

NBS
PUBLICATIONS

A11102 247917

NAT'L INST OF STANDARDS & TECH R.I.C.



A11102247917

Yeh, T. T/A computer code for gas-liquid
QC100 .U56 NO.86-3414 1986 C.2 NBS-PUB-C

NBSIR 8

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

QC

100

.U56

86-3414

1986

C. 2

Aeronautics and Space Administration
Kennedy Space Center
Space Center, FL 32899

NBSIR 86-3414

**A COMPUTER CODE FOR GAS-LIQUID
TWO-PHASE VORTEX MOTIONS:
GLVM**

NBS

86-100

U.S.

86-3414

1986

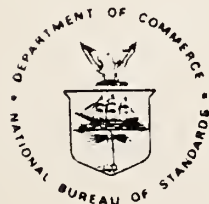
C2

T. T. Yeh

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

July 1986

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



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

Table of Contents

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

Abstract

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

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

List of Notations

A_a, A_{ak}	Added mass coefficients
A_d, A_{dk}	Drag coefficients
\bar{B}_k	Body force density
C_{ij}, C_{pk}, C_{dk}	Generalized coefficients
d_1	Bubble diameter
d_2	Liquid (droplet) diameter
\bar{M}_k	Effective interfacial force density
n_a	Exponent used for w_{ak}
n_d	Exponent used for w_{dk}
p	Pressure
R	Tank radius
Re	$\rho_2 V_s R / \mu_2$, Reynolds number
R_{j1}, R_{j2}	Jet opening, $R_{j1} < r < R_{j2}$
R_1	Minimum radius considered in the numerical analysis
r	Radial coordinate
t	Time
V_j	Averaged jet velocity
V_{r1}	Gas radial velocity
V_{r2}	Liquid radial velocity
V_s	Velocity scale
$V_{\theta 1}$	Gas tangential velocity

$v_{\theta 2}$	Liquid tangential velocity
\bar{v}_1	Gas velocity
\bar{v}_2	Liquid velocity
w_{ak}	Weighting function for added mass coefficients
w_{dk}	Weighting function for drag coefficients
α_1	Gas volume fraction
α_2	$(1-\alpha_1)$, liquid volume fraction
γ	Exponent for diameter variation
θ	Circumferential coordinate
μ_1	Gas dynamic viscosity
μ_2	Liquid dynamic viscosity
μ_k^e	$\mu_k + \mu_k^t$ total effective viscosity
μ_k^t	Turbulence or eddy viscosity
ν_k	μ_k/ρ_k , kinematic viscosity
ρ_1	Gas density
ρ_2	Liquid density
ρ_k	Averaged density of k-phase
$\langle \rho^2 \rangle$	$\alpha_1 \alpha_2 \rho_1 \rho_2 + A_a (\alpha_1 \rho_1 + \alpha_2 \rho_2)$
ω	Angular velocity

List of Figures

	<u>Page</u>
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 detail analysis of the two-phase vortex model can be found elsewhere [1]. In this report, the details of the computer code are described.

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

The program was developed and tested on an HP-1000 minicomputer. The HP-1000's FORTRAN 77 is based on the American National Standards Institute (ANSI) 77 standard programming language FORTRAN (ANSI X3.9-1978). The HP FORTRAN 77 has extensions to provide a more structured approach to program development and more flexibility in computing for scientific applications. It fully implements the Military Standard Definition (MIL-STD-1753) of extensions to the ANSI 77 standard. In order to make the computer code more useful for other computer systems, modifications have been made so that the code is closer to the ANSI 77 standard and thus less system dependent. Some limited extensions are still kept in order to produce the code in the HP-1000. Since the graphic routines are system dependent and must be modified with their equivalents at each computing facility, the original graphic code has not been included in this report. All

lines preceded by "*V" are originally adopted to use the vector operation package supplied by Hewlett Packard. The speed of the code can be increased by replacing many "do-loop" operations in the code with high speed vector operations. Little effort is required to incorporate the vector operation into the code if the vector operation package is in the system.

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

II. Model Equations

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

The equations for the conservation of mass and momentum for the two fluid two-phase model in an one-dimensional, axisymmetric case

(i.e. $\frac{\partial}{\partial z} = \frac{\partial}{\partial \theta} = 0$) are:

$$\alpha_1 + \alpha_2 = 1$$

$$\frac{\partial r \alpha_k}{\partial t} + \frac{\partial r \alpha_k V_{rk}}{\partial r} = 0$$

$$\frac{\partial r \alpha_k V_{rk}}{\partial t} + \frac{\partial r \alpha_k V_{rk}^2}{\partial r} - \alpha_k V_{\theta k}^2 = -\alpha_k C_{pk} r \frac{\partial p}{\partial r}$$

$$+ \alpha_k \sum_{\ell=1}^2 C_{k\ell} \left(\frac{\partial r \alpha_\ell \tau_{r\ell}}{\partial r} - \alpha_\ell \tau_{\theta\ell} + \alpha_\ell \rho_\ell B_{r\ell} r \right)$$

$$+ \alpha_k C_{dk} r (V_{r1} - V_{r2})$$

$$\frac{\partial r \alpha_k V_{\theta k}}{\partial t} + \frac{\partial r \alpha_k V_{rk} V_{\theta k}}{\partial r} + \alpha_k V_{rk} V_{\theta k} = \alpha_k \sum_{\ell=1}^2 C_{k\ell} \left(\frac{\partial r \alpha_\ell \tau_{r\theta\ell}}{\partial r} + \alpha_\ell \tau_{r\theta\ell} + \alpha_\ell \rho_\ell B_{\theta\ell} r \right)$$

$$+ \alpha_k C_{dk} r (V_{\theta 1} - V_{\theta 2})$$

for $k = 1$ and 2 and with

$$C_{p1} = (\alpha_1 \alpha_2 \rho_2 + A_a) / \langle \rho^2 \rangle$$

$$C_{p2} = (\alpha_1 \alpha_2 \rho_1 + A_a) / \langle \rho^2 \rangle$$

$$C_{11} = (\alpha_2 \rho_2 + A_a) / \langle \rho^2 \rangle$$

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

$$C_{22} = (\alpha_1 \rho_1 + A_a) / \langle \rho^2 \rangle$$

$$C_{d1} = -\alpha_2 \rho_2 A_d / \langle \rho^2 \rangle$$

$$C_{d2} = \alpha_1 \rho_1 A_d / \langle \rho^2 \rangle$$

and

$$\langle \rho^2 \rangle = \alpha_1 \alpha_2 \rho_1 \rho_2 + A_a (\alpha_1 \rho_1 + \alpha_2 \rho_2)$$

The effective stresses are modeled as

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

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

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

with

$$\mu_k^e = \mu_k + \mu_k^t$$

and the interfacial forces are modeled in the form of

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

\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 $\alpha_1 V_{r1} + \alpha_2 V_{r2} = Q_r / r$, 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 $\alpha_k \rho_k \bar{B}_k r$ will be replaced by the net momentum gain, $\frac{\alpha_k \rho_k V_j}{2\pi} (V_j \bar{n} - \bar{V})$ at the nozzle location, where V_j is the injection speed.

III. Numerical Method

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

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

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

Here the subscripts (t and r) denote partial differentiation with respect to t and r, respectively, and W, F, P_r , gG_r and S are column matrices with five

elements. All the components of F , P_r , gG_r and S can be regarded as functions of the components of W which are the independent variables. The fundamental theory of the MacCormack's scheme is briefly given below.

For second order accuracy, the solution could be written as

$$\begin{aligned}
 W^1 &= W^0 + \Delta t W_t^0 + \frac{(\Delta t)^2}{2} W_{tt}^0 \\
 &= W^0 + \frac{\Delta t}{2} W_t^0 + \frac{\Delta t}{2} (W_t^0 + \Delta t W_{tt}^0) \\
 &= \frac{1}{2} (W^0 + \Delta t W_t^0) + \frac{1}{2} (W^0 + \Delta t W_t^p) \\
 &= \frac{1}{2} (W^p + W^c)
 \end{aligned}$$

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

$$\begin{aligned} W_{i-1/2}^p &= \frac{1}{2} (W_{i-1}^o + W_i^o) + \Delta t W_t^o \\ &= \frac{1}{2} (W_{i-1}^o + W_i^o) + \frac{\Delta t}{\Delta r} [(F_i^o - F_{i-1}^o) + \frac{(g_i^o + g_{i-1}^o)}{2} (G_i^o - G_{i-1}^o)] \\ &\quad + \frac{\Delta t}{2} (S_i^o + S_{i-1}^o) + \Delta t \hat{P}_{i-1/2}^p \end{aligned}$$

and the corrected value is evaluated at the predicted time place, that is at

$W_{i+1/2}^p$. Thus

$$\begin{aligned} W_i^c &= W_i^o + \Delta t W_t^p \\ &= W_i^o + \frac{\Delta t}{\Delta r} [(F_{i+1/2}^p - F_{i-1/2}^p) + \frac{(g_{i+1/2}^p + g_{i-1/2}^p)}{2} (G_{i+1/2}^p - G_{i-1/2}^p)] \\ &\quad + \frac{\Delta t}{2} (S_{i+1/2}^p + S_{i-1/2}^p) + \Delta t \hat{P}_i^c \end{aligned}$$

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, $i \pm 1$ is the spatial location of regular mesh points immediately to the right and left of the location i , $i \pm 1/2$ 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 Δr , the maximum time step Δt is given by

$$\Delta t_k = 1/[|C_{dk}| + |V_{rk}|/\Delta r + \frac{2}{\Delta r^2} (\alpha_1 \mu_1^e C_{k1} + \alpha_2 \mu_2^e C_{k2})]$$

where $k = 1$ and 2 . The minimum Δt_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

$$W_i^{lD} = W_i^l (1-D) + (W_{i-1}^l + W_{i+1}^l - W_i^l) D$$

where W_i^1 is the value obtained based on the two-step scheme, and W_i^{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 (TO), 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 α_1 , $ALMT(1) < \alpha_1 < ALMT(2)$.
- DAMP Numerical damping factor. Normally set to 0.
- DEN1D2 Density ratio, ρ_1/ρ_2
- DO Base diameter, i.e., $d_k = d_{ok} \alpha_k^\gamma$
- DS Density scale = ρ_2
- EVF μ^t/μ effective eddy viscosity factor.
- GAMMA Diameter exponent γ
- 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, $MM \geq NG$. MM appears in many subroutines.
- MU μ_k , dynamic viscosity for phase k.
- MUEF $(1 + EVF) * MU$. Effective viscosity.
- MU1D2 μ_1/μ_2 , Viscosity ratio.
- NA, ND n_a, n_b , Exponents for weighting function for drag and added mass coefficients, A_a and A_d .

NAMR Data file name for initial condition, if any.

NG Number of grid points used.

OMEGA ω , initial rotation speed, $V_{\theta} = \omega r$, when IVTX > 0.

PS Pressure scale, $DS * VS^2$

QJ,VJ Injection flow rate and speed.

RE Reynolds number, $\rho_2 V_s R / \mu_2$

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 $K, \alpha_1(K), V_{r1}(K), V_{r2}(K), V_{\theta1}(K), V_{\theta2}(K)$, 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, ΔA

To determine the fraction of grid size in which the injection is made. $0 \leq \Delta A \leq 1$. This is used to define the location of jet. The region of injection could cover several full or fractions of grid sizes.

4) DERIV1, Derivative

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

5) DGCOEF, Generalized Coefficients

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

The effective coefficients are modeled as:

$$A_a = A_{a1} w_{a1} + A_{a2} w_{a2}$$

$$A_d = A_{d1} w_{d1} + A_{d2} w_{d2}$$

$$A_{a1} = \alpha_1 \alpha_2 \rho_2 / (\alpha_1 + 2\alpha_2 / (1 + 3\alpha_1))$$

$$A_{a2} = \alpha_1 \alpha_2 \rho_1 / (\alpha_2 + 2\alpha_1 / (1 + 3\alpha_2))$$

$$A_{d1} = 18 \mu_2 \alpha_1 / d_1^2 \alpha_2$$

$$A_{d2} = 18 \mu_1 \alpha_2 / ((1 - \alpha_2 / 0.8)^2 d_2^2)$$

$$w_{a1} = \alpha_2^{na} / (\alpha_1^{na} + \alpha_2^{na})$$

$$w_{a2} = 1 - w_{a1}$$

$$w_{d1} = \alpha_2^{nd} / (\alpha_1^{nd} + \alpha_2^{nd})$$

$$w_{d2} = 1 - w_{d1}$$

6) DWPDE, Partial Differential Equations.

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

7) FNDDT, Δt

To determine the suitable time-step size.

8) FSOFW, Column matrices F and S .

To determine the convective matrix F and the source matrix S .

9) JET, Injection.

To determine the momentum source due to the jet injection.

10) SIZES

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

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

Different models for size distributions could be easily adopted here.

11) TAUOFW

To determine the stress tensor τ and its derivative.

12) UOFW

To convert the column matrix W into the physical independent variables, such as α , V_r , V_θ .

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

1. Yeh, T. T. (1986) "A Vortex-Induced, Gas-Liquid Separation in a Cylindrical Tank at Zero Gravity", NBSIR 86-3322.
2. Drew, D. A. (1983) "Mathematical Modeling of Two-Phase Flow", Annual Review of Fluid Mechanics, Vol. 15.
3. MacCormack, R. W. (1969) "The Effect of Viscosity in Hypervelocity Impact Cratering", AIAA Paper 69-354.
4. Chima, R. V. and Johnson, G. M. (1985) "Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid Algorithm", AIAA Vol. 23, No. 1.
5. Peyret, R. and Taylor, T. D. (1983) "Computational Methods for Fluid Flow" Springer-Verlag, New York.

Appendix A

Code Listing

```

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

0001 FTN77
0002 $EMA /DATA/,/WWW/,/COEFF/,/SOURCE/,/FANDS/,/TAU/
0003 $FILES 1,2
0004 PROGRAM GLVM(,99),(860425.1537)
0005 C
0006 C THIS PROGRAM WAS DEVELOPED TO STUDY THE FLUID DYNAMICAL BEHAVIOR
0007 C OF A ROTATIONAL TWO-PHASE FLUIDS(GAS/LIQUID) IN A CYLINDRICAL TANK.
0008 C THE VORTEX MOTIONS ARE ESTABLISHED BY TANGENTIAL FLUID INJECTION.
0009 C THE PROGRAM WAS DEVELOPED ORIGINALLY BY T.T. YEH OF NBS
0010 C IT WAS BASED ON HP'S FORTRAN 77(ANSI 77+MIL-STD-1753)
0011 C
0012 C WHEN WHO WHAT
0013 C 8502XX TTY ZERO-G FUEL TRANSFER, START-UP STAGE.
0014 C LAX-WENDROFF 2-STEP SCHEME (FULL STEP PREDICTION+CORRECTION)
0015 C WITH NUMERICAL DAMPING FACTOR(NORMALLY SET TO ZERO)
0016 C CONSERVATION FORM, VARIABLE(AUTO) TIME STEP
0017 C REAL*8
0018 C INTERFACIAL FORCES: DRAG,ADDED MASS
0019 C PUMP CONDITION: MOMENTUM SOURCE BUT NO MASS SOURCE
0020 C 850715 TTY GENERALIZED EQUATIONS AND COEFF. Cij
0021 C 851018 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, DGCOEF, DWPDE,FNDDT,FSOFW,
0026 C INIT, JET, SIZES, TAUOFW and UOFW
0027 C *** MOST OF THE LIST OF NOTATIONS ARE GIVEN IN SUBROUTINE..INIT
0028 C
0029 CHARACTER NAMR*16, NOTES*72
0030 INTEGER I,IOS,J,JTIME(5),K,MM,NPRT,NT
0031 Z ,IW,LUP,LUS,NG,NGM1,NGM2
0032 PARAMETER (MM=101)
0033 REAL*8 BA(MM,2),DDT,DT,DTMAX,DTMIN,DTPRT,PZERO
0034 X ,DW1, T,TMAX,TPRT,UJT, UDR(2)
0035 Y ,RJ1,RJ2,TJ1,TJ2,QJ,UJ
0036 1 ,ALMT,U,V,ALP,P,R, W,WP,WN,DW,RDP,RH, F,S
0037 4 ,BR,BRH, RHO,MUEF,U18,NA,ND
0038 5 ,DO,GAMMA, DAMP,DR
0039 6 ,TRR,TRA,TAA,RTRR,RTRA, C,CPA,CD
0040
0041 COMMON
0042 Y /JETS/ RJ1,RJ2,TJ1,TJ2,QJ,UJ
0043 Z /CONTP/ IW,LUP,LUS,NG,NGM1,DAMP,DR
0044 1 /ALPLMT/ ALMT(2)
0045 2 /COEFF/ C(MM,2,2),CPA(MM,2),CD(MM,2)
0046 3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0047 4 /DRAG1/ RHO(4),MUEF(2),U18(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),TRA(MM,2),TAA(MM,2),RTRR(MM,2),RTRA(MM,2)
0051 9 /WWW/ W(MM,5),WP(MM,5),WN(MM,5),DW(MM,5),RDP(MM),RH(MM)
0052
0053 EQUIVALENCE (WN,BA)
0054
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

```

```

0059      LUP=1          ! LU FOR PRINTING DEBUG DATA(=1 TERMINAL)
0060      LUS=6          ! LU FOR STORING DATA(=6 PRINTER)
0061
0062 7    FORMAT(2X,A,3I5)
0063 8    FORMAT(2X,A,3(1PE12.4))
0064      WRITE(1,7) 'Enter lu for saving data. D.F.=',LUS
0065      READ(1,*) LUS
0066
0067 C    KEET JOB TIME FOR FUTURE REFERENCE
0068 C    CALL EXEC(11,JTIME,JTIME(1))
0069
0070      IF(LUS .NE. 1 .AND. LUS .NE. 6) THEN
0071 C    ***** Define a file name for stroing output *****
0072      WRITE(1,'(2A)') 'Enter FILE NAME for saving data.'
0073      READ(1,'(A)') NAMR
0074      LUS=90
0075      OPEN(LUS,FILE=NAMR,IOSTAT=IOS,STATUS='NEW',ERR=999)
0076      ENDIF
0077
0078      WRITE(1,'(A)') ' Enter NOTES((73 CHAR.) for the job'
0079      READ(1,'(A)') NOTES
0080      WRITE(LUS,'(3H1 ,A)') NOTES
0081 C    WRITE(LUS,'(5I4)') JTIME
0082
0083 C    To set-up the initial condition.
0084      CALL INIT
0085      NGM2=NG-2
0086      VDR(1)=2.*MUEF(1)/DR**2      ! For determining time step
0087      VDR(2)=2.*MUEF(2)/DR**2
0088
0089      T=0.          ! INITIAL TIME
0090      TMAX=5.
0091      DTPRT=0.2
0092      WRITE(1,8) 'Enter INITIAL and FINAL TIMES. D.F.=',T,TMAX
0093      READ(1,*) T,TMAX
0094      WRITE(1,8) 'Enter TIME STEP for output. D.F.=',DTPRT
0095      READ(1,*) DTPRT
0096
0097      DTMIN=1.0D-6      ! SET MINIMUM TIME STEP
0098      DTMAX=DTPRT
0099
0100      NT=0          ! Time step number
0101      TPRT=T
0102      NPRT=0
0103      PZERO=0.0D0      ! Pressure at center core
0104      DT=DTMIN
0105
0106 10    NT=NT+1
0107      CALL UDFW(W,R,NG)
0108
0109 C    PRINT OUT AT SELECTED TIME
0110      IF(T .GE. TPRT .OR. T .GT. TMAX) THEN
0111          IF(NT .GT. 1) THEN
0112              NPRT=NPRT+1
0113              TPRT=TPRT+DTPRT*(1.+DNINT((T-TPRT)/DTPRT))
0114 21    FORMAT(1H1,2(A5,I4,A5,1PE9.3))
0115              WRITE(LUP,21) 'NP=',NPRT,'T=',T,'NT=',NT,'DT=',DT
0116              WRITE(LUS,21) 'NP=',NPRT,'T=',T,'NT=',NT,'DT=',DT
0117
0118              WRITE(LUS,'(A5,A6,5A11)') 'J','ALP1','U1','U2','V1','V2','P'

```



```

0119          DO 30 J=1,NG
0120      30      WRITE(LUS,'(I5,F6.4,5(1PE11.3))')
0121      1          J,ALP(J,1),U(J,1),U(J,2),V(J,1),V(J,2),P(J)
0122          ENDIF
0123      ENDIF
0124
0125          IF( T .GT. TMAX) GOTO 9999
0126
0127      C          TO SOLVE THE DIFFERENTIAL EQUATIONS
0128      C          USING 2 STEP LAX-WENDROFF SCHEME
0129      C          MacCormack's method.. BACKWARD PREDICTOR, FORWARD CORRECT
0130      C          CENTER DIFFERENCED ON TAU
0131
0132      C          FIRST TO GET THE SPECIAL VARIABLES AND THEIR SPACIAL DERIVATIVES
0133
0134          CALL DGCOEF(ALP,NG)                ! DRAG AND GENERIZED COEFF.
0135          CALL TAUOFW(MUEF,DR,R,NG)          ! STRESS
0136
0137
0138          VJT=VJ
0139          IF( T .LT. TJ1 .OR. T .GT. TJ2) VJT=0. ! NO INJETCTION
0140          CALL JET(BA,RHO,BR,V,VJT,NG)       ! MOMENTUM SOURCE
0141          CALL FSOFW(W,BA,R,NG)              ! CONVECTIVE -F AND SOURCE-S
0142
0143      C          DETERMINE THE TIME-STEP SIZE
0144          IF( NT .GT. 1) THEN
0145              CALL FNDDT(DT,DR,VDR,NG)
0146              DT=DT+S.*DTMIN/10.**NT        ! INITIAL INPLUSE TREATMENT
0147              IF(DT .GE. DTMIN) THEN
0148                  I=DLOG10(DT/DTMIN)        ! ROUND OFF TIME STEP
0149                  DDT=DTMIN*10.**I
0150                  DT=DINT(DT/DDT+0.001)*DDT
0151                  IF( DT .GT. DTMAX) DT=DTMAX
0152              ELSE
0153                  WRITE(LUP,'(5X,A,1PE13.3)')
0154      1          'STOP DUE TO TOO SMALL TIME STEP. DT=',DT
0155                  GOTO 9999
0156              ENDIF
0157          ENDIF
0158
0159      C          BACKWARD PREDICTOR
0160
0161          CALL DWPDE(DW,RDP, DR,DT,RH,NGM1)   ! INCREMENT
0162
0163          DO 40 J=1,NGM1
0164              DO 40 I=1,5
0165      40          WP(J,I)=0.5*(W(J+1,I)+W(J,I))+DW(J,I) ! BASE+INCREAM
0166
0167      C          PREDICTION DATA COMPLETED, CONTINUE FOR CORRECTION
0168
0169          CALL UOFW(WP,RH,NGM1)
0170          CALL DGCOEF(ALP,NGM1)
0171          CALL TAUOFW(MUEF,DR,RH,NGM1)
0172          CALL JET(BA,RHO,BRH,V,VJT,NG)
0173          CALL FSOFW(WP,BA,RH,NGM1)
0174
0175      C          FORWARD CORRECTION
0176          CALL DWPDE(DW,P(2), DR,DT,R(2),NGM2)
0177          P(1)=PZERO
0178          DO 50 J=1,NGM2

```

```

0179      P(J+1)=P(J)+(0.25*(RDP(J)+RDP(J+1))+0.5*P(J+1))/R(J)
0180      DO 50 I=1,5
0181 50      WN(J+1,I)=.25*(WP(J+1,I)+WP(J,I))+.5*(W(J+1,I)+DW(J,I))
0182      P(NG)=P(NGM1)
0183
0184 C      2ND STEP (PREDICTION+CORRECTION) COMPLETED
0185
0186      IF( NT .EQ. 1) GOTO 10
0187 C      Estimation of initial P completed, return to the initial condition
0188 C      and start to advance the program in time.
0189
0190 *****
0191 C      DATA W AT THE NEW TIME STEP COMPLETED
0192
0193 C      IMPOSED B.C. #6.4
0194      DW1=.5*(WN(2,1)+W(2,1))-0.
0195      ALP(1,1)=(.5*(W(1,5)+W(2,5))-DW1*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
0200      DW1=0.-0.5*(WN(NGM1,1)+W(NGM1,1))
0201      ALP(NG,1)=(.5*(W(NGM1,5)+W(NG,5))-DW1*DT/DR)/RH(NGM1)
0202      IF(ALP(NG,1) .LT. ALMT(1)) ALP(NG,1)=ALMT(1)
0203      IF(ALP(NG,1) .GT. ALMT(2)) ALP(NG,1)=ALMT(2)
0204      WN(NG,5)=ALP(NG,1)*R(NG)
0205
0206      WN(1,1)=0.          ! NO RADIAL VEL.
0207      WN(1,2)=0.
0208      WN(NG,1)=0.
0209      WN(NG,2)=0.
0210      WN(1,3)=WN(2,3)*WN(1,5)/WN(2,5)*R(1)/R(2) ! OMEGA=CON.
0211      WN(1,4)=WN(2,4)*(R(1)-WN(1,5))/(R(2)-WN(2,5))*R(1)/R(2)
0212      IF(IW .EQ. 0) THEN
0213      WN(NG,3)=WN(NGM1,3)*WN(NG,5)/WN(NGM1,5)*R(NG)/R(NGM1)
0214      WN(NG,4)=WN(NGM1,4)*(R(NG)-WN(NG,5))/(R(NGM1)-WN(NGM1,5))
0215 1      *R(NG)/R(NGM1)
0216      ELSE
0217      WN(NG,3)=0.          ! NON-SLIP AT WALL
0218      WN(NG,4)=0.
0219      ENDIF
0220
0221 C      ARTIFICIAL DAMPING
0222      DO 60 I=1,5
0223      W(1,I)=(1.-DAMP)*WN(1,I)+DAMP*W(2,I)
0224      W(NG,I)=(1.-DAMP)*WN(NG,I)+DAMP*W(NGM1,I)
0225      DO 60 J=2,NGM1
0226 60      W(J,I)=(1.-DAMP)*WN(J,I)+DAMP*(WN(J-1,I)+WN(J+1,I)-WN(J,I))
0227
0228 C      SOLUTION FOR THIS TIME STEP COMPLETED
0229
0230
0231 90      T=T+DT          ! UPDATE TIME AND CONTINUE TO THE NEXT STEP
0232      GOTO 10
0233
0234 999      WRITE(LUP,7) 'OPEN FILE FAILED ON FILE:'
0235      WRITE(LUP,7) NAMR
0236      WRITE(LUP,7) 'IOSTAT=',IOS
0237
0238 9999      CONTINUE

```

```

0239 C    CALL EXEC(11,JTIME,JTIME(1))
0240 C    WRITE(LUS,'(35X,5I4)') JTIME
0241     CLOSE(LUS)
0242
0243     END
0244
0245 *****
0246 $EMA /DATA/,/WWW/,/SOURCE/
0247     SUBROUTINE INIT ,(860425.1537)
0248
0249 C    TO SET-UP THE INITIAL CONDITIONS
0250     INTEGER I,IOS,ITLOG,IVTX,J,K,MM
0251     Z          ,ITIME,IW,LUP,LUS,NG,NGM1
0252     PARAMETER (MM=101)
0253     CHARACTER NAMR*16
0254     REAL*8 ALMT, W,WP,WN,DW,RDP,RH
0255     1          ,U,V,ALP,P,R, F,S
0256     4          ,RHO,MUEF,V18,NA,ND, DO,GAMMA
0257     7          ,BR,BRH, DAMP,DR
0258     Y          ,RJ1,RJ2,TJ1,TJ2,QJ,VJ
0259     X          ,A2,DELA,DEN1D2,D1,D2,PI,RE,RMIN,RTANK
0260     X          ,DS,LS,VS,TS,PS
0261     X          ,ALP10,OMEGA,RPEAK,RJB,VPEAK
0262     X          ,EVF(2),VT,MU(2),MU1D2
0263
0264     COMMON
0265     1 /ALPLMT/ ALMT(2)
0266     Y /JETS/ RJ1,RJ2,TJ1,TJ2,QJ,VJ
0267     Z /CONTP/ IW,LUP,LUS,NG,NGM1,DAMP,DR
0268     3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0269     4 /DRAG1/ RHO(4),MUEF(2),V18(2),NA,ND
0270     5 /DSIZE/ DO(2),GAMMA(2)
0271     7 /SOURCE/ BR(MM),BRH(MM)
0272     9 /WWW/ W(MM,5),WP(MM,5),WN(MM,5),DW(MM,5),RDP(MM),RH(MM)
0273
0274     DATA PI/3.141596D0/
0275
0276
0277 C ***** NOTES: PHASE-1=GAS, PHASE-2=LIQUID *****
0278 C RHO DENSITY, RHO(1) ( RHO(2)
0279 C ALMT LIMIT VALUES FOR ALP1, ALM(1)(ALP1(ALM(2)
0280 C DAMP NUMERICAL DAMPING FACTOR, NORMALLY =0.
0281 C DS,LS,VS,TS DENSITY,LENGTH,VELOCITY AND TIME SCALES
0282 C RE REYNOLDS  $\# = V_s * RTANK * RHO(2) / MU(2)$ 
0283 C RTANK TANK RADIUS
0284 C IVTX TYPE OF INITIAL FLOW.
0285 C (0=Rest,1=Pure rotation,2=H.O,3=GIT Vortex)
0286 C RPEAK,VPEAK VORTEX PARAMETERS
0287 C OMEGA PURE ROTATION.  $V = OMEGA * r$ 
0288 C EVF,MUEF EFFECTIVE VISCOSITY,  $MUEF = (1 + EVF) * MU$ 
0289 C DO,GAMMA DIA. PARAMETERS:  $D = DO * ALP * GAMMA$ 
0290 C RJ1,RJ2,TJ1, TO DEFINE JET SIZE, PUMPING TIME,
0291 C TJ2,QJ,VJ VOLUME FLOW RATE AND INJECTION MEAN SPEED
0292 C
0293
0294 5     FORMAT(2X,A,2(X,A))
0295 7     FORMAT(2X,A,3I5)
0296 8     FORMAT(2X,A,3(1PE12.4))
0297 9     FORMAT(A25,2(1PE15.4))
0298 92    FORMAT(X,7(1PE11.4))

```



```

0299
0300 C   DEFINE THE PARAMETERS FOR THE PROBLEM.
0301
0302     RTANK=1.D0           ! COULD BE SET TO 1 (M)
0303     RHO(2)=1.000D+3     ! " (KG/M**3)
0304     MU(2)=1.514D-3     ! " (KG/M-S)
0305
0306 C   WRITE(1,8) 'ENTER RTANK(M) OR DEFAULT ',RTANK
0307 C   READ(1,*) RTANK
0308
0309 C   The values of RTANK,RHO(2),and MU(2) could all be set to 1, since
0310 C   the length and density scales are based on RTANK,and RHO(2) and the
0311 C   value of the viscosity MU(2) can be combined into and specified by the
0312 C   Reynolds number RE. Thus all characteristic scales(LS,VS,TS,and DS)
0313 C   are fixed after RE is given.
0314
0315     RE=1.0D5             ! LS**2/(NU*TS)=LS*VS/NU
0316 C   WRITE(1,8) 'enter Reynolds no.,RE. D.F.=',RE
0317 C   READ(1,*) RE
0318     LS=RTANK             ! LENGTH SCALE (M)
0319     DS=RHO(2)           ! DENSITY SCALE (KG/M**3)
0320     VS=RE*MU(2)/RHO(2)/LS ! VELOCITY SCALE(M/S)
0321     TS=LS/VS           ! TIME SCALE (S)
0322     PS=DS*VS**2       ! PRESSURE SCALE
0323
0324
0325 C   AFTER THIS POINT ALL VARIABLES ARE BASED ON THE CHARA. SCALES
0326 C   i.e. ALL VARIABLES ARE DIMENSIONLESS
0327
0328
0329     DEN1D2=1.293D0/1.000D3 ! D1/D2
0330     MU1D2=1.71D-5/1.514D-3 ! MU1/MU2
0331
0332     RHO(2)=RHO(2)/DS
0333     MU(2)=MU(2)/(DS*LS*VS) ! =1/RE
0334     RHO(1)=DEN1D2*RHO(2)
0335     MU(1)=MU1D2*MU(2)
0336
0337     ALMT(1)=0.00010       ! MIN. OF ALP1
0338     ALMT(2)=0.99999       ! MAX. OF ALP1
0339     DO(1)=1.D-2/LS        ! GAS DIAMETER at ALP1=1
0340     DO(2)=1.D-2/LS        ! LIQUID DIAMETER at ALP2=1
0341     GAMMA(1)=2.D-1        !
0342     GAMMA(2)=2.D-1
0343     EVF(2)=1.D3           ! TURB.+PHASE-DISPERSION EFFECTS
0344     EVF(1)=DEN1D2/MU1D2*EVF(2) ! MODEL
0345     DAMP=0.0              ! NUMERICAL DAMPING FACTOR(e.g.=.2)
0346     NA=4.                 ! WEIGHTING EXP. FOR ADM
0347     ND=4.                 ! WEIGHTING EXP. FOR DRAG
0348
0349 C   WRITE(1,8) 'Enter DENSITY and VISCOSITY ratios.'
0350 C   WRITE(1,8) 'D.F.=',DEN1D2,MU1D2
0351 C   READ(1,*) DEN1D2,MU1D2
0352
0353 C   WRITE(1,8) 'Enter BASE DIAMETERS: DO1,DO2'
0354 C   WRITE(1,8) 'D.F.=',DO
0355 C   READ(1,*) DO
0356
0357 C   WRITE(1,8) 'Enter SIZE EXPONENT: GAMMA1,GAMMA2'
0358 C   WRITE(1,8) 'D. F. =',GAMMA

```

```

0359      READ(1,*) GAMMA
0360
0361      WRITE(1,8) 'Enter weighting exponent: NA,ND. D.F.=',NA,ND
0362      READ(1,*) NA,ND
0363
0364      WRITE(1,8) 'Enter GAS VOLUME FRACTION limits:ALMT1,ALMT2.'
0365      WRITE(1,8) 'D.F.=',ALMT
0366      READ(1,*) ALMT
0367
0368      WRITE(1,8) 'Enter eddy viscosity factor. D.F.=',EVF
0369      READ(1,*) EVF
0370
0371      IW=1
0372      WRITE(1,7) 'Enter wall condition,1=nonslip,0=slip. D.F.',IW
0373      READ(1,*) IW
0374
0375      WRITE(1,8) 'Enter numerical damping factor. D.F.=',DAMP
0376      READ(1,*) DAMP
0377
0378      DO 10 K=1,2
0379          V18(K)=18.*MU(K)
0380      10  MUEF(K)=MU(K)*(1.+EVF(K)) ! EFFECTIVE VISCOSITY FOR STRESS
0381
0382      RHO(3)=RHO(1)*RHO(2)
0383      RHO(4)=RHO(1)-RHO(2)
0384
0385      RMIN=0.1 ! MINIMUM FLOW RADIUS IN THE TANK
0386      NG=101 ! # OF GRID POINTS USED
0387      NGM1=NG-1
0388      DR=(1.-RMIN)/NGM1
0389
0390  C      Initial cleanning-up.
0391          DO 15 J=1,MM
0392              DO 15 K=1,6
0393                  W(J,K)=0.D0
0394      15  U(J,K)=0.D0
0395
0396  C      MOMENTUM SOURCE, JET CONDITIONS
0397          RJ1=8.5D-1
0398          RJ2=9.5D-1
0399          VJ=10. ! TANGENTIAL INJECTION SPEED
0400          TJ1=0.
0401          TJ2=10.
0402
0403          WRITE(1,8) 'Enter JET SIZE defined by RJ1,RJ2. D.F.=',RJ1,RJ2
0404          READ(1,*) RJ1,RJ2
0405          WRITE(1,8) 'Enter INJECTION SPEED AND TIME RANGE, VJ,T1,T2'
0406          WRITE(1,8) 'D.F.=',VJ,TJ1,TJ2
0407          READ(1,*) VJ,TJ1,TJ2
0408          QJ=(RJ2-RJ1)*VJ ! JET VOLUME FLOW RATE
0409
0410          DO 20 J=1,NG
0411              R(J)=RMIN+(J-1)*DR
0412              RH(J)=R(J)+0.5*DR
0413              BR(J)=DELA((R(J)),DR,RJ1,RJ2)/(2.*PI) ! JET DISTRIBUTION
0414      20  BRH(J)=DELA((RH(J)),DR,RJ1,RJ2)/(2.*PI) ! PER RADIAN
0415
0416
0417  C      SETUP INITIAL CONDITIONS
0418          IVTX=0

```

```

0419      OMEGA=0.
0420      VPEAK=0.
0421      RPEAK=RMIN
0422
0423      NAMR='Simple vortex'
0424      WRITE(1,5) 'Enter data FILE NAME for initial cond., if any;'
0425      WRITE(1,5) 'D.F.=',NAMR
0426      READ(1,'(A)') NAMR
0427
0428      IF( NAMR .NE. ',' .AND. NAMR .NE. 'Simple vortex') THEN
0429 C      INITIAL CONDITION FROM A GIVEN FILE NAMR.
0430          OPEN(99,FILE=NAMR,IOSTAT=IOS,STATUS='OLD',ERR=299)
0431          DO 25 J=1,NG          ! INITIAL VALUES FROM FILE NAMR
0432      25      READ(99,*) K,ALP(J,1),U(J,1),U(J,2),V(J,1),V(J,2)
0433          CLOSE(99)
0434
0435      ELSE
0436 C      TO DEFINE INITIAL CONDITION.
0437          ALP10=2.5D-1          ! INITIAL GAS VOL. FRACTION
0438          WRITE(1,8) 'Enter initial value of alp1. D.F.=',ALP10
0439          READ(1,*) ALP10
0440
0441          WRITE(1,7) 'Enter type of vortex: 0=At rest,1=pure rotation'
0442          WRITE(1,7) '2=H.O.,3=GIT. D.F.=',IVTX
0443          READ(1,*) IVTX
0444          IF(IVTX .GT. 0) THEN
0445              IF(IVTX .GT. 1) THEN
0446                  WRITE(1,8) 'Enter PEAK VEL. and LOCATION for classic vortex'
0447                  WRITE(1,8) 'D.F.=',VPEAK,RPEAK
0448                  READ(1,*) VPEAK,RPEAK
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          ,OMEGA
0453                  READ(1,*) OMEGA
0454              ENDIF
0455          ENDIF
0456
0457          DO 30 J=1,NG
0458              ALP(J,1)=ALP10
0459              VT=OMEGA*R(J)          ! PURE ROTATION
0460              IF(IVTX .GT. 0) THEN
0461                  RJB=R(J)/RPEAK
0462                  IF( IVTX .EQ. 1) THEN          ! H.O. VORTEX
0463                      VT=VT+1.398*VPEAK/RJB*(1.-DEXP(-1.25643*RJB**2))
0464                  ELSE          ! G.I.T. VORTEX
0465                      VT=VT+VPEAK*RJB*DEXP((1.-RJB**2)/2.)
0466                  ENDIF
0467              ENDIF
0468              DO 30 K=1,2
0469                  IF(IW .EQ. 0) THEN
0470                      V(J,K)=VT          ! NO WALL
0471                  ELSE
0472                      V(J,K)=VT*(1.-R(J))**0.1          ! BOUNDARY LAYER
0473                  ENDIF
0474      30      U(J,K)=0.          ! NO RADIAL VEL.
0475
0476          ENDIF
0477
0478          DO 40 J=1,NG          ! FORM W FOR NUMERICAL CAL.

```

```

0479     ALP(J,2)=1.-ALP(J,1)
0480     P(J)=0.D0
0481     W(J,5)=R(J)*ALP(J,1)
0482     DO 40 K=1,2
0483         W(J,K)=ALP(J,K)*U(J,K)*R(J)
0484     40   W(J,K+2)=ALP(J,K)*V(J,K)*R(J)
0485
0486
0487 C     PRINTOUT PARAMETERS
0488     WRITE(LUS,5) 'INITIAL CONDITION FILE:',NAMR
0489     WRITE(LUS,5) '**DIMENSION UNITS ARE IN MKS**'
0490     WRITE(LUS,9) 'DENSITY SCALE(kg/m3)',DS
0491     WRITE(LUS,9) 'LENGTH SCALE=RTANK,(m)',LS
0492     WRITE(LUS,9) 'VELOCITY SCALE(m/s)',VS
0493     WRITE(LUS,9) 'TIME SACLE(s)',TS
0494     WRITE(LUS,9) 'PRESSURE SCALE(Pa)',PS
0495     WRITE(LUS,*)
0496     WRITE(LUS,9) 'Reynolds number, Re',RE
0497     WRITE(LUS,9) 'Jet size, RJ1,RJ2',RJ1,RJ2
0498     WRITE(LUS,9) 'Tangential jet, QJ,VJ',QJ,VJ
0499     WRITE(LUS,9) 'Injection time,TJ1,TJ2',TJ1,TJ2
0500
0501     WRITE(LUS,'(/33X,"PHASE-1",8X,"PHASE-2")')
0502     WRITE(LUS,9) 'Density',RHO(1),RHO(2)
0503     WRITE(LUS,9) 'Viscosity',MU
0504     WRITE(LUS,9) 'Eddy viscosity factor',EVF
0505     WRITE(LUS,9) 'Base dia.',DO
0506     WRITE(LUS,9) 'Size exp.',GAMMA
0507     WRITE(LUS,9) 'Phase limits',ALMT(1),1.-ALMT(2)
0508     WRITE(LUS,*)
0509     WRITE(LUS,*)
0510     WRITE(LUS,'(" OTHER CONSTANS: IW,IUTX,NA,ND,DAMP,VPEAK,RPEAK",
0511 1      ",OMEGA,D1/D2,MU1/MU2")')
0512     WRITE(LUS,'(2I3,2F4.0,F5.2)') IW,IUTX,NA,ND,DAMP
0513     WRITE(LUS,92) VPEAK,RPEAK,OMEGA,DEN1D2,MU1D2
0514     WRITE(LUS,*)
0515     WRITE(LUS,'(I10,F10.2)') NG,RMIN
0516
0517     RETURN
0518
0519 299 WRITE(LUP,5) 'OPEN FILE FAILED ON INPUT FILE:',NAMR
0520     WRITE(LUP,7) 'IOSTAT=',IOS
0521     STOP 111
0522     END
0523
0524 *****
0525     REAL*8 FUNCTION DELA(R,DR,RJ1,RJ2),(860425.1537)
0526 C     TO DETERMINE THE EFFECTIVE NOZZLE SIZE AT EACH GRID LOCATION
0527 C     THE SIZE IS IN THE FRACTION OF GRID SIZE DR (i.e. 0<DELA<1)
0528
0529     REAL*8 R,DR,RJ1,RJ2, R1,R2
0530
0531     R1=R-0.5*DR
0532     R2=R1+DR
0533     DELA=0.D0
0534     IF(R1 .GE. RJ2 .OR. R2 .LE. RJ1) RETURN
0535     IF(R1 .LT. RJ1) R1=RJ1
0536     IF(R2 .GT. RJ2) R2=RJ2
0537     DELA=(R2-R1)/DR
0538     RETURN

```



```

0539         END
0540
0541 *****
0542 SUBROUTINE DERIV1(Y,DY,DX,N2),<860425.1537>
0543 C   GET 1ST DERIVATIVE, USING CENTERED DIFFERENCE
0544 REAL*8 DX,Y(1),DY(1),C
0545 EMA Y,DY
0546
0547 C=5.D-1/DX
0548 DO 10 J=2,N2-1
0549 10  DY(J)=C*(Y(J+1)-Y(J-1))
0550 *V  CALL DWSUB(Y(3),1,Y,1,DY(2),1,N2-2)
0551
0552 DY(1)=(Y(2)-Y(1))/DX      ! BASED ON 3-END PTS
0553 DY(N2)=(Y(N2)-Y(N2-1))/DX
0554 DY(1)=2.*DY(1)-DY(2)
0555 DY(N2)=2.*DY(N2)-DY(N2-1)
0556
0557 *V  CALL DWSMY(5.D-1/DX,DY,1,DY,1,N2)
0558
0559 RETURN
0560 END
0561
0562 *****
0563 $EMA /COEFF/
0564 SUBROUTINE DGCOEF(ALP,N2),<860425.1537>
0565 C   CALCULATE THE DRAG, ADDED MASS AND GENERIZED COEFF.
0566
0567 INTEGER J,MM,N2
0568 PARAMETER (MM=101)
0569 REAL*8 ALP(MM,2)
0570 EMA ALP
0571
0572 REAL*8 C,CPA,CD, RHO,MUEF,V18,NA,ND
0573 X   ,AA,AA1,AA2,AD,AD1,AD2,A1,A2,A12,DA12,DB2, D1,D2,WT1,WT2,X
0574 COMMON
0575 1 /COEFF/ C(MM,2,2),CPA(MM,2),CD(MM,2)
0576 4 /DRAG1/ RHO(4),MUEF(2),V18(2),NA,ND
0577
0578 DO 50 J=1,N2
0579 A1=ALP(J,1)
0580 A2=ALP(J,2)
0581 A12=A1*A2
0582
0583
0584 C   TO GET DRAG COEFF. AD
0585 CALL SIZES(D1,D2,A1)
0586
0587 AD=V18(2)*A1/(A2*D1)      ! =AD1 IF A2>.78
0588
0589 IF( A2 .LT. .78) THEN
0590 AD2=V18(1)*A2/((1,-A2/.8)*D2)**2
0591 IF(AD2 .LT. AD) THEN
0592 WT1=A2**ND
0593 WT2=A1**ND
0594 AD=(AD*WT1+AD2*WT2)/(WT1+WT2)
0595 ENDIF
0596 ENDIF
0597
0598 C   ADDED MASS COEFF. AA

```

```

0599      AA1=A12*RHO(2)/(A1+A2/(.5+1.5*A1))
0600      AA2=A12*RHO(1)/(A1/(.5+1.5*A2)+A2)
0601
0602      WT1=A2**NA
0603      WT2=A1**NA
0604      AA=(AA1*WT1+AA2*WT2)/(WT1+WT2)
0605
0606 C      THE GENERALIZED COEFF. CPA,C,AND CD
0607      DB2=A12*RHO(3)+AA*(RHO(1)*A1+RHO(2)*A2)
0608      CPA(J,1)=A1*(A12*RHO(2)+AA)/DB2
0609      CPA(J,2)=A2*(A12*RHO(1)+AA)/DB2
0610      C(J,1,2)=AA/DB2
0611      C(J,2,1)=C(J,1,2)
0612      C(J,1,1)=A2*RHO(2)/DB2+C(J,1,2)
0613      C(J,2,2)=A1*RHO(1)/DB2+C(J,2,1)
0614      CD(J,1)=-A2*RHO(2)*AD/DB2
0615      CD(J,2)=A1*RHO(1)*AD/DB2
0616
0617      50 CONTINUE
0618
0619      RETURN
0620      END
0621
0622 *****
0623 $EMA /DATA/,/FANDS/,/TAU/,/COEFF/
0624      SUBROUTINE DWPDE(DW,RDP, DR,DT,RR,N2),(<860425.1537>)
0625 C
0626 C      TO GET DW OF THE PDEs
0627 C
0628      INTEGER J,JP1,K,KP2,MM,N2
0629      PARAMETER (MM=101)
0630      REAL*8 DW(MM,5),RDP(MM), DR,DT,RR(MM)
0631      EMA DW,RDP,RR
0632
0633 C      NOTES: COEFF. C =C*ALP WHEN THIS IS CALLED
0634      REAL*8 C,CPA,CD, U,V,ALP,P,R, F,S
0635      8      ,TRR,TRA,TAA,RTRR,RTRA
0636      4      ,RHO,MUEF,V18,NA,ND
0637      X      ,ALP1,ALP2,CP1,CP2,DTDR,DW1,DW1DT,G1,G2,HDT,VT,WJ1,WJ3,WJ4
0638
0639      COMMON
0640      1 /COEFF/ C(MM,2,2),CPA(MM,2),CD(MM,2)
0641      3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0642      4 /DRAG1/ RHO(4),MUEF(2),V18(2),NA,ND
0643      6 /FANDS/ F(MM,5),S(MM,5)
0644      8 /TAU/ TRR(MM,2),TRA(MM,2),TAA(MM,2),RTRR(MM,2),RTRA(MM,2)
0645
0646      DTDR=DT/DR
0647      HDT=0.5*DT
0648
0649      DO 10 J=1,N2+1      ! CHANGE C TO ALP*C
0650      C(J,1,1)=ALP(J,1)*C(J,1,1)
0651      C(J,2,1)=ALP(J,2)*C(J,2,1)
0652      C(J,1,2)=ALP(J,1)*C(J,1,2)
0653      10 C(J,2,2)=ALP(J,2)*C(J,2,2)
0654
0655      DO 20 J=1,N2
0656      JP1=J+1
0657      DW(J,5)=DTDR*(-F(JP1,5)+F(J,5))+HDT*(S(JP1,5)+S(J,5))
0658      DO 25 K=1,2

```

```

0659      G1=.5*(C(JP1,K,1)+C(J,K,1))
0660      G2=.5*(C(JP1,K,2)+C(J,K,2))
0661      DW(J,K)=DTDR*(-F(JP1,K)+F(J,K)+G1*(RTRR(JP1,1)-RTRR(J,1))
0662      1          +G2*(RTRR(JP1,2)-RTRR(J,2)))
0663      2          +HDT*(S(JP1,K)+S(J,K))
0664      KP2=K+2
0665      25      DW(J,KP2)=DTDR*(-F(JP1,KP2)+F(J,KP2)
0666      1          +G1*(RTRA(JP1,1)-RTRA(J,1))
0667      2          +G2*(RTRA(JP1,2)-RTRA(J,2)))
0668      3          +HDT*(S(JP1,KP2)+S(J,KP2))
0669
0670      C      DP FOR PRESSURE CORRECTION
0671      CP1=0.5*(CPA(J,1)+CPA(JP1,1))
0672      CP2=0.5*(CPA(J,2)+CPA(JP1,2))
0673
0674      IF( -DW(J,1) .GT. DW(J,2)) DW(J,1)=-DW(J,2) ! DP>=0
0675      RDP(J)=(DW(J,1)+DW(J,2))/(CP1+CP2)
0676      DW(J,1)=DW(J,1)-CP1*RDP(J)
0677
0678      DW(J,2)=-DW(J,1)
0679      20      RDP(J)=RDP(J)/DT
0680
0681      RETURN
0682      END
0683
0684      *****
0685      $EMA /COEFF/,/DATA/,/WWW/
0686      SUBROUTINE FNDDT(DT,DR,VDR,NG),(<860425.1537)
0687
0688      C      DETERMINE THE TIME-STEP SIZE
0689
0690      INTEGER I,J,LUP,MM,NG
0691      PARAMETER (MM=101)
0692      REAL*8 DT,DR,VDR(2)
0693
0694      REAL*8 C,CPA,CD, RHO,MUEF,V18,NA,ND
0695      9      ,W,WP,WN,DW,RDP,RH, U,V,ALP,P,R
0696      X      ,DUM1,DUM2
0697
0698      COMMON
0699      2 /COEFF/ C(MM,2,2),CPA(MM,2),CD(MM,2)
0700      3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0701      4 /DRAG1/ RHO(4),MUEF(2),V18(2),NA,ND
0702      9 /WWW/ W(MM,5),WP(MM,5),WN(MM,5),DW(MM,5),RDP(MM),RH(MM)
0703      DATA LUP/1/
0704
0705      DUM1=0.
0706      DO 10 J=1,NG
0707      DUM2=-CD(J,1)
0708      1          +DABS(U(J,1))/DR ! 1/St
0709      2          +VDR(1)*ALP(J,1)*C(J,1,1) ! CONVECTIVE
0710      3          +VDR(2)*ALP(J,2)*C(J,1,2) ! CROSS VISICOSITY
0711      IF(DUM1 .LT. DUM2) DUM1=DUM2
0712      DUM2=CD(J,2)+DABS(U(J,2))/DR
0713      1          +VDR(2)*ALP(J,2)*C(J,2,2)
0714      2          +VDR(1)*ALP(J,1)*C(J,2,1)
0715      IF(DUM1 .LT. DUM2) DUM1=DUM2
0716      10      CONTINUE
0717
0718      C      FIND THE MAXIMUN OF DW

```

```

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 *V DT=1.D0/DUM1
0724
0725 RETURN
0726 END
0727
0728 *****
0729 $EMA /DATA/,/FANDS/,/TAU/,/COEFF/
0730 SUBROUTINE FSOFW(W,BA,RR,N2),(860425.1537)
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,P,R, F,S, RHO,MUEF,V18,NA,ND
0737 2 ,TRR,TRA,TAA,RTRR,RTRA, C,CPA,CD
0738 X ,ALPD(2),RDU,RDV
0739
0740 REAL*8 W(MM,5),BA(MM,2),RR(1)
0741 EMA W,BA,RR
0742
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 /DRAG1/ RHO(4),MUEF(2),V18(2),NA,ND
0747 6 /FANDS/ F(MM,5),S(MM,5)
0748 8 /TAU/ TRR(MM,2),TRA(MM,2),TAA(MM,2),RTRR(MM,2),RTRA(MM,2)
0749
0750 *V CALL DWMPY(W,1,U,1,F,1,2*MM)
0751 *V CALL DWMPY(W(1,3),1,U,1,F(1,3),1,2*MM)
0752 *V CALL DWMOV(W,1,F(1,5),1,MM)
0753
0754 DO 20 J=1,N2
0755 F(J,5)=W(J,1)
0756 S(J,5)=0.
0757
0758 RDU=RR(J)*(U(J,1)-U(J,2))
0759 RDV=RR(J)*(V(J,1)-V(J,2))
0760 DO 20 K=1,2
0761 KP2=K+2
0762 F(J,K)=W(J,K)*U(J,K)
0763 F(J,KP2)=W(J,KP2)*U(J,K)
0764
0765 S(J,K)=ALP(J,K)*(V(J,K)**2+CD(J,K)*RDU-C(J,K,1)*TAA(J,1)
0766 1 -C(J,K,2)*TAA(J,2))
0767 20 S(J,KP2)=ALP(J,K)*(-U(J,K)*V(J,K)+CD(J,K)*RDV+C(J,K,1)*(BA(J,1)
0768 1 +TRA(J,1))+C(J,K,2)*(BA(J,2)+TRA(J,2)))
0769
0770 RETURN
0771 END
0772
0773
0774 *****
0775 SUBROUTINE JET(BA,RHO,BR,U,UJ,NG)
0776 C
0777 C INJECTION MOMENTUM SOURCE
0778 C

```



```

0779     PARAMETER (MM=101)
0780     REAL*8 BA(MM,2),RHO(1),BR(1),V(MM,2),VJ
0781     EMA BA,BR,V
0782
0783     DO 10 J=1,NG
0784     IF( BR(J) .GT. 0.) THEN
0785         Q=BR(J)*VJ                               ! Flow rate/radian
0786         BA(J,1)=Q*(VJ-V(J,1))*RHO(1)           ! NET MOMENTUM GAIN
0787         BA(J,2)=Q*(VJ-V(J,2))*RHO(2)
0788     ELSE
0789         BA(J,1)=0.D0
0790         BA(J,2)=0.D0
0791     ENDIF
0792 10 CONTINUE
0793     RETURN
0794     END
0795
0796 *****
0797     SUBROUTINE SIZES(D1,D2,ALP1),(<860425.1537>)
0798 C     TO DETERMINE THE PARTICLE DIAMETERS
0799 C
0800     REAL*8 D1,D2,ALP1, DO,GAMMA
0801     COMMON /DSIZE/ DO(2),GAMMA(2)
0802
0803     D1=DO(1)*ALP1**GAMMA(1)
0804     D2=DO(2)*(1.0D0-ALP1)**GAMMA(2)
0805
0806     RETURN
0807     END
0808
0809 *****
0810 $EMA /DATA/,/TAU/
0811     SUBROUTINE TAUOFW(MU,DR,RR,N2),(<860425.1537>)
0812 C     STRESSES AND THEIR DERIVATIVES
0813 C
0814     REAL*8 MU(2),DR,RR(1)
0815     EMA RR
0816
0817     PARAMETER (MM=101)
0818     REAL*8 U,V,ALP,P,R
0819     2     ,TRR,TRA,TAA,RTRR,RTRA
0820     X     ,TAMU,TMU
0821
0822     COMMON
0823     3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0824     8 /TAU/ TRR(MM,2),TRA(MM,2),TAA(MM,2),RTRR(MM,2),RTRA(MM,2)
0825
0826     DO 50 K=1,2
0827     CALL DERIV1(U(1,K),TRR(1,K),DR,N2)           ! dU/dR
0828 *V     CALL DWMPY(ALP(1,K),1,TRR(1,K),1,TRR(1,K),1,N2) ! *ALP
0829 *V     CALL DWSMY(2.*MU(K),TRR(1,K),1,TRR(1,K),1,N2) ! *2*MU=TRR
0830 *V     CALL DWDIV(V(1,K),1,RR,1,TAA(1,K),1,N2) ! V/R
0831     DO 10 J=1,N2
0832 10     TAA(J,K)=V(J,K)/RR(J)
0833     CALL DERIV1(TAA(1,K),TRA(1,K),DR,N2)         ! d()/dR
0834
0835     TMU=2.*MU(K)
0836     DO 20 J=1,N2
0837     TAMU=TMU*ALP(J,K)
0838     TRR(J,K)=TAMU*TRR(J,K)

```

```

0839          TRA(J,K)=MU(K)*ALP(J,K)*RR(J)*TRA(J,K)
0840          TAA(J,K)=TAMU*U(J,K)/RR(J)
0841          RTRR(J,K)=RR(J)*TRR(J,K)
0842          20  RTRA(J,K)=RR(J)*TRA(J,K)
0843
0844 *V        CALL DWMPY(RR,1,TRA(1,K),1,TRA(1,K),1,N2)          ! *R
0845 *V        CALL DWMPY(ALP(1,K),1,TRA(1,K),1,TRA(1,K),1,N2)    ! *ALP
0846 *V        CALL DWSMY(MU(K),TRA(1,K),1,TRA(1,K),1,N2)        ! *MU=TRA
0847
0848 *V        CALL DWDIV(U(1,K),1,RR,1,TAA(1,K),1,N2)! U/R
0849 *V        CALL DWMPY(ALP(1,K),1,TAA(1,K),1,TAA(1,K),1,N2)    ! *ALP
0850 *V        CALL DWSMY(2.*MU(K),TAA(1,K),1,TAA(1,K),1,N2)      ! *2*MU=TAA
0851
0852 *V        CALL DWMPY(RR,1,TRR(1,K),1,RTRR(1,K),1,N2)          ! R*TRR
0853 *V        CALL DWMPY(RR,1,TRA(1,K),1,RTRA(1,K),1,N2)          ! R*TRA
0854          50  CONTINUE
0855          RETURN
0856          END
0857
0858 *****
0859 $EMA /DATA/
0860          SUBROUTINE UOFW(W,RR,N2),(<860425.1537>)
0861 C          CONVERTS W TO THE INDEPENDENT VARIABLES(U,V,ALP)
0862
0863          PARAMETER (MM=101)
0864          REAL*8 W(MM,5),RR(MM)
0865          EMA W ,RR.
0866
0867          REAL*8 ALMT, U,V,ALP,P,R
0868          X          ,WJ6
0869          COMMON
0870          X /ALPLMT/ ALMT(2)
0871          3 /DATA/ U(MM,2),V(MM,2),ALP(MM,2),P(MM),R(MM)
0872
0873 *V        CALL DWDIV(W(1,5),1,RR,1,ALP,1,N2)
0874
0875 C        CHECK VOLUME FRACTION & FLOW DIRECTIONS
0876          DO 50 J=1,N2
0877          ALP(J,1)=W(J,5)/RR(J)
0878          IF(ALP(J,1) .LE. ALMT(1) .OR. ALP(J,1) .GE. ALMT(2)) THEN
0879          IF(ALP(J,1) .LE. ALMT(1)) THEN
0880          ALP(J,1)=ALMT(1)
0881          W(J,3)=W(J,4)*ALMT(1)/(1.D0-ALMT(1))
0882          ELSE
0883          ALP(J,1)=ALMT(2)
0884          W(J,4)=W(J,3)*(1.D0-ALMT(2))/ALMT(2)
0885          ENDIF
0886          W(J,1)=0.D0
0887          W(J,2)=0.D0
0888          W(J,5)=ALP(J,1)*RR(J)
0889
0890          ENDIF
0891
0892          ALP(J,2)=1.D0-ALP(J,1)
0893
0894          IF( W(J,2) .LT. 0.) THEN          ! PHASE-2 DOES NOT MOVE IN
0895          W(J,2)=0.
0896          W(J,1)=0.
0897          ENDIF
0898

```

```

0899      U(J,1)=W(J,1)/W(J,5)
0900      V(J,1)=W(J,3)/W(J,5)
0901      WJ6=RR(J)*ALP(J,2)
0902      U(J,2)=W(J,2)/WJ6
0903      V(J,2)=W(J,4)/WJ6
0904      50  CONTINUE
0905
0906      *V  CALL DWDIV(W,1,W(1,5),1,U,1,N2)
0907      *V  CALL DWDIV(W(1,3),1,W(1,5),1,V,1,N2)
0908      *V  CALL DWMPY(ALP(1,2),1,RR,1,W(1,6),1,N2)
0909      *V  CALL DWDIV(W(1,2),1,W(1,6),1,U(1,2),1,N2)
0910      *V  CALL DWDIV(W(1,4),1,W(1,6),1,V(1,2),1,N2)
0911
0912      RETURN
0913      END
0914

```

Exhibit A

A Sample Input

```

:GLVM
  Enter lu for saving data. D.F.= 6
90
Enter FILE NAME for saving data.
TS153::LB
  Enter NOTES(<73 CHAR.) for the job
SAMPLE RUN OF TEST 153
  enter Reynolds no.,RE. D.F.= 1.0000E+05
.
  Enter DENSITY and VISCOSITY ratios.
  D.F.= 1.2930E-03 1.1295E-02
.
  Enter BASE DIAMETERS: DO1,DO2
  D.F.= 1.0000E-02 1.0000E-02
.
  Enter SIZE EXPONENT: GAMMA1,GAMMA2
  D.F. = 2.0000E-01 2.0000E-01
.
  Enter weighting exponent: NA,ND. D.F.= 4.0000E+00 4.0000E+00
.
  Enter GAS VOLUME FRACTION limits:ALMT1,ALMT2.
  D.F.= 1.0000E-04 9.9999E-01
.
  Enter eddy viscosity factor. D.F.= 1.1448E+02 1.0000E+03
1000,1000
  Enter wall condition,1=nonslip,0=slip. D.F. 1
.
  Enter numerical damping factor. D.F.= 0.0000E+00
.
  Enter JET SIZE defined by RJ1,RJ2. D.F.= 8.5000E-01 9.5000E-01
.
  Enter INJECTION SPEED AND TIME RANGE, VJ,T1,T2
  D.F.= 1.0000E+01 0.0000E+00 1.0000E+01
1,0,1
  Enter data FILE NAME for initial cond., if any:
  D.F.= Simple vortex
.
  Enter initial value of alpl. D.F.= 2.5000E-01
.
  Enter type of vortex: 0=At rest,1=pure rotation
  2=H.O.,3=GIT. D.F.= 0
.
  Enter INITIAL and FINAL TIMES. D.F.= 0.0000E+00 5.0000E+00
0.,.01
  Enter TIME STEP for output. D.F.= 2.0000E-01
.01
1 NP= 1 T=0.000E+00 NT= 2 DT=1.000E-06
1 NP= 2 T=1.200E-02 NT= 6 DT=3.000E-03
:

```

Exhibit B

A Sample Output

TS153 T=00004 IS ON CR LB USING 00024 RLKS R=0000

```

0001 1 SAMPLE RUN OF TEST 153
0002 INITIAL CONDITION FILE: .
0003 **DIMENSION UNITS ARE IN MKS**
0004 DENSITY SCALE(kg/m3) 1.0000E+03
0005 LENGTH SCALE=RTANK,(m) 1.0000E+00
0006 VELOCITY SCALE(m/s) 1.5140E-01
0007 TIME SACLE(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 1.0014E-05
0022
0023
0024 OTHER CONSTANS: IW,IVTX,NA,ND,DAMP,VPEAK,RPEAK,OMEGA,D1/D2,MUI/M
0025 1 0 4. 4. 0.00
0026 0.0000E+00 1.0000E-01 0.0000E+00 1.2930E-03 1.1295E-02
0027
0028 101 .10
0029 1 NP= 1 T=0.000E+00 NT= 2 DT=1.000E-06
0030 J ALP1 U1 U2 V1 V2
0031 1 .2500 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0032 2 .2500 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
----- OUTPUT IN THE BETWEEN OMITTED -----
0230 97 .2500 -8.685E-10 2.895E-10 3.513E-04 3.474E-04 3.969E-0
0231 98 .2500 -1.775E-10 5.918E-11 1.268E-04 1.250E-04 3.972E-0
0232 99 .2500 -2.785E-11 9.284E-12 4.319E-05 4.256E-05 3.973E-0
0233 100 .2500 -5.225E-12 1.742E-12 1.553E-05 1.531E-05 3.973E-0
0234 101 .2500 0.000E+00 0.000E+00 0.000E+00 0.000E+00 3.973E-0
----- THE REST OF THE OUTPUT IS OMITTED -----

```

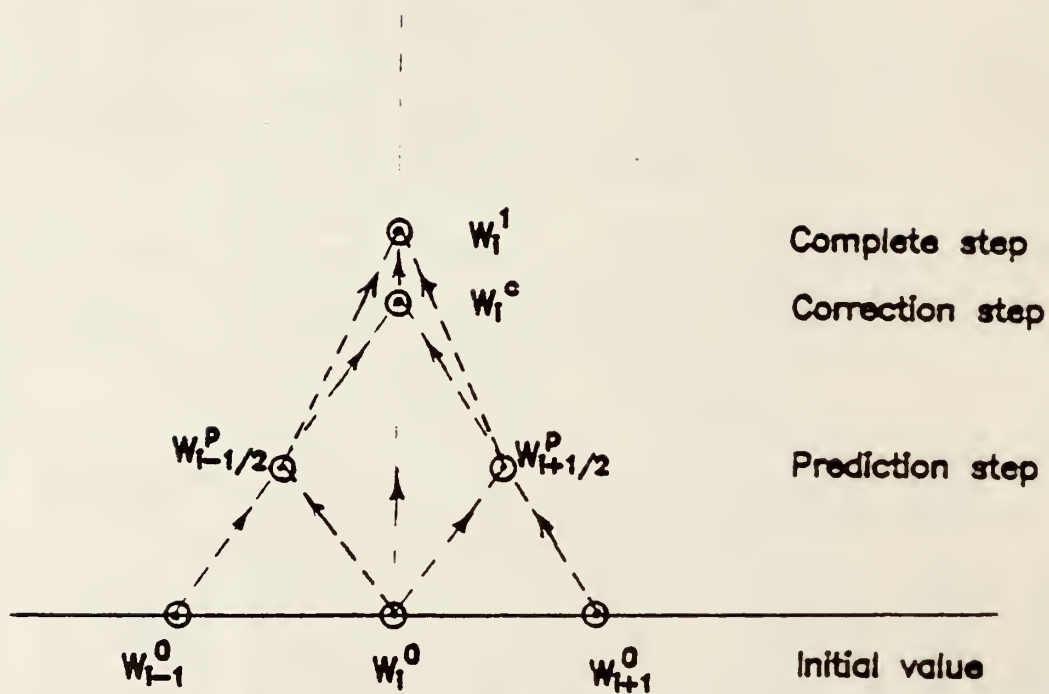
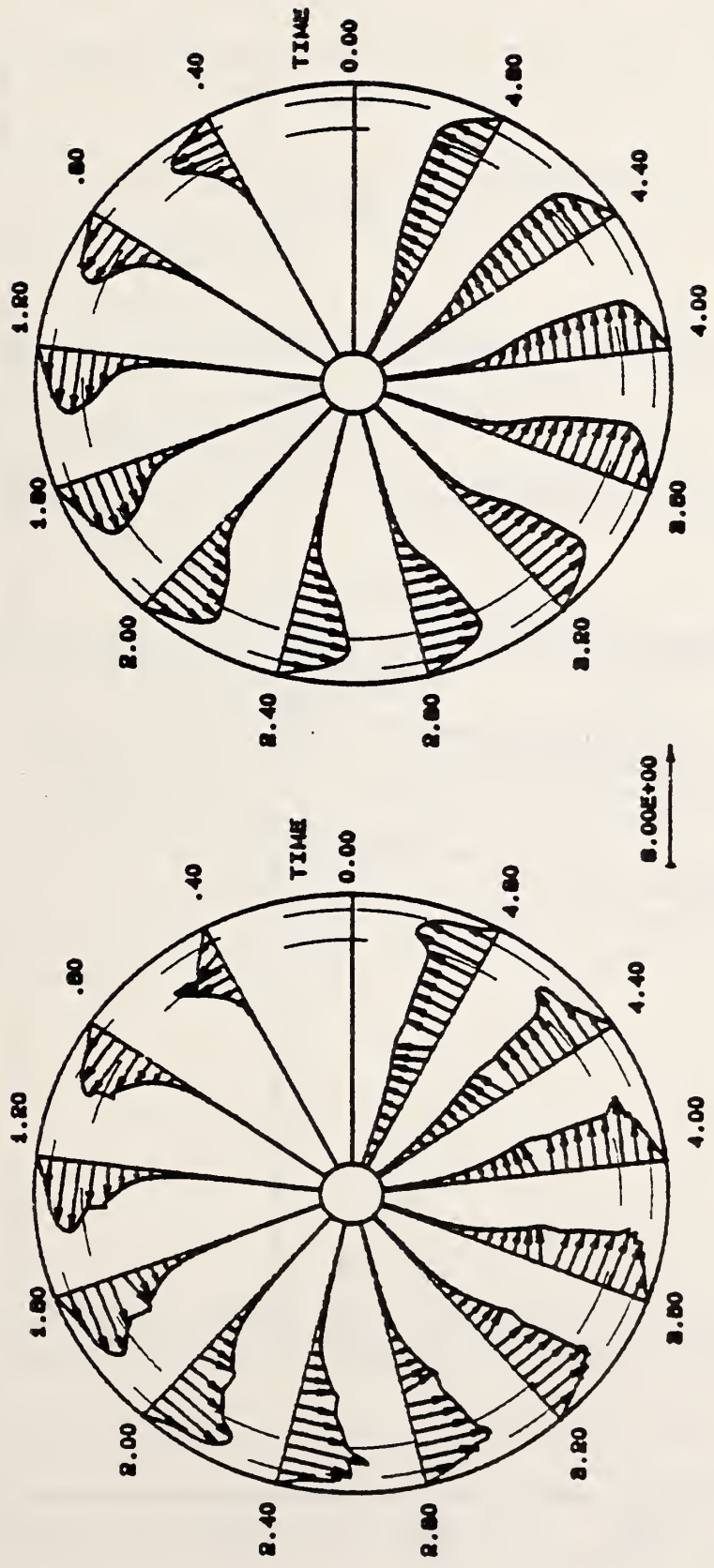


Figure 1. Two Step Difference Scheme (Backward Predictor - Forward Corrector Version)



b. LIQUID PHASE

a. GAS PHASE

Figure 2. Velocity Vector Distributions
 The annular region between two dashed circles is
 the region of injection.

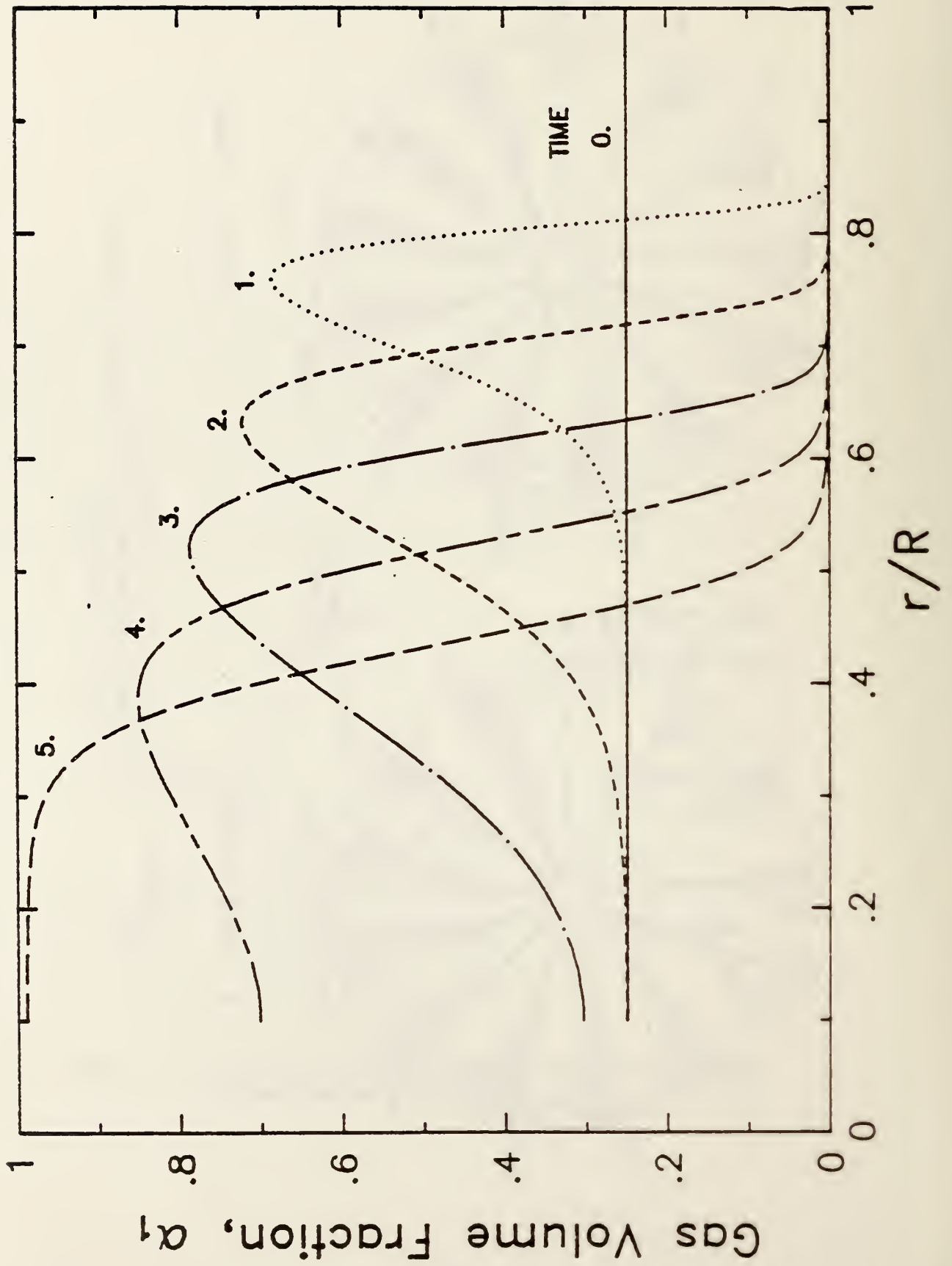


Figure 3. Gas Volume Fraction Distributions

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)		1. PUBLICATION OR REPORT NO. NBSIR 86-3414	2. Performing Organ. Report No.	3. Publication Date JULY 1986
4. TITLE AND SUBTITLE A Computer Code for Gas-Liquid Two-Phase Vortex Motions: GLVM				
5. AUTHOR(S) T. T. Yeh				
6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions) NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			7. Contract/Grant No.	
			8. Type of Report & Period Covered	
9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP) National Aeronautics and Space Administration John F. Kennedy Space Center Kennedy Space Center, FL 32899				
10. SUPPLEMENTARY NOTES <input type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.				
11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here) A computer program aimed at the phase separation between gas and liquid at zero gravity, induced by vortex motion, is developed. It utilizes an explicit solution method for a set of equations describing rotating gas-liquid flows. The vortex motion is established by a tangential fluid injection. A Lax-Wendroff two-step (McCormack's) numerical scheme is used. This program can be used to study the fluid dynamical behavior of the rotational two-phase fluids in a cylindrical tank. It provides a quick/easy sensitivity test on various parameters and thus provides the guidance for the design and use of actual physical systems for handling two-phase fluids.				
12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons) computer code; gas-liquid separation; numerical modeling; two-phase vortex motions				
13. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA. 22161			14. NO. OF PRINTED PAGES 44	
			15. Price \$9.95	

