=1,N2
TR(J,K)=MU(K)*ALP(J,K)*RR(J)*TRA(J,K)
TAA(J,K)=TAMU(J,K)/RR(J)
RTR(J,K)=RR(J)*TR(J,K)
20 RTR(J,K)=RR(J)*TR(J,K)

CALL DUMPY(RR,1,TRA(1,K),1,TRA(1,K),1,N2) ! R
CALL DUMPY(ALP(1,K),1,TRA(1,K),1,TRA(1,K),1,N2) ! ALP
CALL DUMPY(MU(K),TRA(1,K),1,TRA(1,K),1,N2) ! MU=TRA
CALL DUMPY(U(K),1,RR(1,K),1,TRA(1,K),1,N2) ! U/P
CALL DUMPY(ALP(1,K),1,TRA(1,K),1,TRA(1,K),1,N2) ! ALP
CALL DUMPY(U(K),1,RR(1,K),1,TRA(1,K),1,N2) ! U/P
50 CONTINUE
RETURN
END

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

EMA /DATA/

C SUBROUTINE UDFW(W,RR,N2),(860425.1537)

PARAMETER (MM=101)

REAL=W(MM,5),RR(MM)

EMA W ,RR

REAL=ALMT, U,V,ALP,P,R

X ,WJ6

COMMON

X /ALPLMT/ ALMT(2)

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

DO 50 J=1,N2

IF(ALP(J,1).LE. ALMT(1) .OR. ALP(J,1).GE. ALMT(2)) THEN

IF(ALP(J,1).LE. ALMT(1)) THEN

ALP(J,1)=ALMT(1)

W(J,3)=W(J,4)*ALMT(1)/(1.0+ALMT(1))

ELSE

ALP(J,1)=ALMT(2)

W(J,4)=W(J,3)/(1.0+ALMT(2))/ALMT(2)

ENDIF

W(J,1)=0.0

W(J,2)=0.0

W(J,3)=ALP(J,1)*RR(J)

ENDIF

ALP(J,2)=1.0+ALP(J,1)

IF( W(J,2) .LT. 0.) THEN ! PHASE-2 DOES NOT MOVE IN

W(J,2)=0

W(J,1)=0.

ENDF

398
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  *V CALL DWDIV(W(I, 1), W(I, 5), 1, U, 1, N2)
0907  *V CALL DWDIV(W(I, 3), W(I, 5), 1, V, 1, N2)
0908  *V CALL DWDIV(ALP(I, 1, 2), 1, RR, 1, W(I, 6), 1, N2)
0909  *V CALL DWDIV(W(I, 2), 1, W(I, 6), 1, U(I, 2), 1, N2)
0910  *V CALL DWDIV(W(I, 4), 1, W(I, 5), 1, V(I, 2), 1, N2)
0911
0912  RETURN
0913  EHD
0914
A Sample Input

:GLVM
Enter Iu for saving data. D.F. = 6
30
Enter FILE NAME for saving data.
TS155::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. = 1.143E+02 1.0000E+03

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

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

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

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

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

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

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

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

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

A Sample Output

\[
\begin{array}{c}
\text{TSM33 } \text{T=0.0004 IS ON OR LB USING 0.0024 RLKS } R=0.0003
\end{array}
\]

0001 1 SAMPLE RUN OF TEST 153
0002 INITIAL CONDITION FILE:
0003 **DIMENSION UNITS ARE IN MKS**
0004 DENSITY SCALE (kg/m$^3$) 1.0000E+03
0005 LENGTH SCALE=RTANK.(m) 1.0000E+00
0006 VELOCITY SCALE(m/s) 1.5140E-01
0007 TIME SCALE(s) 6.6050E+00
0008 PRESSURE SCALE(Pa) 2.2922E+01
0009
0010 Reynolds number, \(Re\) 1.0000E+05
0011 Jet size, RJ1,RJ2 8.5000E-01 9.5000E-01
0012 Tangential jet, QJ,VJ 1.0000E-01 1.0000E+00
0013 Injection time,TJ1,TJ2 0.0000E+00 1.0000E+00
0014
0015 PHASE-1 PHASE-2
0016 Density 1.2930E-03 1.0000E+00
0017 Viscosity 1.1295E-07 1.0000E-05
0018 Eddy viscosity factor 1.0000E+03 1.0000E+03
0019 Base dia. 1.0000E-02 1.0000E-02
0020 Size exp. 2.0000E-01 2.0000E-01
0021 Phase limits 1.0000E-04 1.0000E-05
0022
0023 OTHER CONSTANTS: J,W,JVTX,NA,ND,DAMP.VPEAK.RPEAK.OMEGA,D1/D2,MU/MU
0024 1 0 4 4 0.00
0025 0.0000E+00 1.0000E-01 0.0000E+00 1.2930E-03 1.1295E-02
0026
0027
0028
0029 1 NP= 1 T=0.0000E+00 NT= 2 DT=1.0000E-06
0030 J ALP1 \(U1\) \(U2\) \(V1\) \(V2\) P
0031 1 2.5000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0032 2 2.5000 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

---------- OUTPUT IN THE BETWEEN OMITTED ----------
0033
0034 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00

---------- 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.