NASA Contractor Report 3652

Development and Application of the GIM Code for the CYBER 203 Computer

J. F. Stalnaker, M. A. Robinson,
E. G. Rawlinson, P. G. Anderson,
A. W. Mayne, and L. W. Spradley

CONTRACTS NAS1-15783 and NAS1-15795
DECEMBER 1982

NASA

NASA Contractor Report 3652

Development and Application of the GIM Code for the CYBER 203 Computer

J. F. Stalnaker, M. A. Robinson,
E. G. Rawlinson, P. G. Anderson,
A. W. Mayne, and L. W. Spradley

*Lockheed Missiles & Space Company, Inc.
Huntsville, Alabama*

Prepared for
Langley Research Center
under Contracts NAS1-15783 and NAS1-15795

NASA
National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1982

Use of trade names or names of manufacturers in this report does not constitute an official endorsement of such products or manufacturers, either expressed or implied, by the National Aeronautics and Space Administration.

FOREWORD

This report constitutes interim documentation of efforts performed on Contracts NAS1-15783 and NAS1-15795. The purpose of this report is the presentation of the GIM code modifications and the latest flowfield calculations performed with the General Interpolants Method (GIM) computer code for NASA-Langley Research Center. The work is comprised of code description, algorithm development, turbulence modeling, reacting chemistry and interactive inputs in addition to flowfield calculations of various problems of interest. Inquiries concerning this report should be directed to:

Lawrence W. Spradley
Lockheed-Huntsville Research & Engineering Center
4800 Bradford Drive
Huntsville, AL 35807

Telephone (205) 837-1800, ext. 249

ACKNOWLEDGMENT

The author gratefully acknowledges the contribution to and continued support of this work by the NASA-Langley Contract Monitors James L. Hunt and J. Phillip Drummond. Other members of the Lockheed-Huntsville computational fluid dynamics group also provided valuable assistance during the course of this study.

CONTENTS

1. INTRODUCTION AND SUMMARY

The General Interpolants Method (GIM) code was developed to analyze complex flow fields which defy solution by simple methods. The code uses numerical difference techniques to solve the full three-dimensional time-averaged Navier-Stokes equations in arbitrary geometric domains. The numerical analogs of the differential equations are derived by representing each flow variable with general interpolant functions. The point of departure then requires that a weighted integral of interpolants be zero over the flow domain. By choosing the weight functions to be the interpolants themselves, the GIM formulation can produce the classical implicit difference schemes. Choosing the weight functions to be orthogonal to the interpolant functions produces explicit finite difference type discrete analogs. By appropriate choice of constants in the weight functions, the GIM becomes analogous to standard finite difference schemes such as centered, backward, forward, windward and multi-step predictor-corrector schemes. The GIM analogs, however, are automatically produced for arbitrary geometric flow domains and hence is a general point of departure and provides flexibility in the choice of differencing schemes.

The GIM computer code was originally written for the CDC 7600 machine. The first effort that was accomplished under Contract to NASA-Langley was the conversion and reprogramming of the code for the CDC-STAR (now termed CYBER 203) vector processor. The GIM-STAR code was then exercised for three-dimensional exhaust flows for application to Scramjet engine studies. The next sequential study in this computational fluid dynamics effort consisted of the development and application of a parabolized GIM algorithm, computation of the flow, including spillage, in a model aircraft inlet and investigation of linearized block implicit schemes for GIM application. These tasks were accomplished under the two subject contracts through a cooperative effort of the Hypersonic Aerodynamics and Hypersonic Propulsion Branches.

The most current effort, which is the subject of this report, is a continuation of the GIM code development and application on the CYBER 203 machine. Objectives of this effort include the following.

- Complete the development of the quasi-parabolic scheme for solving the parabolized Navier-Stokes equations using spatial marching/relaxation.
- Adapt and verify the quasi-parabolic algorithm for computing subsonic forebody flow fields.
- Complete the adaptation of the quasi-parabolic code to hyperbolic flow field computation.
- Continue the investigation and implementation of linearized block implicit schemes for the GIM elliptic and parabolic codes.
- Complete incorporation of a two-equation turbulent kinetic energy transport model and a multiple length scale model.
- Incorporate interactive computer technology for user-oriented input of the geometry description for the GIM code.
- Complete incorporation of a hydrogen-air equilibrium chemistry model.
- Incorporate a hydrogen-air and a hydrocarbon-air finite rate chemistry model and synthesize the model into a global reaction model with one or more equations.
- Compute the flow field for several configurations including a missile fuselage, a wing/body case, a turbulent boundary layer in a model inlet and mixing/reaction of hydrogen and air in a duct.

Certain of these stated objectives were accomplished in full and others partially. The plan of attack was to organize a set of overall tasks, some of which are interrelated, aimed at meeting the major objectives. This report is organized into sections with each section independently presenting details of the major tasks. The following is a summary of these sections:

SECTION 2: THE GIM CODE – VERSION D

Formulation, coding and check out of GIM parabolized Navier-Stokes code was completed. The major effort on this task was expended on obtaining reliable coding for the marching algorithms in vector FORTRAN language. A relatively large amount of logic was necessary to implement the elliptic, hyperbolic and parabolic code in one package. The GEOM module was extensively modified to compute the geometry data one-plane-at-a-time for use in the spatial marchers. The INTEG integration module can now operate with a full elliptic flow field or in a spatial marching mode, either parabolic or hyperbolic. The GIMPLT plotting module was also upgraded to handle the output of the new code. The version which is now online at the Langley Center is termed Version D. The input data for each module of this version can be generated interactively on a remote terminal. Version D contains many features not reported in the previous user's guide. The following items are especially noteworthy.

- Elliptic, parabolic, hyperbolic solvers operational
- Interactive input and runstream generation available
- Explicit finite difference algorithms available in all three solvers; MacCormack implicit (MI) scheme operational in elliptic solver
- Algebraic eddy viscosity and two-equation TKE turbulence models coded in the elliptic and parabolic solvers. Algebraic (Baldwin-Lomax) model only is currently checked out.
- Ideal gas option operational for one or two components. Equilibrium, complete reaction, model in code for hydrogen-air mixture.
- Geometry module can generate grids for all three solvers.

The GIM/Version D code is available on permanent files at NASA-Langley.

SECTION 3: IMPLICIT ALGORITHM DEVELOPMENT

This task was concentrated on developing and implementing a linearized block implicit algorithm for the GIM code. The specific algorithm chosen is termed the MacCormack implicit (Ref. 1.1). The two-dimensional version of the scheme presented in the MacCormack AIAA paper was coded using the GIM methodology. Boundary conditions were not fully presented in the paper, thus development of general boundaries was required. Formulation and coding of a three-dimensional version of the MI scheme were also accomplished. Check out and investigation of the new algorithm were done on a simple Couette flow problem. The following are among the major findings of this task.

- The MI scheme is not readily vectorizable.
- The overall numerical stability is governed by the streamwise CFL number = 1.0. This can still be many times the global CFL based on a tight normal-direction mesh.
- The method appears to have good potential for computing high Reynolds number boundary layer problems.
- With both the explicit and implicit algorithms available in GIM, a large variety of problems can be attacked.

SECTION 4: TURBULENCE MODEL IMPLEMENTATION

This task consists of implementing a number of turbulence models in the GIM code and applying these models to a two-dimensional simulation of an aircraft inlet. The first type of model is termed "algebraic" and consists of the Baldwin-Lomax (B-L) eddy viscosity model. This scheme has been implemented, in general, in the GIM code elliptic and parabolic solvers. The user has at his option, no viscosity, artificial viscosity only, laminar constant or Sutherland's law values or turbulent (B-L) coefficients. Also coded in general format in GIM is a two-equation turbulent kinetic energy (TKE) model. Conservation law form differential equations are written for the turbulent kinetic energy and for the dissipation of TKE. The B-L algebraic model has been checked out for an incompressible flow over a flat plate at several Reynolds numbers. The TKE models have not been fully verified at the time of this writing. The same case solved with laminar and the B-L model is being repeated with the two equation TKE scheme.

SECTION 5: CHEMISTRY MODEL DEVELOPMENT

To enable the computation of chemically reacting flows, the code was modified under this task to include the option of ideal gas or a chemical equilibrium reaction scheme. The basic premise of chemical equilibrium treatment in flow fields is that the local reaction rates are much faster than the flow residence time. The consequence is that the reactions proceed to completion at each point within the flow field. It has been shown that for chemical equilibrium the thermochemistry computations can be uncoupled from the flowfield solution. The thermochemistry data can then be communicated to the flowfield code via one of two methods. First, the flow state variables (P, ρ, T), thermodynamics (γ, M_W), and species distribution can be generated a priori for an isentropic expansion process from a given stagnation condition and the results stored in tabular form for use via a table look-up procedure. Second, the chemical equilibrium properties can be computed as an integral part of the flowfield solution by using the equilibrium calculation (Ref. 1.2) as a subprogram. In either case the results are the same. A third choice is to specify a specific gas system, simplify the reaction model and code the equations. Rather than solving a full set of species equations, a global specie conservation law is solved. In this task, we have accomplished the following in terms of chemistry models:

- The simple "complete reaction" model (type 3 above) has been coded for the hydrogen-air system. The coding has been checked out on a simple case and found to be correct. This simple model is now being used to compute the mixing and reacting flow of air over a plate with hydrogen injection.
- The full equilibrium model for any gas system has been previously developed for other codes. These subprograms are currently being added to GIM to allow equilibrium computations for any gas system. No results have been obtained at this writing.
- The finite rate reaction model has been examined for inclusion in GIM. The equations and solution scheme can be coded in general, with the specific reactions to be allowed and the rate data being an input to the code. This development has been previously completed and is shown in Section 5. No GIM/finite rate coding has yet been done. This will be the major emphasis during the coming year's program.

SECTION 6: INTERACTIVE INPUT MODULE

A new module has been added to the GIM code. The purpose of the module is to render the input and operation of GIM much easier on the CYBER 203. This interactive program, entitled "RUNGIM," uses the CDC FORTRAN Extended (FTN) Version 4.6 language and CDC NOS.1.3 CYBER control language (CCL). RUNGIM essentially replaces the report-format input guides in Ref. 1.3. Upon entering RUNGIM, via an interactive terminal, for example, questions are asked and the user must respond with answers and data. The interactive module performs the following jobs:

- Supplies actual input data for the geometry and integration modules
- Provides program updates to use as input for the CDC UPDATE processor
- Supplies dynamic dimension data to set the size of GIM for a specific problem
- Sets up runstreams, including control cards, file sizes, etc., for executing a GIM run.

The interactive module is currently operational at Langley for the GIM GEOM and INTEG modules Version D. Section 6 of this report gives details on executing RUNGIM and shows examples of its use.

SECTION 7: FLOW OVER A WING-BODY CONFIGURATION

In addition to the development aspects of this work, some effort was expended on applying the code to problems of interest at Langley. A number of cases were run in conjuction with the development work and under a separate "applications" contract with NASA-Langley. One of the main pure calculations associated with this cooperative effort is a "wing-body" problem. The inviscid, supersonic, three-dimensional flow was computed with the GIM hyperbolic marching solver. The configuration consists of an ogive-cylinder fuselage with a wedge-shaped wing attached. The flow was assumed to be inviscid for this initial solution. The problem is currently being repeated with viscous effects. The wing-body interference flow field was computed with the GIM/QH code using approximately 20,000 grid points and required 210 seconds to solve the full three-dimensional field on the CYBER 203. The solution of the full flow field is shown in Section 7 as computer generated contour maps. The regions of wing-body interference are clearly seen. The solution for surface pressures are compared to measured data for this configuration. The agreement is reasonable in some regions and shows some deviations in others. The inviscid treatment of leading edges and corners contributed to this disagreement.

An important part of a research effort is to publish the findings and results of the studies. In addition to the NASA Contractor Reports, the results of the GIM code development and applications have been published at professional society meetings and in the open literature. During the past year, two papers were presented at AIAA meetings, both of which were published in the AIAA Journal, one paper in the NASA-Langley Grid Generation Conference and a presentation at the American Physical Society-Division of Fluid Dynamics Meeting.

To further enhance the GIM code's capability and to utilize its current potential, the following items are recommended.

Development

- Formulate and code the MacCormack implicit method for the GIM forward marching solvers
- Complete the full equilibrium chemistry model coding and checkout
- Write a full finite rate model subprogram with specific reaction data as input
- Synthesize a global hydrogen-air finite rate model
- Incorporate triangular elements into the geometry package to facilitate easier modeling and to allow better grid control

Applications

- Compute parallel mixing of hydrogen and air in a duct, for equilibrium and finite rate
- Determine the flow field, with spillage, in a three-dimensional inlet
- Perform a viscous calculation with the GIM/QP solver over the wing/body configuration.

An important part of the next year's effort should be directed toward assisting Langley personnel in running the GIM code. A course is planned to acquaint users at Langley with aspects of the code that have changed or are new since the previous user-orientation given last year.

SECTION 1 REFERENCES

1-1. MacCormack, R. W., "A Numerical Method for Solving the Equations of Compressible Viscous Flow," AIAA Paper 81-0110, January 1981.

1-2. Svehla, R. A., and B. J. McBride, "FORTRAN IV Computer Program for Calculation of Thermodynamic and Transport Properties of Complex Chemical Systems," NASA TN D-7056, January 1973.

1-3. Spradley, L. W., J. F. Stalnaker and A. W. Ratliff, "Hyperbolic/Parabolic Development for the GIM/STAR Code," NASA CR-3369, Langley Research Center, Hampton, Va., December 1980.

2. THE GIM CODE – VERSION D

The completion of the GIM code parabolized algorithms was a major part of this year's effort. Inclusion of elliptic, hyperbolic and parabolic solvers was accomplished. As a result of these updates to the code, a number of modifications were made to each module:

- GEOM - Generates geometric boundaries of the problem and the finite difference grid
- MATRIX - Performs matrix operations to obtain finite difference operators for the GEOM grid
- INTEG - Integrates the flow field with user specified initial and boundary conditions
- GIMPLT - Displays grid and/or flow field in contour maps
- RUNGIM - Interactively produces a runstream and input data for the GIM code

With the inclusion of the updated capability, the lastest version (as of this writing) of the code is termed GIM/Version D. The following paragraphs summarize the coding changes made to achieve Version D. Charts showing the revised input formats are also given. Sections 3 through 7 of this report presents the technical aspects of each task in detail.

2.1 PROGRAM MODIFICATIONS

Several modifications were made to the CYBER 203 version of the GIM code geometry module (GEOM). Some of the modifications were necessary to remain compatible with Version D of the integration module, and some were made to increase user facility and reliability.

The most significant modification to the geometry module involved changing the output file logic so that data could be output a line (2-D) or plane (3-D) at a time as required by the QP version of the integrator module. This modification greatly reduced the storage requirements of the geometry module for QP-type analyses. In order to be able to compute in either the elliptic or QP modes the output logic for both the geometry data and the analog data had to be modified extensively.

Another modification was to incorporate logic to count the number of boundary nodes in a given geometry. The number is used in dimensioning the integer module but is difficult to determine manually for complex geometries.

The final set of modifications involved rework on some of the logic in the module that was too difficult for users to understand. Changes were made to the circular arc, conical arc and edge of revolution algorithms to increase their utility and reliability.

The MATRIX module for the elliptic solver remains essentially intact from the previous version. The reading/writing of files has been changed for compatibility with GEOM D. At this writing, there are two matrix modules on file at the computer center.

- MATRIX D - For use with elliptic solver
- MATQP D - For use with spatial marching solvers

The coding of the matrix operations for the two different solvers necessitated construction of separate modules.

The major portion of the coding changes for GIM/D was made to the integration module, INTEG. Considerable logic is necessary to allow integration in arbitrary three-dimensional regions. The INTEG D module contains:

- Elliptic, hyperbolic and parabolic solves
- Two and three dimensions

- Inviscid, laminar or turbulent flow
- One or two-gas (ideal) or simplified equilibrium chemistry
- Explicit finite difference or the new "MacCormack" implicit algorithm
- Interactive input of data.

The storage and dimensioning of INTEG was also reworked to make optimum use of the high speed memory of the CYBER 203.

The GIMPLT graphics module required some extensive reworking to make it compatible with the spatial marching routines. First, the program was simplified by eliminating all unnecessary coding. The original plotting routine had several options which are currently unused and thus these were eliminated. Next, internal variable names corresponded to symbols which were also confusing. These variable names have been changed and are now representative of actual usage as it applies to the GIM code. Some external variable names used as input were changed for consistency and readability. However, the order of input and location of variables has remained the same as the STAR version of the plotter. This is consistent with published documentation. Also, the subroutines have been resequenced to aid tracability of "GO TO" and "DO" statements. Then, comments were placed throughout the subroutines to aid users in locating regions of interest for debugging or modifying purposes. The program has been simplified, rewritten, and internally documented for ease of usage.

Plot documentation has been improved by the addition of subtitles for contour and velocity vector plots. The subtitles are located at the bottom of each plot frame to the right of the main title.

The size of the model that GIMPLT can handle has increased dramatically from about 5,000 nodes last year to over 28,000 nodes currently. The primary limitations are the core size of CDC 7600 and the unavailability of the CY203 for plot runs. The alternative has been to rewrite the plot program or to do cumbersome external data file manipulation to reduce the number of nodes which can be plotted for a given run. Fortunately, it has been possible to develop a way for the plotter to handle models of up to 28,000 nodes through variable manipulation within the plotter. This approach maximizes the usage of available core. To further increase the plotter capacity would require having more core available since the plotter must have at least enough room to store the X, Y and Z positions for each node. This modification allows one version of the plotter to be used for both small and large models without external data file manipulation for models up to 28,000 nodes.

2.2 STATE OF THE CODE

The GIM/D code is cataloged on permanent files at NASA-Langley. It is recommended at the current time that the interactive input module described in Section 6 be used for inputting the GIM code. For completeness and for information on the required program inputs, the following charts are included here. These summary input guides can be used in conjunction with the previous GIM documentation to fully describe the codes operation:

GIM Manual – NASA CR 3157; 1979
Hyperbolic/Parabolic – NASA CR 3369; 1980

CHARTS: Input Guides

2-1 DYNMAT D
2-2 DYNDIM D
2-3 GEOM D
2-4 MATRIX D
2-5 MATQP D
2-6 INTEG D
2-7 Notes on INTEG D
2-8 GIMPLT D

Cards/Format

NX IDIM NSPEC NXMAX NMATE NSAV METHOD

(7I5)

NX = total number of nodes (elliptic)
= number of nodes in one cross-plane (QP)

IDIM = dimensionality (= 2 or 3)

NSPEC = number of special nodes
= 0 for GEOMD runs

NXMAX = maximum number of nodes in a cross-plane

NMATE = number of nodes to be checked for mating
e.g., for a surface of 20 nodes mated to a surface of 50 nodes NMATE $\geq$ 50. When in doubt set NMATE to the number of nodes in the largest zone.

NSAV = number of planes processed per record on the geometry files. The actual number of nodes processed per record is

(NSAV + 1) *NXMAX - elliptic
2 *NXMAX - QP

METHOD = 0 for elliptic runs
> 0 for QP runs

Chart 2-1 - DYNMAT D Input Data

Cards/Format

MN IDIM ISPEC NSP MNB METHOD IREC
(7I5)

MN = total number of nodes (elliptic)
= number of nodes in one cross-plane (QP)

IDIM = dimensionality

ISPEC = two-gas flag = 0 1 ideal gas
= 1 2 ideal gas

NSP = number of special nodes
(equal to NSPEC input to matrix module for QP)

MNB = number of boundary terms (IB $\neq$ 9)
= 0 program sets MNB = MN

METHOD = $\leq$ 2 elliptic
$>$ 2 parabolic

IREC = number of records on the GEOM file
= 1 for QP

Chart 2-2 - DYNDIM D Input Data

Card Type	Parameter List/Format
1	HEADER(I),[*] I = 1,72 (12A6)
2	NZONES, IDIM, ISTEP, IMATRX, IMATE (5I5)
3	IWRITE, LWRITE, NWRITE (3I5)
4	KC(I), I=1,6 (6A5)
5	NSECTS (I5)
6	MAPE(I), I = 1,12 (12I5)
7	MAPS(I), I = 1,6 (6I5)
8	(IBWL(I), I = 1,6), ITRAIN (7I5)
9	(NNOD(I), I = 1,3), (ISTRCH(I), I = 1,3) (6I5)
10	DIVPI(I), I = 1,3 (3E10.4)
11	[AETA(J,I), I = 1, NNOD(J)], J = 1, IDIM (8E10.4)
12	[(AC(I,K,J), I = 1,8), K = 1,5], J = 1,4 or 12 (8E10.4)
13	[AS(I,J), I = 1,8], J = 1,6 (8E10.4)
14	(PT(I,J), I = 1,5), J = 1,4 or 12 (8E10.4)
15	[(PMAX(I,K,J), I = 1,5), ETAMAX(K,J), K = 1,4], J = 1,4 or 12 (6E10.4)

Chart 2-3 - GEOMD Input Guide

[*] See GIM documentation for explanation of FORTRAN symbols, NASA CR 3157 and CR 3369.

Card	Parameters	Format
1	NDX, NDY, NDZ, ISNOPT[*]	(4I5)
2	KC(I), I = 1,6	(6A5)
3	N1,IC,NT	(3I5)

Chart 2-4 - MATRIX D Input Guide

[*] See GIM documentation for explanation of FORTRAN symbols, NASA CR 3157 and CR 3369.

Card Type	Parameter List/Format
1	ITITLE(I)[*], I = 1, 78 (13A6)
2	KC1(I), I = 1, 3, KC2(I) (I = 1, 3) (6A5)
3	NPLT(I), MI2(I), MI3(I), NATT(I), L2(I), L3(I) (I = 1, NZONES + 1) (6I5)
4	IN1, IIC, INPP, IPRMAX (4I5)

Cards 1 and 2 are as previously defined.

Card 3: ZONE/PLANE Data

NPLT(I) - Number of cross planes in Zone I

MIZ(I) - Number of nodes in η_2 in Zone I

MI3(I) - Number of nodes in η_3 in Zone I

NATT(I) - Plane of attachment of Zone I = 0 (or blank) for I = 1

L2(I), L3(I) - Grid coordinates for first mated node in Zone I=0 (or blank) for I=1

EXAMPLE: 5 x 5 node cross plane mated to 7 x 7 node cross plane (x's are mated nodes)

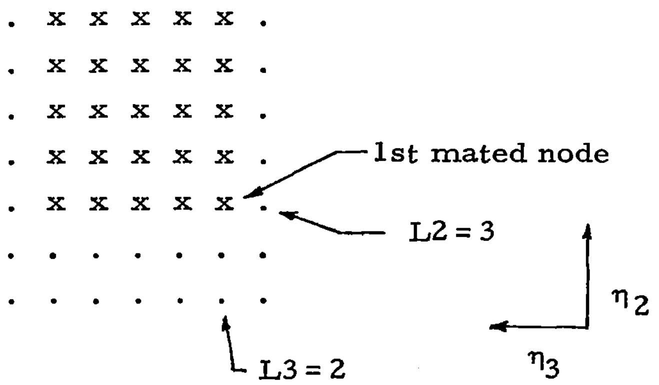


Chart 2-5 - MATQP D Input Guide

[*] See GIM documentation for explanation of FORTRAN symbols, NASA CR 3157 and CR 3369.

This input is terminated with a -1 in Columns 4 and 5.

Card 4: Analog Print Specifications

IN1 - First node to be printed
= 0 no nodes printed
= -1 terminates input

IIC - Nodal increment
= 0 only IN1st node printed

INPP - Number of planes to which this specification applies

IPRMAX - Maximum number of print nodes
= 0 all nodes possible under this specification will be printed

Notes: 1. All the INPP must sum up to the total number of planes in the problem
2. If only a -1 appears in Columns 4 and 5 no analogs will be printed.

EXAMPLE

```
 0  0  21   1
 2  2   5  10  10
 1  2   6   6
-1
```

No analogs are printed for the first 21 planes; analogs for the 2nd, 4th, 6th,... 20th nodes are printed for the next five planes; analogs for the 1st, 3rd, 5th,... node are printed for the last six planes.

Notes on MATQPD Output

1. No output appears for plane 1 (the initial input plane)
2. For plane 2 or plane NATT(I) +1 for which the number of nodes in a cross-plane in zone I is greater than that in Zone I-1 (i.e., planes which require flowfield input), analogs are printed for nodes IN1 through 2*NX (where NX is the number of nodes in a cross-plane) incremented as specified.
3. For all other planes the node number printed on the output is increased by NX since nodes 1 to NX are in the preceding plane.

Chart 2-5 - (Concluded)

Card Type	Parameter List/Format
1	ITITLE(I)[*], I = 1,80 (80 A1)
2	IDIM, METHOD, ITMAX, IPRNT, ITSAVE, ISTART, IOTYPE, IUNITS, ITSTRT, IVISC, IDIST, ISPEC, IDS, IBOUND, ITHERM (15I5)
2a	IPSQ, INORM, ISBSM (3I5)
3	INFOUT, IJUMPO, JJUMPO, NIOUT, NJOUT, ICALC, AMFLW (6I5, E10.0)
3a	INFINL, IJUMPI, JJUMPI, NIIN, NJIN, ICALC, OUTMFL (6I5, E10.0)
3b	INFINL, IJUMPI, JJUMPI, NIIN, NJIN, INFOUT, IJUMPO, JJUMPI, NIOUT, NJOUT, ICALC (11I5)
4[*]	IORD, IPCVG, SUMCHK, SLPCHK (2I5, 2E10.0)

4a[*]	NPLT(I), MI2(I), MI3(I), NATT(I), L2(I), L3(I)
	I = 1,NZONES + 1
	(6I5)

5[+]	NN, NNX, NDX, NNY, NDY, NNZ, NDZ, NPM, KZONES (9I5)
5a[+]	KST, KNX, KDX, KNY, KDY, KNZ, KDZ (7I5)
6	DTIME, DTFAC, INCDT (2E10.0, I5)
7	REALMU, REALK, GAMS1, GAMS2, WM1, WM2, DK,RK (8E10.0)
7a	VF(I), I = 1, 8 (8E10.0)
8	EMU, ELAM, ERHO, ESPEC (4E10.0)
9	KC(I), I = 1, 6 (6A5)

10[+]	NNPM(I), NCPM(I), (NNCPM(I, J), J = 1, 5), ANGPM(I);
	I = 1,NPM
	(7I5, E10.0)

Chart 2-6 - INTEG D Input Guide

[*] See GIM documentation for explanation of FORTRAN symbols, NASA CR 3157 and CR 3369.

Card Type	
11^{+}	NCT(I, J, K), PXPM(I, J, K), PYPM(I, J, K),* K = 1,4; J = 1, NCPM(I); I = 1, NPM (I5, 2E10.0)
12	RHOZ, PZ, ASTAR, NINC, A, B (3E10.0, I5, 2E10.0)
13	NJ, INC, NTOT, ITAN, ITYPE (5I5)
14	RI, UI, VI, WI, PI, CSI (6E10.0)
15^{+}	N1, IC, NT (3I5)
15a*	IN1, IIC, INPP, IPRMAX (4I5)
16*	NJ, INC, NTOT, ITAN, ITYPE (5I5)
16a*	RI, UI, VI, WI, PI, CSI (6E10.0)

*QP Only

$^{+}$Elliptic Only

Chart 2-6 - INTEG D Input Guide (Concluded)

*See GIM documentation for explanation of FORTRAN symbols, NASA CR 3157 and CR 3369.

Card 2:

IVISC = 0 - Inviscid Euler Equations
= 1 - Numerical damping only
= 2 - Laminar viscosity and damping
= 3 - Baldwin/Lomax turbulence model

ITHERM = Thermal boundary conditions for no-slip walls
= 0 - Free thermal boundaries
= 1 - Adiabatic walls
= 2 - Constant temperature walls

Card 2a:

IPSQ = SUMSQ print increment
= N, SUMSQ's printed every Nth iteration

INORM = 0 SUMSQ's normalized by 1st iteration values
> 0 SUMSQ's not normalized

ISBSM = 0 SUMSQ's of conserved variables calculated
> 0 SUMSQ's of primitive variables calculated

Card 4:

IORD = 1 1st order backward
2 2nd order backward

IPCVG = Convergence check increment
= N, convergence checked at every Nth iteration

SUMCHK = SUMSQ tolerance
Convergence assumed if all SUMSQ < tolerance

SLPCHK = SUMSQ slope tolerance
Convergence assumed if all slopes of the SUMSQ curves are < SLPCHK

Chart 2-7 - Notes on the New Input for INTEGD

Cards 4a:

These are the same as for MATQPD input. This sequence must be terminated with a -1 in Cards 4 and 5.

Card 6:

INCDT < 0 - Implicit technique

Card 7a: (If IVISC $\leq$ 1 omit Card 7a)

Sutherland's law:

VF(1) = Sutherland's Constant
= 1.45804E-5 g/cm-sec ($K^{1/2}$)
= 7.30350E-7 lb_m/ft-sec ($R^{1/2}$)

VF(2) = T_*
= 110.33 K
= 198.6 R

Baldwin/Lomax Turbulence Model

VF(3) = CCP = 1.6
VF(4) = CKLEB = 0.3
VF(5) = $(CWK)^{-1/2}$ = 2.0
VF(6) = Pr_T = 0.9
VF(7) = CMUTM = 14.0

Note on variable viscosity:

The viscosity at any node is computed as

$$\mu = \mu_d + \mu_L + \mu_T$$

where

μ_d = numerical damping coefficient

μ_L = REALMU + VF(1) * $T^{3/2}/(T + T^*)$

μ_T = Turbulent viscosity.

The laminar thermal conductivity is calculated through the laminar Prandtl number $\frac{\mu C_p}{k}$ which is now entered as REALK.

Card 15a: QP print control

$$\left.\begin{array}{l}\text{IN1}\\ \text{IIC}\\ \text{INPP}\\ \text{IPRMAX}\end{array}\right\} \quad \text{as described for MATQPD}$$

Note on plane numbering:

The plane number which appears in the QP integration is the number of integration planes. That is, one less than the actual geometric planes since the first plane is input. The total number of integrated planes, MAXPLN, is thus one less than the total number of planes. All of the INPP in the print specifications above must sum up to MAXPLN.

Cards 16 & 16a:

Input initial conditions for added zone input surfaces. These assume the same form as cards 13 and 14. Only card input and USER1P options are available for these. This input is read at the time that the plane is integrated. The information is stored at the end of the primitive variable vectors. As a result the first node to be initialized, NJ, is

$$NJ = 2 * NN + 1$$

where NN is the number of non-input nodes in the cross-plane.

Chart 2-7 - (Concluded)

Card Type	Parameter List/Format
1	ITITLE (A40)
2	NN, ITRSTR, ITRBLK, KDIM, ISPC (5I5)
3	GAMMA, FACTOR, RGAS, PO, TO, RHOO (6E10.0)
Specs.	
S-1	NPLT, STITLE, IVIEW, ISYM, ITHET1, IAXIS1, ITHET2, IAXIS2, IXTABL, IYTABL, VFAC (I5, 5X, A20, 8I5, E10.0)
S-2	NTYPE, JO, IJUMP, JJUMP, NI, NJ, IPRNT, JOO (8I5)
Grid	
G-1	'GRID,' IOPT, ICSCLE, NSPECS, (ISPEC(I), I=1, NSPECS) (A4, 1X, I5, 25X, 2I5, 7I5)
VVEC	
V-1	'VVEC,' IOPT, NITER, ICSCLE, NSPECS, (ISPEC(I), I=1, NSPECS) (A4, IX, 2I5, 20X, 2I5, 7I5) (NITER = ƀ for QP)

Chart 2-8 - GIMPLTD Input Guide

VVEC

V-2	(ISPEC(I), I=8, NSPECS) (if NSPECS>7) (45X, 7I5)
I-1	(ITER(I), I=1, NITER) (16I5)

Contours

C-1	ITYPE, IPOT, NITER, NC, ITABLE, INCR, ICSCLE, NSPECS, ISPEC(I), I=1, NSPECS) (A4, 1X, 5I5, 5X, 2I5, 7I5)
C-2	(ISPEC(I), I=1, NSPECS) (if NSPECS>7) (45X, 7I5)
I-1	(ITER(I), I=1, NITER) (16I5)
L-1	(CVAL(I), I=1, NC) (8E10.0) (No I-1 cards for QP)

Description of Input Data

Card 1

ITITLE - This problem identification title appears on the bottom left of the plot frame for Grid, Contour, and Velocity Vector plots as one line of forty characters.

Card 2

NN - The total number of nodes in the model which is being plotted. NN must be consistent with the GEOMD and INTEGD runs which generated the grid and flow field.

Chart 2-8 - (Continued)

ITRSTR - The iteration number of the first set of flow field information on the INTEGD File 22 which is being used as input to the GIMPLTD module. (= -1 for QP).

ISPC - The two gas flag used in INTEGD

ISPC = 0 for a single gas

= 1 for two gases

Card 3

RGAS - The gas constant as input in INTEGD

Card S-1

STITLE - The plot specification title. This subtitle will appear on the bottom right of grid, contour and velocity vector plots as one line of twenty characters.

Card S-2 - Element generation control parameter

NTYPE = 0 a single element is input

= 2 a network of two-node line connectors is generated

= 3 a network of 4-node elements is generated conconnecting two zones with different numbering schemes

= 4 a network of 4-node elements is generated

Default NTYPE = 0

JO - For NTYPE = 3, JO is the first node in the first string of nodes.

IJUMP - For NTYPE = 3, IJUMP is the nodal increment for the first string of nodes.

JJUMP - For NTYPE = 3, JJUMP is the nodal increment for the second string of nodes.

NI - For NTYPE = 3, NI is the number of elements generated.

NJ - For NTYPE = 3, NJ = 1

JOO - For NTYPE = 3 only, JOO is the first node in the second string of nodes.

Example

For NTYPE = 3, a network of 4-node elements, are generated connecting two zones with different numbering schemes as illustrated on the following page.

Where J1, J2, J3 and J4 are the nodes connected by the element that the algorithm generates for I = 1 through NI.

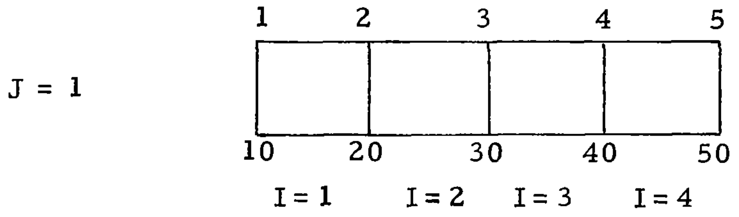
J1 = J0 + (I-1) * IJUMP

J2 = J00 + (I-1) * JJUMP

J3 = J2 + JJUMP

J4 = J1 + IJUMP

NTYPE = 3, J0 = 1, IJUMP = 1, JJUMP = 10, NI = 4, NJ = 1, J00 = 10



Note on calculation of core sizes

The total core requirements for plotting including the program storage, system library, and all data arrays is given by the following formula.

$$KFL_{10} = KMAX_{10} + 24000_{10}$$

where,

$$\begin{aligned} KMAX_{10} &= NWRDS + 3 * NN + 1 \\ &= 1000 + 3 * NN + 1 \text{ (NN = number of nodes)} \end{aligned}$$

KMAX is the dimension of the array "A" in the GIMPLTD MAIN program which is normally set to 46001 and must be changed to plot a larger model. KMAX is also a program variable which must be reset in the MAIN program for a larger model.

For KMAX = 46001, the maximum number of nodes in a model to be plotted is

$$NN = \frac{KMAX - NWRDS - 1}{3}$$

$$NN = \frac{46{,}001 - 1000 - 1}{3} = 15000 \text{ nodes}$$

The field length requirement for KMAX = 46,001 is

$$\begin{aligned} KFL_{10} &= 46{,}001_{10} + 24{,}000_{10} \\ &= 20{,}001_{10} \\ &= 210{,}561_{8} \end{aligned}$$

A field length of $330{,}000_8$ ($110{,}592_{10}$) permits the following

$$KMAX_{10} = 110{,}592_{10} - 24{,}000_{10} = 86{,}592_{10}$$

$$NN = \frac{86{,}592 - 1000 - 1}{3} = 28{,}530 \text{ nodes.}$$

Chart 2-8 - (Concluded)

3. IMPLICIT ALGORITHM DEVELOPMENT

3.1 INTRODUCTION

Numerical solution of the unsteady Navier-Stokes equations by explicit finite difference techniques has a number of disadvantages. The most serious one, from a practical engineering viewpoint, is the small time steps which are usually required to maintain stability. Computation of boundary layer flows at high Reynolds number, for example, requires a fine grid near solid boundaries, hence very small time steps and long computer run times.

Numerical treatment of the steady state parabolic form of the Navier-Stokes equations face many of the same difficulties as the elliptic form. The spatial marching step size is constrained by the small grid spacing required to resolve boundary layers normal to a solid wall. Thus, marching downstream great distances can result in impractically long run times.

One apparent solution for these difficulties is the use of implicit methods, some of which are unconditionally stable for any size time step/spatial marching step. These schemes are not without problems of their own in terms of their practical use. Among the major difficulties are the following:

1. Implicit finite differences, in general, lead to systems of nonlinear algebraic equations when applied to the Navier-Stokes equations. These must either be solved directly or linearized in some manner.

2. Multi-dimensional implicit methods lead to very large systems of simultaneous algebraic equations. Even for linear systems, the efficient solution is not practical due to large size of the coefficient matrix.

3. Fully implicit methods cannot be programmed as efficiently as explicit methods on advanced vectorized machines such as the CYBER 203.

In a recent paper, MacCormack (Ref. 3-1) presented a new implicit method which circumvents the first three of the above mentioned difficulties. The new method contains two stages. The first stage utilizes the explicit, second order, predictor-corrector finite difference method developed by MacCormack in 1969 (Ref. 3-2) which is widely used in many codes, including the GIM code. The second stage removes the explicit stability requirements by numerically transforming the finite difference equations into an implicit form. The resulting matrix equations are either upper or lower block bi-diagonal equations and are easily solved. Morever, the method preserves the conservation form of the Navier-Stokes equations so that shock capturing techniques can be employed. Because of these advantages along with its straightforward extension to three dimensions, the MacCormack implicit method was selected for implementation in the GIM code.

3.2 TWO-DIMENSIONAL DEVELOPMENT

The GIM code formulation of the implicit MacCormack method begins with the Navier-Stokes equations in conservation law form. In two dimensions and by neglecting body force terms and heat sources, these equations can be written as

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho \mathcal{E} \end{bmatrix}$$

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + P - \tau_{xx} \\ \rho uv - \tau_{xy} \\ (\rho\mathcal{E} + P)\,u - u\tau_{xx} - v\tau_{xy} - q_x \end{bmatrix}$$

$$F = \begin{bmatrix} \rho v \\ \rho vu - \tau_{xy} \\ \rho v^2 + P - \tau_{yy} \\ (\rho\mathcal{E} + P)\,v - u\tau_{xy} - v\tau_{yy} - q_y \end{bmatrix}$$

where

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial y}{\partial y}\right)$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)$$

$$\tau_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

$$q_x = k \frac{\partial T}{\partial x}$$

$$q_y = k \frac{\partial T}{\partial y}$$

and where

$\boldsymbol{p}$ = mass density
v = y-component of velocity
$\mathcal{E}$ = total energy
λ = bulk viscosity coefficient
T = temperature
x, y = space coordinates

u = x-component of velocity
p = pressure
μ = kinematic viscosity coefficient
k = thermal conductivity
t = time coordinate

$$P = (\gamma - 1)\,\rho \left[\mathcal{E} - \frac{u^2 + v^2}{2}\right] \qquad \text{(ideal gas law)}$$

The subsequent development of the GIM code formulation parallels that of MacCormack up to the presentation of the finite difference equations (Eq. (8) of Ref. 3-1). Here the finite difference equations are written in a form compatible with the GIM code as follows:

$$P \begin{cases} \Delta U_i^n = -\left(\dfrac{\Delta_+ E_i^n}{\Delta x} + \dfrac{\Delta_+ F_i^n}{\Delta y}\right) & \text{(3.1a)} \\[2ex] \left(I - \Delta t\, \dfrac{\Delta_+ |A|}{\Delta x} \cdot\right)\left(I - \Delta t\, \dfrac{\Delta_+ |B|}{\Delta y} \cdot\right) \delta U_i^{\overline{n+1}} = \Delta U_i^n & \text{(3.1b)} \\[2ex] U_i^{\overline{n+1}} = U_i^n + \Delta t\, \delta U_i^{\overline{n+1}} & \text{(3.1c)} \end{cases}$$

$$C \begin{cases} \Delta U_i^{\overline{n+1}} = -\left(\dfrac{\Delta_- E_i^{\overline{n+1}}}{\Delta x} + \dfrac{\Delta_- F_i^{\overline{n+1}}}{\Delta y}\right) & \text{(3.2a)} \\[2ex] \left(I + \Delta t\, \dfrac{\Delta_- |A|}{\Delta x} \cdot\right)\left(I + \Delta t\, \dfrac{\Delta_- |B|}{\Delta y} \cdot\right) \delta U_i^{n+1} = \Delta U_i^{\overline{n+1}} & \text{(3.2b)} \\[2ex] U_i^{n+1} = U_i^{\overline{n+1}} + \dfrac{\Delta t}{2}\left(\delta U_i^{n+1} - \delta U_i^{\overline{n+1}}\right) & \text{(3.2c)} \end{cases}$$

where

n = time level $\qquad$ i = node number

$$\Delta U_i^n = \left(\frac{\partial U}{\partial t}\right)^n_{i,\ \text{explicit}} \qquad \delta U_i^{n+1} = \left(\frac{\partial U}{\partial t}\right)^{n+1}_{i,\ \text{implicit}}$$

and where $\frac{\Delta_+}{\Delta x}$, $\frac{\Delta_-}{\Delta x}$, $\frac{\Delta_+}{\Delta y}$ and $\frac{\Delta_-}{\Delta y}$ are difference operators given by (for any property Q)

$$
\begin{aligned}
\frac{\Delta_+ Q_i}{\Delta x} &= \sum_{j=0}^{NCON} a^+_{i+j} Q_{i+j} \equiv \text{Forward analog of } \left(\frac{\partial Q}{\partial x}\right)_i \\
\frac{\Delta_- Q_i}{\Delta x} &= \sum_{j=0}^{NCON} a^-_{i-j} Q_{i-j} \equiv \text{Backward analog of } \left(\frac{\partial Q}{\partial x}\right)_i \\
\frac{\Delta_+ Q_i}{\Delta y} &= \sum_{j=0}^{NCON} b^+_{i+j} Q_{i+j} \equiv \text{Forward analog of } \left(\frac{\partial Q}{\partial y}\right)_i \\
\frac{\Delta_- Q_i}{\Delta y} &= \sum_{j=0}^{NCON} b^-_{i-j} Q_{i-j} \equiv \text{Backward analog of } \left(\frac{\partial Q}{\partial y}\right)_i
\end{aligned}
\qquad (3.3)
$$

The coefficients a^+_i, a^-_i, b^+_i and b^-_i used above are the finite difference coefficients at node i obtained from the GIM interpolant scheme for the explicit MacCormack method on an arbitrary mesh with NCON connections to adjacent nodes. The Jacobian matrices $|A|$ and $|B|$ are given by

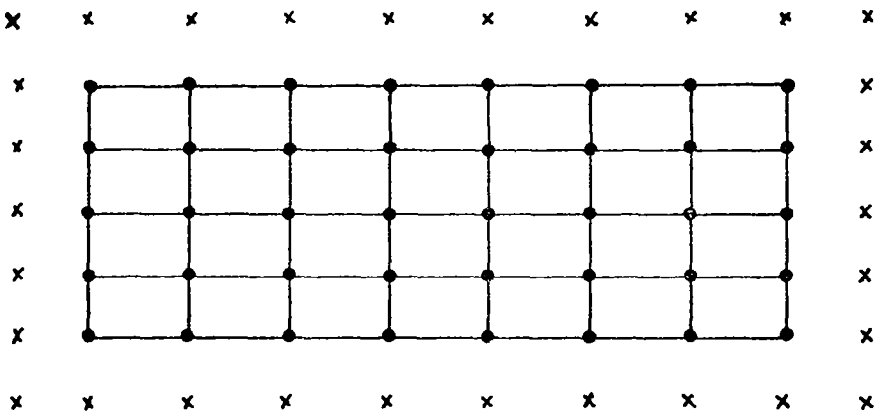
$$|A| = S_x^{-1} D_A S_x \qquad \text{and} \qquad |B| = S_y^{-1} D_B S_y$$

with S_x, S_y, D_A and D_B as defined in Ref. 3-1 and presented in Appendix 3.A at the end of this section.

The solution of Eqs. (3.1) and (3.2) proceeds as outlined in Ref. 3-1 with two exceptions. First, the use of the general difference operators given by Eq. (3.3) produces a set of difference equations that is either upper or lower block triangular. These equations are solved directly via backward or forward substitution with little more computational work than for the block bidiagonal system, which is a special case for rectangular meshes. The second deviation from the process described in Ref. 3-1 involves the implementation of boundary conditions and is discussed next.

3.3 BOUNDARY CONDITION DEVELOPMENT

The solution process in the published implicit MacCormack method involves forward and backward "sweeping" over the finite-difference mesh points up to but not including the boundary points. Boundary values are fixed in terms of U and/or ΔU. Note, however, that in the GIM code formulation, the explicit-implicit sequence, Eqs. (3.1a), (3.1b) and (3.2a), (3.2b), operate with and on $\frac{\partial U}{\partial t}$ instead of ΔU. Boundary values are set in terms of $\frac{\partial U}{\partial t}$ via constraints derived from a quasi-variational technique (Ref. 3-3). To accommodate these boundary conditions, the GIM code formulation employs a set of pseudo-mesh points around the periphery of the actual mesh, as illustrated below:



● Real Mesh Points x Pseudo-Mesh Points

At these pseudo-mesh points, the value of $\frac{\partial U}{\partial t}$ is set to zero for all time; the value of U is not set nor indeed ever required. The GIM code explicit step, Eqs.(3.1a) and (3.2a), proceeds as usual, utilizing values at real mesh points only. The implicit step, Eqs.(3.1b) and (3.2b), now involves forward and backward sweeping on all real mesh points including the boundary points. Boundary conditions are imposed on $\frac{\partial U}{\partial t}$ after the explicit step, Eqs. (3.1a) and (3.2a), and again after the implicit step, Eqs.(3.1b) and (3.2b). The pseudo-nodes are used in the implicit differencing scheme to ensure that the resulting difference matrix is a strictly upper or lower block triangular matrix. The boundary value of $\frac{\partial U}{\partial t} \equiv 0$ at the pseudo-nodes is used to drive the solution to the steady-state condition $\frac{\partial U}{\partial t} = 0$ everywhere.

The boundary condition treatment described above is easily extended to non-rectangular and non-orthogonal meshes in both two and three dimensions. This flexibility is inherent in the general difference operators, Eqs.(3.3), employed in the GIM code and permits consideration of a wide range of geometrical configurations in a very convenient manner.

3.4 THREE-DIMENSIONAL DEVELOPMENT

The three-dimensional implicit GIM code formulation is very similar to the two-dimensional formulation. The factored operator and Jacobian diagonalization approach used in the implicit MacCormack method allow a simple straightforward extension to three dimensions with only a relatively small increase in computational labor. The three-dimensional implicit GIM code equations are presented in Appendix 3.B at the end of this section.

3.5 RESULTS AND DISCUSSIONS

The implicit MacCormack method has been formulated and coded to be compatible with the GIM code. Both the two- and three-dimensional versions are available as an option in the current INTEG module. Only limited experience with the method has been acquired thus far and this almost exclusively with simple two-dimensional problems on rectangular meshes. However, this has been sufficient to establish several important characteristics of the method.

3.5.1 Vectorization

Like other implicit methods, the implicit MacCormack method does not appear to be amenable to vectorization for use on the STAR or other vector machines. Thus the GIM code implicit routines execute much more slowly than do the highly vectorized explicit routines. Some gain has been achieved by partially vectorizing the computation of the eigenvalue vectors and the components of the S matrices. However, the bulk of the implicit routines including the backward and forward mesh sweeps and imposition of boundary conditions remains in scalar coding. As a result, it appears that implicit runs will execute about three times slower per iteration than explicit runs. Some additional gain may yet be achieved through refinement of the implicit coding as more experience is acquired.

3.5.2 Stability

The published implicit MacCormack method of Ref. 3-1 is described as being stable for unbounded Δt. However, this has not been the case with the implicit GIM code formulation. Thus far, the upper bound on Δt for stability appears to be the value of Δt at which the flow field computation begins to become implicit in the streamwise or dominant flow direction, i.e., $CFL_{streamwise} = 1.0$. For a two-dimensional problem on a rectangular mesh with the streamwise direction in the x-coordinate direction, this upper bound is given by Eq. (9) of Ref. 3-1 as

$$\Delta t \leq \frac{1/2}{(|u| + c)/\Delta x + (2\nu/\rho\Delta x^2)}$$

In no instance has an implicit GIM code calculation been able to exceed this upper-bound and remain stable. Even calculations which have been integrated in time to an apparent steady state quickly become unstable if Δt is increased beyond the $CFL_{streamwise} \leq 1.0$ limit.

While there is yet no explanation for the apparent descrepancy between the reported and observed stability of the method, several items are being investigated. Variations in the boundary condition implementation, modification of the eigenvalue terms, and permutations of the x-y operator sequence and forward-forward/backward-backward sequence are all being examined to determine their effect on the stability limit.

An important point to note is that even if the $CFL_{streamwise} \leq 1.0$ stability limit cannot be relaxed or circumvented, the implicit GIM code option will remain a very useful and valuable addition to the code. Many problems of current interest require a highly refined mesh in directions normal to the dominant flow direction in order to resolve boundary layers and other viscous phenomena. In these instances, the implicit GIM code option can be used to run with a time step many times larger than that imposed by explicit stability requirements ($CFL \leq 1.0$) and yet remain within the Δt bound imposed by the $CFL_{streamwise} \leq 1.0$ limit. As an example, a simple two-dimensional Couette flow has been integrated using the implicit GIM code option with time steps as large as 18 times that required by explicit stability requirements and remained stable. It is envisioned that the gain over the explicit stability limit could be much larger for highly refined meshes.

3.5.3 Recommendations

While the vectorization possibilities and stability limit should continue to be investigated, the implicit GIM code option can be used to calculate several more difficult and realistic flow fields, including a shock wave-boundary layer interaction problem, a two-dimensional inlet flow with spillage, a three-dimensional internal duct flow, and a turbulent boundary layer problem. In the near future, the implicit GIM code option can be coupled to the Quasi-Parabolic (QP) routines to allow efficient computation of the flow over wing-body and missile configurations. Other possibilities included coupling the implicit routines with the chemistry routines or the two-equation turbulent kinetic energy model.

SECTION 3 REFERENCES

3.1 MacCormack, R. W., "A Numerical Method for Solving the Equations of Compressible Viscous Flow," AIAA Paper No. 81-0110, January 1981.

3.2 MacCormack, R. W , "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper No. 69-354, May 1969.

3.3 Prozan, R. J., L. W. Spradley, P. G. Anderson and M. L. Pearson, "The General Interpolants Method: A Procedure for Generating Numerical Analogs of the Conservation Laws," AIAA Paper 77-642, June 1977.

Appendix 3.A
TWO-DIMENSIONAL IMPLICIT GIM CODE EQUATIONS

Governing Equations

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho \mathcal{E} \end{bmatrix}$$

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + P - \tau_{xx} \\ \rho uv - \tau_{xy} \\ (\rho \mathcal{E} + P)\, u \qquad - u\tau_{xx} - v\tau_{xy} - q_x \end{bmatrix}$$

$$F = \begin{bmatrix} \rho v \\ \rho vu - \tau_{xy} \\ \rho v^2 + P - \tau_{yy} \\ (\rho \mathcal{E} + P)\, v - u\tau_{xy} - v\tau_{yy} - q_y \end{bmatrix}$$

where

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)$$

$$\tau_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

$$q_x = k \frac{\partial T}{\partial x}$$

$$q_y = k \frac{\partial T}{\partial y}$$

and where

ρ =	mass density	u =	x-component of velocity
v =	y-component of velocity	P =	pressure
$\mathcal{E}$ =	total energy	μ =	kinematic viscosity coefficient
λ =	bulk viscosity coefficient	k =	thermal conductivity
T =	temperature	t =	time coordinate
x, y =	space coordinates		

$$P = (\gamma - 1)\,\rho \left[\mathcal{E} - \frac{u^2 + v^2}{2}\right] \qquad \text{(ideal gas law)}$$

Finite Difference Equations

$$P \left\{ \begin{aligned}
&\Delta U_i^n = -\left(\frac{\Delta_+ E_i^n}{\Delta x} + \frac{\Delta_+ F_i^n}{\Delta y}\right) \\
&\left(I - \Delta t \frac{\Delta_+ |A|}{\Delta x} \cdot\right)\left(I - \Delta t \frac{\Delta_+ |B|}{\Delta y} \cdot\right) \delta U_i^{\overline{n+1}} = \Delta U_i^n \\
&U_i^{\overline{n+1}} = U_i^n + \Delta t\, \delta U_i^{\overline{n+1}}
\end{aligned} \right.$$

$$C \left\{ \begin{aligned}
&\Delta U_i^{\overline{n+1}} = -\left(\frac{\Delta_- E_i^{\overline{n+1}}}{\Delta x} + \frac{\Delta_- F_i^{\overline{n+1}}}{\Delta y}\right) \\
&\left(I + \Delta t \frac{\Delta_- |A|}{\Delta x} \cdot\right)\left(I + \Delta t \frac{\Delta_- |B|}{\Delta y} \cdot\right) \delta U_i^{n+1} = \Delta U_i^{\overline{n+1}} \\
&U_i^{n+1} = U_i^{\overline{n+1}} + \frac{\Delta t}{2}\left(\delta U_i^{n+1} - \delta U_i^{\overline{n+1}}\right)
\end{aligned} \right.$$

where

$$n = \text{time level} \qquad i = \text{node number}$$

$$\Delta U_i^n = \left(\frac{\partial U}{\partial t}\right)_{i,\,\text{explicit}}^n \qquad \delta U_i^{n+1} = \left(\frac{\partial U}{\partial t}\right)_{i,\,\text{implicit}}^{n+1}$$

and where $\frac{\Delta_+}{\Delta x}$, $\frac{\Delta_-}{\Delta x}$, $\frac{\Delta_+}{\Delta y}$ and $\frac{\Delta_-}{\Delta y}$ are difference operators given by (for any property Q)

$$\frac{\Delta_+ Q_i}{\Delta x} = \sum_{j=0}^{NCON} a_{i+j}^+ Q_{i+j} \equiv \text{Forward analog of } \left(\frac{\partial Q}{\partial x}\right)_i$$

$$\frac{\Delta_- Q_i}{\Delta x} = \sum_{j=0}^{NCON} a_{i-j}^- Q_{i-j} \equiv \text{Backward analog of } \left(\frac{\partial Q}{\partial x}\right)_i$$

$$\frac{\Delta_+ Q_i}{\Delta y} = \sum_{j=0}^{NCON} b_{i+j}^+ Q_{i+j} \equiv \text{Forward analog of } \left(\frac{\partial Q}{\partial y}\right)_i$$

$$\frac{\Delta_- Q_i}{\Delta y} = \sum_{j=0}^{NCON} b_{i-j}^- Q_{i-j} \equiv \text{Backward analog of } \left(\frac{\partial Q}{\partial y}\right)_i$$

with finite difference coefficients a_i^+, a_i^-, b_i^+ and b_i^-.

Jacobian Matrices

$$|A| = S_x^{-1} D_A S_x \qquad |B| = S_y^{-1} D_B S_y$$

where

$$S_x = \begin{bmatrix} 1 & 0 & 0 & -1/c^2 \\ 0 & \rho c & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & -\rho c & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -u/\rho & 1/\rho & 0 & 0 \\ -v/\rho & 0 & 1/\rho & 0 \\ \alpha\beta & -u\beta & -v\beta & \beta \end{bmatrix}$$

$$S_y = \begin{bmatrix} 1 & 0 & 0 & -1/c^2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \rho c & 1 \\ 0 & 0 & -\rho c & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -u/\rho & 1/\rho & 0 & 0 \\ -v/\rho & 0 & 1/\rho & 0 \\ \alpha\beta & -u\beta & -v\beta & \beta \end{bmatrix}$$

$$c = \sqrt{\gamma P/\rho} \qquad \alpha = 1/2\,(u^2 + v^2) \qquad \beta = \gamma - 1$$

$$D_A = \begin{bmatrix} \lambda_{A1} & 0 & 0 & 0 \\ 0 & \lambda_{A2} & 0 & 0 \\ 0 & 0 & \lambda_{A3} & 0 \\ 0 & 0 & 0 & \lambda_{A4} \end{bmatrix}$$

$$D_B = \begin{bmatrix} \lambda_{B1} & 0 & 0 & 0 \\ 0 & \lambda_{B2} & 0 & 0 \\ 0 & 0 & \lambda_{B3} & 0 \\ 0 & 0 & 0 & \lambda_{B4} \end{bmatrix}$$

$$\lambda_{A1} = \max \left\{ |u| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t} , 0.0 \right\}$$

$$\lambda_{A2} = \max \left\{ |u + c| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t} , 0.0 \right\}$$

$$\lambda_{A3} = \max \left\{ |u| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t} , 0.0 \right\}$$

$$\lambda_{A4} = \max \left\{ |u - c| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t} , 0.0 \right\}$$

$$\lambda_{B1} = \max \left\{ |v| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t} , 0.0 \right\}$$

$$\lambda_{B2} = \max \left\{ |v| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t} , 0.0 \right\}$$

$$\lambda_{B3} = \max \left\{ |v + c| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t} , 0.0 \right\}$$

$$\lambda_{B4} = \max \left\{ |v - c| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t} , 0.0 \right\}$$

$$\nu = \max(\mu , \lambda + 2\mu, k/c_v)$$

Appendix 3.B

THREE-DIMENSIONAL IMPLICIT GIM CODE EQUATIONS

Govering Equations

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = 0$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho \mathcal{E} \end{bmatrix}$$

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + P - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ (\rho \mathcal{E} + P)\, u - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} - q_x \end{bmatrix}$$

$$F = \begin{bmatrix} \rho v \\ \rho vu - \tau_{xy} \\ \rho v^2 + P - \tau_{yy} \\ \rho vw - \tau_{yz} \\ (\rho \mathcal{E} + P)\, v - u\tau_{xy} - v\tau_{yy} - w\tau_{yz} - q_y \end{bmatrix}$$

$$G = \begin{bmatrix} \rho w \\ \rho w u - \tau_{xz} \\ \rho w v - \tau_{yz} \\ \rho w^2 + P - \tau_{zz} \\ (\rho \mathcal{E} + P)\, w - u\tau_{xz} - v\tau_{yz} - w\tau_{zz} - q_z \end{bmatrix}$$

where

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$

$$\tau_{zz} = 2\mu \frac{\partial w}{\partial z} + \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$

$$\tau_{xy} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

$$\tau_{xz} = \mu \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)$$

$$\tau_{yz} = \mu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)$$

$$q_x = k \frac{\partial T}{\partial x}$$

$$q_y = k \frac{\partial T}{\partial y}$$

$$q_z = k \frac{\partial T}{\partial z}$$

and where

ρ	=	mass density	u	=	x-component of velocity
$\mathcal{E}$	=	total energy	w	=	z-component of velocity
P	=	pressure	v	=	y-component of velocity
μ	=	kinematic viscosity coefficient	λ	=	bulk viscosity coefficient
k	=	thermal conductivity	T	=	temperature

x, y, z, t = space and time coordinates

$$P = (\gamma - 1)\,\rho \left[\mathcal{E} - \frac{u^2 + v^2 + w^2}{2}\right] \quad \text{(ideal gas law)}$$

Finite Difference Equations

$$P \left\{ \begin{aligned} &\Delta U_i^n = -\left(\frac{\Delta_+ E_i^n}{\Delta x} + \frac{\Delta_+ F_i^n}{\Delta y} + \frac{\Delta_+ G_i^n}{\Delta z}\right) \\ &\left(I - \Delta t\,\frac{\Delta_+ |A|}{\Delta x}\,\cdot\right)\left(I - \Delta t\,\frac{\Delta_+ |B|}{\Delta y}\,\cdot\right)\left(I - \Delta t\,\frac{\Delta_+ |C|}{\Delta z}\right) \delta U_i^{\overline{n+1}} = \Delta U_i^n \\ &U_i^{\overline{n+1}} = U_i^n + \Delta t\,\delta U_i^{\overline{n+1}} \end{aligned} \right.$$

$$P \left\{ \begin{aligned} &\Delta U_i^{n+1} = -\left(\frac{\Delta_- E_i^n}{\Delta x} + \frac{\Delta_- F_i^n}{\Delta y} + \frac{\Delta_- G_i^n}{\Delta z}\right) \\ &\left(I + \Delta t\,\frac{\Delta_- |A|}{\Delta x}\,\cdot\right)\left(I + \Delta t\,\frac{\Delta_- |B|}{\Delta z}\,\cdot\right)\left(I + \Delta t\,\frac{\Delta_- |C|}{\Delta z}\,\cdot\right) \delta U_i^{n+1} = \Delta U_i^{\overline{n+1}} \\ &U^{n+1} = U^{\overline{n+1}} + \frac{\Delta t}{2}\left(\delta U_i^{n+1} - \delta U_i^{\overline{n+1}}\right) \end{aligned} \right.$$

where

n = time level $\qquad$ i = node number

$$\Delta U_i^n = \left(\frac{\partial U}{\partial t}\right)_{i,\,\text{explicit}}^{n} \qquad \delta U_i^{n+1} = \left(\frac{\partial U}{\partial t}\right)_{i,\,\text{implicit}}^{n+1}$$

and where $\frac{\Delta_+}{\Delta x}$, $\frac{\Delta_-}{\Delta x}$, $\frac{\Delta_+}{\Delta y}$, $\frac{\Delta_-}{\Delta y}$, $\frac{\Delta_+}{\Delta z}$ and $\frac{\Delta_-}{\Delta z}$ are difference operators given by (for any property Q)

$$\frac{\Delta_+ Q_i}{\Delta x} = \sum_{j=0}^{NCON} a^+_{i+j} Q_{i+j} \equiv \text{Forward analog of } \left(\frac{\partial Q}{\partial x}\right)_i$$

$$\frac{\Delta_- Q_i}{\Delta x} = \sum_{j=0}^{NCON} a^-_{i-j} Q_{i-j} \equiv \text{Backward analog of } \left(\frac{\partial Q}{\partial x}\right)_i$$

$$\frac{\Delta_+ Q_i}{\Delta y} = \sum_{j=0}^{NCON} b^+_{i+j} Q_{i+j} \equiv \text{Forward analog of } \left(\frac{\partial Q}{\partial y}\right)_i$$

$$\frac{\Delta_- Q_i}{\Delta y} = \sum_{j=0}^{NCON} b^-_{i-j} Q_{i-j} \equiv \text{Backward analog of } \left(\frac{\partial Q}{\partial y}\right)_i$$

$$\frac{\Delta_+ Q_i}{\Delta z} = \sum_{j=0}^{NCON} c^+_{i+j} Q_{i+j} \equiv \text{Forward Analog of } \left(\frac{\partial Q}{\partial z}\right)_i$$

$$\frac{\Delta_- Q_i}{\Delta z} = \sum_{j=0}^{NCON} c^-_{i-j} Q_{i-j} \equiv \text{Backward Analog of } \left(\frac{\partial Q}{\partial z}\right)_i$$

with finite difference coefficients a^+_i, a^-_i, b^+_i, b^-_i, c^+_i and c^-_i.

Jacobian Matrices

$$|A| = S_x^{-1} D_A S_x \qquad |B| = S_y^{-1} D_B S_y \qquad |C| = S_z^{-1} D_C S_z$$

where

$$S_x = \begin{bmatrix} 1 & 0 & 0 & 0 & -1/c^2 \\ 0 & \rho c & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -\rho c & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -u/\rho & 1/\rho & 0 & 0 & 0 \\ -v/\rho & 0 & 1/\rho & 0 & 0 \\ -w/\rho & 0 & 0 & 1/\rho & 0 \\ \alpha\beta & -u\beta & -v\beta & -w\beta & \beta \end{bmatrix}$$

$$S_y = \begin{bmatrix} 1 & 0 & 0 & 0 & -1/c^2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \rho c & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\rho c & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -u/\rho & 1/\rho & 0 & 0 & 0 \\ -v/\rho & 0 & 1/\rho & 0 & 0 \\ -w/\rho & 0 & 0 & 1/\rho & 0 \\ \alpha\beta & -u\beta & -v\beta & -w\beta & \beta \end{bmatrix}$$

$$S_z = \begin{bmatrix} 1 & 0 & 0 & 0 & -1/c^2 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \rho c & 1 \\ 0 & 0 & 0 & -\rho c & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -u/\rho & 1/\rho & 0 & 0 & 0 \\ -v/\rho & 0 & 1/\rho & 0 & 0 \\ -w/\rho & 0 & 0 & 1/\rho & 0 \\ \alpha\beta & -u\beta & -v\beta & -w\beta & \beta \end{bmatrix}$$

$$c = \gamma P/\rho \qquad \alpha = 1/2\,(u^2 + v^2 + w^2) \qquad \beta = \gamma - 1$$

$$D_A = \begin{bmatrix} \lambda_{A1} & 0 & 0 & 0 & 0 \\ 0 & \lambda_{A2} & 0 & 0 & 0 \\ 0 & 0 & \lambda_{A3} & 0 & 0 \\ 0 & 0 & 0 & \lambda_{A4} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{A5} \end{bmatrix}$$

$$D_B = \begin{bmatrix} \lambda_{B1} & 0 & 0 & 0 & 0 \\ 0 & \lambda_{B2} & 0 & 0 & 0 \\ 0 & 0 & \lambda_{B3} & 0 & 0 \\ 0 & 0 & 0 & \lambda_{B4} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{B5} \end{bmatrix}$$

$$D_C = \begin{bmatrix} \lambda_{C1} & 0 & 0 & 0 & 0 \\ 0 & \lambda_{C2} & 0 & 0 & 0 \\ 0 & 0 & \lambda_{C3} & 0 & 0 \\ 0 & 0 & 0 & \lambda_{C4} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{C5} \end{bmatrix}$$

$$\lambda_{A1} = \max \left\{ |u| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0.0 \right\}$$

$$\lambda_{A2} = \max \left\{ |u + c| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0.0 \right\}$$

$$\lambda_{A3} = \max \left\{ |u| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0.0 \right\}$$

$$\lambda_{A4} = \max \left\{ |u| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0.0 \right\}$$

$$\lambda_{A5} = \max \left\{ |u - c| + \frac{2\nu}{\rho\Delta x} - \frac{1}{2}\frac{\Delta x}{\Delta t},\ 0.0 \right\}$$

$$\lambda_{B1} = \max \left\{ |v| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t},\ 0.0 \right\}$$

$$\lambda_{B2} = \max \left\{ |v| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t},\ 0.0 \right\}$$

$$\lambda_{B3} = \max \left\{ |v + c| + \frac{2\nu}{\rho\Delta y} - \frac{1}{2}\frac{\Delta y}{\Delta t},\ 0.0 \right\}$$

$$\lambda_{B4} = \max \left\{ \left| v \right| + \frac{2\nu}{\rho \Delta y} - \frac{1}{2} \frac{\Delta y}{\Delta t}, 0.0 \right\}$$

$$\lambda_{B5} = \max \left\{ \left| v - c \right| + \frac{2\nu}{\rho \Delta y} - \frac{1}{2} \frac{\Delta y}{\Delta t}, 0.0 \right\}$$

$$\lambda_{C1} = \max \left\{ \left| w \right| + \frac{2\nu}{\rho \Delta z} - \frac{1}{2} \frac{\Delta z}{\Delta t}, 0.0 \right\}$$

$$\lambda_{C2} = \max \left\{ \left| w \right| + \frac{2\nu}{\rho \Delta z} - \frac{1}{2} \frac{\Delta z}{\Delta t}, 0.0 \right\}$$

$$\lambda_{C3} = \max \left\{ \left| w \right| + \frac{2\nu}{\rho \Delta z} - \frac{1}{2} \frac{\Delta z}{\Delta t}, 0.0 \right\}$$

$$\lambda_{C4} = \max \left\{ \left| w + c \right| + \frac{2\nu}{\rho \Delta z} - \frac{1}{2} \frac{\Delta z}{\Delta t}, 0.0 \right\}$$

$$\lambda_{C5} = \max \left\{ \left| w - c \right| + \frac{2\nu}{\rho \Delta z} - \frac{1}{2} \frac{\Delta z}{\Delta t}, 0.0 \right\}$$

$$\nu = \max(\mu, \lambda + 2\mu, k/c_v)$$

4. TURBULENCE MODEL IMPLEMENTATION

4.1 REVIEW

The ability to compute actual time-dependent turbulent flow fields is beyond the range of current and foreseeable computer resources. It is impossible to refine the computational mesh to the extent necessary to resolve the wide range of eddy length scales involved without exceeding computer speed and memory bounds. Such is also the case with any attempt to determine the instantaneous behavior of the rapid turbulent fluctuations about the mean flowfield solution. As a result the effects of turbulence must be described by modeling.

To avoid resolving every possible mode of turbulent flow, the first step in turbulence modeling is to time-average the Navier-Stokes equations. In this process the conserved variables are written as a sum of a time-average mean flow and a fluctuating part. Casting the Navier-Stokes equations in terms of these averaged variables results in a set of governing equations similar to the original set with some new terms referred to as the Reynolds stress and turbulent heat transfer terms. Rubesin and Rose (Ref. 4-1) use time-mass-averaged variables to arrive at a set of governing equations which are particularly useful in compressible flow calculations.

The Reynolds stress and turbulent heat transfer terms (hereafter referred to only as Reynolds stress terms) are now unknowns in the governing equations and closure of the system must be provided through empirical relationships or "turbulence models." Marvin (Ref. 4-2) provides an excellent review of turbulence modeling in compressible flows. There are two major classes of turbulence models available. The first of these models takes advantage of the Boussinesq hypothesis which states that the Reynolds stress terms in the Navier-Stokes equations can be modeled as the product of the phenomenological "eddy" viscosity and the appropriately averaged local velocity gradient. These models then depend only on local mean flow quantities and empirically determined constants. The second class involves models that use fundamental transport equations to obtain more directly the spatial variation of the Reynolds' stress throughout the flow. These models relate the turbulent shear stress (turbulent eddy viscosity) to the turbulent kinetic energy and are called the TKE models.

Eddy Viscosity Models: There are many algebraic eddy viscosity models available to directly relate the Reynolds stress terms to the mean flow conditions. One of the most widely used is that developed by Cebeci (Ref. 4-3). It is a two-layer model which employs a Prandtl-Van Driest mixing length formulation in the near-wall region and the Clauser wake formulation in the outer region (or in wakes). This model and, in general, most of the algebraic closures perform well for calculating attached turbulent boundary layers in zero or mild pressure gradients.

The Cebeci-Smith model was modified by Shang and Hankey (Ref. 4-4) to include a relaxation length for the eddy viscosity. This model was found to give good predictions in regions of flow separation under the influences of strong adverse pressure gradients.

Another eddy viscosity model in wide use is that of Baldwin and Lomax (Ref. 4-5) This model was developed for use in two- and three-dimensional Navier-Stokes codes and is applicable to separated flows in shear layers, wakes and wall boundary layers. This model is similar to that of Cebeci but it avoids the necessity of determining the outer edge of the boundary layer (or wake) by employing an approximation to the Clauser formulation in the outer region. This approximation amounts to the use of a vorticity distribution function rather than the boundary layer thickness to determine the length scales.

Turbulent Kinetic Energy (TKE) Models: The modeling of turbulent kinetic energy in compressible turbulent shear flows is discussed in detail in Ref. 4-2. In these models the governing equations are supplemented by additional transport equations for turbulent kinetic energy and the various Reynolds stresses (Ref. 4-3). For computational efficiency, the most widely used models add only one or two extra transport equations. These are commonly referred to as one-equation and two-equation models.

One-Equation Models — In its original form, the one-equation model involves the solution of the turbulent kinetic energy equations. The values for the constants involved in the method were obtained through correlation of a variety of experimental data (Ref. 4-6). However, the model did not incorporate the compressibility corrections described in Ref. 4-7. This correction involves the modification of the dissipation constant. The introduction of the compressibility correlation provides a radical improvement in the ability of the one-equation model to predict flowfield development at supersonic Mach numbers. This corrected correlation is described in Ref. 4-8. Rubesin (Ref. 4-9) has also devised a one-equation model for compressible flows from an extension of Glushko's (Ref. 4-10) incompressible model.

Two-Equation Models — There have been a number of two-equation turbulent kinetic energy models developed. The fundamental differences between the various two-equation models and the one-equation model just described are that in the two-equation models a transport equation is written for the turbulence energy dissipation length scale, (or equivalently, the turbulence energy dissipation rate) and the Prandtl-Kolmogorov relation is used to obtain the turbulent eddy viscosity. The most highly developed two-equation model is that given in Launder et al. (Ref. 4-11). The use of this so-called "k-ϵ" model is advocated and described by Launder and Spalding (Ref. 4-12). However, this two-equation model does not include a compressibility correction, and predictions made using this model for highly compressible flows are not accurate. Pan describes an extension of this model to three-dimensional steady compressible flows (Ref. 4-13).

In general, TKE models provide more flexibility and give better predictions of turbulent flows than do the algebraic models. However, adjustments of the empirical constants in these models are required depending on flow conditions. The user must weigh the increased accuracy of the TKE models against the increase in required computation time and storage in order to choose between the methods.

4.2 MODELS CURRENTLY IN GIM

At the time of this writing the GIM code user may choose between the following turbulence models:

- The Baldwin-Lomax algebraic eddy viscosity model for elliptic and QP calculations in the presence of one wall.

- The "k-ϵ" two-equation TKE model for two- and three-dimensional elliptic and QP calculations.

The Algebraic Model: The eddy viscosity model presently in the code is a variation of the Baldwin-Lomax model (Ref. 4-5). The turbulence effects are modeled by an eddy viscosity coefficient μ_T which enters into the flow equations through a modified total viscosity:

$$\mu = \mu_L + \mu_T \quad , \tag{4.1}$$

where the laminar viscosity μ_L can assume a constant value or be calculated from the temperature via Sutherland's law. The thermal conductivity is determined through the input of a laminar and turbulent Prandtl number (Pr and Pr_T, respectively) as

$$K/C_p = \mu/Pr + \mu_T/Pr_T \tag{4.2}$$

The eddy viscosity is given by a two-layer model:

$$\mu_T = \begin{cases} (\mu_T)_{inner} & r \leq r_c \\ (\mu_T)_{outer} & r > r_c \end{cases} \tag{4.3}$$

The description of the formulae used to calculate the inner and outer solutions is given in Ref. 4-5 and will not be repeated here. The only variation of the model in the code from that in Ref. 4-5 is in the definition of r (called y in the reference). In the GIM code r is the distance to the node point of interest in the computational grid from the node point on the solid wall which lies along the row of nodes including the point of interest. The symbol r_c denotes the least value of r for which the inner and outer viscosities are equal. Thus, the GIM code version and the model in Ref. 4-5 are equivalent if one set of grid lines is maintained perpendicular to the wall, i.e., orthogonal grids. It is planned to extend the present model to multiple walls.

The TKE Model: The differential equation model presently in the code is a time-dependent extension of that proposed by Pan (Ref. 4-13). This model requires the solution of two additional equations. The first one is for the turbulent kinetic energy

$$k = \frac{1}{2} (\overline{u'^2} + \overline{v'^2} + \overline{w'^2}) \tag{4.4}$$

where $\overline{u'^2}$, etc., are the mean square turbulent fluctuations of the velocity about the mean velocity profile. The other equation describes the transport of the rate of dissipation of turbulent kinetic energy, ϵ. The variables k and ϵ are related through an extension of the Prandtl-Kolmogorov relation for the turbulent eddy viscosity

$$\mu_T = C_\mu \rho k^2/\epsilon \tag{4.5}$$

where C_μ is an empirical constant. The governing equations for the flow field are given in Fig. 4-1. $C_{\epsilon 1}$ and $C_{\epsilon 2}$ are empirical constants. This model is in the code and has been compiled but no test cases have been completed as of this writing. A few questions remain to be addressed on the use of this model. The exact treatment of the boundary conditions on k and ϵ at solid walls is in question since the codes which use these models do not treat real solid walls but use extrapolated "wall functions." This boundary treatment has been known

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} + H = 0$$

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ \rho k \\ \rho \epsilon \end{bmatrix}$$

$$E = \begin{bmatrix} \rho u - \sigma_x \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ (\rho E+p)u - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} - q_x \\ \rho uk - \mu_T \; \partial k/\partial x \\ \rho u\epsilon - \mu_T \; \partial \epsilon/\partial x \end{bmatrix}$$

$$F = \begin{bmatrix} \rho v - \sigma_y \\ \rho uv - \tau_{yx} \\ \rho v^2 + p - \tau_{yy} \\ \rho vw - \tau_{yz} \\ (\rho E+p) v - u\tau_{yx} - v\tau_{yy} - w\tau_{yz} - q_y \\ \rho vk - \mu_T \; \partial k/\partial y \\ \rho v\epsilon - \mu_T \; \partial \epsilon/\partial y \end{bmatrix}$$

$$G = \begin{bmatrix} \rho w - \sigma_z \\ \rho uw - \tau_{zx} \\ \rho vw - \tau_{zy} \\ \rho w^2 + p - \tau_{zz} \\ (\rho E+p) w - u\tau_{zx} - v\tau_{yz} - w\tau_{zz} - q_z \\ \rho wk - \mu_T \; \partial k/\partial z \\ \rho w\epsilon - \mu_T \; \partial \epsilon/\partial z \end{bmatrix}$$

$$H = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \rho\epsilon - \xi \\ \frac{\epsilon}{k}(C_{\epsilon 2}\,\rho\epsilon - C_{\epsilon 1}\,\xi) \end{bmatrix}$$

where

$$\mu_T = C_\mu \rho k^2/\epsilon$$

$$\mu = \mu_L + \mu_T + \mu_d$$

$$k = \frac{\mu_L C_p}{Pr} + \frac{\mu_T C_p}{Pr_T} + C_2 \mu_d$$

$$\lambda = C_1 \mu$$

$$\xi = 2\mu_T \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2\right] + \mu_T \left[\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2\right]$$

Fig. 4-1 - TKE Model in the GIM Code

to yield poor results in the near-wall region where the model predictions of the turbulent length scale, $k^{3/2}/\epsilon$, vary dramatically from experimental results (Ref. 4-14). The ability of the model to compute turbulence in the presence of pressure gradients strong enough to cause separation or in recirculating flow is in doubt. The greatest obstacle to proper boundary treatment at solid walls is a paucity of experimental information from which to determine the behavior of the dissipation at the wall. For compressible flows it is not known how much information about density fluctuations is obscured by mass-weighted averaging; nor, has the ability of models averaged in this way to compute flows with large density or pressure gradients been verified.

The need for artificial damping to stabilize the time-iterative solution of these equations has also not been studied. The specific form of the damping terms needs to be derived.

4.3 RESULTS OF COMPUTATION

The two-dimensional spillage problem defined in Ref. 4.15 was used as a test case.

The turbulent flow field over the leading edge of this 25-deg compression surface of the model inlet was calculated. Figure 4-2 shows the 840-node computational grid. The velocity vectors, pressure, and Mach contours are shown in Figs. 4-3, 4-4 and 4-5, respectively.

The flow was initialized at freestream values everywhere except at the walls where a zero velocity condition was imposed. All features of the flow were then calculated by the GIM code. Convergence was assumed after 1500 iterations when the sum of squares of the unsteady derivatives were decreasing by less than 1 percent. The turbulent viscosity was determined with the Baldwin-Lomax turbulence model and the laminar viscosity was held constant. The freestream Reynolds number per unit length was approximately 2×10^5.

These results will be used to form the initial conditions for the calculation of the remainder of the turbulent flow field.

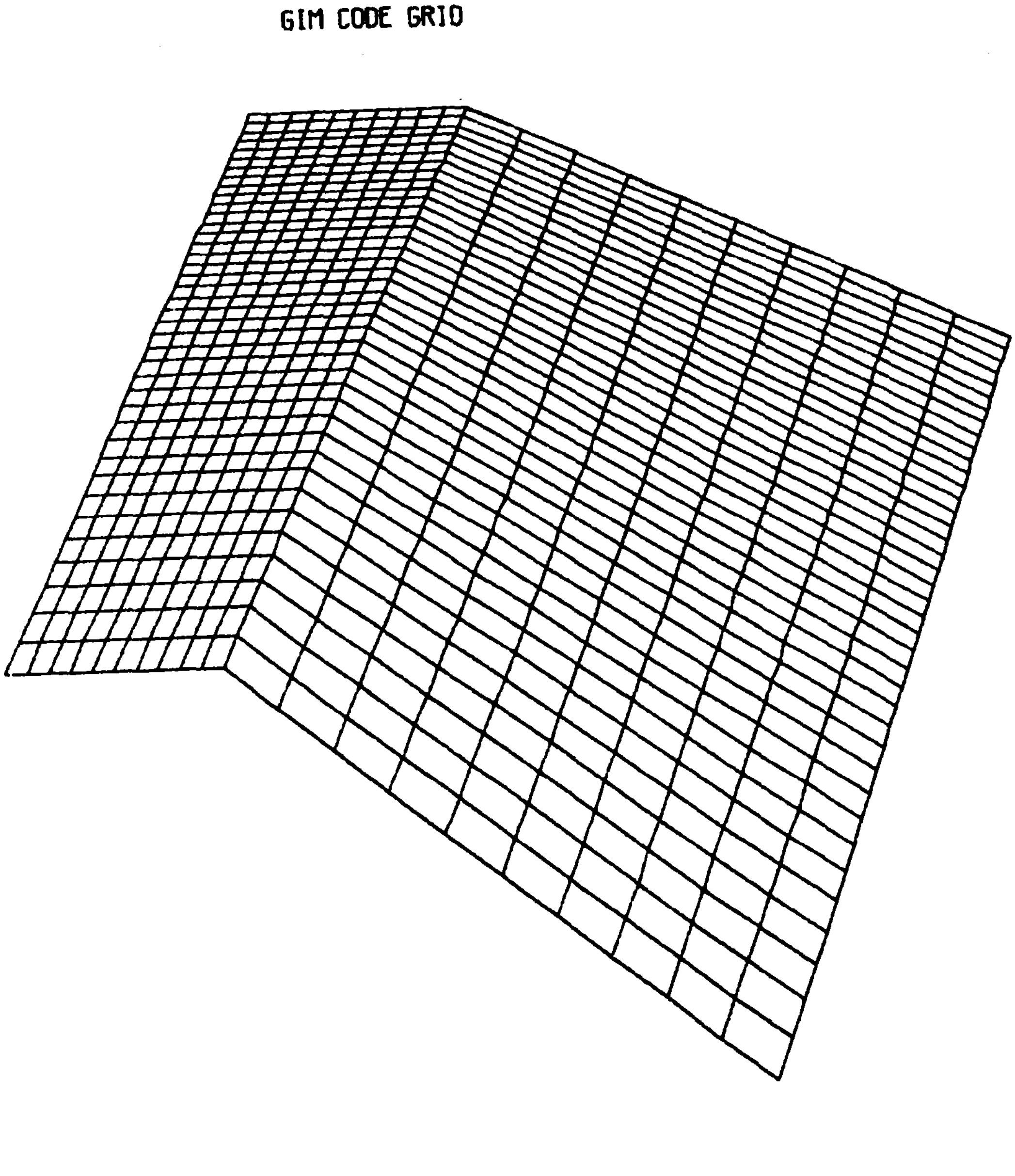


Fig. 4-2 - Computational Grid for the Leading Edge of the Turbulent Spillage Problem

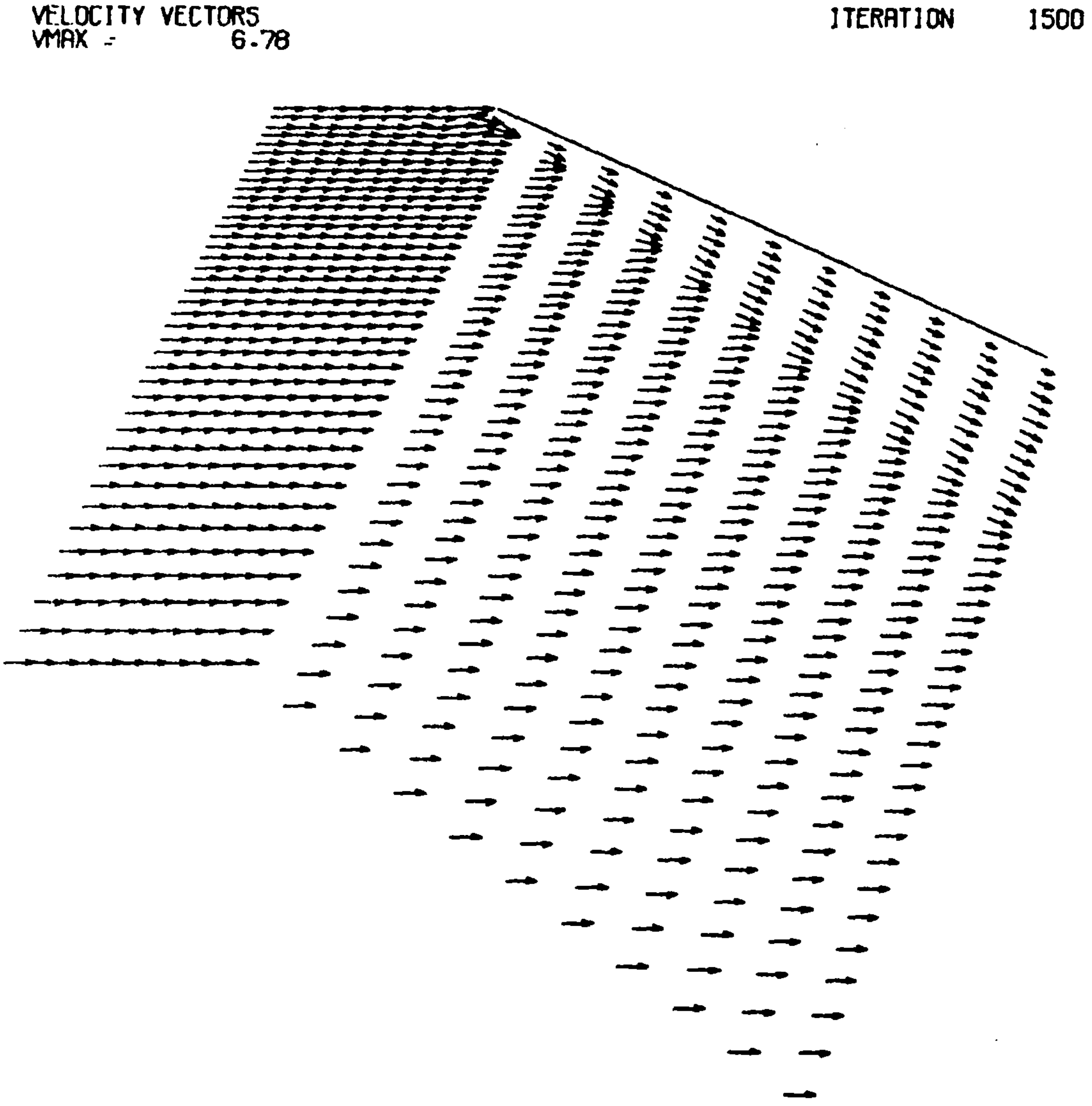


Fig. 4-3 - Spillage Problem — Leading Edge Adiabatic Walls (Turbulent)

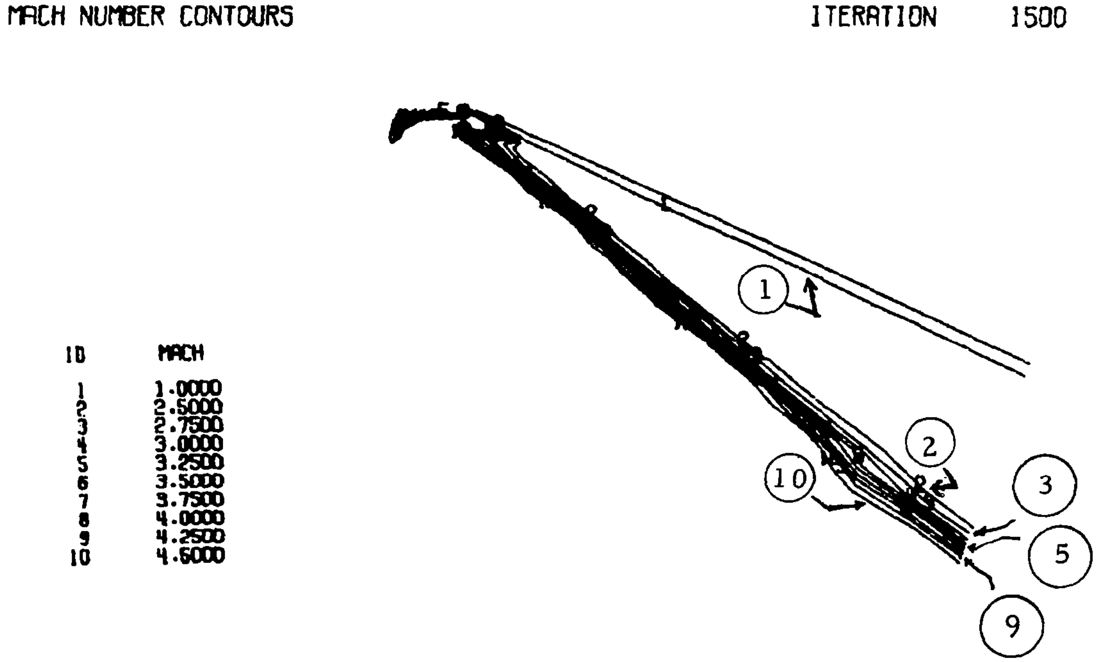
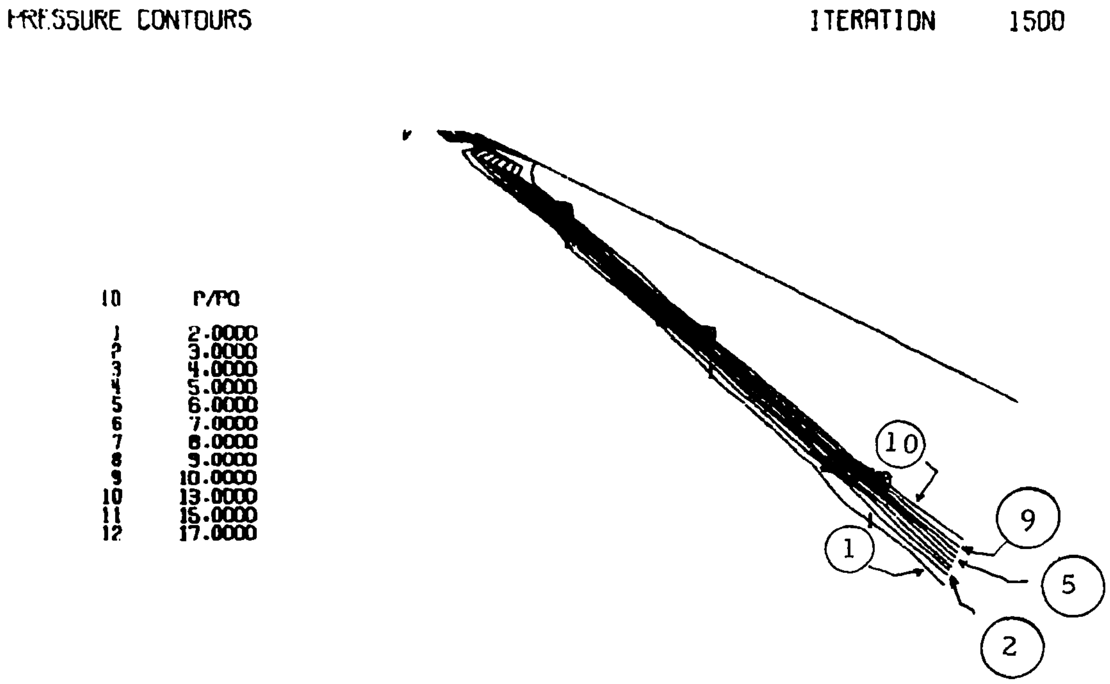


Fig. 4-4 - Spillage Problem — Leading Edge Adiabatic Walls (Turbulent)



ID	P/PO
1	2.0000
2	3.0000
3	4.0000
4	5.0000
5	6.0000
6	7.0000
7	8.0000
8	9.0000
9	10.0000
10	13.0000
11	15.0000
12	17.0000

Fig. 4-5 - Spillage Problem — Leading Edge Adiabatic Walls (Turbulent)

SECTION 4 REFERENCES

4.1 Rubesin, M. W., and W. C. Rose, "The Turbulent Mean-Flow, Reynolds-Stress, and Heat-Flux Equations in Mass-Averaged Dependent Variables," NASA TM-62248, March 1973.

4.2 Marvin, J. G., "Turbulence Modeling for Compressible Flows," NASA TM X-73, 188, January 1977.

4.3 Cebeci, T., "Calculation of Compressible Turbulent Boundary Layers with Heat and Mass Transfer," AIAA Paper 70-74, Los Angeles, June 1970.

4.4 Shang, J. S., and W. L. Hankey, "Numerical Solution for Supersonic Turbulent Flow over a Compression Ramp," AIAA J., Vol. 13, No. 10, October 1975, pp. 1368-1374.

4.5 Baldwin, B. S., and H. Lomax, "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flow," AIAA Paper 78-257, Huntsville, Ala., January 1978.

4.6 Ferri, A. et al., "Theoretical and Experimental Investigation of Supersonic Combustion," ARL-62-467, Aerospace Research Laboratories, San Bernadino, Calif., September 1962.

4.7 Harsha, P. T., and S. C. Lee, "Correlation Between Turbulent Shear Stress and Turbulent Kinetic Energy," AIAA J., Vol. 8, No. 8, August 1970.

4.8 Harsha, P. T., "A General Analysis of Free Turbulent Mixing," AEDC TR-73-177, May 1974.

4.9 Rubesin, M. W., "A One-Equation Model of Turbulence for Use with the Compressible Navier-Stokes Equation," NASA TM X-73, 128, April 1976.

4.10 Glushko, G. S., "Turbulent Boundary Layer on a Flat Plate in an Incompressible Fluid," Bull. Acad. Sci. USSR Mech., 1965, pp. 13-23.

4.11 Launder, B. E. et al., "Prediction of Free Shear Flows — A Comparison of the Performance of Six Turbulence Models. Free Turbulent Shear Flows. Volume 1- Conference Proceedings, NASA SP-321, pp. 361-426 (1973).

4.12 Launder, B. E., and D. B. Spalding, "The Numerical Computation of Turbulent Flows," Computer Methods in Applied Mechanics and Engineering, Vol. 3, 1974, pp. 264-289.

4.13 Pan, Y. S., "The Development of Three-Dimensional Partially Elliptic Flow Computer Program for Combustor Research," NASA CR-3057, Langley Research Center, Hampton, Va., November 1978.

4.14 East, L. F., and W. G. Sawyer, "An Investigation of the Structure of Turbulent Boundary Layers," Paper 6, AGARD Conference Proceedings, 1979, p. 271.

4.15 Spradley, L. W., J. F. Stalnaker and A. W. Ratliff, "Hyperbolic/Parabolic Development for the GIM-STAR Code," NASA CR-3369, December 1980.

5. CHEMISTRY MODEL DEVELOPMENT

The original GIM algorithms were developed and the code written for an ideal gas only. The need for computing chemically reacting flow fields for application to engine combustion problems necessitates the development and implementation of a reacting gas capability in GIM. Most of the work during this contract has been limited to equilibrium considerations, but some attention has been given to finite rate models. The equilibrium work is discussed first (Section 5.1) followed by a general formulation of a finite rate scheme for GIM.

5.1 EQUILIBRIUM CHEMISTRY

The basic premise of chemical equilibrium treatment in flow fields is that the local reaction rates are much faster than the flow residence time. The consequence is that the reactions proceed to completion at each point within the flow field. It has been shown in Ref 5-1 that for chemical equilibrium the thermochemistry computations can be uncoupled from the flowfield solution. The thermochemistry data can then be communicated to the flowfield code via one of two methods. First, the flow state variables (P, ρ, T), thermodynamics (γ, M_w) and species distribution can be generated a priori for an isentropic expansion process from a given stagnation condition and the results stored in tabular form for use via a table look-up procedure. Second, the chemical equilibrium properties can be computed as an integral part of the flowfield solution by using the equilibrium calculation code as a subprogram. In either case the results are the same. Depending upon the size of the system, i.e., number of species and reactions involved, it has been found that, in general, the uncoupled approach saves computer time. A third approach is to simplify the reaction model to a complete reaction application. The approach is now discussed.

5.1.1 Complete Reaction Model

Adding additional species equations to the GIM code and tracking each of the species present increases both the storage and computer time used in performing calculations for a reacting flow situation. As a first step, an effective global species conservation equation is solved for the mass fraction of the hydrogen fuel. This fuel mass fraction is the sum of the hydrogen component over all species; it is assumed that the remainder of the element mass fractions are those of nitrogen and oxygen (again summed over all species) in the proportions found in air. This approach allows the use of the two-gas species equation already in the GIM code.

In the one and two ideal gas versions of the GIM code presently used, the thermodynamic and transport properties of the fluid have been computed locally whenever needed. This approach is efficient in the use of computer memory; however, when considering more complex chemistry models, this procedure can become too costly in the use of computer time. For this reason, the logic of the GIM code has been changed to perform the thermodynamic and transport property calculations only once per time step in a new fluid properties subroutine. For the sake of consistency and ease of use, the fluid property calculations for the one and two ideal gas cases have also been included in this subroutine, with the choice of paths through the subroutine being determined by an input indicator.

The complete reaction model used to describe hydrogen-air chemistry assumes that all of the hydrogen fuel or oxygen present goes completely into water, depending upon whether there is an excess of oxygen or fuel, respectively. Following Ref. 5-2, reaction is not permitted when the volume fraction of hydrogen in the mixture is less than a threshold level of 0.04. Only the four species H_2, O_2, N_2 and H_2O are considered in the chemistry model described herein, and only when the amount of hydrogen present is less than the threshold level will both H_2 and O_2 be present. The primary input to this phase of the complete reaction model is the mass fraction of hydrogen fuel (summed over all hydrogen-containing species) which is the dependent variable associated with the species equation.

After the species fractions have been determined, the temperature is found from the internal energy (obtained from the energy equation) using the enthalpy-temperature curve fits for the individual species from Ref. 5-3. Because these curve fits are higher order polynomials in temperature, an iterative procedure is required to do this. This iteration can be avoided as in Ref. 5-2 through the use of a quadratic equation to represent the enthalpy-temperature variation. This permits direct solution for temperature using the standard inversion procedure for quadratic equations.

Knowing the composition, temperature, and density (from the continuity equation), the mixture molecular weight can be obtained, and the pressure from the equation of state for a mixture of ideal gases.

The transport properties are obtained by considering the fluid as an effective binary mixture for the purposes of computing a viscosity and thermal conductivity. Water, oxygen and nitrogen (H_2O, O_2 and N_2) are lumped as one effective species and hydrogen (H_2) is the second species. Actual values of the transport properties are based on data from Refs. 5-4 and 5-5. Wilke's rule is used to obtain the mixture viscosity and Mason and Saxena's relation for the mixture conductivity. These relations were taken from Ref. 5-6. The transport coefficients needed for the species equation are obtained from constant Prandtl number and Lewis number.

Since the speed of sound value used at various places in the GIM code need not be rigorously precise, an approximate value is computed from the perfect gas expression using a value for the ratio of specific heats of 1.4 and a gas constant value obtained using the universal gas constant and the local mixture molecular weight. Frozen specific heat values are obtained using the curve fits in Ref. 5-2.

The previous discussion has described how the complete reaction model composition and the associated thermodynamic and transport property data are computed. As noted earlier, the one and two ideal gas case fluid property calculations have also been consolidated into the new fluid properties subroutine. These ideal gas properties are computed as was done previously in the GIM code and are not described here.

Initial checkout of the logic changes involved in implementing the fluid properties subroutine has been performed using a previously run one-ideal gas case, essentially reproducing the results obtained earlier. The hydrogen-air capability is being tested by running a flat-plate slot-injection test case (Ref. 5-7 as given in Ref. 5-2). Initially, results will be for non-reacting flow (obtained by setting the hydrogen fraction threshold level required for reaction to a value greater than one), and then for a reacting flow following the complete reaction model.

5.1.2 Arbitrary Gas Full Equilibrium Model

The mathematical formulation of a full chemical equilibrium calculation is well documented (Ref. 5-3) and a computer code is available to provide thermochemistry information. The computer code is fast computationally, reliable and accurate. Its output has been made compatible with our other flow field codes at Lockheed and are communicated to the codes via tabulated data.

In the flow solution, the governing equations are cast in terms of independent variables such as velocity, entropy and total enthalpy. For isoenergetic isentropic flow there is no change in either the entropy or total enthalpy level during a flow field expansion process. Computationally, a combustion calculation (specified reactant formulation and pressure-temperature conditions) locates the appropriate isentrope where the flow field expansion is initiated (Fig. 5-1). Succeeding points on the isentrope are obtained by specifying one state variable (normally pressure) and the remaining properties computed from chemical equilibrium considerations and isentropic expansion calculations from reference conditions.

Shock waves calculated in the discrete fashion are easily treated since the shock wave represents a discontinuity in the flow across which there is an increase in the entropy level as a result of loss in local total pressure.

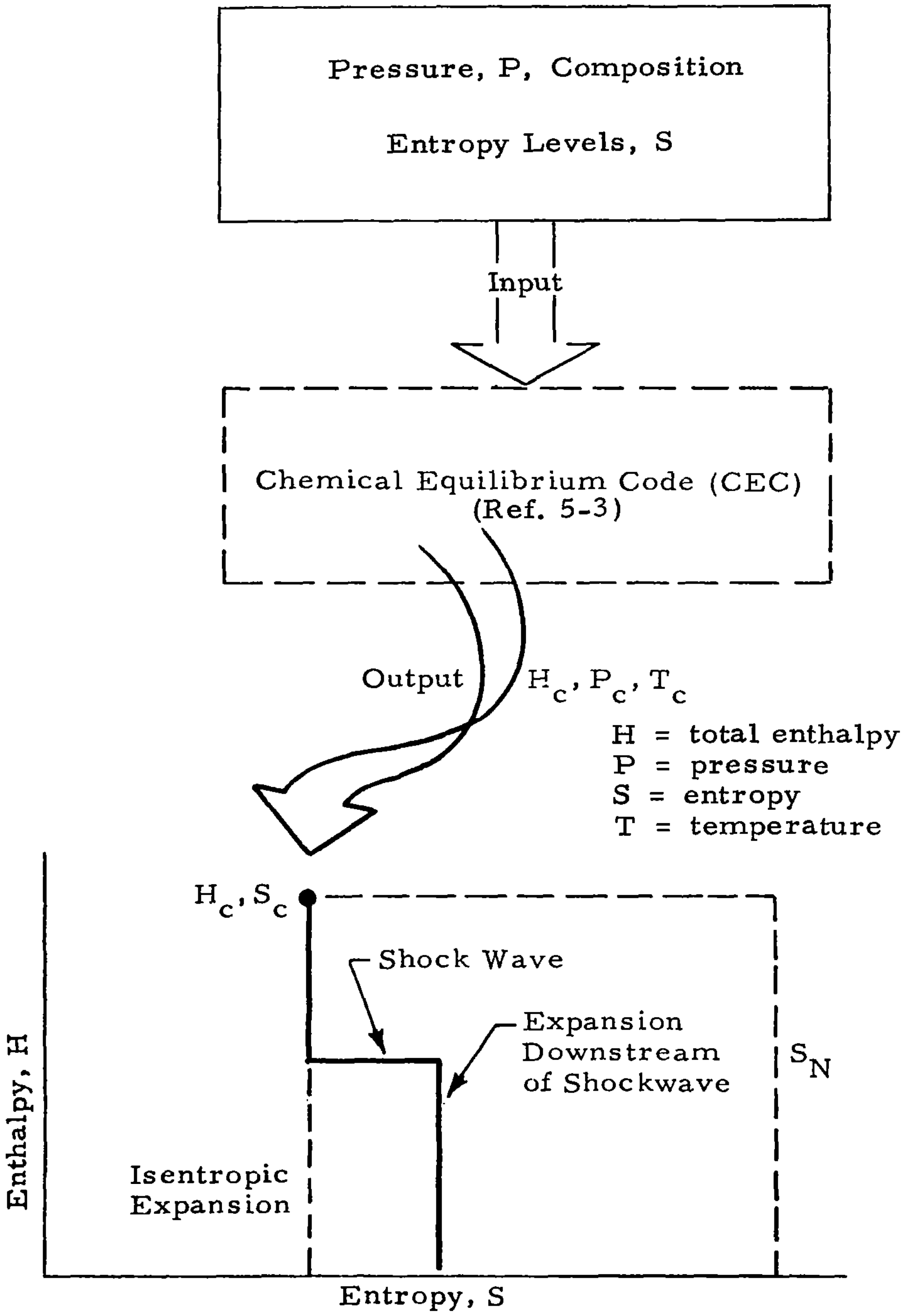


Fig. 5-1 - Schematic of Equilibrium Thermochemistry Table Construction for Isoenergetic Flow

This means simply that the expansion along an isentrope changes to a new isentrope (Fig. 5-1) when the streamline of interest passes through a shock wave. To locate this isentrope, the change in entropy across the shock is computed from the shock relations.

Chemical equilibrium is treated in the flowfield codes by generating tables of isentropic expansions a priori from specified reference conditions, where non-isentropic effects are obtained by expanding the isentropic relations. Most practical flow fields which are reacting have a non-constant energy field. This is treated in our equilibrium code as another parameter in a table look-up. An a priori knowledge of the range of energy distributions can be used to set up another entry in the table. This is commonly referred to as Oxidizer/Fuel (O/F) ratio (see Fig. 5-2). Utilization of the equilibrium routine then proceeds as described above.

Shock capturing finite difference schemes are formulated by casting the set of governing partial equations in "conservative form" and then constructing the difference equation such that the conservative nature of the governing equations is maintained. However, it is necessary to add artificial viscosity to the solution in the vicinity of shockwave for numerical stability. If too much artificial viscosity is induced, the damping is too great and the jump conditions across the shock are computed incorrectly. Consequently, obtaining the correct thermodynamic properties through the shock depends on correctly constructing the shock (i.e., computing the correct "jump" conditions across the shock). In any event, the chemical equilibrium properties are obtained as discussed above.

5.2 FINITE RATE MODEL DEVELOPMENT

In flow problems where the gas may be considered in equilibrium (chemical and thermodynamic) at every point, two parameters are sufficient to define any of the other thermodynamic variables, either by assuming a perfect gas or by utilizing the results of chemical equilibrium calculations of the gases involved. If the gases cannot be considered to be in equilibrium,

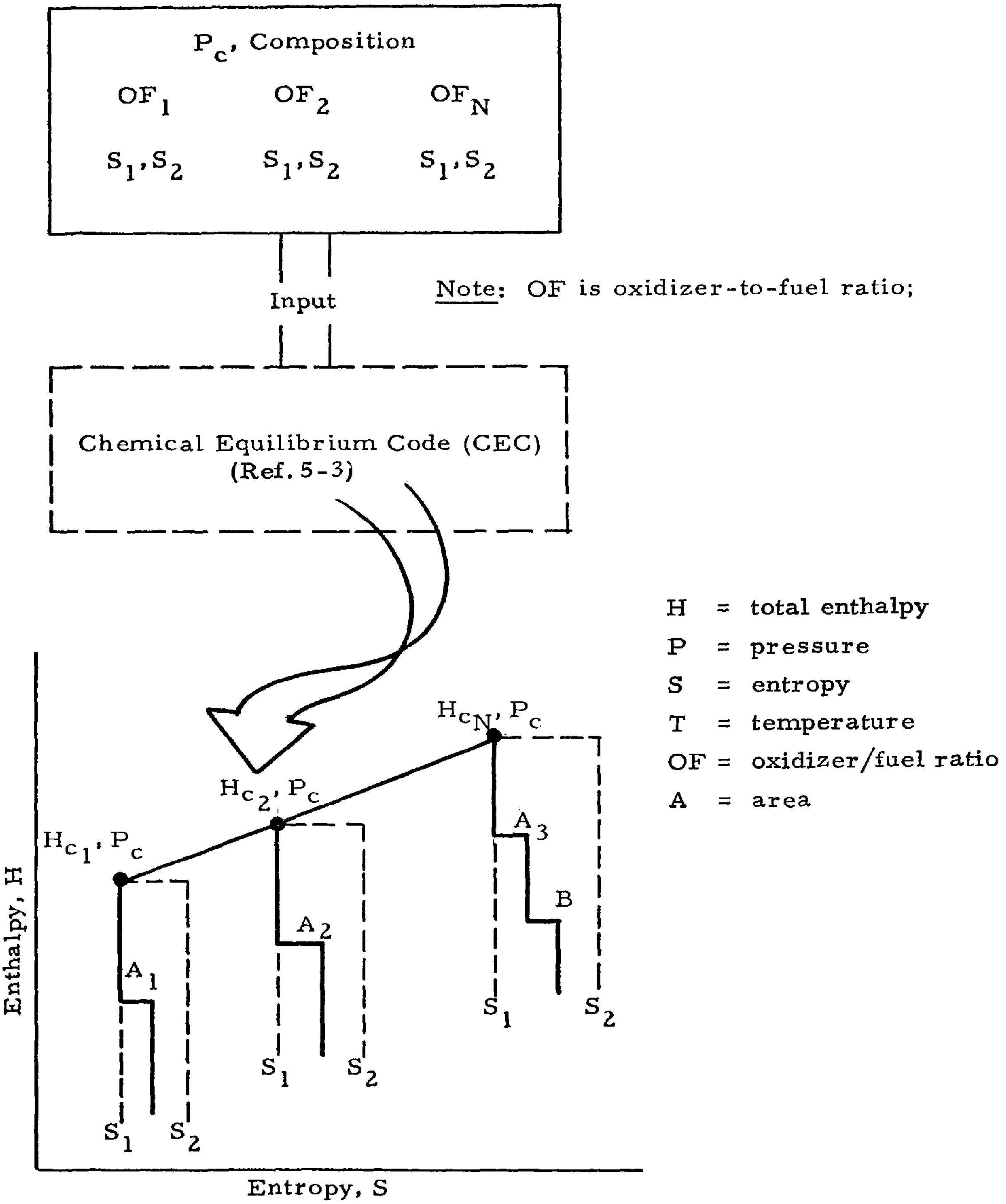


Fig. 5-2 - Schematic of Equilibrium Thermochemistry Table Construction for Non-Isentropic and Non-Isoenergetic Flow

a point-by-point evaluation of its composition via integrating the individual species continuity equations is required. This subsection addresses a generalized procedure used to perform such calculations for an inviscid analysis This general procedure with addition of viscous terms will be used in the GIM code.

A detailed description of the rate processes that occur in reacting flows requires that a myriad of mechanisms be considered to include all the possible chemical reactions of dissociation, formation, recombination. All of these, however, can be treated with a very general formalism. In the form usually quoted in chemical kinetics (Ref. 5-8) the phenomenological law of mass action states that the rate of a reaction is proportional to the product of the concentrations of the reactants. Thus, for a general reaction of the form

$$\sum_{i=1}^{N} \nu_i' A_i \rightleftharpoons \sum_{i=1}^{N} \nu_i'' A_i \qquad (5.1)$$

the net rate of production $\dot{w}_i$ for any participating species for which the stoichiometric coefficients ν_i' and ν_i'' are not equal can then be written as

$$\dot{w} = k_f \prod_{i=1}^{N} (C_i)^{\nu_i'} - k_b \prod_{i=1}^{N} (C_i)^{\nu_i''} \qquad (5.2)$$

where C_i is the species concentration defined as $C_i = \rho F_i$ with ρ being the density and F_i being the species mol/mass ratio, respectively. Assuming small deviations from equilibrium, the forward and backward reaction rate constants, k_f and k_b, respectively, can be related to the concentration equilibrium constant and to the pressure equilibrium constant as follows:

$$\frac{k_f}{k_b} = K_c = K_p \, (\Re T)^{\sum_{i=1}^{N} (\nu_i' - \nu_i'')} \qquad (5.3)$$

The significance of the pressure equilibrium constant K_p is that it can be easily evaluated for any reaction using tabulated values of K_f the equilibrium constant for formation from the elements. Values of K_f are commonly tabulated in conjunction with specific heats, entropies and enthalpies as a function of temperature, and are available in general for most species. An equally convenient method exists for determining K_p from the change of free energy accompanying the reaction, i.e.,

$$K_p = \exp(-\Delta G/\Re T) \qquad (5.4)$$

where ΔG is the change in free energy during the reaction process. Free energy values are also available for most species in tabular form. This is the method most commonly used to compute K_p.

Using Eq. (5.3) to eliminate the backward rate constant from Eq. (5.2) the general production rate equation can be written as

$$\dot{w} = k_f \left\{ \prod_{i=1}^{N} (C_i)^{\nu_i'} - \frac{(\Re T)^{\sum_{i=1}^{N} (\nu_i'' - \nu_i')}}{K_p} \prod_{i=1}^{N} (C_i)^{\nu_i''} \right\} \qquad (5.5)$$

Finally, the net rate of production for any species participating in the reaction, either as a reactant or as a reaction product, is then given by

$$\dot{w}_i = (\nu_i'' - \nu_i')\,\dot{w} \qquad (5.6)$$

Since most reaction systems involve a large number of simultaneous reactions, the net rate of production of species i usually equals a sum of terms. Thus, for an arbitrary number of M simultaneous reactions, all involving species i,

$$\dot{w}_i = \sum_{k=1}^{M} \dot{w}_{i,k} \qquad k = 1, \ldots, M \qquad (5.7)$$

where $\dot{w}_{i,k}$ denotes the net rate of production of species i due to reaction K only.

For reasons of computational speed and efficiency, a computer program can contain explicit expressions, as obtained from Eq. (5.5), for the most commonly encountered reaction mechanisms. Twelve types of reaction mechanisms are generally considered as possible contributors to the calculation of the net rate of production, $\dot{w}_i$:

Reaction type

$$\begin{array}{llcl}
(1, 7) & A + B & \rightleftharpoons & C + D \\
(2, 8) & A + B + M & \rightleftharpoons & C + M \\
(3, 9) & A + B & \rightleftharpoons & C + D + E \\
(4, 10) & A + B & \rightleftharpoons & C \\
(5, 11) & A + M & \rightleftharpoons & C + D + M \\
(6, 12) & A + M & \rightleftharpoons & C + M
\end{array} \qquad (5.8)$$

Reaction types (7) through (12) correspond to reaction types (1) through (6), but proceed in the forward direction only.

To reduce roundoff and truncation errors, the forward and backward rates for each reaction are computed separately. All contributions to the molar rate of production of a given species are then computed and added algebraically to form matrix coefficients. Since reaction types (7) through (12) proceed in the forward direction only, the second term on the right-hand side of Eq. (5.5) is disregarded in calculating the contributions to the coefficient matrix.

In reactions (2), (5) and (6), as well as in (8), (11) and (12), M denotes a third body species which can be specified. For these reactions the situation often occurs where for various third bodies the respective rate constants differ only by a constant multiplier. These multipliers can be considered as third body efficiencies or weighting factors. If such a case is encountered, the third body species mole mass ratio F_M becomes effectively a fictitious mole mass ratio, consisting of the weighted sum over all those species having a nonzero weighting factor, i.e.,

$$F_M = \sum_i f_i F_{M_i} \tag{5.9}$$

where f_i are the weighting factors.

The forward rate constant, k_f, is generally a function of temperature. It is most often expressed in Arrhenius form. Again, for speed and efficiency in computation, the rate constants are divided into five types:

Rate Constant Type

$$\begin{array}{ll}
(1) & k_f = A \\
(2) & k_f = AT^{-N} \\
(3) & k_f = A \exp(B/\Re T) \\
(4) & k_f = AT^{-N} \exp(B/\Re T) \\
(5) & k_f = AT^{-N} \exp(B/\Re T^M)
\end{array} \tag{5.10}$$

The equations presented in this section provide a very general formalism for the evaluation of various rate processes. The specification of particular systems and associated rate constants is left to the program user to provide input data.

Considering now the general species continuity equation

$$\rho\vec{q} \cdot \nabla C_i = \dot{w} \tag{5.11}$$

and making use of the foregoing discussion of the rate process we now proceed to describe a calculational technique for determining the individual species composition on a point-by-point basis. The description of this process is substantially simplified if Eq. (5.5) is specialized to a particular reaction type, say number (7) from Eq.(5.8) which is a one-way, two-body reaction.

$$A + B \rightarrow C + D \tag{5.12}$$

the net production rate for this process is

$$\dot{w} = - k_f \rho^2 F_A F_B \tag{5.13}$$

and the species continuity equation for species B then becomes

$$\rho\vec{q} \cdot \nabla F_B = - k_f \rho^2 F_A F_B \tag{5.14}$$

which along streamlines becomes

$$\rho q \frac{\partial F_B}{\partial S} = - k_f \rho^2 F_A F_B \tag{5.15}$$

This equation can readily be solved using finite difference techniques employing explicit relationships, such as Euler or more sophisticated schemes, such as Predictor-Corrector. The step size for integrating this equation however is severely limited by stability criteria. It can be seen from Eq. (5.15) that the rate of change of a species along the streamline becomes increasingly larger as the flow speed is slowed, the density increased, or for fast reaction rates. In many flow problems, combinations of slow speeds, high densities and fast reaction rates (i.e., quasi-equilibrium) are quite common and integration step sizes so small (i.e., $< 10^{-8}$ meters) are encountered that the solution becomes impractical in terms of computation time.

For this reason, the technique described in Ref. 5-9, based on a linearization of the production rates, is a good choice. Writing Eq. (5.15) in finite difference form over a streamline step from station n to n+1.

$$F_{B_{n+1}} = F_{B_n} - \frac{k_f \Delta S \rho}{q} F_{A_{n+1}} F_{B_{n+1}} \tag{5.16}$$

And evaluating all the species concentrations at the downstream point results in a set of simultaneous nonlinear algebraic equations. In order to solve these equations, we must then linearize the term $F_{A_{n+1}} F_{B_{n+1}}$ which is accomplished following Ref. 5-9. By expanding in terms of values at station n along with the increments over n to n+1 and neglecting products which are of second order, we can get the following expression using algebra.

$$F_{A_{n+1}} F_{B_{n+1}} = F_{A_n} F_{B_{n+1}} + F_{B_n} F_{A_{n+1}} - F_{A_n} F_{B_n} \tag{5.17}$$

Equation (5.16) can now be written in its linearized form. Let $C = \Delta S\, k_f\, \rho/q$

$$F_{B_{n+1}} = F_{B_n} - C \left[F_{A_n} F_{B_{n+1}} + F_{B_n} F_{A_{n+1}} - F_{A_n} F_{B_n}\right]$$

and

$$F_{A_{n+1}} = F_{A_n} - C \left[F_{A_{n+1}} F_{B_n} + F_{B_{n+1}} F_{A_n} - F_{A_n} F_{B_n}\right] \tag{5.18}$$

Equation (5.18) can then be expressed in terms of a set of unknowns and calculable coefficients, C. Rewriting these we obtain

$$F_{B_{n+1}} = Q_{B_n} - C\, F_{A_n} (F_{B_{n+1}}) - C\, F_{B_n} (F_{A_{n+1}})$$

$$F_{A_{n+1}} = Q_{A_n} - C\, F_{B_n} (F_{A_{n+1}}) - C\, F_{A_n} (F_{B_{n+1}}) \tag{5.19}$$

where

$$Q_{i_n} = F_{i_n} + C\, F_{i_n} F_{j_n} \tag{5.20}$$

$$F_{A_{n+1}} (1 + C\, F_{B_n}) + (C\, F_{A_n})\, F_{B_{n+1}} = Q_{A_n}$$

$$F_{A_{n+1}} (C\, F_{B_n}) + (1 + C\, F_{A_n})\, F_{B_{n+1}} = Q_{B_n} \tag{5.21}$$

A matrix can now be formed using totally known information.

$$\begin{bmatrix} 1 + C\,F_{B_n} & C\,F_{A_n} \\ C\,F_{B_n} & 1 + C\,F_{An} \end{bmatrix} \begin{bmatrix} F_{A_{n+1}} \\ F_{B_{n+1}} \end{bmatrix} = \begin{bmatrix} Q_{A_n} \\ Q_{B_n} \end{bmatrix}$$

The algebraic equations $[A]\,[X] = [B]$ can then be solved for the unknown compositions $F_{A_{n+1}}$, $F_{B_{n+1}}$ via an elimination technique. Although consuming more time per integration step than an explicit formulation, the implicit technique employed here is unconditionally stable permitting much larger step sizes.

5.3 RECOMMENDATIONS

Based on the development during the past year's work, the following recommendations are offered.

- The "complete reaction" equilibrium model should be checked out for hydrogen-air mixing in a duct.
- The full, arbitrary gas, equilibrium model discussed in Section 5.1 should be completed and a solution compared to the complete reaction case.
- Parallel to item 2 above, the full formal finite rate model should be coded in GIM.
- A global finite rate system should be synthesized for hydrogen air to provide input for checkout of the finite rate coding. Comparison with the equilibrium cases should then be made.

SECTION 5 REFERENCES

5-1. Prozan, R. J., "Solution of Non-Isoenergetic, Supersonic Flows by the Method of Characteristics," LMSC-HREC D162220-III-A, Contract NAS7-761, Lockheed Missiles & Space Company, Huntsville, Alabama, July 1971. (Available as NASA CR-1322, 1974.)

5-2. Drummond, J. P., and E. H. Weidner, "Numerical Study of a Scramjet Engine Flow Field," AIAA Paper 81-0186, January 1981.

5-3. Gordon, S., and B. J. McBride, "Computer Program for Calculation of Complex Chemical Equilibrium Compositions, Rocket Performance, Incident and Reflected Shocks, and Chapman-Jouguet Detonations," NASA SP-273, 1971.

5-4. Kumar, A., R. A. Graves, Jr., and K. J. Weilmuenster, "User's Guide for Vectorized Code EQUIL for Calculating Equilibrium Chemistry on Control Data STAR-100 Computer," NASA TM 80193, April 1980.

5-5. Eckert, E. R. G., and R. M. Drake, Jr., Heat and Mass Transfer, Second ed., McGraw-Hill Book Co., Inc., 1959.

5-6. Dorrance, W. H., Viscous Hypersonic Flow, McGraw-Hill Book Co., Inc., c.1962.

5-7. Thayer, W. J., "The Two-Dimensional Separated Flow Region Upstream of Inert and Chemically Reactive Transverse Jets," Ph.D. Thesis, Department of Mechanical Engineering, University of Washington, 1971.

5-8. Williams, F. A., Combustion Theory, Addison-Wesley Publishing Co., Inc., c.1965.

5-9. Moretti, G., "A New Technique for the Numerical Analysis of Non-Equilibrium Flows," AIAA J., Vol. 3, No. 2, February 1965.

6. INTERACTIVE GIM CODE INPUT PROGRAM – RUNGIM

6.1 INTRODUCTION

RUNGIM is an interactive program package designed to automate the preparation of GIM code program input and to effectively eliminate a percentage of the mistakes made by the novice as well as the experienced user. The prevention of gross errors will result in great savings of both engineering hours and computer time. In addition, the use of RUNGIM will reduce the required training period for a new user, make the GIM code easier to use, and significantly decrease the overall time required for setting up a given problem.

RUNGIM combines the interactive features of FORTRAN Extended (FTN) Version 4.7 and Nos. 1.3 CYBER Controls Language (CCL) to efficiently automate the GIM code input procedures. One of the most important of these features is the acceptance of list-directed input from a remote terminal. This type of input accepts integer, floating point, exponential, or alphanumeric data if separated by a comma or space. Thus the physical portion of the input phase becomes a function of user convenience and is quick, easy, and relatively free of possible error. To facilitate this type of input, RUNGIM is self-directing and prompts the user for input data. The "prompt" supplies the user with minimal necessary information about the input and then invites a response. RUNGIM contains appropriate default values for those inputs which have a frequently assumed value and checks user-supplied numerical inputs for proper format (i.e., integer or real) and range. If the user enters an improper input value or attempts a default where none is allowed, RUNGIM responds with appropriate messages and continues to prompt the user until an appropriate response is entered. Since many GIM code runs involve only minor changes in previously submitted input data, RUNGIM allows the user the access and XEDIT previously created data files. Finally, RUNGIM provides the user with the ability to interrupt execution of the interactive program at any point by entering a special "abort code." The user is then able to resume execution at the point of interruption, restart at the beginning, or terminate execution completely.

There are basically three types of data that must be generated for use in a GIM code run. These three types are job control cards (runstreams), UPDATE directives, and the actual program input data. Control card types are generally independent of the problem under consideration; therefore, RUNGIM can supply all necessary control card information except for account and charge numbers, core storage and run time requirements, and file name designations. By limiting user responsibility for knowledge of the job control language to such easily understood information most of the control card errors, especially those committed by novice users, are eliminated.

Since modification of the GIM code is often necessary to accommodate special applications, RUNGIM allows the user to supply directives to the UPDATE processor from the remote terminal or from previously built files. This requires some user knowledge of both the UPDATE processor and the GIM code but provides a convenient means of assembling the UPDATE directives.

The primary input to the GIM code is, of course, the program input data and this is quite naturally the greatest source of potential error for the user. RUNGIM prevents many of these errors by ensuring that the proper input loops are chosen and explained at each logical step according to the options requested. This eliminates the omission of data or improperly placed data. The use of the free formatted input scheme, which allows RUNGIM to properly format the information on a data file, eliminates the danger of badly formatted data stopping the job or producing erroneous results.

6.2 RUNGIM DETAILS AND FEATURES

RUNGIM actually consists of seven separate programs which are executed interactively under the control of the main program, as illustrated in Fig. 6-1. These seven individual programs, GEOM, MATRIX, INTEG, GIMPLT, UPDATE, DYNDIM, and RNSTRM, perform the following tasks:

GEOM - supplies input data for the GIM code GEOM modules;

MATRIX - supplies input data for the GIM code MATRIX modules (not yet operational);

INTEG - supplies input data for the GIM code INTEG modules

GIMPLT - supplies input data for the GIM code plotting program; (not yet operational)

UPDATE - supplies modification directives to the CDC UPDATE processor;

DYNDIM - supplies input data for the GIM code dynamic dimensioning programs; and

RNSTRM - supplies the control cards required to execute any GIM code module on the NOS/CY203 computers.

The basic layout of the individual programs is the same for each one and is illustrated for GEOM in Fig. 6-2. Each program has the capability of using previously created data files or interactively creating new one(s). The interactive FORTRAN portion of each program can supply the control cards required to invoke the XEDIT'er and/or save the data files as permanent files. Note that the user can interrupt the execution of each program by entering an "abort code" in response to any prompt. The user then has the option of resuming execution at the point of interruption, restarting the program, or terminating execution of that particular program. This feature could be expanded to allow branching to various points within a program during interrupt processing.

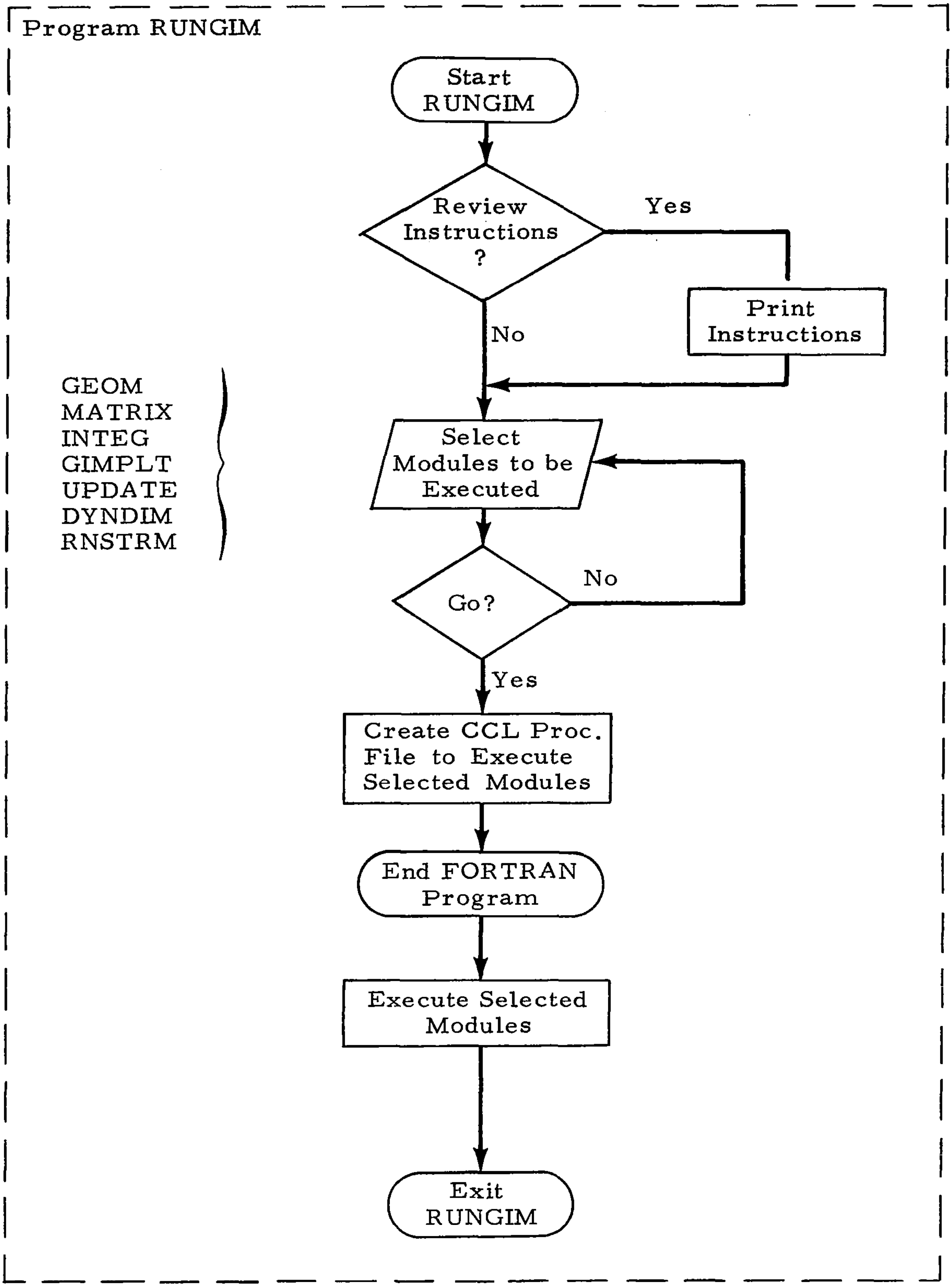
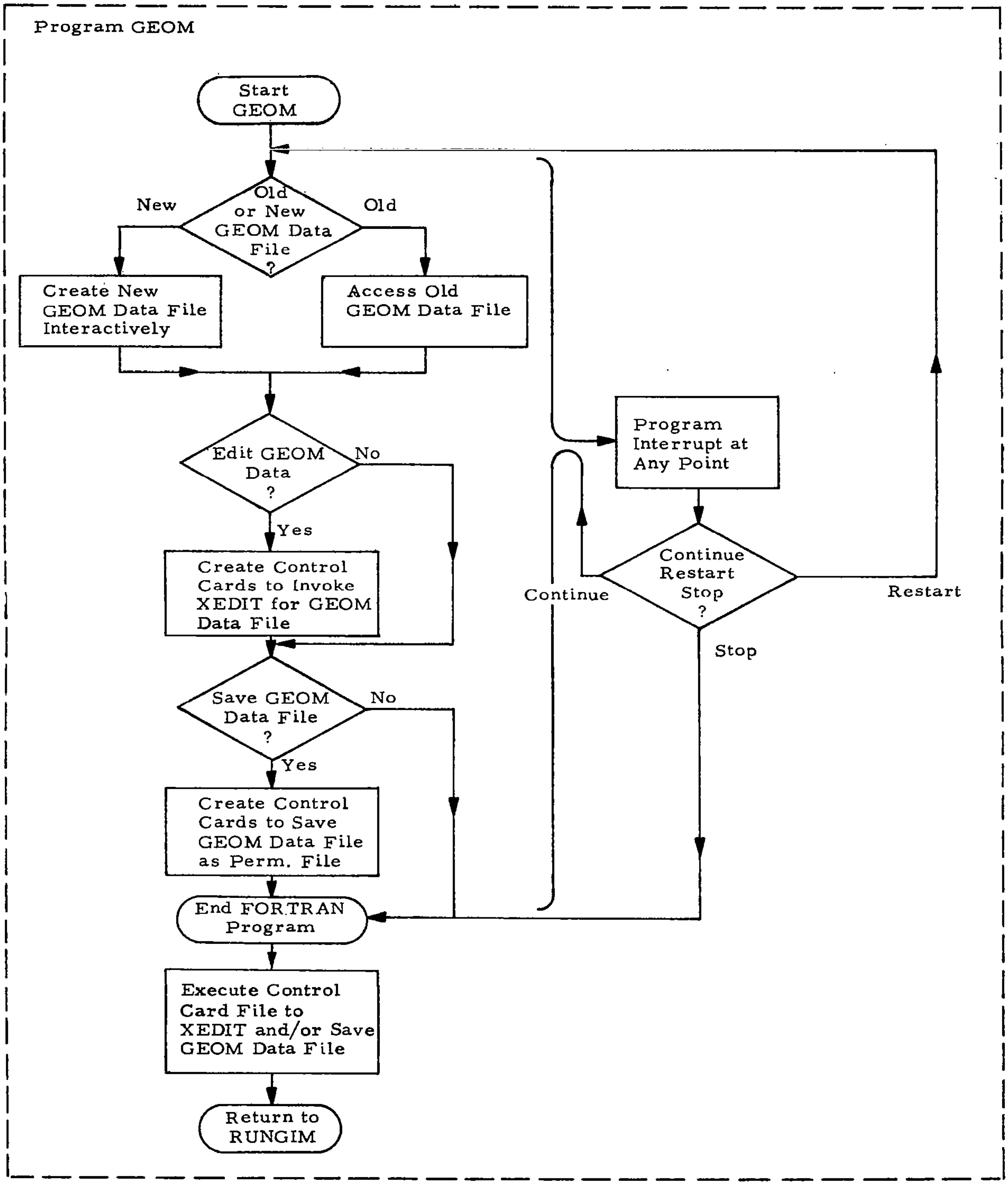


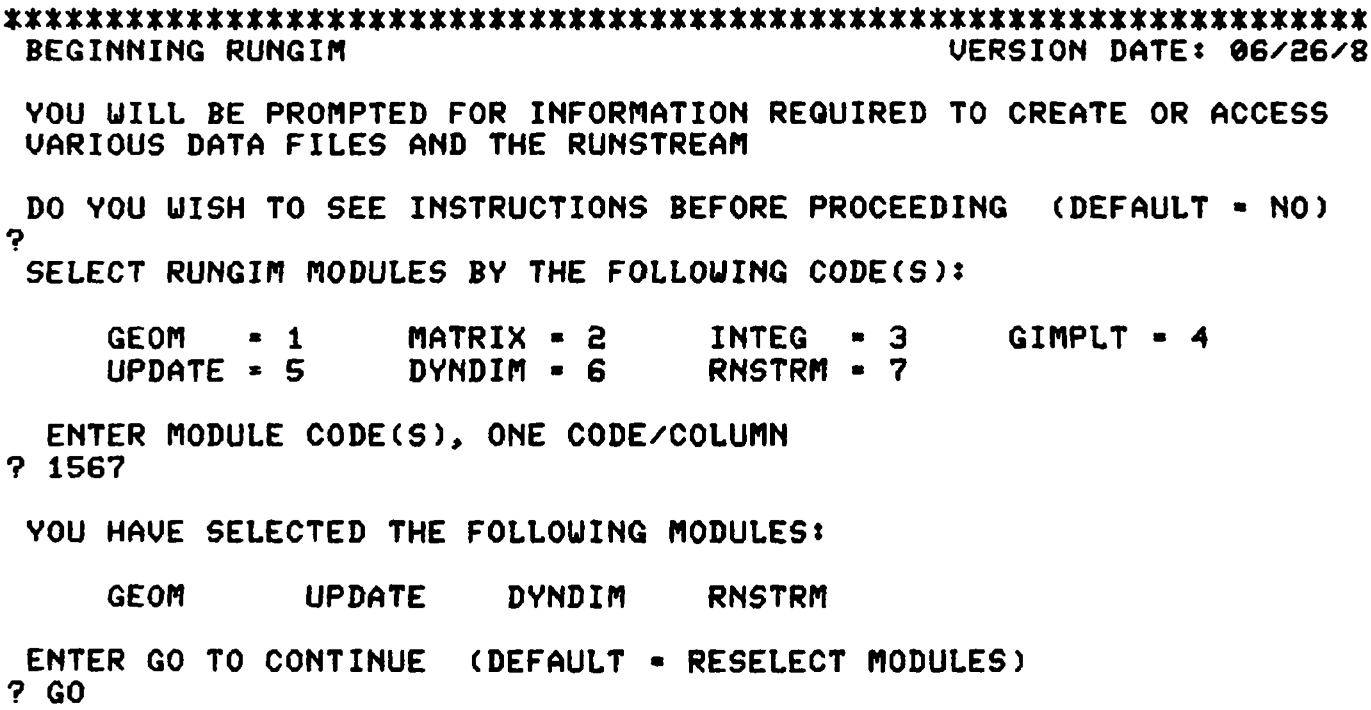
Fig. 6-1 - Schematic of RUNGIM



Program GEOM
Start GEOM
Old or New GEOM Data File ?
New
Old
Create New GEOM Data File Interactively
Access Old GEOM Data File
Edit GEOM Data ?
No
Yes
Create Control Cards to Invoke XEDIT for GEOM Data File
Save GEOM Data File ?
No
Yes
Create Control Cards to Save GEOM Data File as Perm. File
End FORTRAN Program
Execute Control Card File to XEDIT and/or Save GEOM Data File
Return to RUNGIM
Program Interrupt at Any Point
Continue Restart Stop ?
Continue
Restart
Stop

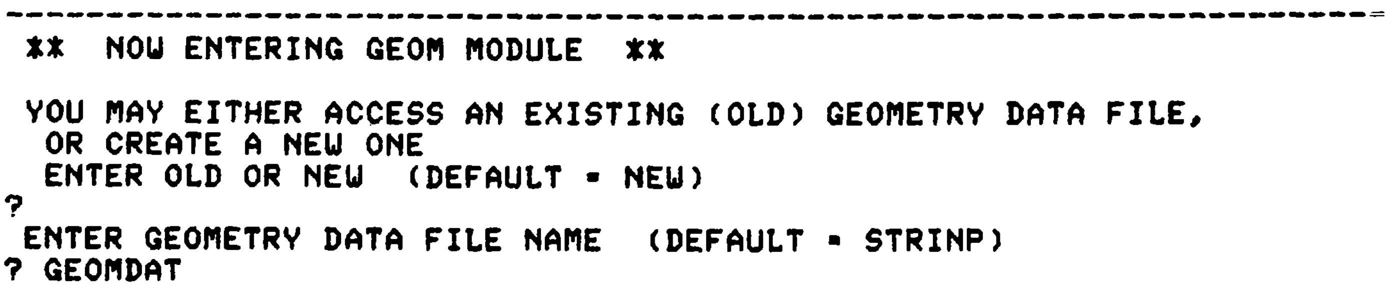
Fig. 6-2 - Schematic of GEOM Interactive Module

The various features of RUNGIM are illustrated in the following examples of actual RUNGIM input and output. The program begins by displaying a banner message and prompts for display of instructions and execution code:



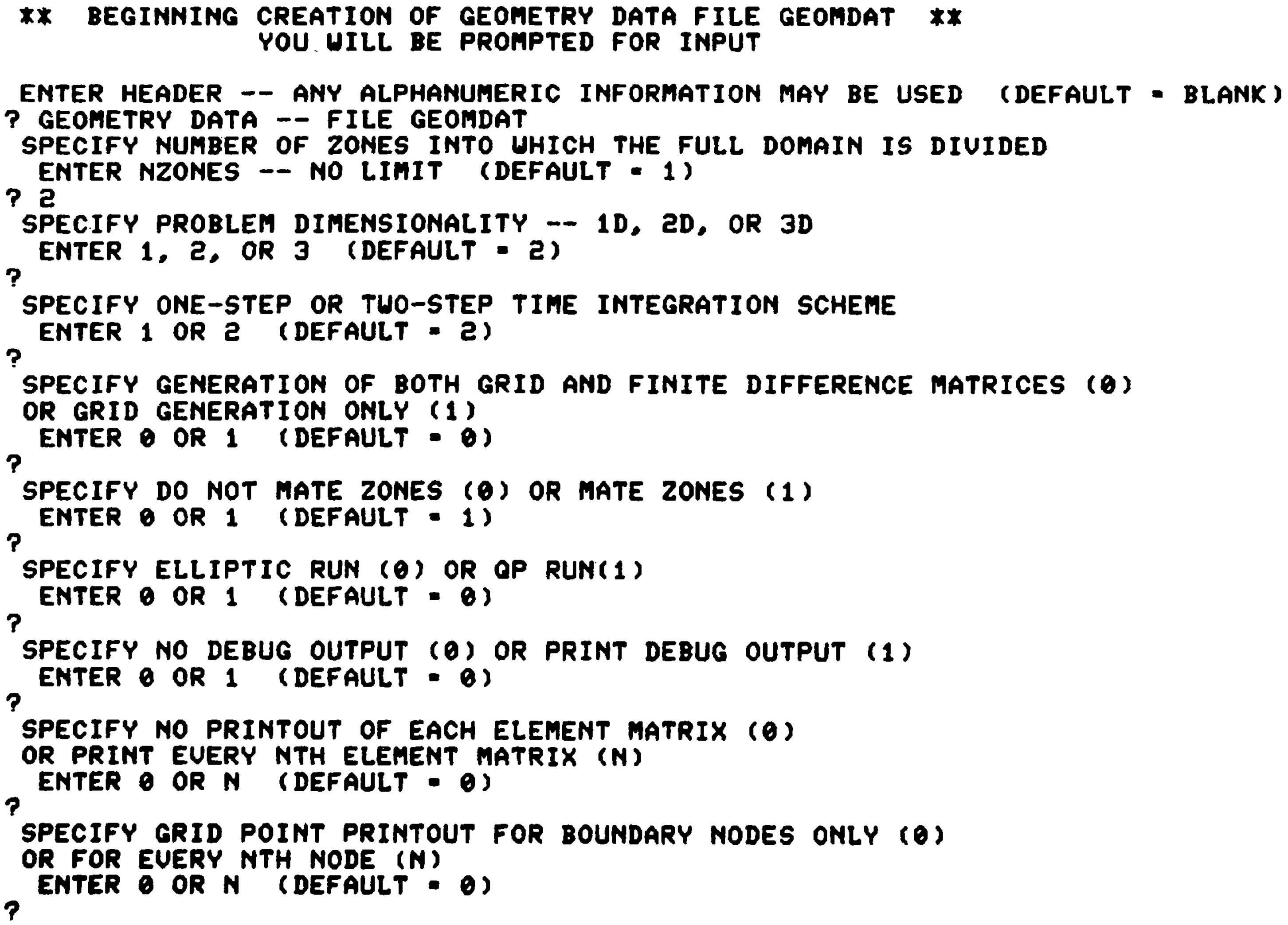
```
*********************************************************************
 BEGINNING RUNGIM                                VERSION DATE: 06/26/8

 YOU WILL BE PROMPTED FOR INFORMATION REQUIRED TO CREATE OR ACCESS
 VARIOUS DATA FILES AND THE RUNSTREAM

 DO YOU WISH TO SEE INSTRUCTIONS BEFORE PROCEEDING  (DEFAULT = NO)
?
 SELECT RUNGIM MODULES BY THE FOLLOWING CODE(S):

     GEOM   = 1     MATRIX = 2     INTEG  = 3     GIMPLT = 4
     UPDATE = 5     DYNDIM = 6     RNSTRM = 7

  ENTER MODULE CODE(S), ONE CODE/COLUMN
? 1567

 YOU HAVE SELECTED THE FOLLOWING MODULES:

     GEOM      UPDATE    DYNDIM    RNSTRM

 ENTER GO TO CONTINUE  (DEFAULT = RESELECT MODULES)
? GO
```

If the user had responded "YES "to the prompt for instructions, then RUNGIM would have displayed a description of the program and instructions for its use (see Section 3.2). In this particular case, the user has selected the GEOM, UPDATE, DYNDIM, and RNSTRM modules. RUNGIM begins execution with the GEOM module:



```
---------------------------------------------------------------------
 **  NOW ENTERING GEOM MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) GEOMETRY DATA FILE,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT = NEW)
?
 ENTER GEOMETRY DATA FILE NAME  (DEFAULT = STRINP)
? GEOMDAT
```

The user has selected creation of a new geometry input data file to be named "GEOMDAT." GEOM then begins to prompt for various input items:



```
 **  BEGINNING CREATION OF GEOMETRY DATA FILE GEOMDAT  **
               YOU WILL BE PROMPTED FOR INPUT

 ENTER HEADER -- ANY ALPHANUMERIC INFORMATION MAY BE USED  (DEFAULT = BLANK)
? GEOMETRY DATA -- FILE GEOMDAT
 SPECIFY NUMBER OF ZONES INTO WHICH THE FULL DOMAIN IS DIVIDED
  ENTER NZONES -- NO LIMIT  (DEFAULT = 1)
? 2
 SPECIFY PROBLEM DIMENSIONALITY -- 1D, 2D, OR 3D
  ENTER 1, 2, OR 3  (DEFAULT = 2)
?
 SPECIFY ONE-STEP OR TWO-STEP TIME INTEGRATION SCHEME
  ENTER 1 OR 2  (DEFAULT = 2)
?
 SPECIFY GENERATION OF BOTH GRID AND FINITE DIFFERENCE MATRICES (0)
 OR GRID GENERATION ONLY (1)
  ENTER 0 OR 1  (DEFAULT = 0)
?
 SPECIFY DO NOT MATE ZONES (0) OR MATE ZONES (1)
  ENTER 0 OR 1  (DEFAULT = 1)
?
 SPECIFY ELLIPTIC RUN (0) OR QP RUN(1)
  ENTER 0 OR 1  (DEFAULT = 0)
?
 SPECIFY NO DEBUG OUTPUT (0) OR PRINT DEBUG OUTPUT (1)
  ENTER 0 OR 1  (DEFAULT = 0)
?
 SPECIFY NO PRINTOUT OF EACH ELEMENT MATRIX (0)
 OR PRINT EVERY NTH ELEMENT MATRIX (N)
  ENTER 0 OR N  (DEFAULT = 0)
?
 SPECIFY GRID POINT PRINTOUT FOR BOUNDARY NODES ONLY (0)
 OR FOR EVERY NTH NODE (N)
  ENTER 0 OR N  (DEFAULT = 0)
?
```

Note the extensive use of defaults for input items with commonly assumed values. This is further illustrated below:

```
 SPECIFY FORWARD (F) OR BACKWARD (B) FINITE DIFFERENCES
 FOR EACH TIME STEP AND EACH COORDINATE DIRECTION
  ENTER F OR B FOR:
   STEP 1 -- X DIRECTION  (DEFAULT = F)
?
   STEP 1 -- Y DIRECTION  (DEFAULT = F)
?
   STEP 2 -- X DIRECTION  (DEFAULT = B)
?
   STEP 2 -- Y DIRECTION  (DEFAULT = B)
?

 BEGIN INPUT DATA FOR ZONE    1

 SPECIFY NUMBER OF SECTIONS WITHIN ZONE    1
  ENTER NSECTS -- NO LIMIT  (DEFAULT = 1)
?
 SPECIFY NUMBER OF SEGMENTS IN EDGE  1
  ENTER NSEGS -- 1 TO 5  (DEFAULT = 1)
?
 SPECIFY EDGE SHAPE FUNCTION FOR EDGE  1
 USE THE FOLLOWING CODES:
      LINEAR         = 1          TRIG FUNCTION
      CIRCULAR ARC   = 2              OF X          = 5
      CONIC SECTION  = 3          TRIG FUNCTION
      HELICAL ARC    = 4            OF THETA        = 6
                    SPECIAL FUNCTION = 7
  ENTER NSHAPE -- 1 TO 7  (DEFAULT = 1)
?
 SPECIFY NUMBER OF SEGMENTS IN EDGE  2
  ENTER NSEGS -- 1 TO 5  (DEFAULT = 1)
?
 SPECIFY EDGE SHAPE FUNCTION FOR EDGE  2
  ENTER NSHAPE -- 1 TO 7  (DEFAULT = 1)
? 2
```

```
 SPECIFY NUMBER OF SEGMENTS IN EDGE  3
  ENTER NSEGS -- 1 TO 5  (DEFAULT = 1)
?
 SPECIFY EDGE SHAPE FUNCTION FOR EDGE  3
  ENTER NSHAPE -- 1 TO 7  (DEFAULT = 1)
?
 SPECIFY NUMBER OF SEGMENTS IN EDGE  4
  ENTER NSEGS -- 1 TO 5  (DEFAULT = 1)
?
 SPECIFY EDGE SHAPE FUNCTION FOR EDGE  4
  ENTER NSHAPE -- 1 TO 7  (DEFAULT = 1)
? 2
```

Some input prompts display tables of appropriate values for the initial prompt only; subsequent prompts for the same input do not display the table. The edge shape function prompt shown above and the boundary condition indicator prompt shown below are examples of this "table on first prompt" procedure.

```
SPECIFY BOUNDARY CONDITION INDICATOR FOR EDGE 1
USE THE FOLLOWING CODES:
     CONSTANT NODES = 0            FREE-SLIP/
     AXIS NODES     = 1             TANGENCY       = 4
     NO-SLIP/STAG-                 SPECIAL CASE    = 5-7
      NATION NODES  = 2            ONE-SIDED
     CORNER NODES   = 3             DIFFERENCES    = 8
                  INTERIOR NODES = 9
 ENTER IBWL -- 0 TO 9  (DEFAULT = 9)
```

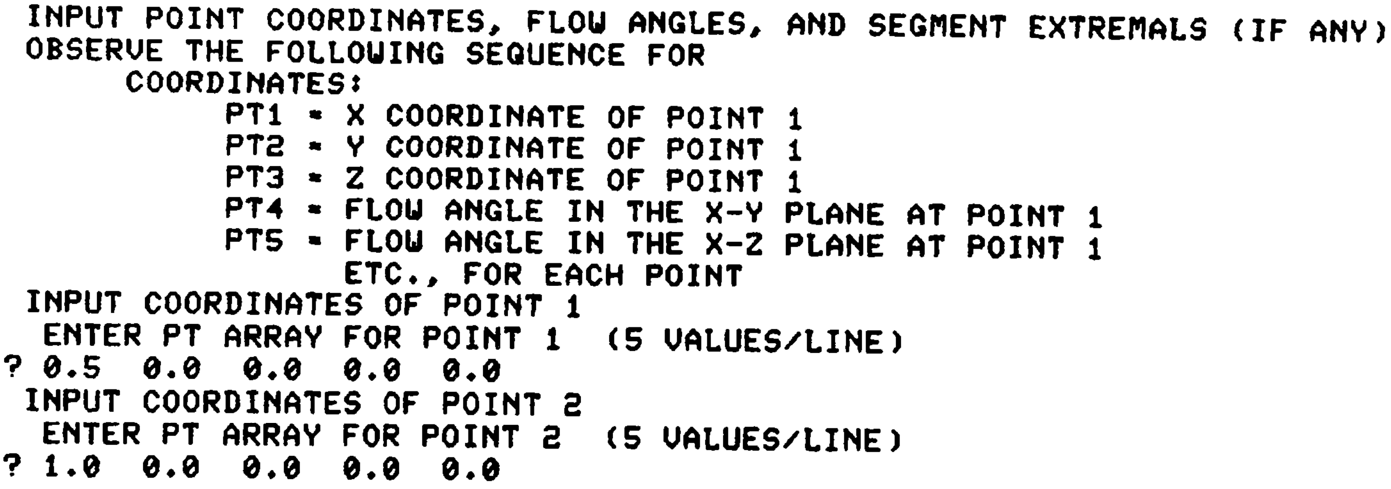
Subsequent prompts for NSHAPE and IBWL will not display the table of values in order to expedite the input process.

The following sequence illustrates the program's ability to respond to improper input:

```
 SPECIFY NUMBER OF NODES IN THE ETA1 DIRECTION
  ENTER NNOD -- 2 TO 100  (NO DEFAULT)
?
 NO DEFAULT FOR NNOD
 SPECIFY NUMBER OF NODES IN THE ETA1 DIRECTION
  ENTER NNOD -- 2 TO 100  (NO DEFAULT)
? 0
 INPUT ERROR -- NNOD
 PLEASE OBSERVE:     2 .LE. NNOD   .LE.    100
 SPECIFY NUMBER OF NODES IN THE ETA1 DIRECTION
  ENTER NNOD -- 2 TO 100  (NO DEFAULT)
? 99.95
      99. < ERROR, RETYPE RECORD AT THIS FIELD
? 99
```

At the first request for the number of nodes (NNOD), the user attempts to default. GEOM responds by indicating that there is no default for NNOD and repeats the prompt. This time the user enters an illegal value (0). GEOM indicates an error, displays proper bounds for NNOD, and repeats the prompt. The user next enters a value of improper format (real value instead of the integer value implied by NNOD). Again GEOM indicates the error and re-prompts. This time the user enters a legal value in the format consistent with the input variable and the program continues.

The virtues of the list-directed input mode are illustrated below where several values are requested simultaneously:



```
  INPUT POINT COORDINATES, FLOW ANGLES, AND SEGMENT EXTREMALS (IF ANY)
  OBSERVE THE FOLLOWING SEQUENCE FOR
       COORDINATES:
            PT1 = X COORDINATE OF POINT 1
            PT2 = Y COORDINATE OF POINT 1
            PT3 = Z COORDINATE OF POINT 1
            PT4 = FLOW ANGLE IN THE X-Y PLANE AT POINT 1
            PT5 = FLOW ANGLE IN THE X-Z PLANE AT POINT 1
                  ETC., FOR EACH POINT
  INPUT COORDINATES OF POINT 1
   ENTER PT ARRAY FOR POINT 1  (5 VALUES/LINE)
? 0.5  0.0  0.0  0.0  0.0
  INPUT COORDINATES OF POINT 2
   ENTER PT ARRAY FOR POINT 2  (5 VALUES/LINE)
? 1.0  0.0  0.0  0.0  0.0
```

Here several values are input on the same line without regard to column location. Either blanks or commas may be used to separate values. Default, range, and format checking and error processing are invoked for each value.

GEOM continues prompting for and accepting input data for the grid generation program, executing all of the loops required to create a complete input data file. The beginning of each major loop in the program is indicated so that the user knows precisely what portion of the geometry the subsequent input describes:

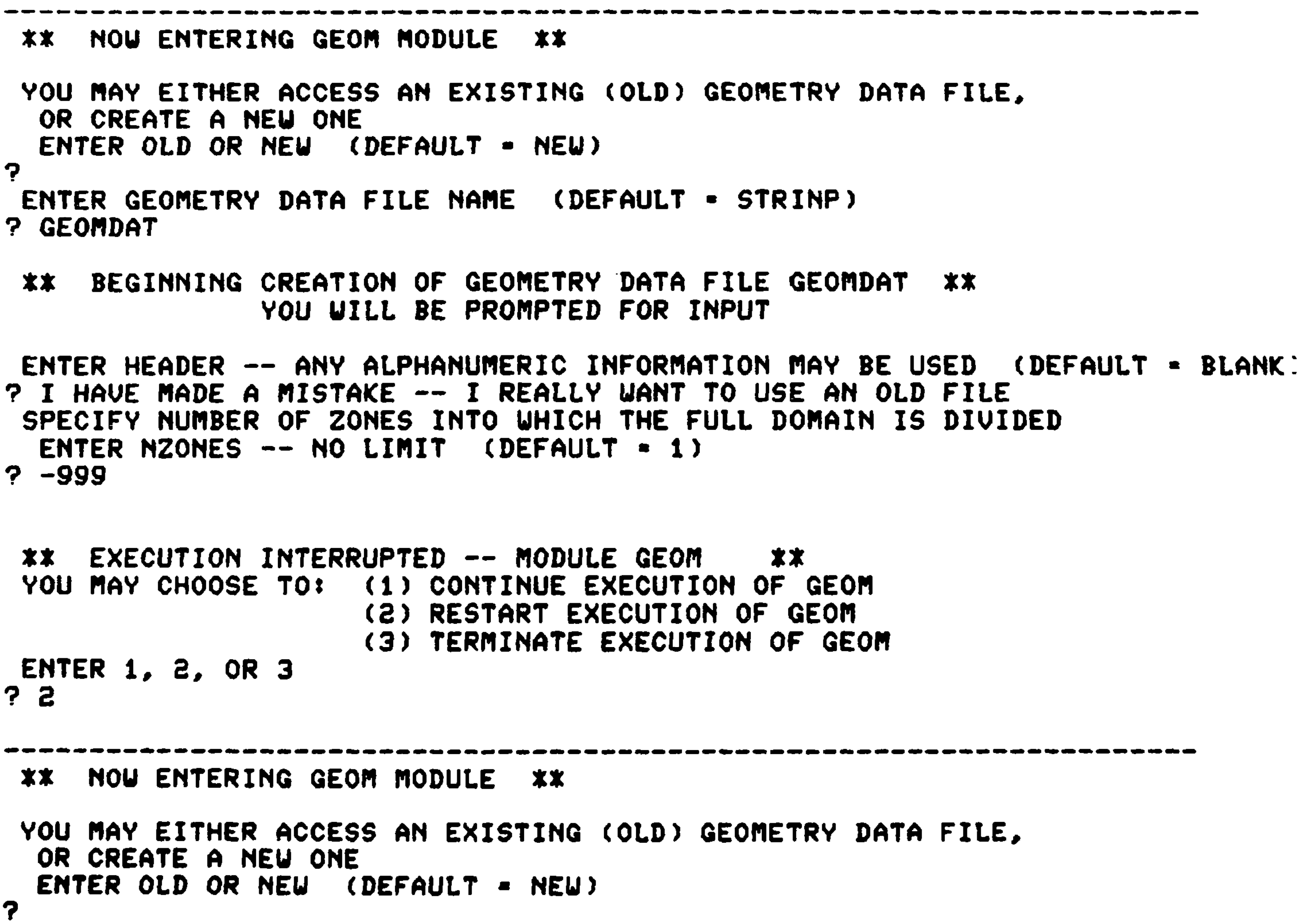
```
  BEGIN INPUT DATA FOR ZONE     2

  SPECIFY NUMBER OF SECTIONS WITHIN ZONE     2
   ENTER NSECTS -- NO LIMIT  (DEFAULT = 1)
 ?
  SPECIFY NUMBER OF SEGMENTS IN EDGE  1
   ENTER NSEGS -- 1 TO 5  (DEFAULT = 1)
 ?
```

The next example illustrates how GEOM can access an existing geometry input data file and then edit it via XEDIT.

```
-----------------------------------------------------------------------
 **  NOW ENTERING GEOM MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) GEOMETRY DATA FILE,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT = NEW)
? OLD
 ENTER GEOMETRY DATA FILE NAME/USER NUMBER  (DEFAULT = */LOGIN NUMBER)
? GEOMDAT
 DO YOU WISH TO XEDIT FILE GEOMDAT  (DEFAULT = NO)
? YES
 DO YOU WISH TO SAVE FILE GEOMDAT  (DEFAULT = NO)
?

PREPARE TO XEDIT FILE GEOMDAT
XEDIT 2.1.7
?? P*
2-D SUPERSONIC SOURCE FLOW CASE -- A TEST FOR RUNGIM
    1    2    2    0    1
    0    0    1
    F    F         B    B
    1    1
    1    2    1    2
    4    8    4    0    0    0    1
   11   11    0    0    0    0
       0.0       0.0       0.0       0.0
       0.0       0.0       0.0       0.0
        .5       0.0       0.0       0.0       0.0
        1.       0.0       0.0       0.0       0.0
    .96593    .25882       0.0       15.       0.0
    .48296    .12941       0.0       15.       0.0
/EOR
END OF FILE
??
```

Here the data file can be modified using XEDIT to affect any changes desired. After editing, the file could be saved; however, in this instance the user relied on the default "NO SAVE" option so that the file will remain as a local file only.

The interrupt capability of RUNGIM is illustrated in the next example. After several prompts, the user decides that he really wanted to use an existing data file instead of creating a new one. In response to the next prompt, the user enters the "abort code" (-999 in this version):



```
------------------------------------------------------------------------
 **  NOW ENTERING GEOM MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) GEOMETRY DATA FILE,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT • NEW)
?
 ENTER GEOMETRY DATA FILE NAME  (DEFAULT • STRINP)
? GEOMDAT

 **  BEGINNING CREATION OF GEOMETRY DATA FILE GEOMDAT  **
              YOU WILL BE PROMPTED FOR INPUT

 ENTER HEADER -- ANY ALPHANUMERIC INFORMATION MAY BE USED  (DEFAULT • BLANK:
? I HAVE MADE A MISTAKE -- I REALLY WANT TO USE AN OLD FILE
 SPECIFY NUMBER OF ZONES INTO WHICH THE FULL DOMAIN IS DIVIDED
  ENTER NZONES -- NO LIMIT  (DEFAULT • 1)
? -999


 **  EXECUTION INTERRUPTED -- MODULE GEOM    **
 YOU MAY CHOOSE TO:  (1) CONTINUE EXECUTION OF GEOM
                     (2) RESTART EXECUTION OF GEOM
                     (3) TERMINATE EXECUTION OF GEOM
 ENTER 1, 2, OR 3
? 2

------------------------------------------------------------------------
 **  NOW ENTERING GEOM MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) GEOMETRY DATA FILE,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT • NEW)
?
```

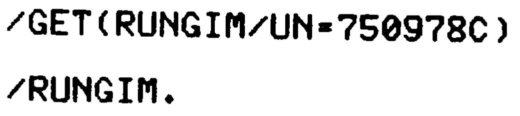
GEOM responds by interrupting execution and giving the user several options, i.e., continuing execution, restarting, or terminating the program. The user selects a restart and the program restarts at the beginning of GEOM.

All of the preceding examples have illustrated the GEOM program. However, the others are similar in procedures and features though they perform different tasks. Several examples are given on the following pages to demonstrate the features and capabilities of the various RUNGIM modules. The sample problem used in these examples is the two-dimensional supersonic source flow case of Ref. 1 (Section 8.1, Ref. 1).

6.3 RUNGIM EXAMPLES

6.3.1 Access and Execution

The RUNGIM program package is available to any remote terminal user and may be accessed and executed using the following commands (in batch mode):



```
/GET(RUNGIM/UN=750978C)
/RUNGIM.
```

6.3.2 RUNGIM Instructions

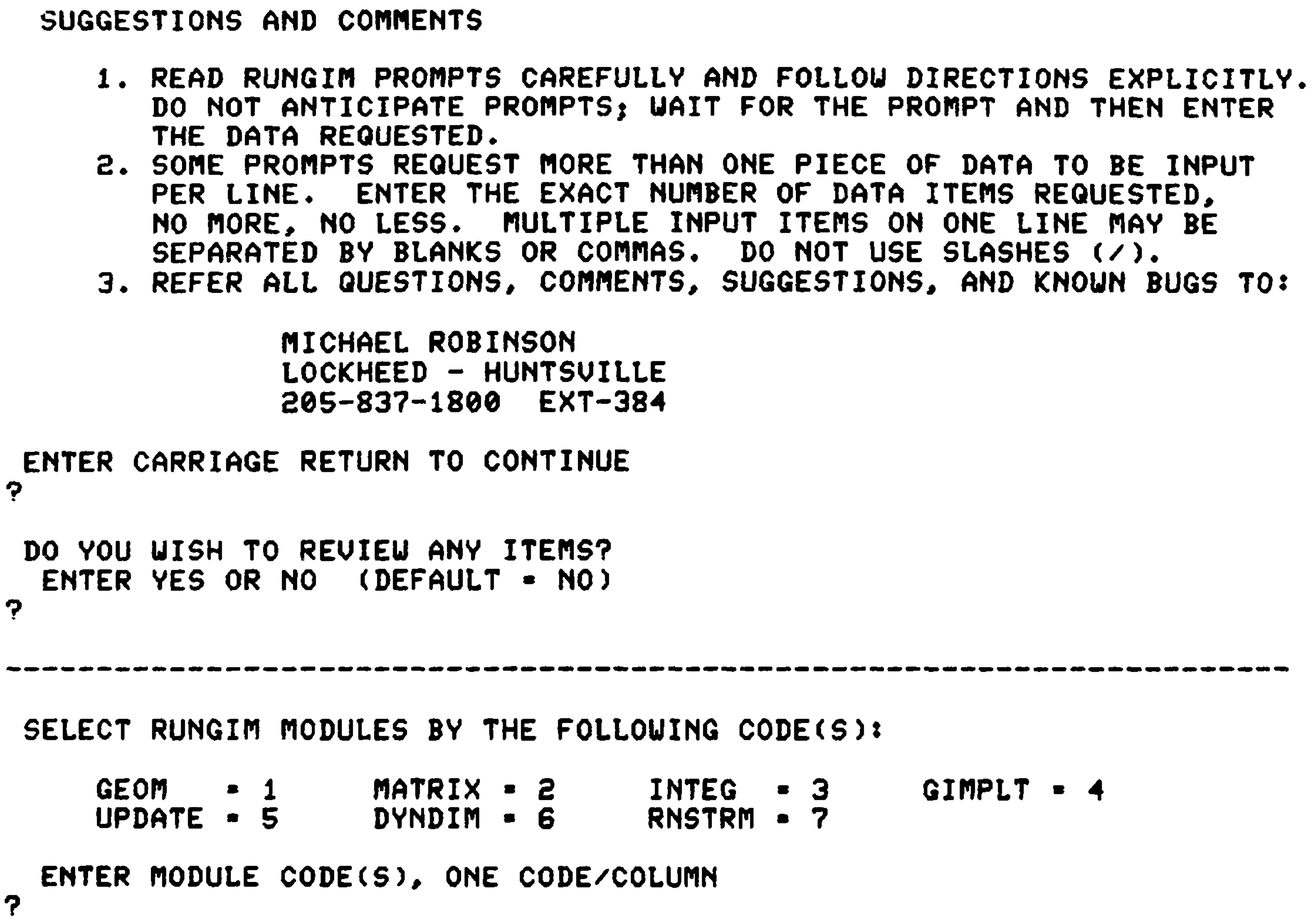
The following are the instructions issued by RUNGIM if requested by the user:

```
/RUNGIM.

**********************************************************************
 BEGINNING RUNGIM                              VERSION DATE: 06/26/81

 YOU WILL BE PROMPTED FOR INFORMATION REQUIRED TO CREATE OR ACCESS
 VARIOUS DATA FILES AND THE RUNSTREAM

 DO YOU WISH TO SEE INSTRUCTIONS BEFORE PROCEEDING  (DEFAULT = NO)
? YES

----------------------------------------------------------------------
 **  RUNGIM INSTRUCTIONS  **


 PLEASE SELECT AN ITEM FROM THE FOLLOWING MENU:

           1.  GENERAL DESCRIPTION OF RUNGIM
           2.  CATALOG OF RUNGIM MODULES
           3.  FREE-FORMAT INPUT DESCRIPTION
           4.  ERROR PROCESSING (ABORT MODE)
           5.  RUNGIM DEFAULTS
           6.  COMPATIBILITY WITH CURRENT GIM CODE DECKS
           7.  KNOWN "BUGS" IN RUNGIM
           8.  SUGGESTIONS AND COMMENTS
           9.  ALL OF THE ABOVE
          10.  RETURN TO RUNGIM

  ENTER ITEM NUMBER  (DEFAULT = 10)
? 9
```

```
  GENERAL DESCRIPTION OF RUNGIM

     RUNGIM IS A COLLECTION OF FORTRAN PROGRAMS AND CCL PROCEDURES
     WHICH PERMIT THE USER TO INTERACTIVELY CREATE INPUT DATA FILES
     AND RUNSTREAMS FOR THE VARIOUS GIM CODE MODULES.  RUNGIM QUERIES
     THE USER FOR INPUT DATA VIA SELF-EXPLANATORY PROMPTS.  AS THE USER
     ENTERS THE DATA, RUNGIM ARRANGES THE DATA IN THE PROPER SEQUENCE
     AND FORMAT AND WRITES IT OUT ONTO USER-NAMED FILES.  ALL USER-
     SUPPLIED INPUT IS SCANNED FOR PROPER FORMAT (INTEGER, REAL, OR
     ALPHANUMERIC) AND RANGE; THE USER IS NOTIFIED OF ERRORS AND
     INVITED TO REENTER CORRECT DATA.  AFTER CREATION OF A DATA FILE
     OR RUNSTREAM, THE USER HAS THE OPTION OF 'XEDIT'ING THE FILE
     AND/OR 'SAVE'ING IT.

 ENTER CARRIAGE RETURN TO CONTINUE
?

  CATALOG OF RUNGIM MODULES

     RUNGIM HAS THE FOLLOWING MODULES:

        1. GEOM   -- CREATES INPUT DATA FILE FOR GEOMETRY MODULES
        2. MATRIX -- CREATES INPUT DATA FILE FOR MATRIX MODULES (NA)
        3. INTEG  -- CREATES INPUT DATA FILE FOR INTEGRATOR MODULES
        4. GIMPLT -- CREATES INPUT DATA FILE FOR PLOT MODULES (NA)
        5. UPDATE -- CREATES DIRECTIVE FILE FOR THE UPDATE PROCESSOR
        6. DYNDIM -- CREATES INPUT DATA FILE FOR THE DYNAMIC DIMENSIONERS
        7. RNSTRM -- CREATES RUNSTREAMS FOR ALL GIM CODE MODULES

 ENTER CARRIAGE RETURN TO CONTINUE
?
```

```
  FREE-FORMAT INPUT DESCRIPTION

     RUNGIM USES FREE-FORMAT (LIST-DIRECTED) INPUT SO THAT THE USER
     NEED NOT BE CONCERNED ABOUT COLUMN NUMBERS OR SPACING.  RUNGIM
     PROMPTS ARE OF THE SAME TYPE (INTEGER OR REAL) AS THE NUMERIC
     DATA TO BE ENTERRED.  THUS THE USER MIGHT RESPOND AS FOLLOWS:

        . . . ENTER IDIM?   2  (INTEGER INPUT FOR INTEGER PROMPT)

        . . . ENTER EMU?  0.1  (REAL INPUT FOR REAL PROMPT)

     REAL NUMBERS MAY BE ENTERRED AS DECIMAL NUMBERS OR WITH E-TYPE
     NOTATION, E.G.,  0.1  OR  1.E-1.  ALPHANUMERIC INPUT DOES NOT
     REQUIRE ENCLOSING QUOTES OR HOLLERITH COUNT - ALL THIS IS HANDLED
     AUTOMATICALLY BY RUNGIM.

 ENTER CARRIAGE RETURN TO CONTINUE
?

  ERROR PROCESSING (ABORT MODE)

     AT ANY TIME DURING THE USE OF RUNGIM, THE USER MAY ELECT TO HALT
     EXECUTION OF THE PROGRAM BYE ENTERRING THE ABORT CODE: -999
     RUNGIM WILL THEN SUSPEND EXECUTION AND QUERY THE USER AS TO HOW
     TO PROCEED.

 ENTER CARRIAGE RETURN TO CONTINUE
?
```

```
  RUNGIM DEFAULTS

     MANY INPUT ITEMS HAVE BUILT-IN DEFAULT VALUES FOR COMMONLY SELECTED
     VALUES; THESE DEFAULTS MAY BE SELECTED BY TYPING A CARRIAGE RETURN
     IN RESPONSE TO A PROMPT.  DEFAULT VALUES ARE INDICATED IN THE
     PROMPT BY THE MESSAGE:  (DEFAULT = 0.1)  (FOR EXAMPLE).
     SOME PROMPTS HAVE NO DEFAULT; THE USER MUST ENTER SOME VALUE.
     RUNGIM WILL NOT PERMIT THE USER TO CIRCUMVENT NO-DEFAULT ITEMS.

 ENTER CARRIAGE RETURN TO CONTINUE
?

  COMPATIBILITY WITH CURRENT GIM CODE DECKS

     RUNGIM MODULES ARE COMATIBLE WITH CURRENT GIM CODE DECKS AS
     INDICATED BELOW:

          GEOM   -- GEOMC
          INTEG  -- INTEGD & INTEGE
          UPDATE -- ALL VERSIONS
          DYNDIM -- DYNMATC/DYNDIMC
          RNSTRM -- VERSIONS C AND D OF ALL DECKS

 ENTER CARRIAGE RETURN TO CONTINUE
?

  KNOWN "BUGS" IN RUNGIM

     THE FOLLOWING ARE KNOWN "BUGS" IN THE RUNGIM MODULES:

        1. GEOM, UPDATE, DYNDIM, AND RNSTRM HAVE NO KNOWN BUGS.
        2. INTEG IS A NEW MODULE AND MAY BE ERROR PRONE INITIALLY
                 QP ROUTES MAY BE PARTICULARLY VULNERABLE.

 ENTER CARRIAGE RETURN TO CONTINUE
?
```



```
  SUGGESTIONS AND COMMENTS

     1. READ RUNGIM PROMPTS CAREFULLY AND FOLLOW DIRECTIONS EXPLICITLY.
        DO NOT ANTICIPATE PROMPTS; WAIT FOR THE PROMPT AND THEN ENTER
        THE DATA REQUESTED.
     2. SOME PROMPTS REQUEST MORE THAN ONE PIECE OF DATA TO BE INPUT
        PER LINE.  ENTER THE EXACT NUMBER OF DATA ITEMS REQUESTED,
        NO MORE, NO LESS.  MULTIPLE INPUT ITEMS ON ONE LINE MAY BE
        SEPARATED BY BLANKS OR COMMAS.  DO NOT USE SLASHES (/).
     3. REFER ALL QUESTIONS, COMMENTS, SUGGESTIONS, AND KNOWN BUGS TO:

               MICHAEL ROBINSON
               LOCKHEED - HUNTSVILLE
               205-837-1800  EXT-384

 ENTER CARRIAGE RETURN TO CONTINUE
?

 DO YOU WISH TO REVIEW ANY ITEMS?
  ENTER YES OR NO  (DEFAULT = NO)
?
---------------------------------------------------------------------

 SELECT RUNGIM MODULES BY THE FOLLOWING CODE(S):

     GEOM   = 1     MATRIX = 2     INTEG  = 3     GIMPLT = 4
     UPDATE = 5     DYNDIM = 6     RNSTRM = 7

  ENTER MODULE CODE(S), ONE CODE/COLUMN
?
```

6.3.3 Supersonic Source Flow (2-D) — GEOM

The following illustrates a complete RUNGIM setup of the input data and runstream required to execute the GIM code GEOM module for the supersonic source flow example (Section 8.1, Ref. 6-1).

```
/RUNGIM.

**********************************************************************
 BEGINNING RUNGIM                              VERSION DATE: 06/26/81

 YOU WILL BE PROMPTED FOR INFORMATION REQUIRED TO CREATE OR ACCESS
 VARIOUS DATA FILES AND THE RUNSTREAM

 DO YOU WISH TO SEE INSTRUCTIONS BEFORE PROCEEDING  (DEFAULT = NO)
?
 SELECT RUNGIM MODULES BY THE FOLLOWING CODE(S):

     GEOM   = 1     MATRIX = 2     INTEG  = 3     GIMPLT = 4
     UPDATE = 5     DYNDIM = 6     RNSTRM = 7

  ENTER MODULE CODE(S), ONE CODE/COLUMN
? 1567

 YOU HAVE SELECTED THE FOLLOWING MODULES:

     GEOM      UPDATE    DYNDIM    RNSTRM

 ENTER GO TO CONTINUE  (DEFAULT = RESELECT MODULES)
? GO

----------------------------------------------------------------------
 **  NOW ENTERING GEOM MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) GEOMETRY DATA FILE,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT = NEW)
?
 ENTER GEOMETRY DATA FILE NAME  (DEFAULT = STRINP)
? GEOMDAT
```

```
 **  BEGINNING CREATION OF GEOMETRY DATA FILE GEOMDAT  **
               YOU WILL BE PROMPTED FOR INPUT

 ENTER HEADER -- ANY ALPHANUMERIC INFORMATION MAY BE USED  (DEFAULT = BLANK)
? 2-D SUPERSONIC SOURCE FLOW CASE -- A TEST FOR RUNGIM
 SPECIFY NUMBER OF ZONES INTO WHICH THE FULL DOMAIN IS DIVIDED
  ENTER NZONES -- NO LIMIT  (DEFAULT = 1)
?
 SPECIFY PROBLEM DIMENSIONALITY -- 1D, 2D, OR 3D
  ENTER 1, 2, OR 3  (DEFAULT = 2)
?
 SPECIFY ONE-STEP OR TWO-STEP TIME INTEGRATION SCHEME
  ENTER 1 OR 2  (DEFAULT = 2)
?
 SPECIFY GENERATION OF BOTH GRID AND FINITE DIFFERENCE MATRICES (0)
 OR GRID GENERATION ONLY (1)
  ENTER 0 OR 1  (DEFAULT = 0)
?
 SPECIFY DO NOT MATE ZONES (0) OR MATE ZONES (1)
  ENTER 0 OR 1  (DEFAULT = 1)
?
 SPECIFY ELLIPTIC RUN (0) OR QP RUN(1)
  ENTER 0 OR 1  (DEFAULT = 0)
?
 SPECIFY NO DEBUG OUTPUT (0) OR PRINT DEBUG OUTPUT (1)
  ENTER 0 OR 1  (DEFAULT = 0)
?
 SPECIFY NO PRINTOUT OF EACH ELEMENT MATRIX (0)
 OR PRINT EVERY NTH ELEMENT MATRIX (N)
  ENTER 0 OR N  (DEFAULT = 0)
?
 SPECIFY GRID POINT PRINTOUT FOR BOUNDARY NODES ONLY (0)
 OR FOR EVERY NTH NODE (N)
  ENTER 0 OR N  (DEFAULT = 0)
? 1
```

```
 SPECIFY FORWARD (F) OR BACKWARD (B) FINITE DIFFERENCES
 FOR EACH TIME STEP AND EACH COORDINATE DIRECTION
  ENTER F OR B FOR:
   STEP 1 -- X DIRECTION  (DEFAULT = F)
?
   STEP 1 -- Y DIRECTION  (DEFAULT = F)
?
   STEP 2 -- X DIRECTION  (DEFAULT = B)
?
   STEP 2 -- Y DIRECTION  (DEFAULT = B)
?

 BEGIN INPUT DATA FOR ZONE    1

 SPECIFY NUMBER OF SECTIONS WITHIN ZONE    1
  ENTER NSECTS -- NO LIMIT  (DEFAULT = 1)
?
 SPECIFY NUMBER OF SEGMENTS IN EDGE  1
  ENTER NSEGS -- 1 TO 5  (DEFAULT = 1)
?
 SPECIFY EDGE SHAPE FUNCTION FOR EDGE  1
 USE THE FOLLOWING CODES:
      LINEAR         = 1          TRIG FUNCTION
      CIRCULAR ARC   = 2              OF X          = 5
      CONIC SECTION  = 3          TRIG FUNCTION
      HELICAL ARC    = 4            OF THETA        = 6
                    SPECIAL FUNCTION = 7
  ENTER NSHAPE -- 1 TO 7  (DEFAULT = 1)
?
 SPECIFY NUMBER OF SEGMENTS IN EDGE  2
  ENTER NSEGS -- 1 TO 5  (DEFAULT = 1)
?
 SPECIFY EDGE SHAPE FUNCTION FOR EDGE  2
  ENTER NSHAPE -- 1 TO 7  (DEFAULT = 1)
? 2
```

```
 SPECIFY NUMBER OF SEGMENTS IN EDGE  3
  ENTER NSEGS -- 1 TO 5  (DEFAULT = 1)
?
 SPECIFY EDGE SHAPE FUNCTION FOR EDGE  3
  ENTER NSHAPE -- 1 TO 7  (DEFAULT = 1)
?
 SPECIFY NUMBER OF SEGMENTS IN EDGE  4
  ENTER NSEGS -- 1 TO 5  (DEFAULT = 1)
?
 SPECIFY EDGE SHAPE FUNCTION FOR EDGE  4
  ENTER NSHAPE -- 1 TO 7  (DEFAULT = 1)
? 2
 SPECIFY BOUNDARY CONDITION INDICATOR FOR EDGE 1
 USE THE FOLLOWING CODES:
      CONSTANT NODES = 0          FREE-SLIP/
      AXIS NODES     = 1           TANGENCY       = 4
      NO-SLIP/STAG-               SPECIAL CASE    = 5-7
       NATION NODES  = 2          ONE-SIDED
      CORNER NODES   = 3           DIFFERENCES    = 8
                    INTERIOR NODES = 9
  ENTER IBWL -- 0 TO 9  (DEFAULT = 9)
? 4
 SPECIFY ZONE TO WHICH EDGE 1 IS MATED  (0 FOR NO MATE)
  ENTER MATE -- 0 TO  1  (DEFAULT = 0)
?
 SPECIFY BOUNDARY CONDITION INDICATOR FOR EDGE 2
  ENTER IBWL -- 0 TO 9  (DEFAULT = 9)
? 8
 SPECIFY ZONE TO WHICH EDGE 2 IS MATED  (0 FOR NO MATE)
  ENTER MATE -- 0 TO  1  (DEFAULT = 0)
?
 SPECIFY BOUNDARY CONDITION INDICATOR FOR EDGE 3
  ENTER IBWL -- 0 TO 9  (DEFAULT = 9)
? 4
 SPECIFY ZONE TO WHICH EDGE 3 IS MATED  (0 FOR NO MATE)
  ENTER MATE -- 0 TO  1  (DEFAULT = 0)
?
```

```
 SPECIFY BOUNDARY CONDITION INDICATOR FOR EDGE 4
  ENTER IBWL -- 0 TO 9  (DEFAULT = 9)
? 0
 SPECIFY ZONE TO WHICH EDGE 4 IS MATED  (0 FOR NO MATE)
  ENTER MATE -- 0 TO  1  (DEFAULT = 0)
?
 SPECIFY NODAL NUMBERING SEQUENCE ALONG COORDINATE DIRECTIONS
 USE THE FOLLOWING CODES:
               ETA2, ETA1 = 1
               ETA1, ETA2 = 2
  ENTER ITRAIN -- 1 TO 2  (DEFAULT = 1)
?
 SPECIFY NUMBER OF NODES IN THE ETA1 DIRECTION
  ENTER NNOD -- 2 TO 100  (NO DEFAULT)
? 11
 SPECIFY NODAL DISTRIBUTION OPTION IN THE ETA1 DIRECTION
 USE THE FOLLOWING CODES:
          UNIFORM SPACING        =  0
          REDUCE ETA1 SPACING    = -1
          INCREASE ETA1 SPACING  = +1
          INPUT LOCATION OF NODES = NNOD
  ENTER ISTRCH -- 0 TO 11  (DEFAULT = 0)
?
 SPECIFY NUMBER OF NODES IN THE ETA2 DIRECTION
  ENTER NNOD -- 2 TO 100  (NO DEFAULT)
? 11
 SPECIFY NODAL DISTRIBUTION OPTION IN THE ETA2 DIRECTION
  ENTER ISTRCH -- 0 TO 11  (DEFAULT = 0)
?
 INPUT AC COEFFICIENTS FOR ZONE  1
 REFER TO TABLE 4-2 IN THE BLUE BOOK FOR CODES
  ENTER 4 AC VALUES FOR EDGE 2
? 0.0  0.0  0.0  0.0
  ENTER 4 AC VALUES FOR EDGE 4
? 0.0  0.0  0.0  0.0
```

```
 INPUT POINT COORDINATES, FLOW ANGLES, AND SEGMENT EXTREMALS (IF ANY)
 OBSERVE THE FOLLOWING SEQUENCE FOR
      COORDINATES:
           PT1 = X COORDINATE OF POINT 1
           PT2 = Y COORDINATE OF POINT 1
           PT3 = Z COORDINATE OF POINT 1
           PT4 = FLOW ANGLE IN THE X-Y PLANE AT POINT 1
           PT5 = FLOW ANGLE IN THE X-Z PLANE AT POINT 1
                 ETC., FOR EACH POINT
 INPUT COORDINATES OF POINT 1
  ENTER PT ARRAY FOR POINT 1  (5 VALUES/LINE)
? 0.5  0.0  0.0  0.0  0.0
 INPUT COORDINATES OF POINT 2
  ENTER PT ARRAY FOR POINT 2  (5 VALUES/LINE)
? 1.0  0.0  0.0  0.0  0.0
 INPUT COORDINATES OF POINT 3
  ENTER PT ARRAY FOR POINT 3  (5 VALUES/LINE)
? 0.96593  0.25882  0.0  15.0  0.0
 INPUT COORDINATES OF POINT 4
  ENTER PT ARRAY FOR POINT 4  (5 VALUES/LINE)
? 0.48296  0.12941  0.0  15.0  0.0
 DO YOU WISH TO XEDIT FILE GEOMDAT  (DEFAULT = NO)
? YES
 DO YOU WISH TO SAVE FILE GEOMDAT  (DEFAULT = NO)
?


 GEOM COMPLETE  --  ESTIMATED NUMBER OF NODES =   121

PREPARE TO XEDIT FILE GEOMDAT
XEDIT 2.1.7
?? P*
```

```
2-D SUPERSONIC SOURCE FLOW CASE -- A TEST FOR RUNGIM
    1    2    2    0    1
    0    0    1
    F    F         B    B
    1    1
    1    2    1    2
    4    8    4    0    0    0    1
   11   11    0    0    0    0
       0.0       0.0       0.0       0.0
       0.0       0.0       0.0       0.0
        .5       0.0       0.0       0.0       0.0
        1.       0.0       0.0       0.0       0.0
    .96593    .25882       0.0       15.       0.0
    .48296    .12941       0.0       15.       0.0
/EOR
--EOR--
END OF FILE
?? Q
GEOMDAT IS A LOCAL FILE

------------------------------------------------------------------
 **  NOW ENTERING UPDATE MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) UPDATE DIRECTIVE FILE,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT = NEW)
?
 ENTER UPDATE DIRECTIVE FILE NAME  (DEFAULT = UPDATE)
? UPDSRC
```

```
 **  BEGINNING CREATION OF UPDATE DIRECTIVE FILE UPDSRC   **
 A BLANK LINE WILL TERMINATE DIRECTIVE INPUT
 ENTER ONE DIRECTIVE PER LINE
? *IDENT,SOURCE
? *
? *
? *     ANY UPDATE DIRECTIVES
? *
? *
?
 DO YOU WISH TO XEDIT FILE UPDSRC   (DEFAULT = NO)
?
 DO YOU WISH TO SAVE FILE UPDSRC   (DEFAULT = NO)
?

--------------------------------------------------------------
 **  NOW ENTERING DYNMAT/DYNDIM MODULE  **


 SPECIFY TOTAL NUMBER OF NODES
  ENTER NN  (DEFAULT =   121)
?
 SPECIFY MAXIMUM NUMBER OF SPECIAL NODE TERMS
  ENTER NSP -- 0 TO   121  (DEFAULT =   121)
?
 SPECIFY MAXIMUM NUMBER OF NODES IN A CROSS-PLANE
  ENTER NXMAX  (DEFAULT =    11)
?
 SPECIFY MAXIMUM NUMBER OF NODES IN MATED PLANES
  ENTER NMATE  (DEFAULT =    11)
?
 SPECIFY NUMBER OF PLANES OF NODES SAVED PER RECORD
  ENTER NSAV  (DEFAULT = 1)
? 11
```

```
 **  THE UPDATE DIRECTIVES AND DYNMAT INPUT DATA HAVE BEEN COMBINED  **
                     THE COMBINED FILE NAME IS UPDSRC

 DO YOU WISH TO XEDIT FILE UPDSRC   (DEFAULT = NO)
? YES
 DO YOU WISH TO SAVE FILE UPDSRC   (DEFAULT = NO)
?

PREPARE TO XEDIT FILE UPDSRC
XEDIT 2.1.7
?? P*
*IDENT,SOURCE
*
*
*     ANY UPDATE DIRECTIVES
*
*
/EOR
  121    2  121   11   11   11    0
/EOR
END OF FILE
?? Q
UPDSRC  IS A LOCAL FILE

-------------------------------------------------------------------------
 **  NOW ENTERING RNSTRM MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) RUNSTREAM,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT = NEW)
?
 ENTER RUNSTREAM FILE NAME  (DEFAULT = RNSTRM)
? RUNSRC
```

```
 **  BEGINNING CREATION OF RUNSTREAM FILE RUNSRC   **
     YOU WILL BE PROMPTED FOR INPUT


 SPECIFY JOBNAME, FIELD LENGTH, AND TIME LIMIT
 ENTER JOBNAME/FL/TL  (DEFAULT IS MODULE DEPENDENT)
? RUNSRC/60000/100
 ENTER USER NUMBER
? 750978C
 ENTER CHARGE NUMBER
? 101857
 ENTER GIM MODULE(S) TO BE RUN
? GEOMC
 ENTER STAR FILE16 LENGTH  (DEFAULT =     4)
?
 ENTER STAR FILE18 LENGTH  (DEFAULT =    12)
?
 ENTER STAR FILE20 LENGTH  (DEFAULT =     4)
?
 ENTER STAR GEOMB CONTROLLEE FILE LENGTH
? 600
 IF YOU WISH TO SAVE THE NOS DAYFILE, ENTER THE DAYFILE NAME
? NOSDAY
 IF YOU WISH TO SAVE THE STAR DAYFILE, ENTER THE DAYFILE NAME
? STRDAY
 DO YOU WISH TO XEDIT FILE RUNSRC   (DEFAULT = NO)
? YES
 DO YOU WISH TO SAVE FILE RUNSRC   (DEFAULT = NO)
?
 DO YOU WISH TO AUTOMATICALLY SUBMIT THIS JOB  (DEFAULT = NO)
?

PREPARE TO XEDIT FILE RUNSRC
XEDIT 2.1.7
?? P*
```

```
/JOB
/NOSEQ
RUNSRC,CM60000,T100.
USER
CHARGE(LRC)
GET(DYNMAT=DYNMATC/UN=838700C)
GET(OLDPL=GEOMC/UN=838700C)
UPDATE(F,C=TAPE8)
DYNMAT.
COPYCF(TAPE3,GEOMS)
REWIND(GEOMS)
RETURN(OLDPL,TAPE3,TAPE8)
TOSTAR(INPUT,GEOMS)
DAYFILE(NOSDAY)
REPLACE(NOSDAY)
EXIT.
DAYFILE(NOSDAY)
REPLACE(NOSDAY)
/EOR
/READ,UPDSRC
STORE 400SDS RUNSRC    B
STRSIDE,T100.
FORTRAN(I=GEOMS,B=GEOMB,O=LB)
REQUEST(FILE16/4,T=P)
REQUEST(FILE18/12,T=P)
REQUEST(FILE19/12,T=P)
REQUEST(FILE20/4,T=P)
LOAD(GEOMB,CN=GEOMGO,600,GRLPALL= )
GEOMGO.
TOAS(Z=750978C,FILE18=BI,FILE20)
DAYFILE(STRDAY)
EXIT.
TOAS(Z=750978C,FILE18=BI,FILE20)
DAYFILE(STRDAY)
EXIT.
/EOR
/READ,GEOMDAT
```

```
/EOR
REVERT. FROM EXRNST
END OF FILE
?? Q
RUNSRC  IS A LOCAL FILE
REVERT.   RUNGIM  COMPLETE
/
```

6.3.4 Supersonic Source Flow (2-D) – INTEG

The following illustrates a complete RUNGIM setup of the input data and runstream required to execute the GIM code INTEG module for the supersonic source flow example (Section 8.1, Ref. 6-1).

```
/RUNGIM.

************************************************************************
 BEGINNING RUNGIM                                     VERSION DATE: 06/26/81

 YOU WILL BE PROMPTED FOR INFORMATION REQUIRED TO CREATE OR ACCESS
 VARIOUS DATA FILES AND THE RUNSTREAM

 DO YOU WISH TO SEE INSTRUCTIONS BEFORE PROCEEDING  (DEFAULT = NO)
?
 SELECT RUNGIM MODULES BY THE FOLLOWING CODE(S):

     GEOM   = 1     MATRIX = 2     INTEG  = 3     GIMPLT = 4
     UPDATE = 5     DYNDIM = 6     RNSTRM = 7

  ENTER MODULE CODE(S), ONE CODE/COLUMN
? 3567

 YOU HAVE SELECTED THE FOLLOWING MODULES:

     INTEG     UPDATE    DYNDIM    RNSTRM

 ENTER GO TO CONTINUE  (DEFAULT = RESELECT MODULES)
? GO

------------------------------------------------------------------------
 **  NOW ENTERING INTEG MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) INTEGRATOR DATA FILE,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT = NEW)
?
 ENTER INTEGRATOR DATA FILE NAME  (DEFAULT = STRINP)
? INTDAT
```

```
 **  BEGINNING CREATION OF INTEGRATOR DATA FILE INTDAT   **
                YOU WILL BE PROMPTED FOR INPUT

 ENTER ITITLE -- ANY ALPHANUMERIC INFORMATION MAY BE USED  (DEFAULT = BLANK)
? 2-D SUPERSONIC SOURCE FLOW CASE -- A TEST FOR RUNGIM
 SPECIFY PROBLEM DIMENSIONALITY -- 1D, 2D, OR 3D
  ENTER 1, 2, OR 3  (DEFAULT = 2)
?
 SPECIFY TIME INTEGRATION SCHEME:
           (1) ONE-STEP ELLIPTIC
           (2) TWO-STEP ELLIPTIC
           (3) QUASI-PARABOLIC
  ENTER 1, 2, OR 3  (DEFAULT = 2)
?
 SPECIFY MAXIMUM NUMBER OF TIME-STEP ITERATIONS
  ENTER ITMAX -- NO LIMIT  (NO DEFAULT)
? 200
 SPECIFY FLOW FIELD PRINT CONTROL FLAG
 FLOW FIELD WILL BE PRINTED EVERY IPRNT-TH ITERATION
  ENTER IPRNT -- 0 TO   200  (DEFAULT = 0)
? 200
 SPECIFY FLOW FIELD SAVE CONTROL FLAG
 FLOW FIELD WILL BE SAVED (ON FILE 22) EVERY ITSAV-TH ITERATION
  ENTER ITSAV -- 0 TO   200  (DEFAULT = 0)
? 200
 SPECIFY COLD START RUN (0) OR RESTART RUN (N),
 WHERE N IS THE RECORD NUMBER OF THE STARTING DATA ON FILE 22
  ENTER ISTART  (DEFAULT = 0)
?
```

```
 SPECIFY PRINT OUTPUT TYPE:
    (1)  ONE-LINE OUTPUT -- NODE, X, Y, Z, RHO, U, V, W, Q, E, P, & IB
    (2)  TWO-LINE OUTPUT -- AS ABOVE PLUS GAM, CS, PHIX, PHIY, SOS, T, & M
    (3)  SIMPLIFIED OUTPUT -- NODE, RHO, P, M, PHIX, PHIY, & IB
  ENTER IOTYPE -- 1 TO 3  (DEFAULT = 1)
? 3
 DO YOU WISH TO PRINT THE ZERO-TH ITERATION?
  ENTER YES OR NO  (DEFAULT = YES)
?
 SPECIFY ENGLISH UNITS (1) OR CGS METRIC/DIMENSIONLESS UNITS (2)
  ENTER IUNITS -- 1 OR 2  (DEFAULT = 2)
?
 SPECIFY STARTING ITERATION NUMBER
  ENTER ITSTRT  (DEFAULT = 0)
?
 SPECIFY TREATMENT OF VISCOSITY TERMS:
           (0) INVISCID EULER EQUATIONS
           (1) NUMERICAL VISCOSITY (DAMPING) ONLY
           (2) LAMINAR VISCOSITY AND DAMPING
           (3) BALDWIN/LOMAX TURBULENCE MODEL
  ENTER IVISC -- 0 TO 3  (DEFAULT = 0)
?
 SPECIFY INVISCID TYPE STARTING SOLUTIONS (0) OR VISCOUS/
 BOUNDARY LAYER TYPE STARTING SOLUTION (1) FOR NOZZLE FLOWS
  ENTER IDIST -- 0 OR 1  (DEFAULT = 0)
?
 SPECIFY ONE IDEAL GAS (0) OR A TWO-GAS SYSTEM (1)
  ENTER ISPEC -- 0 OR 1  (DEFAULT = 0)
?
 SPECIFY SUPERSONIC-TYPE/ONE-SIDED DIFFERENCES (0) OR SUBSONIC-TYPE/
 MASS BALANCING TECHNIQUE (1) FOR INFLOW AND OUTFLOW BOUNDARIES
  ENTER IDS -- 0 OR 1  (DEFAULT = 0)
?
```

```
 SPECIFY PRINT SUMSQ INFORMATION EVERY N-TH ITERATION
  ENTER IPSQ  (DEFAULT = 1)
?
 SPECIFY SUMSQ NORMALIZATION BY 1ST ITERATION VALUES (0),
 OR NO SUMSQ NORMALIZATION (1)
  ENTER INORM -- 0 OR 1  (DEFAULT = 0)
?
 SPECIFY CALCULATION OF SUMSQ FOR CONSERVED VARIABLES (0),
 OR FOR PRIMITIVE VARIABLES (1)
  ENTER ISBSM -- 0 OR 1  (DEFAULT = 0)
?
 SPECIFY TOTAL NUMBER OF NODES
  ENTER NN -- NO LIMIT  (NO DEFAULT)
? 121
 SPECIFY NUMBER OF NODES IN THE X-COORDINATE DIRECTION  (ZONE  1)
  ENTER NNX -- 2 TO 100  (NO DEFAULT)
? 11
 SPECIFY NODAL POINT INCREMENT IN THE X-COORDINATE DIRECTION  (ZONE  1)
  ENTER NDX  (NO DEFAULT)
? 11
 SPECIFY NUMBER OF NODES IN THE Y-COORDINATE DIRECTION  (ZONE  1)
  ENTER NNY -- 2 TO 100  (NO DEFAULT)
? 11
 SPECIFY NODAL POINT INCREMENT IN THE Y-COORDINATE DIRECTION  (ZONE  1)
  ENTER NDY  (NO DEFAULT)
? 1
 SPECIFY NO SPECIAL TREATMENT OF SHARP EXPANSION CORNERS (0)
 OR NPM SHARP CORNERS ARE TO BE TREATED EXPLICITLY
  ENTER NPM -- 0 TO 10  (DEFAULT = 0)
?
 SPECIFY NUMBER OF ZONES (IF A MULTI-ZONE PROBLEM)
  ENTER KZONES -- 1 TO 10  (DEFAULT = 1)
?
```

```
 IF A CONSTANT TIME STEP IS DESIRED,
  ENTER DTIME  (DEFAULT = VARIABLE TIME STEP SELECTED)
? 0.5E-2
 SPECIFY CFL FACTOR -- DTMIN = DTFAC * CFL
  ENTER DTFAC -- 0.0 TO 1.0  (DEFAULT = 1.0)
?
 SPECIFY TIME STEP UPDATE EVERY INCDT-TH ITERATION
  ENTER INCDT -- 0 TO   200  (DEFAULT = 1)
?
 SPECIFY EXPLICIT METHOD (0) OR IMPLICIT METHOD (1)
  ENTER IMPL -- 0 OR 1  (DEFAULT = 0)
?
 SPECIFY PRANDTL NUMBER
  ENTER REALK  (DEFAULT = 0.72)
?
 SPECIFY SPECIFIC HEAT RATIO
  ENTER GAM  (DEFAULT = 1.4)
?
 SPECIFY MOLECULAR WEIGHT OF THE GAS
  ENTER WM  (DEFAULT = GAS CONSTANT TO BE SPECIFIED NEXT)
?
 SPECIFY IDEAL GAS CONSTANT FOR THIS GAS
  ENTER RK  (NO DEFAULT)
? 1.0
 SPECIFY FORWARD (F) OR BACKWARD (B) FINITE DIFFERENCES
 FOR EACH TIME STEP AND EACH COORDINATE DIRECTION
  ENTER F OR B FOR:
   STEP 1 -- X DIRECTION  (DEFAULT = F)
?
   STEP 1 -- Y DIRECTION  (DEFAULT = F)
?
   STEP 2 -- X DIRECTION  (DEFAULT = B)
?
   STEP 2 -- Y DIRECTION  (DEFAULT = B)
?
```

```
 DO YOU WISH TO USE NOZZLE-FLOW TYPE INITIALIZATION?
  ENTER YES OR NO  (DEFAULT = NO)
?

 BEGIN FLOW FIELD INITIALIZATION -- USE AS MANY SPECIFICATIONS AS REQUIRED
 SPECIFY FIRST NODE TO BE INITIALIZED BY SPECIFICATION  1
  ENTER NJ -- 1 TO   121  (DEFAULT = TERMINATE FLOW FIELD INITIALIZATION)
? 1
 SPECIFY NODE NUMBER INCREMENT
  ENTER INC  (DEFAULT = 0)
? 1
 SPECIFY TOTAL NUMBER OF NODES SET BY SPECIFICATION  1
  ENTER NTOT -- 1 TO   121  (DEFAULT = 1)
? 121
 SPECIFY INPUT OF VELOCITY COMPONENTS (0) OR TOTAL VELOCITY (1)
  ENTER ITAN -- 0 OR 1  (DEFAULT = 1)
?
 SPECIFY TYPE OF INITIALIZATION TO BE DONE BY SPECIFICATION  1
           (0) INPUT INITIAL CONDITIONS NEXT
           (1) USE NOZZLE FLOW INITIALIZATION OPTION  (ELLIPTIC ONLY)
           (2) USE CONDITIONS ENTERRED FOR PREVIOUS SPECIFICATION
           (3) USE THE USERIP OPTION
  ENTER ITYPE -- 0 TO 3  (DEFAULT = 0)
?
 SPECIFY FLOW FIELD INITIAL CONDITIONS -- SPECIFICATION  1
  ENTER THE FOLLOWING:  (NO DEFAULT)
   MASS DENSITY
? 1.4
   TOTAL VELOCITY
? 2.0
   STATIC PRESSURE
? 1.0
```

```
 SPECIFY FIRST NODE TO BE INITIALIZED BY SPECIFICATION  2
  ENTER NJ -- 1 TO   121  (DEFAULT = TERMINATE FLOW FIELD INITIALI
?
 BEGIN NODAL OUTPUT CONTROL SPECIFCATIONS -- USE AS MANY AS REQUIR
 OUTPUT IS RESTRICTED TO   121 NODES
 SPECIFY FIRST NODE TO BE PRINTED BY SPECIFICATION  1
  ENTER N1 -- 1 TO   121  (DEFAULT = TERMINATE FLOW FIELD OUTPUT S
? 1
 SPECIFY NODE NUMBER INCREMENT
  ENTER IC  (DEFAULT  = 0)
? 1
 SPECIFY TOTAL NUMBER OF NODES TO BE PRINTED BY SPECIFICATION  1
  ENTER NT -- 1 TO   121  (DEFAULT = 1)
? 121
 SPECIFY FIRST NODE TO BE PRINTED BY SPECIFICATION  2
  ENTER N1 -- 1 TO   121  (DEFAULT = TERMINATE FLOW FIELD OUTPUT S
?
 DO YOU WISH TO XEDIT FILE INTDAT   (DEFAULT = NO)
? YES
 DO YOU WISH TO SAVE FILE INTDAT   (DEFAULT = NO)
?

PREPARE TO XEDIT FILE INTDAT
XEDIT 2.1.7
?? P*
```

```
2-D SUPERSONIC SOURCE FLOW CASE -- A TEST FOR RUNGIM
    2    2  200  200  200    0   -3    2    0    0    0    0    0    0    0
    1    0    0
  121   11   11   11    1    1    0    1
 .5000E-05  1.000000    1
       0.0       .72       1.4       0.0       0.0       0.0       0.0        1.
       0.0       0.0       0.0       0.0       0.0       0.0       0.0
       0.0
       0.0       0.0       0.0       0.0
    F    F         B    B
       0.0       0.0       0.0    0       0.0       0.0
    1    1  121    1    0
       1.4        2.       0.0       0.0        1.       0.0
   -1
    1    1  121
   -1
/EOR
--EOR--
END OF FILE
?? Q
INTDAT  IS A LOCAL FILE

----------------------------------------------------------------------
 **  NOW ENTERING UPDATE MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) UPDATE DIRECTIVE FILE,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT = NEW)
? OLD
 ENTER UPDATE DIRECTIVE FILE NAME/USER NUMBER  (DEFAULT = */LOGIN NUMBER)
? UPDSRC
 DO YOU WISH TO XEDIT FILE UPDSRC   (DEFAULT = NO)
?
```

```
-----------------------------------------------------------------------
 **  NOW ENTERING DYNMAT/DYNDIM MODULE  **


 ARE THE UPDATE DIRECTIVES AND DYNDIM INPUT DATA BOTH ON FILE UPDSRC
  ENTER YES OR NO  (NO DEFAULT)
? NO
 SPECIFY ELLIPTIC RUN (0) OR QP RUN (1)
  ENTER 0 OR 1  (DEFAULT = 0)
?
 SPECIFY TOTAL NUMBER OF NODES
  ENTER NN  (DEFAULT =   121)
?
 SPECIFY MAXIMUM NUMBER OF SPECIAL NODE TERMS
  ENTER NSP -- 0 TO   121  (DEFAULT =   121)
?
 SPECIFY MAXIMUM NUMBER OF BOUNDARY NODES
  ENTER MNB -- 0 TO   121  (DEFAULT =   121)
?
 SPECIFY NUMBER OF RECORDS OF DATA ON FILE20
  ENTER IREC  (DEFAULT = 1)
?

 **  THE UPDATE DIRECTIVES AND DYNDIM INPUT DATA HAVE BEEN COMBINED  **
                  THE COMBINED FILE NAME IS UPDSRC

 DO YOU WISH TO XEDIT FILE UPDSRC   (DEFAULT = NO)
? YES
 DO YOU WISH TO SAVE FILE UPDSRC   (DEFAULT = NO)
?

PREPARE TO XEDIT FILE UPDSRC
XEDIT 2.1.7
??
```

```
*IDENT,SOURCE
*
*
*     ANY UPDATE DIRECTIVES
*
*
/EOR
  121    2    0  121  121    0    1
/EOR
END OF FILE
?? Q
UPDSRC  IS A LOCAL FILE

-----------------------------------------------------------
 **  NOW ENTERING RNSTRM MODULE  **

 YOU MAY EITHER ACCESS AN EXISTING (OLD) RUNSTREAM,
  OR CREATE A NEW ONE
  ENTER OLD OR NEW  (DEFAULT = NEW)
?
 ENTER RUNSTREAM FILE NAME  (DEFAULT = RNSTRM)
? RUNSRC

 **  BEGINNING CREATION OF RUNSTREAM FILE RUNSRC   **
     YOU WILL BE PROMPTED FOR INPUT


 SPECIFY JOBNAME, FIELD LENGTH, AND TIME LIMIT
 ENTER JOBNAME/FL/TL  (DEFAULT IS MODULE DEPENDENT)
? RUNSRC/60000/100
 ENTER USER NUMBER
?- XXXXXXC
 ENTER CHARGE NUMBER
? XXXXXX
```

```
 ENTER GIM MODULE(S) TO BE RUN
? INTEGD
 ENTER NOS FILE17 NAME
? TAPE17
 ENTER NOS FILE20 NAME
? TAPE20
 ENTER STAR FILE22 LENGTH  (DEFAULT =     3)
?
 ENTER STAR INTGB BINARY FILE LENGTH  (DEFAULT = 100)
?
 ENTER STAR INTGB CONTROLLEE FILE LENGTH
? 1000
 IF YOU WISH TO SAVE THE NOS DAYFILE, ENTER THE DAYFILE NAME
? NOSDAY
 IF YOU WISH TO SAVE THE STAR DAYFILE, ENTER THE DAYFILE NAME
? STRDAY
 DO YOU WISH TO XEDIT FILE RUNSRC   (DEFAULT = NO)
? YES
 DO YOU WISH TO SAVE FILE RUNSRC   (DEFAULT = NO)
?
 DO YOU WISH TO AUTOMATICALLY SUBMIT THIS JOB  (DEFAULT = NO)
?

PREPARE TO XEDIT FILE RUNSRC
XEDIT 2.1.7
?? P*
```

```
/JOB
/NOSEQ
RUNSRC,CM60000,T100.
USER
CHARGE(LRC)
GET(DYNDIM=DYNDIMC/UN=838700C)
GET(OLDPL=INTEGD/UN=838700C)
UPDATE(F,C=TAPE8)
DYNDIM.
COPYCF(TAPE3,INTGS)
REWIND(INTGS)
RETURN(OLDPL,TAPE3,TAPE8)
ATTACH(FILE17=TAPE17)
ATTACH(FILE20=TAPE20)
TOSTAR(INPUT,INTGS,FILE20,FILE17=BI//U)
DAYFILE(NOSDAY)
REPLACE(NOSDAY)
EXIT.
DAYFILE(NOSDAY)
REPLACE(NOSDAY)
/EOR
/READ,UPDSRC
STORE 400SDS RUNSRC   B
STRSIDE,T100.
REQUEST(FILE22/3,T=P)
FORTRAN(I=INTGS,B=INTGB/100,O=LB)
LOAD(INTGB,CN=INTEGO,1000
,GRLP=*TMVEC,*PRIM,*EBUF,*TURB,*VPROP,*UBUF,*BOUND,*STEP
,GRLP=*XBUF1,GRLP=*XBUF2,GRLP=*TAU,*QPNOD,*SECORD
,GRSP=*CNTRL,*VCOEF,*TDATA,*VECP,*SQ,*PM,*SUBSBC,*USER
,*STEPI,*QPCOM,*QPPRNT,*CVGCOM,*AXSYM
,GROL=*Q3MAP)
INTEGO.
TOAS(Z=FILE22)
DAYFILE(STRDAY)
EXIT.
TOAS(Z= FILE22)
```

```
DAYFILE(STRDAY)
EXIT.
/EOR
/READ,STRINP
/EOR
REVERT. FROM EXRNST
END OF FILE
?? Q
RUNSRC  IS A LOCAL FILE
REVERT.   RUNGIM  COMPLETE
/
```

6.4 CURRENT STATUS OF RUNGIM

RUNGIM is operational and available to all remote terminal GIM code users who wish to use it. The MATRIX and GIMPLT modules of RUNGIM are not yet operational but will be in the near future. All RUNGIM users should occasionally check the "KNOWN BUGS" section of the instructions for any recently discovered problems in the code. Any problems, questions, comments, or suggestions should be directed to Michael Robinson, Lockheed-Huntsville, 205-837-1800, ext 384.

SECTION 6 REFERENCE

6.1. Spradley, Lawrence, W., and Mark Pearson, "GIM Code User's Manual for the STAR-100 Computer," NASA CR 3157, LMSC-HREC TR D568850, November, 1979.

7. CALCULATION OF THE FLOW FIELD ABOUT A WING-BODY CONFIGURATION

7.1 INTRODUCTION

Figure 7-1 depicts the wing body configuration for which the flow field was computed using the hyperbolic GIM/STAR computer code. The complete configuration is 0.355 m in length with a 70 deg swept wing. The upper wing surface is a flat plate aligned with the longitudinal center line of the body. The lower wing surface makes a 3-deg angle with the longitudinal center and a 8.712 deg normal to the wing leading edge. The forebody, which is 0.173 m in length, is described by a power law function. The cylindrical afterbody is 0.1822 m in length. The freestream conditions, which are shown in Fig. 7-1, represent a flight altitude of approximately 30,480 m. Since these calculations represent the initial attempt to compute a wing-body configuration with GIM all computations were done for zero angle of attack.

Section 7.2 presents a discussion of the development of the solution. An analysis of the final solution is given in Section 7.3. This discussion is divided into two parts: (1) the flow field over the forebody, and (2) the flow field over the afterbody.

7.2 DEVELOPMENT OF THE SOLUTION

7.2.1 Forebody Flow Field

As a matter of convenience, the computational domain was constructed with the input boundary beginning .003 m aft of the forebody nose. The stagnation region and leading edges could be treated with code, but it is not necessary here. We can easily use tables for these inputs. The outer boundary of the freestream was made to conform to the anticipated shape of the bow shock.

Since all computations were made for a pitch plane orientation (no yaw), a symmetry plane was employed. Figure 7-2 depicts the computational mesh on the body and symmetry plane. Figure 7-3 shows the body and outer surface grids while Fig. 7-4 presents the body, symmetry plane, and outer surface mesh points. The computational grid for a typical cross-plane is shown in Fig. 7-5. Equal nodal spacing was used in the cross-plane directions, while a mild stretching function was employed in the main flow direction in order to obtain the desired resolution of the bow shock in the nose region.

7.2.2 Afterbody Flow Field

The computational domain for the afterbody was constructed to adequately describe the wing without becoming prohibitatively large. As with the forebody a symmetry plane was employed, and since base flow calculations are not feasible with a hyperbolic or parabolic method, the computational domain extends only to the wing trailing edge. A sharp wing leading edge was employed and the wing-body juncture has been modeled as a distinct three-dimensional external corner. Figures 7-6 and 7-7 present perspective views of the computational mesh on the wing and body surfaces. Figure 7-8 and 7-9 show the computational mesh on the upper and lower wing surfaces. Figure 7-10 is a side view of the mesh on the afterbody. The thickness of the wing can be seen in this view. In Fig. 7-11 a three-dimensional view of the computational mesh on the wing, the body, and the outer surfaces of the computational domain is shown. Figure 7-12 is a plot of the complete mesh on the final three computational planes.

In order to accurately describe the wing, the upper and lower wing surface mesh points had to be at the same spatial location on the initial plane. This results in several dual points occuring at the intersection of the wing with the body (See Fig. 7-13). As Figs. 7-14 through 7-19 depict, the wing mesh spacing becomes larger as the trailing edge is approached.

Since the computational mesh was very dense on the forward 20% of the wing, the finite difference analogs had to be examined closely in this area.

Because of the presence of the dual points, the analogs on the initial plane are somewhat suspect. However, since the flowfield values on the initial plane are input and not computed, these analogs were not used directly in the calculations. The analogs on the second plane, which are influenced by the initial plane geometry, were examined carefully and are correct.

7.3 RESULTS AND DISCUSSION

7.3.1 Forebody Flow Field

A 10830-node computational grid with 361 nodes in a cross-plane was used to compute the flow field about the forebody. The solid boundaries were treated in an inviscid manner. The bow shock was attached to the nose and held fixed for the initial plane. For subsequent planes the shock location and jump conditions were determined from the governing equations. These computations required 210.0 CPU seconds on the CYBER 203 to obtain the final solution. Figure 7-20 shows the velocity vectors on the symmetry plane, and Fig. 7-21 provides a magnified view of the nose region. The bow shock lies close to the body near the nose, and dissipates as the flow expands back to freestream near the base. Figure 7-22 depicts the static pressure contours over the symmetry plane. Figure 7-23 shows the cross-plane static pressure contours on a plane that is .022 m from the nose. This figure depicts the near proximity of the bow shock to the missile body. Figure 7-24 presents the static pressures on the final cross-plane. The bow shock has dissipated by this time, and the flow has expanded back to freestream.

7.3.2 Afterbody Flow Field

The computational grid for the afterbody contained 9796 nodes with 316 nodes in each cross plane. The solid boundaries with the exception of the wing leading edge and the wing-body juncture point were treated in an inviscid manner. The wing leading edges were initialized to inviscid post shock values and held fixed during the calculations. This was done for convenience only since the downstream shock conditions can be determined from shock tables. Tangency conditions could also have been used; however, this would require additional mesh points to permit the formation of the shock. The wing-body juncture presented a unique problem and was dealt with in a special manner. Inviscid corner flow conditions were used at the wing-body juncture. As subsequent planes were computed the mesh spacing in the vicinity of the corner became larger and larger. The divergence of the mesh is due to the incompatibility of the grid structure over different components of aircraft configurations. The solution to this problem is to use grid imbedding techniques to achieve grid resolution where it is required; however, time and budget restrictions did not permit this refinement. Inviscid boundary conditions could normally be applied if the mesh were tight enough. Since this refinement was not possible, the corner flow field conditions were held fixed in time at the final values of the first five corner points. The first five points along the wing/body juncture appeared to be fine because they were not affected by the mesh divergence problem. The implementation of the above two "special" boundary conditions did not create any perturbations in the computed flow field. These computations required 334 CPU seconds on the CYBER 203 to obtain the final solution. For more details of the computational procedure utilized in the wing/body problem please refer to Ref. 7-1.

Cross-plane velocity vector plots for planes 5, 10, 15, 20, 25 and the final plane are presented in Figs. 7-25 to 7-30, respectively. The fact that the velocity vectors on the lower wing surface seem to be coming directly out of the wing is an illusion created by the fact that only cross-plane components are plotted. Each of these vectors has a much larger component directed into the plane of the plots. These plots depict the presence of the formation of vortices particularly from the lower wing surface.

Static pressure contours are presented for planes 5, 10, 15, 20, 25 and the final plane in Figs. 7-31 through 7-36, respectively. A high pressure region is produced on the lower wing surface while the upper surface remains virtually at freestream conditions.

Figures 7-37 and 7-38 show the velocity vectors and static pressure contours, respectively on the wing upper surface for the grid of Fig. 7-8. Figures 7-39 and 7-40, respectively, depict the same type of plots on the wing lower surface for the grid of Fig. 7-9. The velocity vectors and pressure contours are shown in Figs. 7-41 and 7-42, respectively, for the grid of Fig. 7-10.

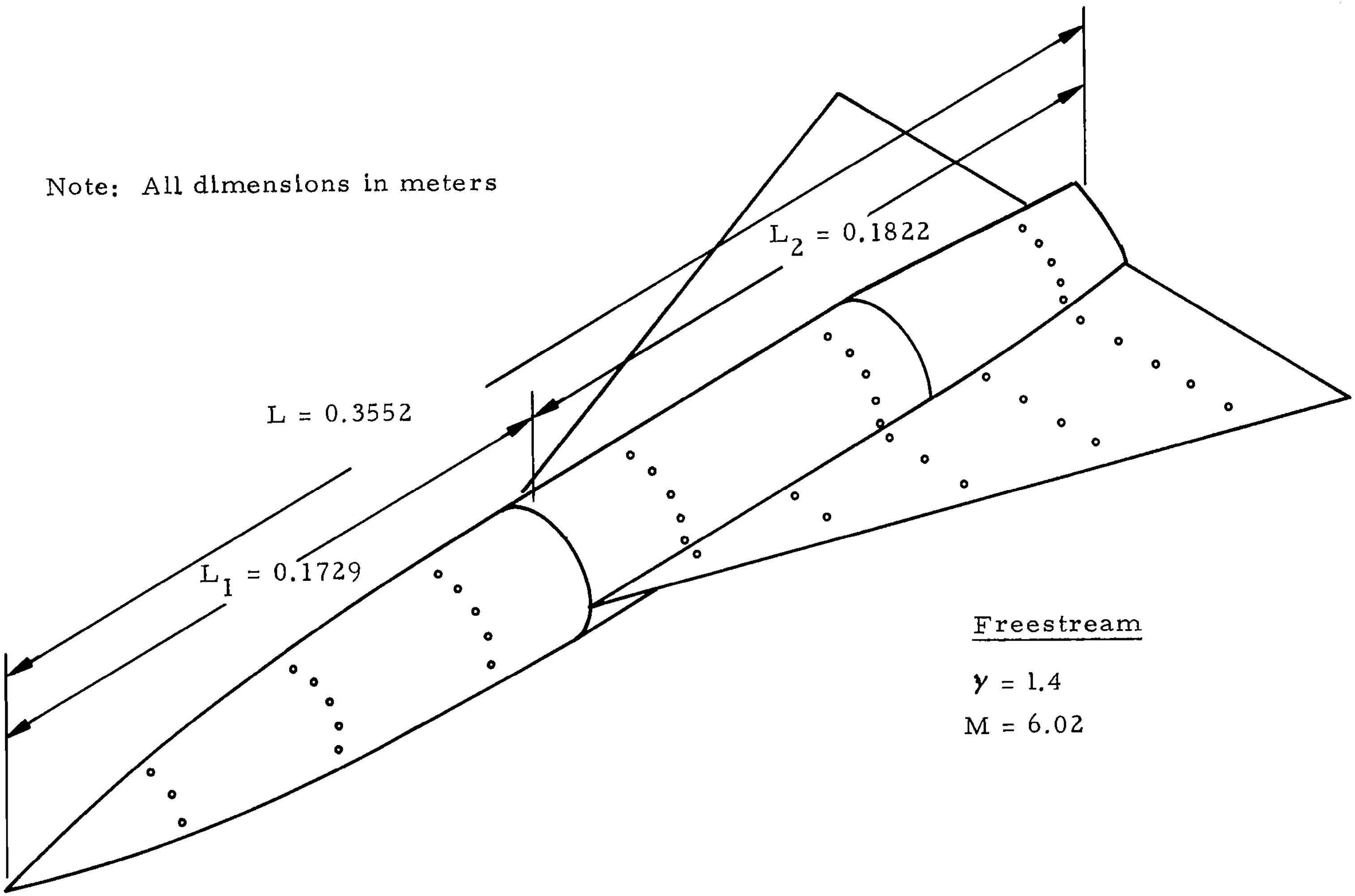
7.3.3 Comparison with Experimental Data

Experimental data for this configuration was obtained from the NASA-Langley Research Center.

Figures 7-43 and 7-44 present a comparison between GIM results and this experimental data for the wing/body configuration. As Fig. 7-43 depicts, the GIM results compare extremely well with the experimental data with GIM predicting slightly more over-expansion than the data. The spanwise pressure distribution in Fig. 7-44 shows good comparison between GIM and experiment. The slight deviation between the GIM predictions and the experimental data near the wing leading edge is due to the imposition of the fixed post shock boundary condition on the wing leading edge.

SECTION 7 REFERENCE

7.1. Xiques, K. E. et al., "Computation of Three-Dimensional Inviscid Flow over a Hypersonic Missile Configuration Using the GIM Code," AIAA Paper 82-0248, 20th Aerospace Sciences Meeting, Orlando, Fla., 11/14 January 1982.



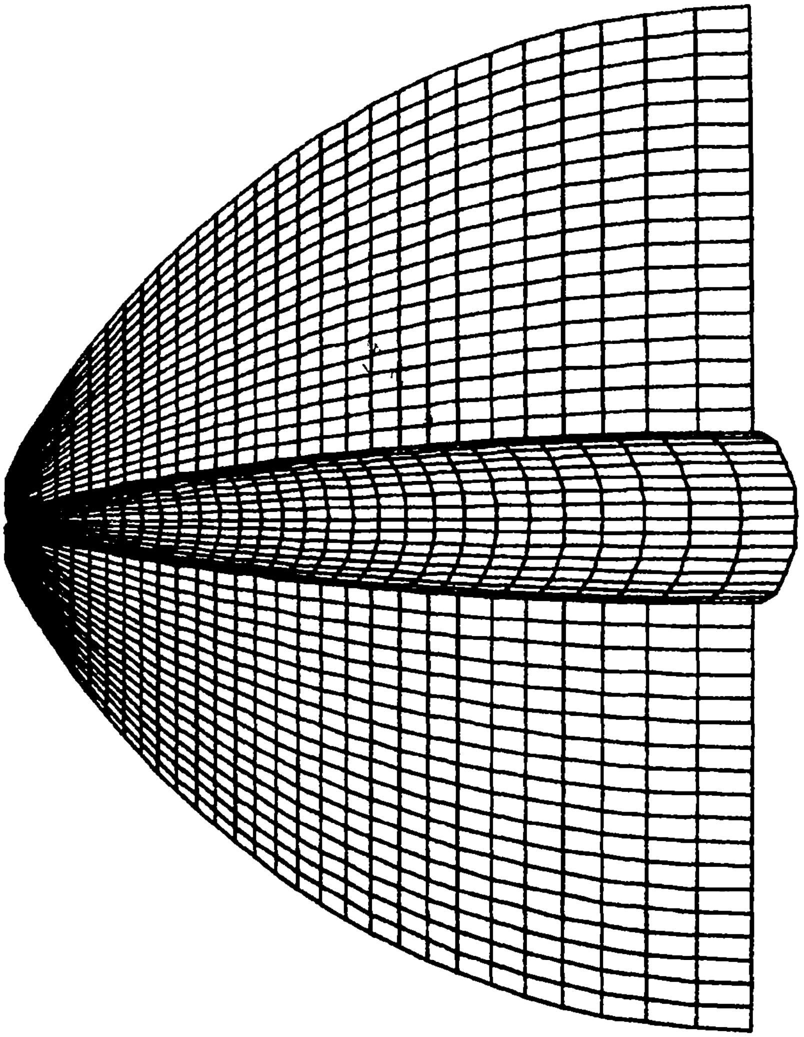
Note: All dimensions in meters

Freestream

$\gamma = 1.4$

$M = 6.02$

Fig. 7-1 - Wing-Body Configuration



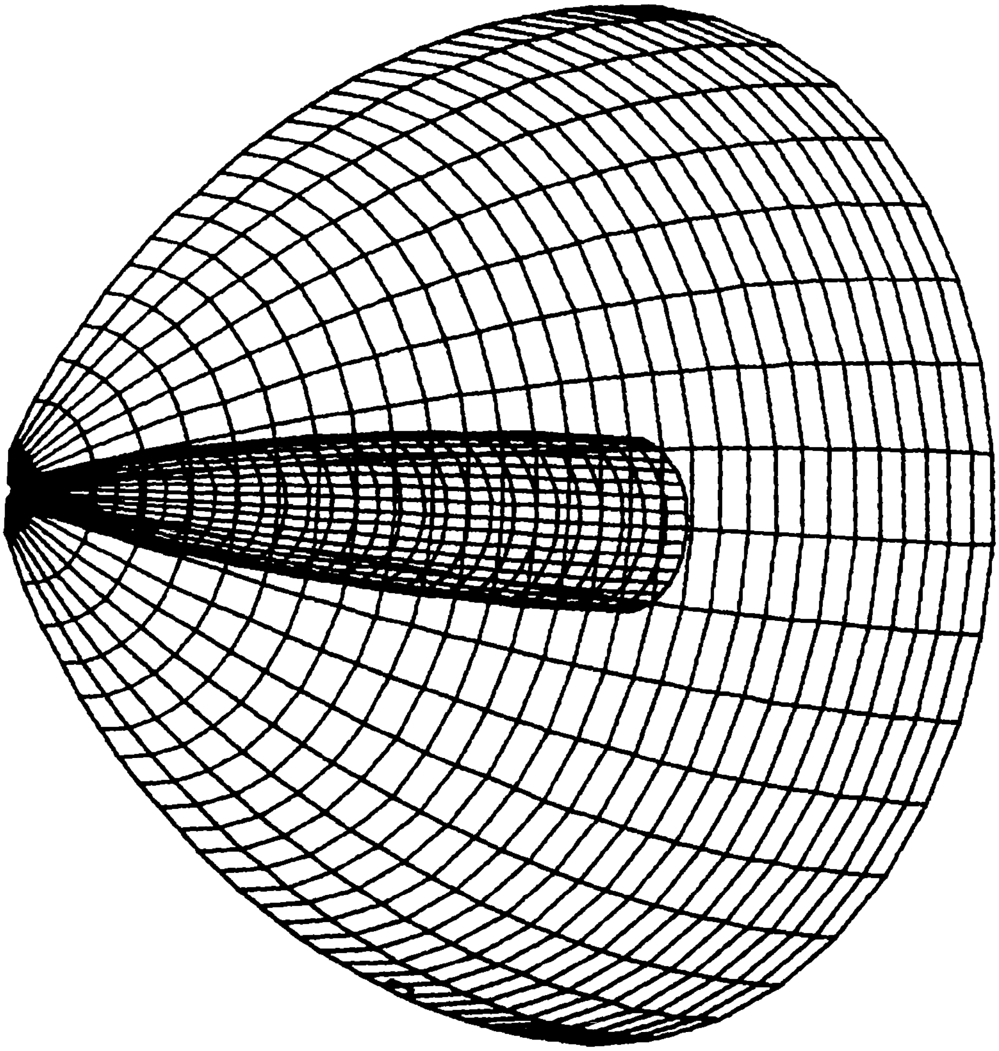


Fig. 7-3 - Forebody Computational Mesh (Body Surface and Outer Surface)

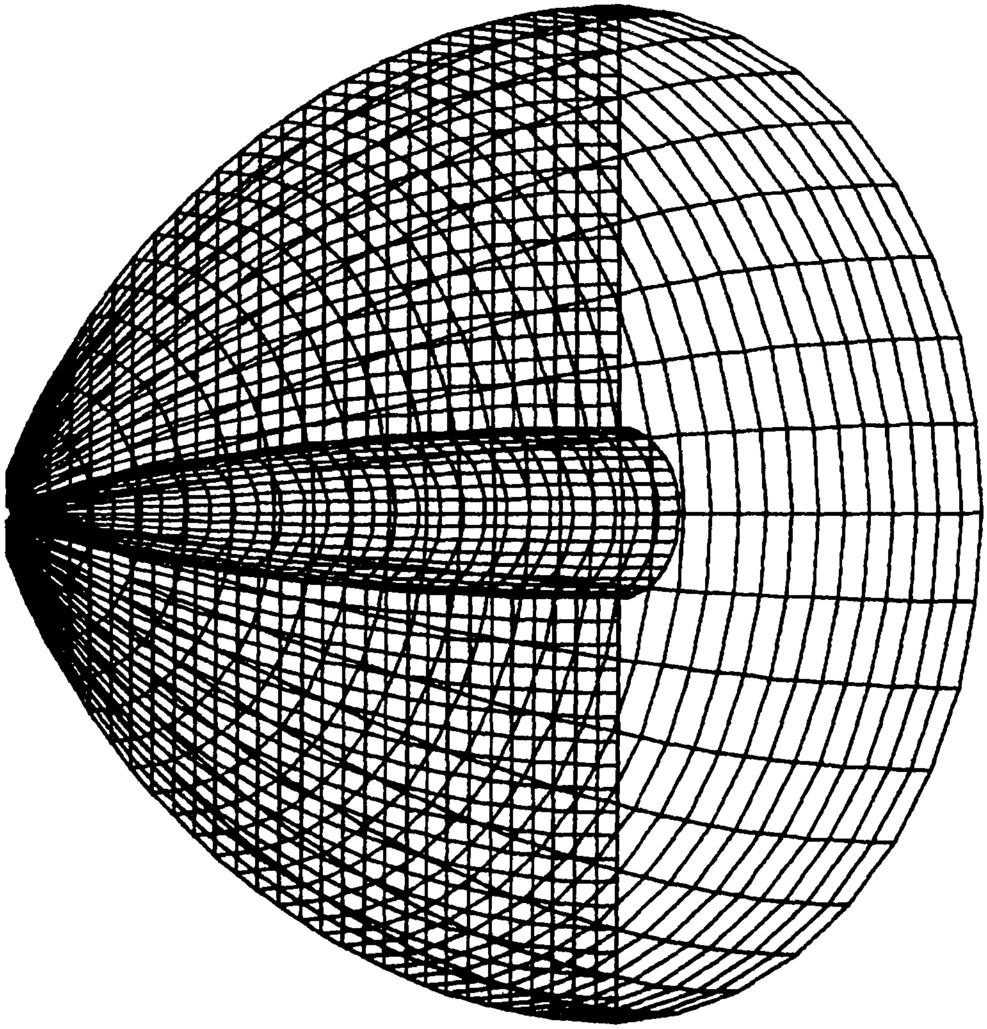


Fig. 7-4 - Forebody Computational Mesh (Body Surface, Symmetry Plane, and Inter Surface)

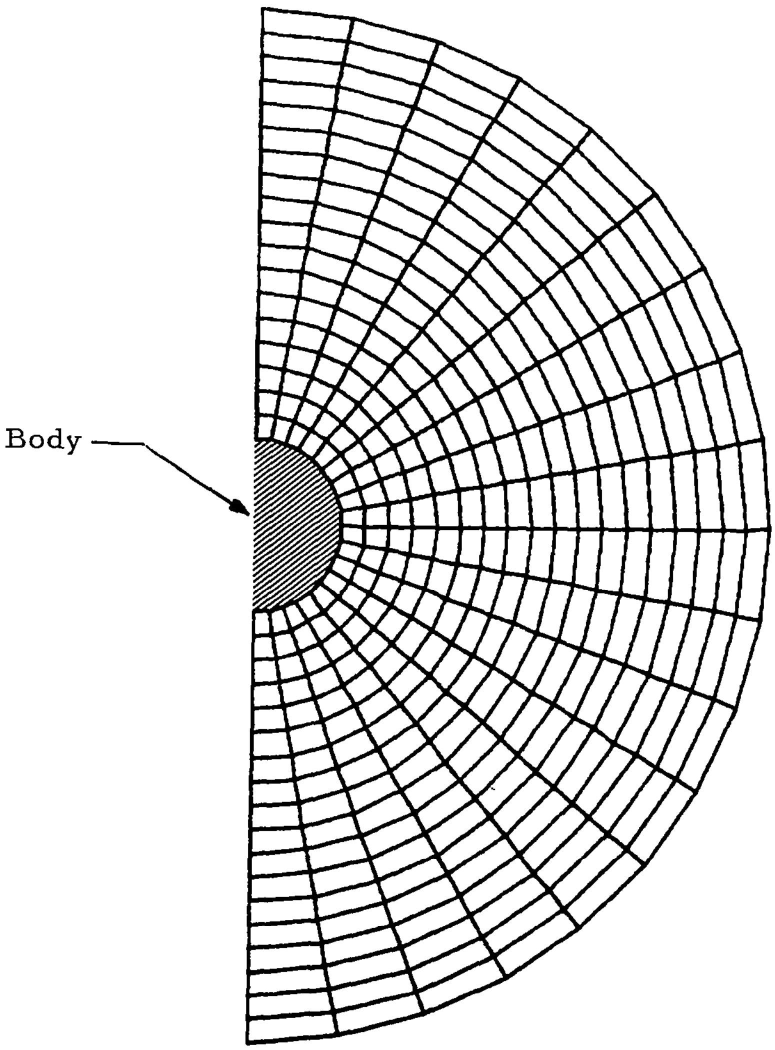


Fig. 7-5 - Computational Mesh for a Typical Cross-Plane

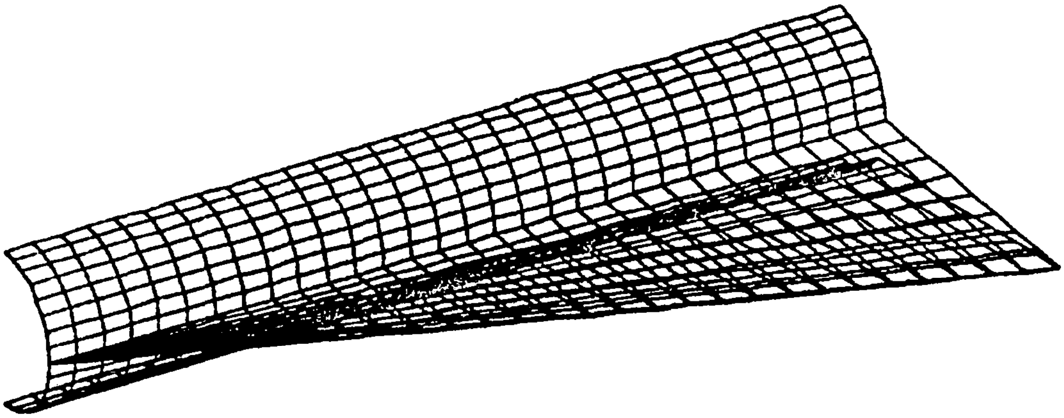


Fig. 7-6 - Computational Mesh on Wing-Body Surface

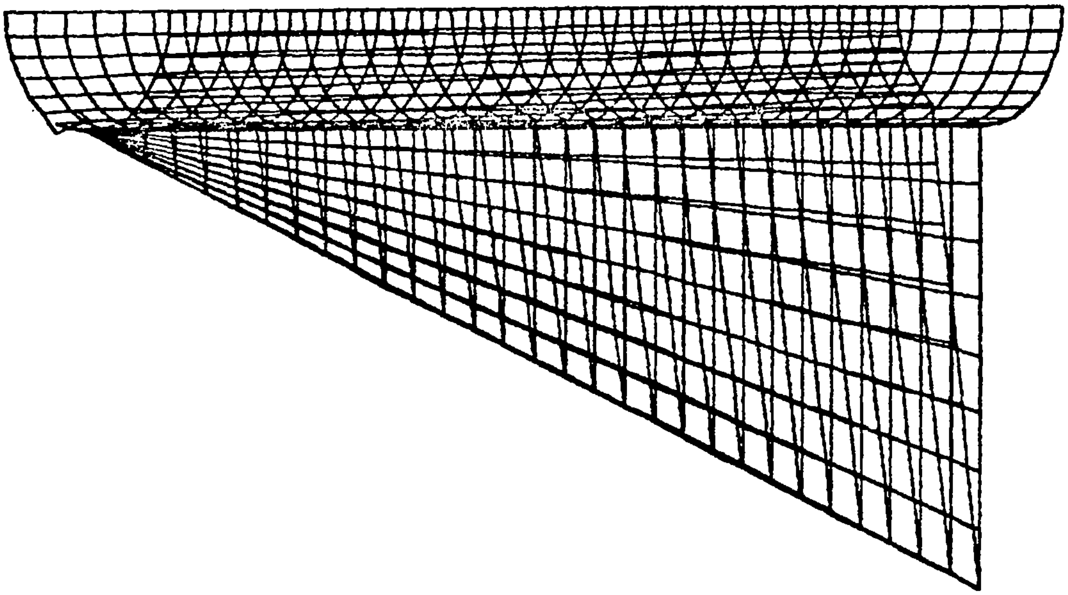


Fig. 7-7 - Computational Mesh on Wing-Body Surface (aft view)

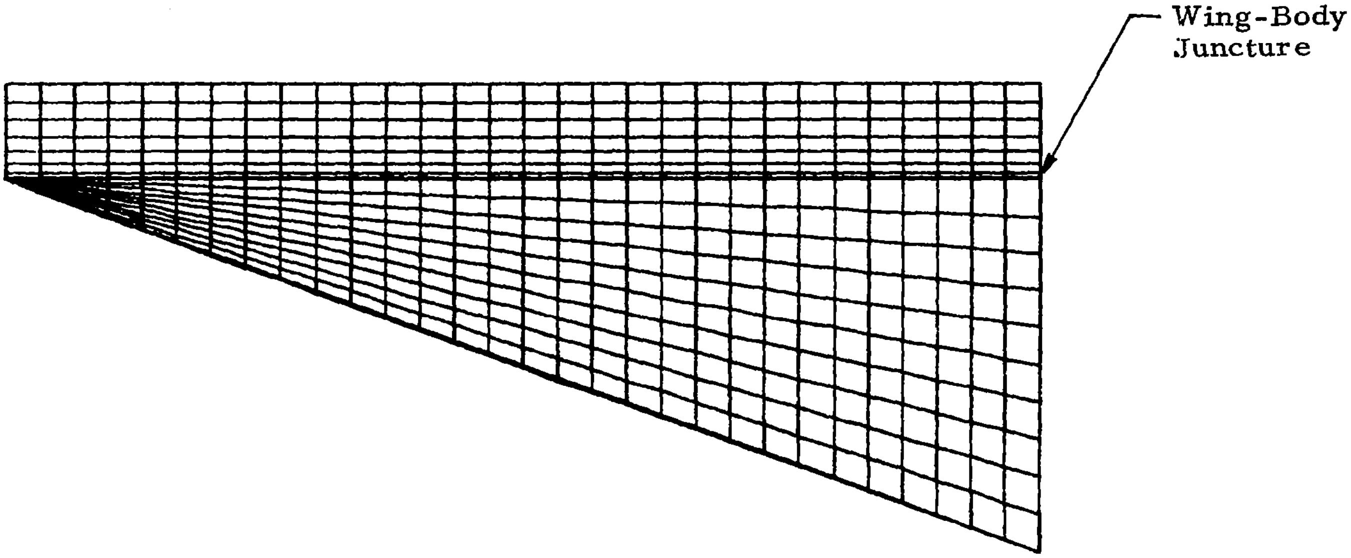


Fig. 7-8 - Computational Mesh on the Upper Wing Surface Viewed from Above

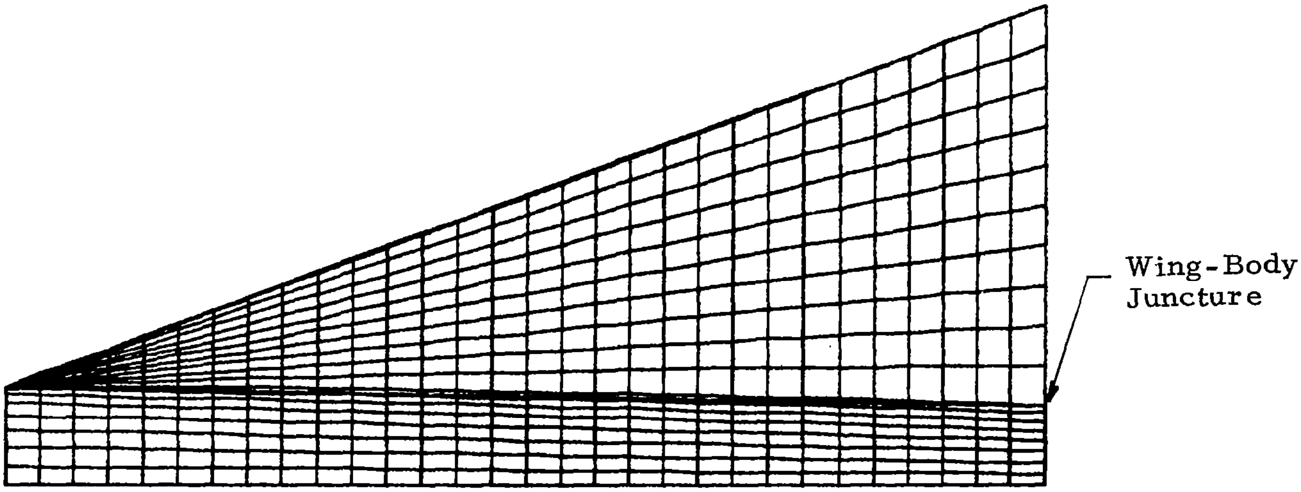


Fig. 7-9 - Computational Mesh on the Lower Wing Surface Viewed from Below

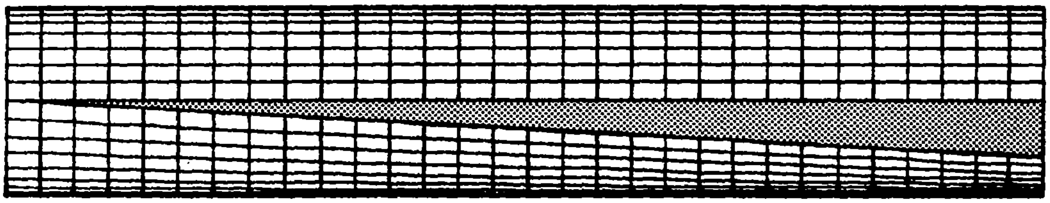


Fig. 7-10 - Computational Mesh on Afterbody Surface

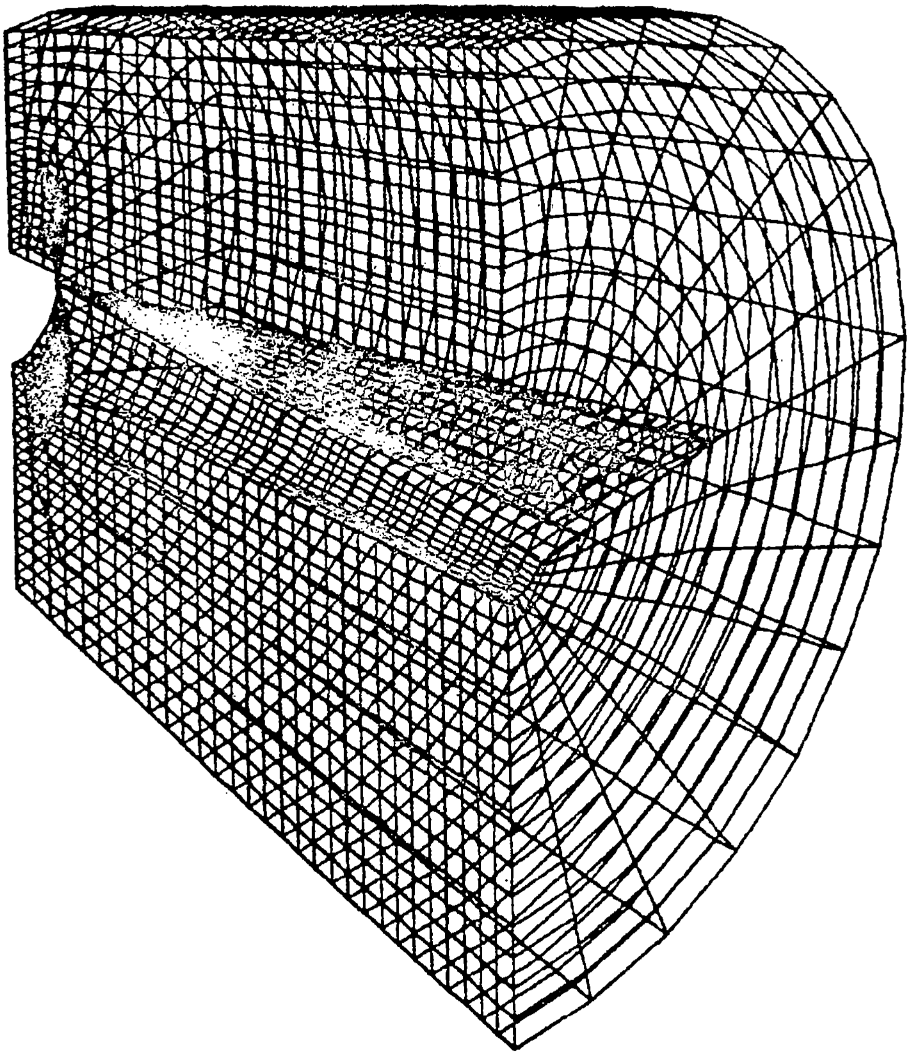


Fig. 7-11 - Computational Mesh on the Wing-Body and Outer Surfaces (Aft View)

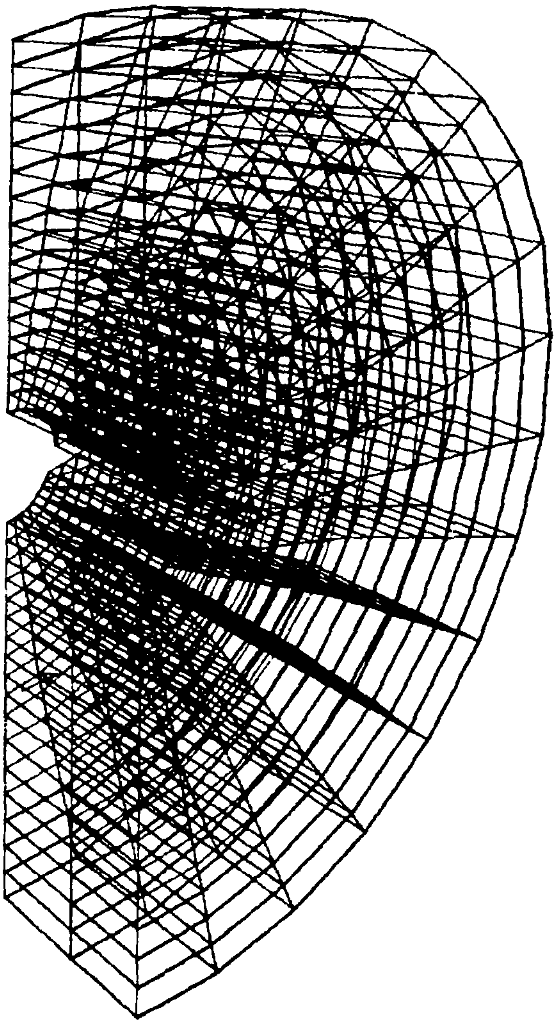


Fig. 7-12 - Three-Dimensional Grid for Final Three Computation Planes (Aft View)

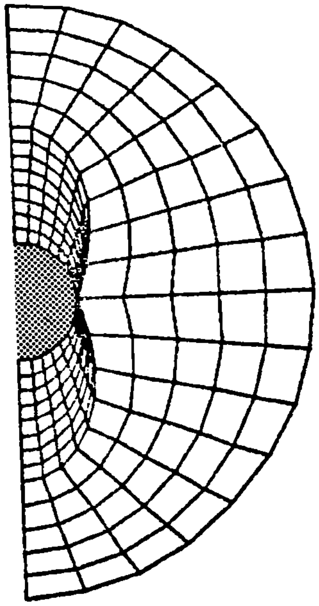


Fig. 7-13 - Afterbody Cross-Plane Grid, Initial Plane, X = 0.1730 m

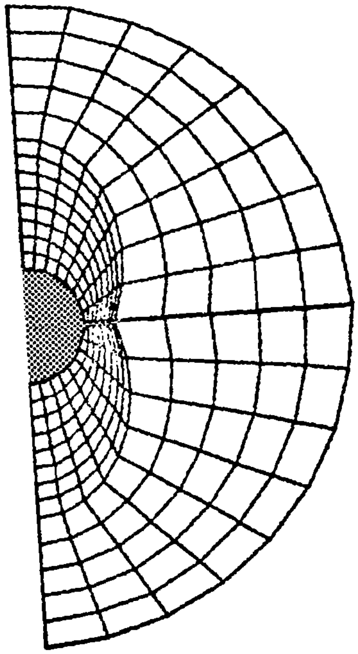


Fig. 7-14 - Afterbody Cross-Plane Grid, Plane 5, X = 0.1973 m

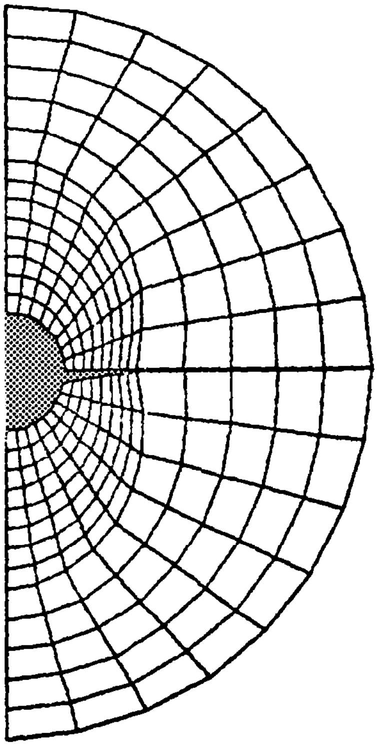


Fig. 7-15 - Afterbody Cross-Plane Grid, Plane 10, X = 0.2276 m

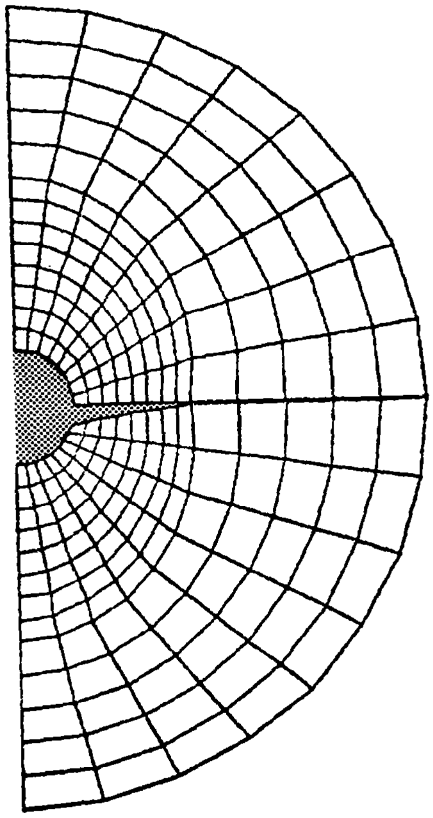


Fig. 7-16 - Afterbody Cross-Plane Grid, Plane 15, X = 0.2641 m

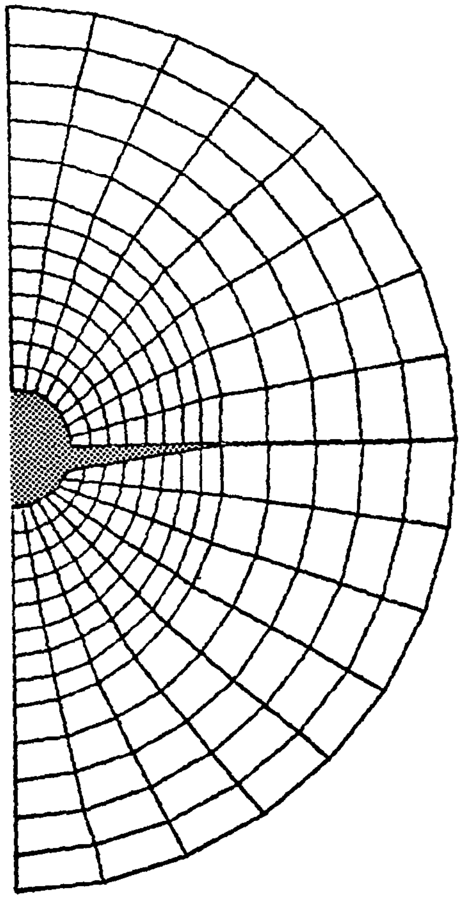


Fig. 7-17 - Afterbody Cross-Plane Grid, Plane 20, X = 0.2884 m

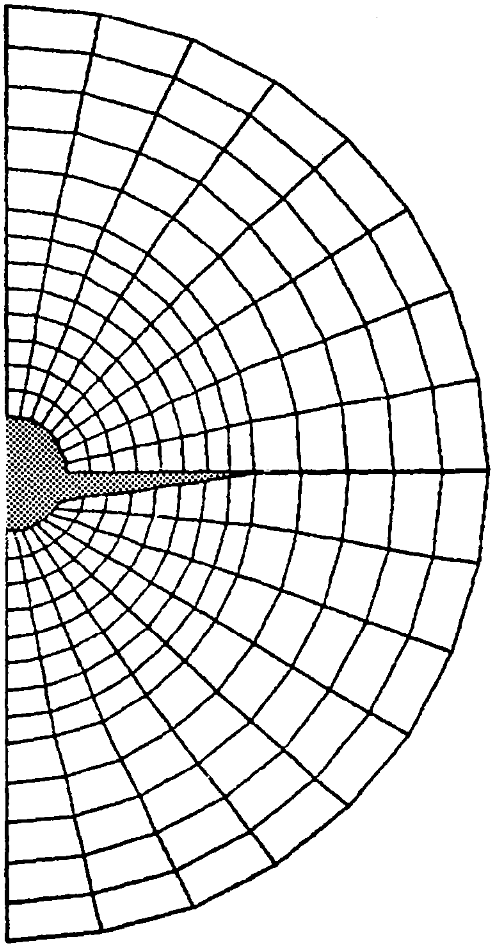


Fig. 7-18 - Afterbody Cross-Plane Grid, Plane 25, X = 0.3188 m

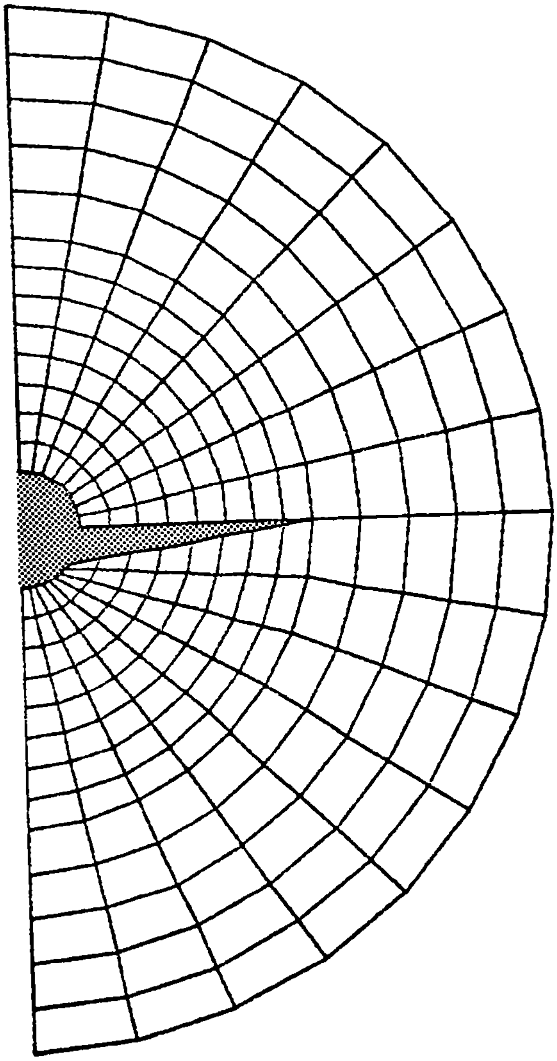


Fig. 7-19 - Afterbody Cross-Plane Grid, Final Plane, X = 0.3551 m

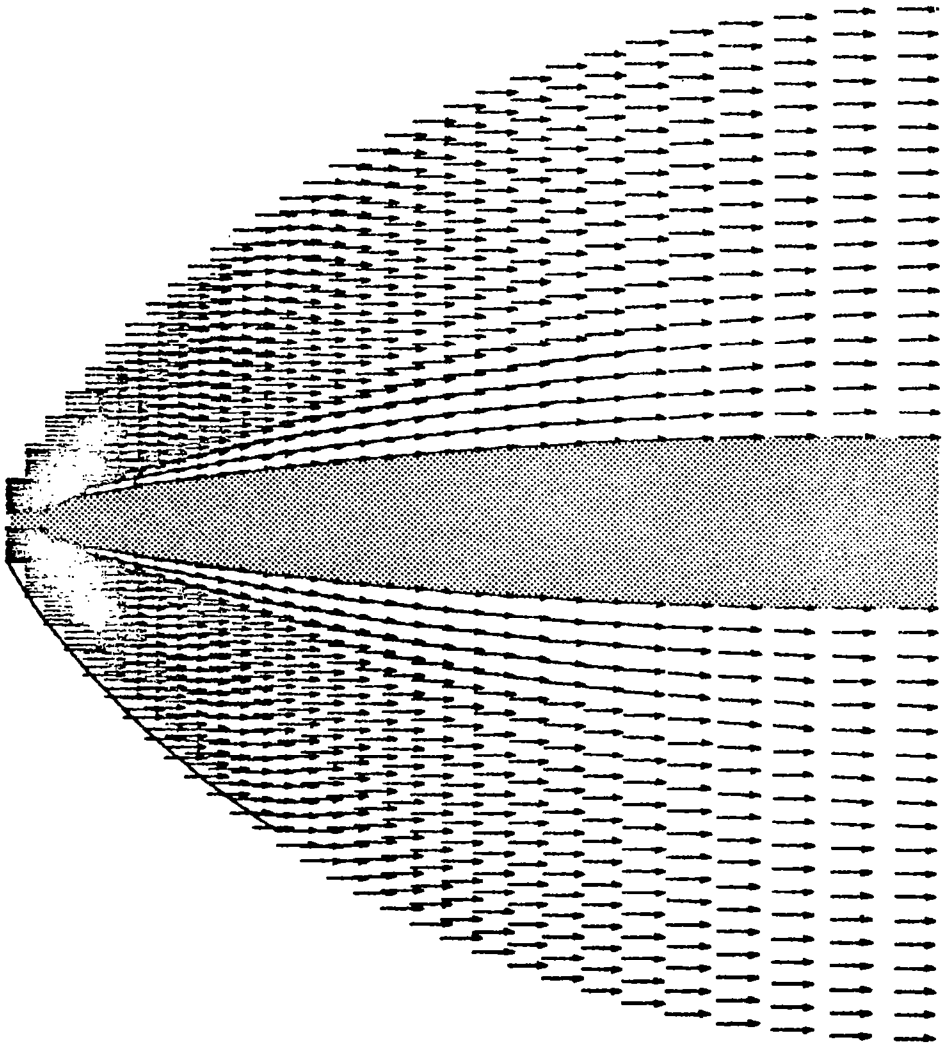


Fig. 7-20 - Forebody Velocity Vectors (Symmetry Plane)

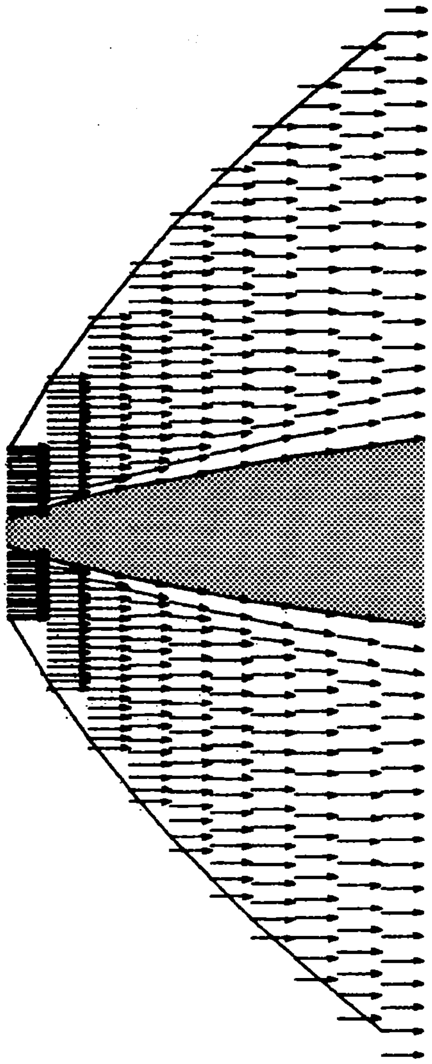


Fig. 7-21 - Forebody Velocity Vectors, Nose Region (Symmetry Plane)

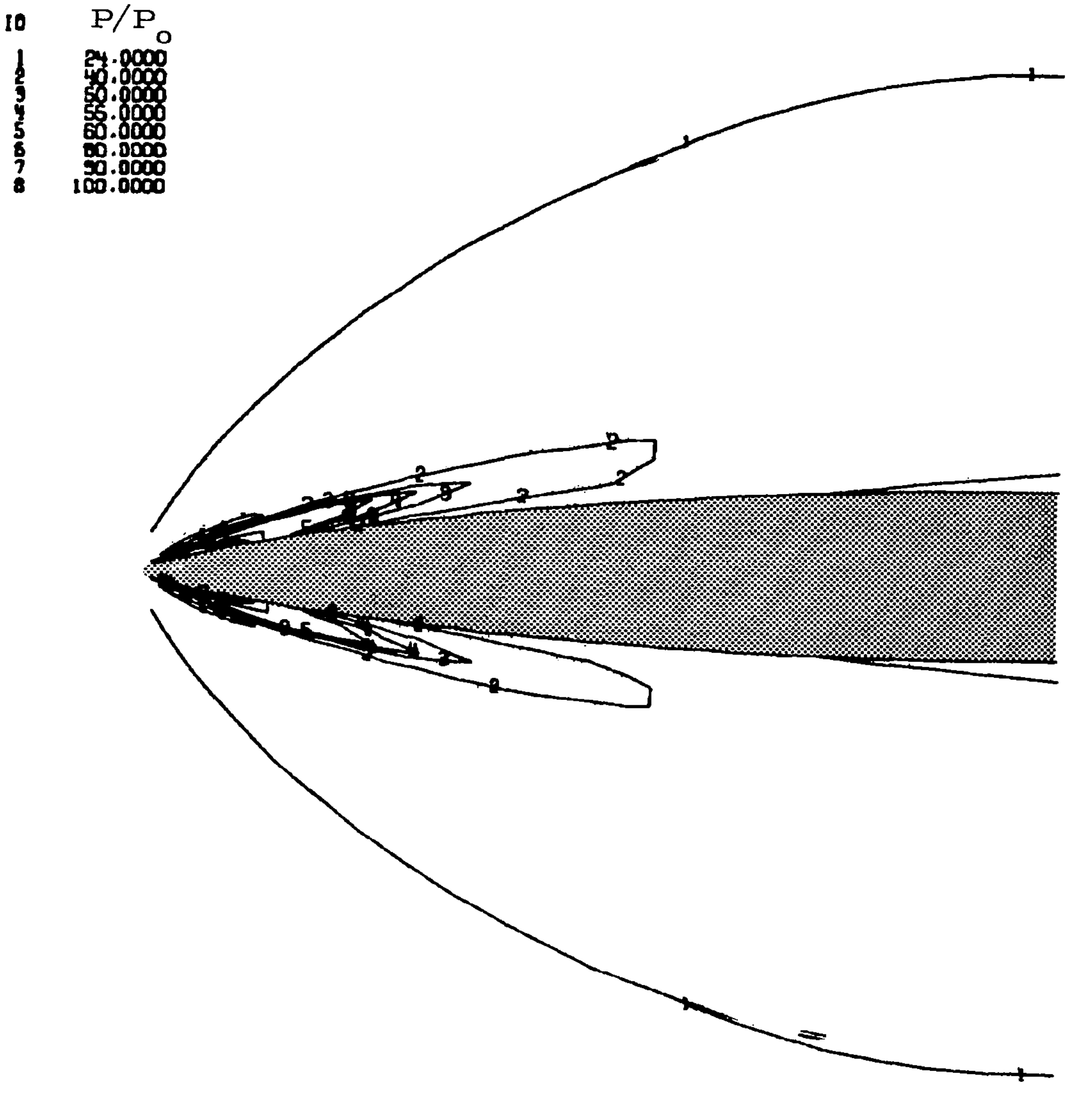


Fig. 7-22 - Static Pressure Contours (Symmetry Plane)
(P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

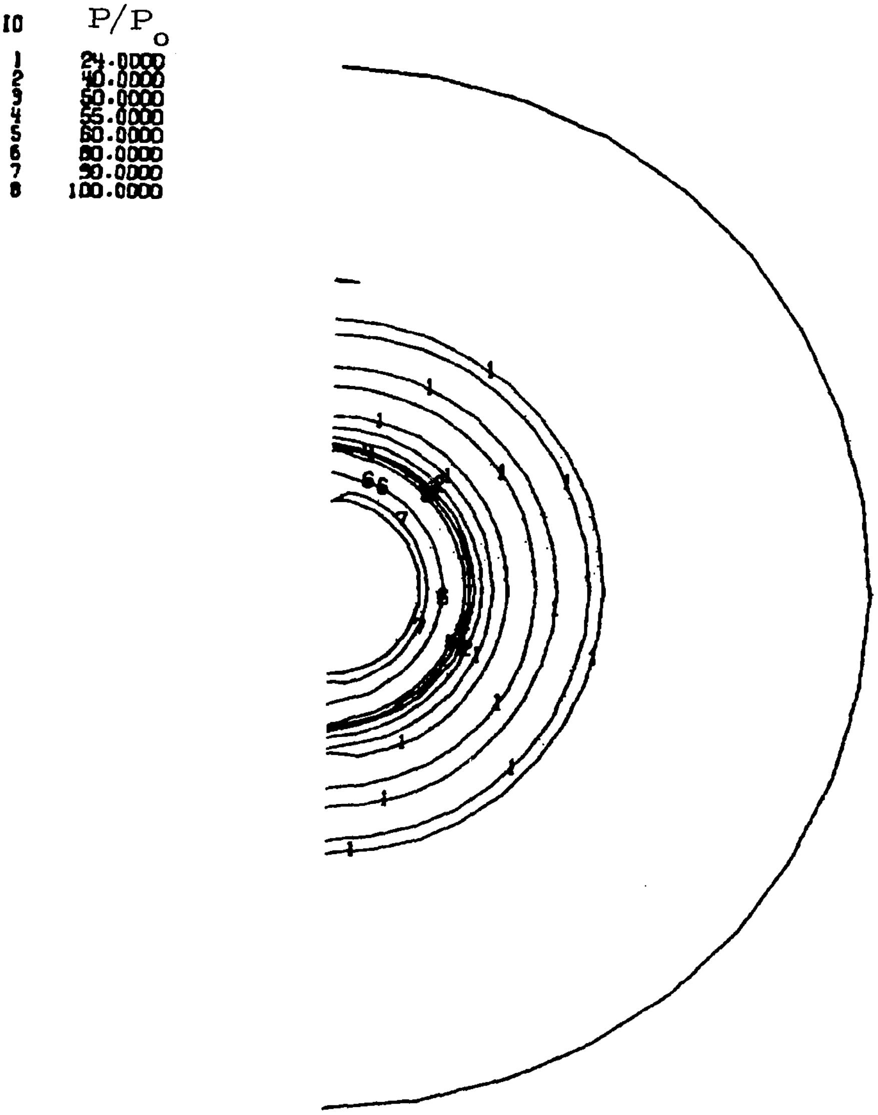


Fig. 7-23 - Cross-Plane Static Pressure Contours (X = 0.02178 m) (P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

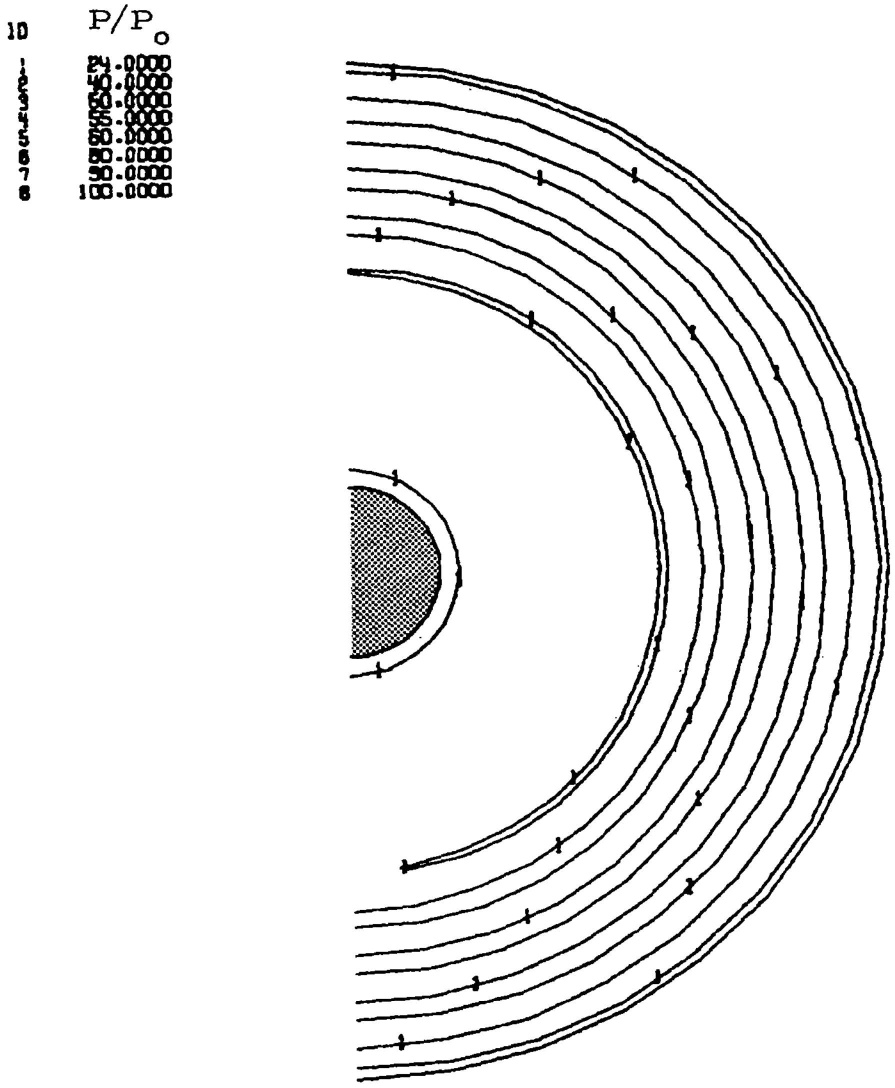


Fig. 7-24 - Cross-Plane Static Pressure Contours (Final Plane)
(P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

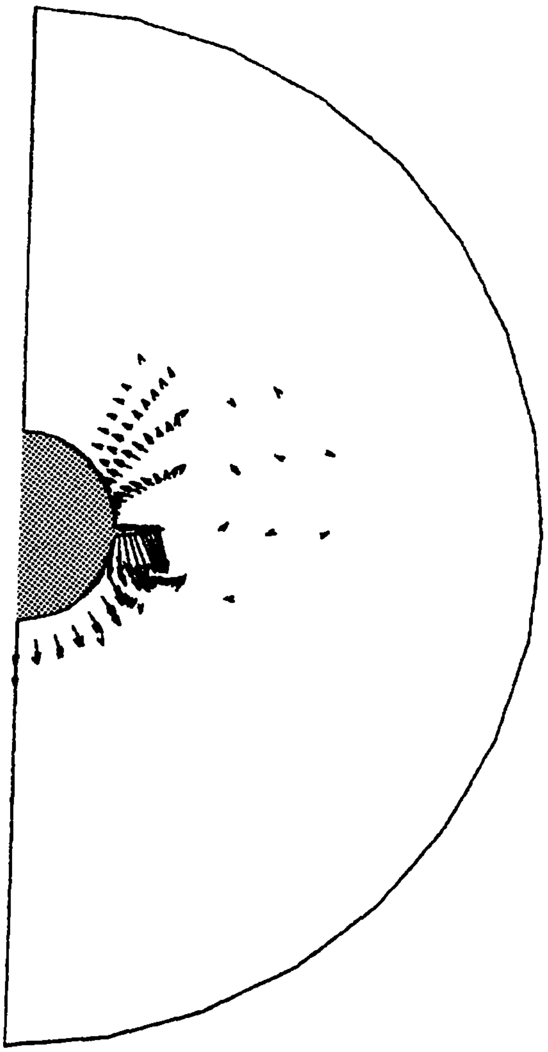


Fig. 7-25 - Afterbody Cross-Plane Velocity Vector Plots (Plane 5)
X = 0.1972 m

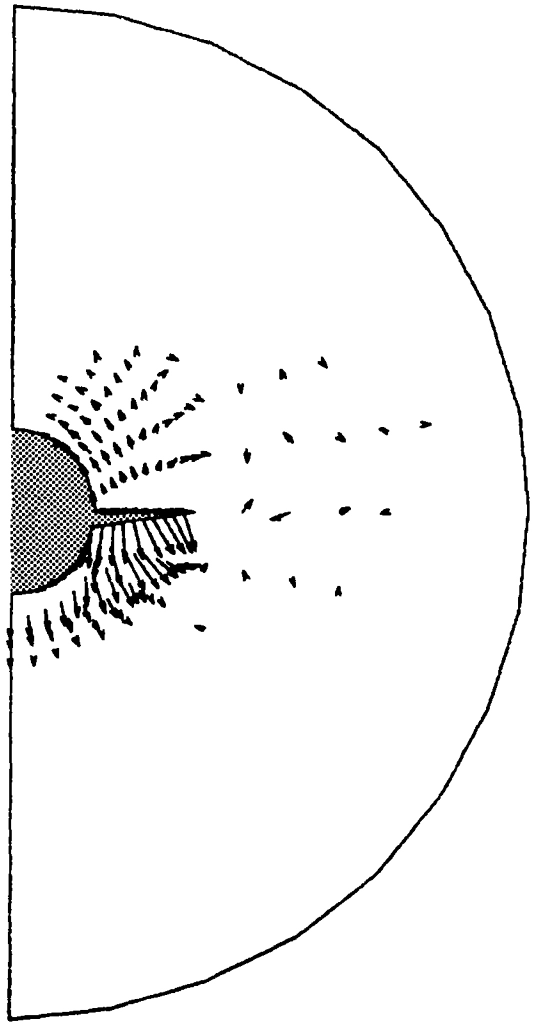


Fig. 7-26 - Afterbody Cross-Plane Velocity Vector Plots (Plane 10)
X = 0.2276 m

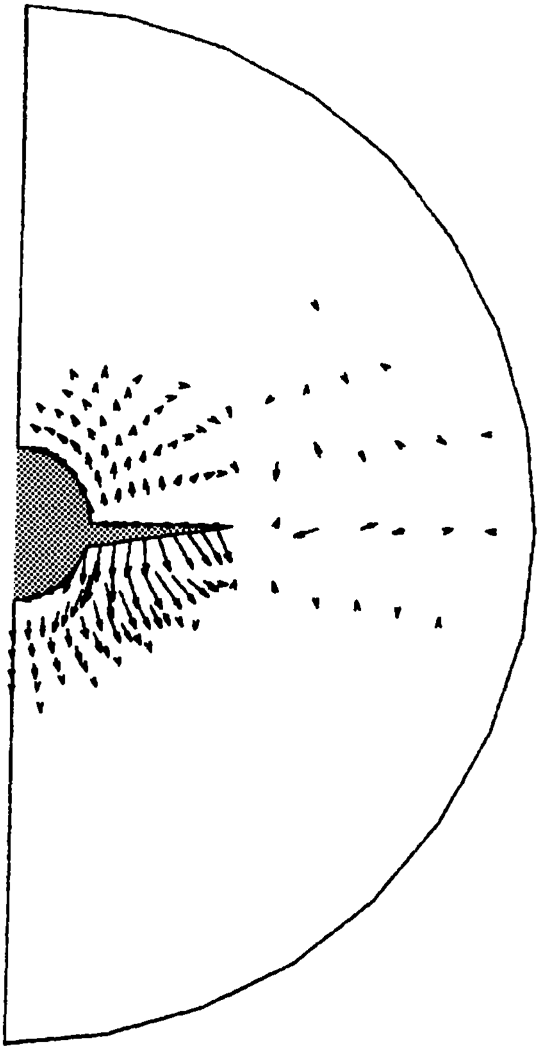


Fig. 7-27 - Afterbody Cross-Plane Velocity Vector Plots (Plane 15)
X = 0.2641 m

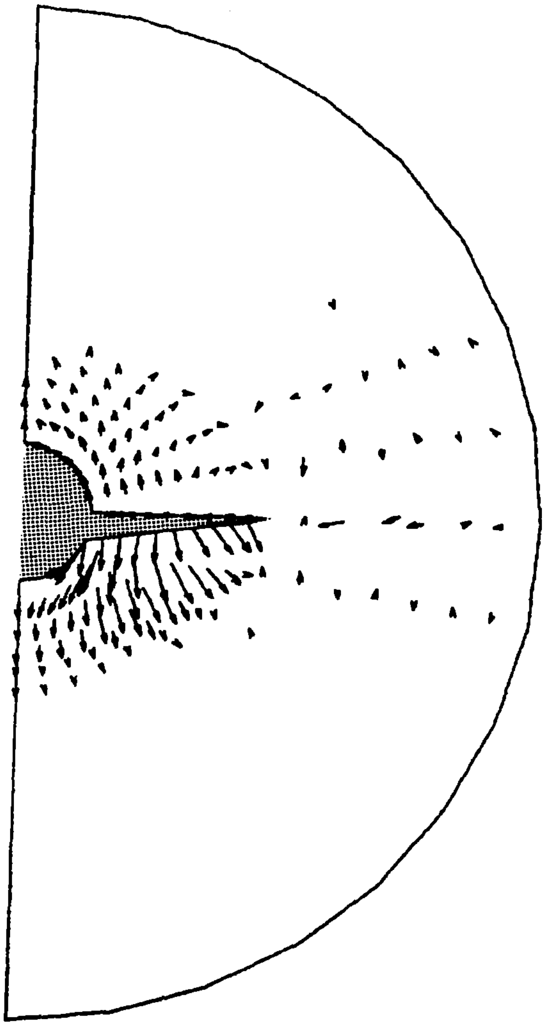


Fig. 7-28 - Afterbody Cross-Plane Velocity Vector Plots (Plane 20)
X = 0.2883 m

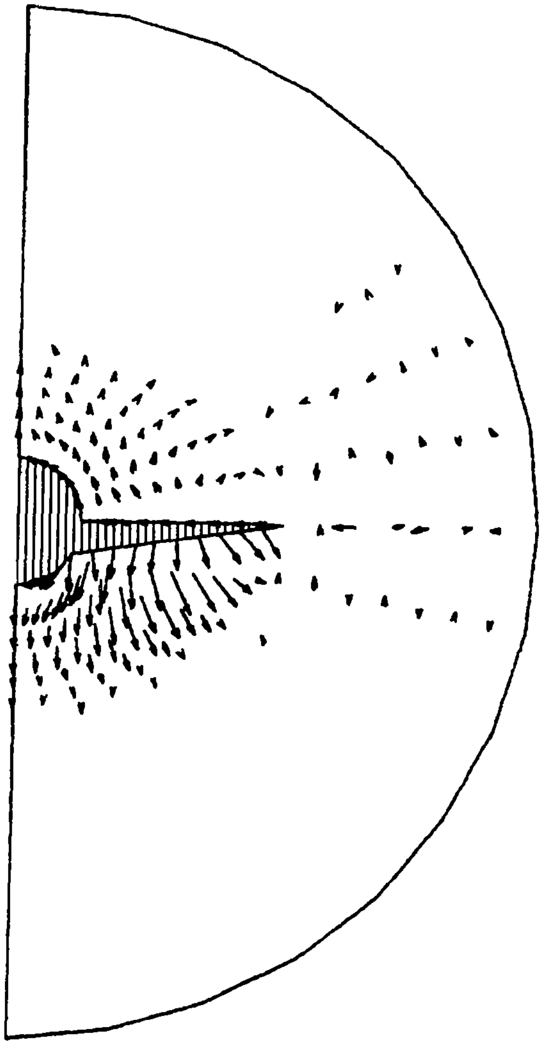


Fig. 7-29 - Afterbody Cross-Plane Velocity Vector Plots (Plane 25)
X = 0.3188 m

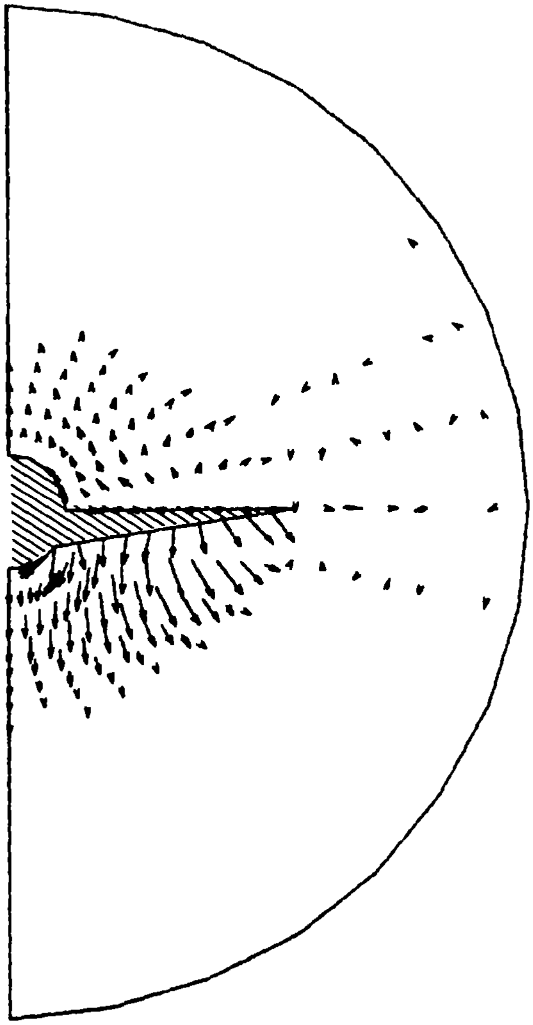


Fig. 7-30 - Afterbody Cross-Plane Velocity Vector Plots (Final Plane)
X = 0.3551 m

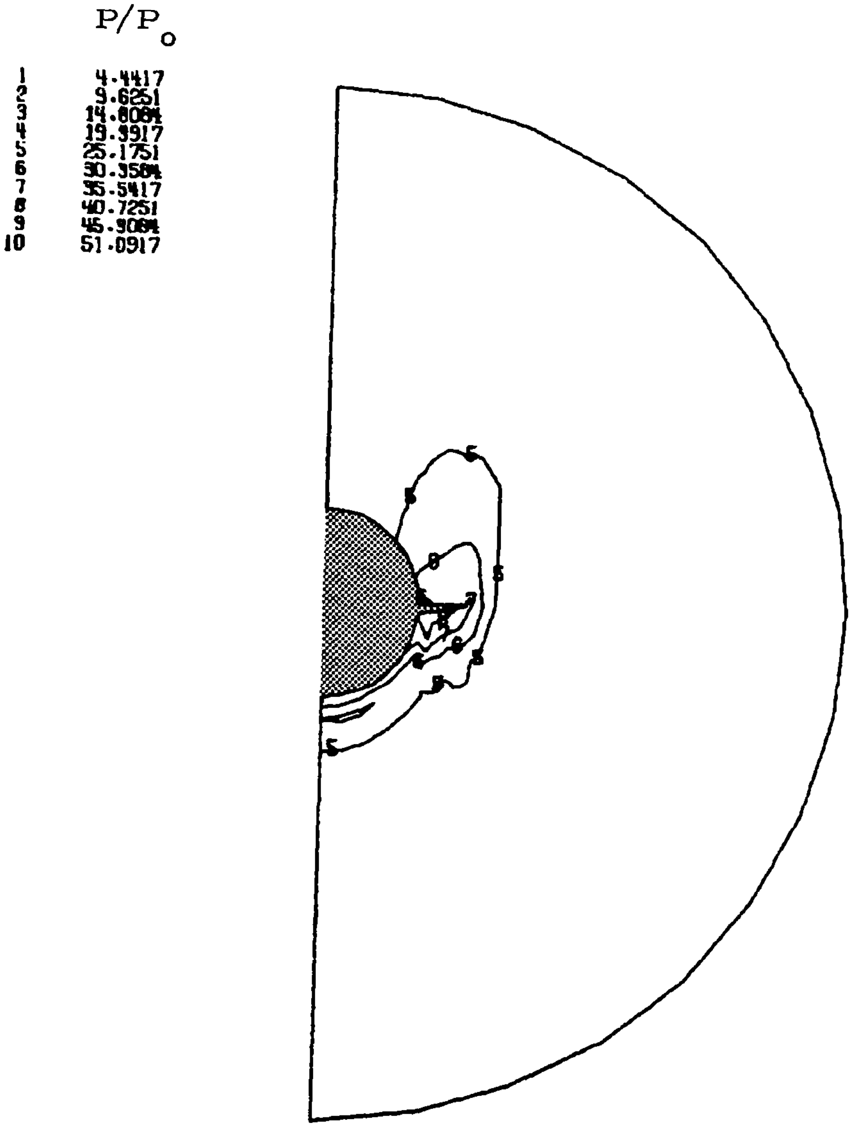


Fig. 7-31 - Static Pressure Contours for Plane 5, X = 0.1973 m (P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

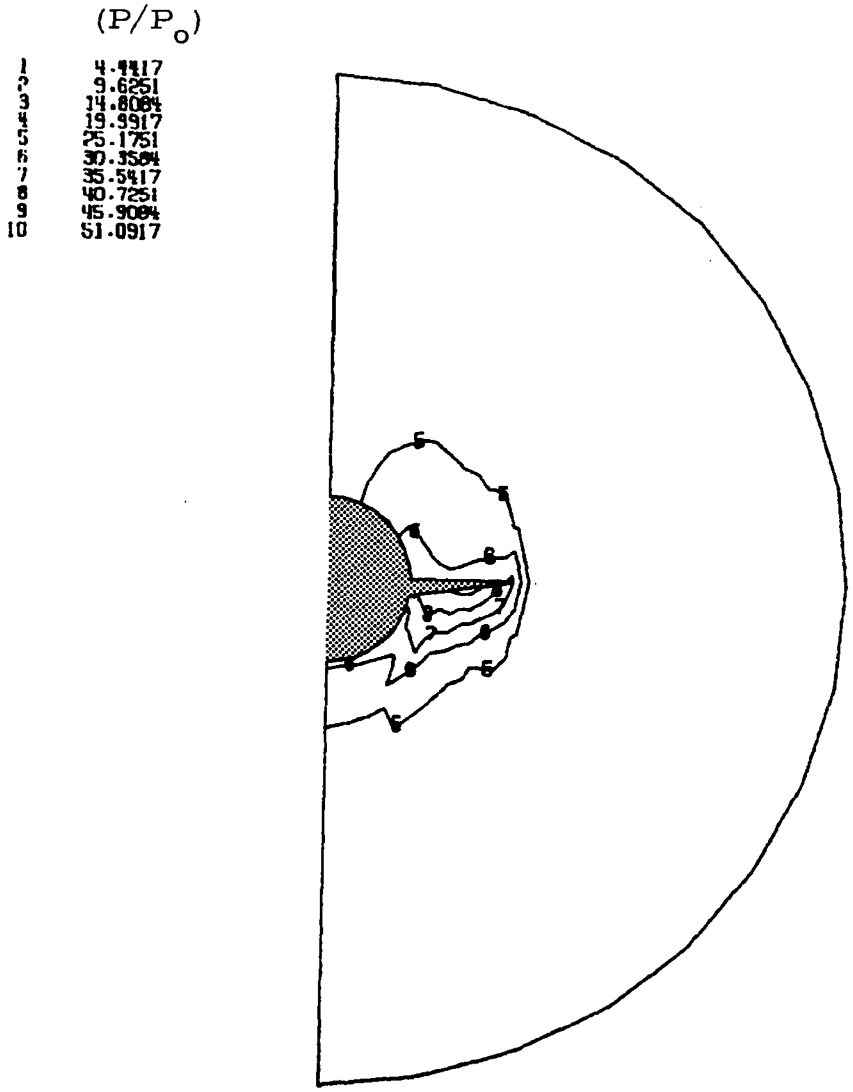


Fig. 7-32 - Static Pressure Contours for Plane 10, X = 0.2276 m
(P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

ID	P/P_o
1	4.4417
2	9.6251
3	14.8084
4	19.9917
5	25.1751
6	30.3584
7	35.5417
8	40.7251
9	45.9084
10	51.0917

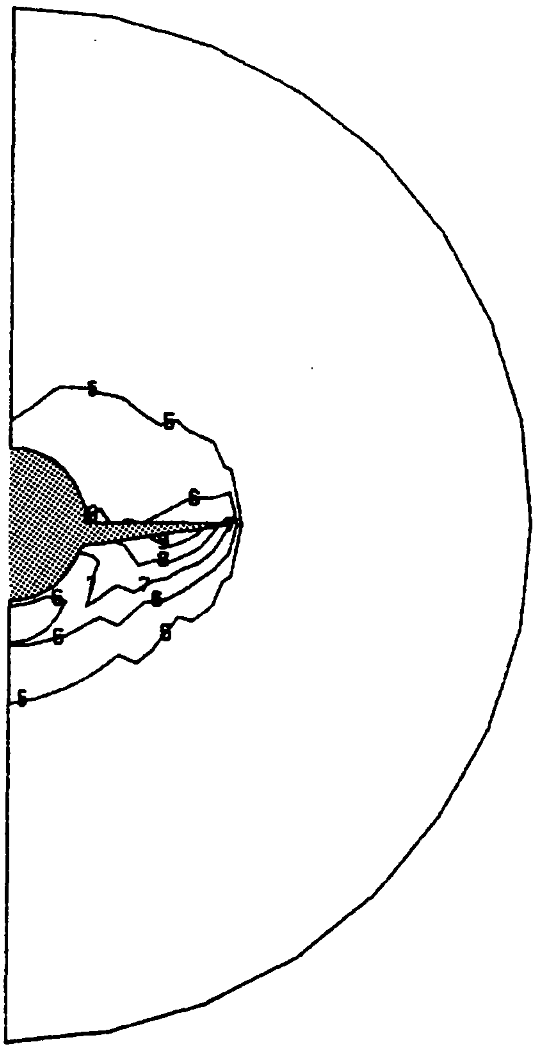


Fig. 7-33 - Static Pressure Contours for Plane 15, X = 0.2641 m
(P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

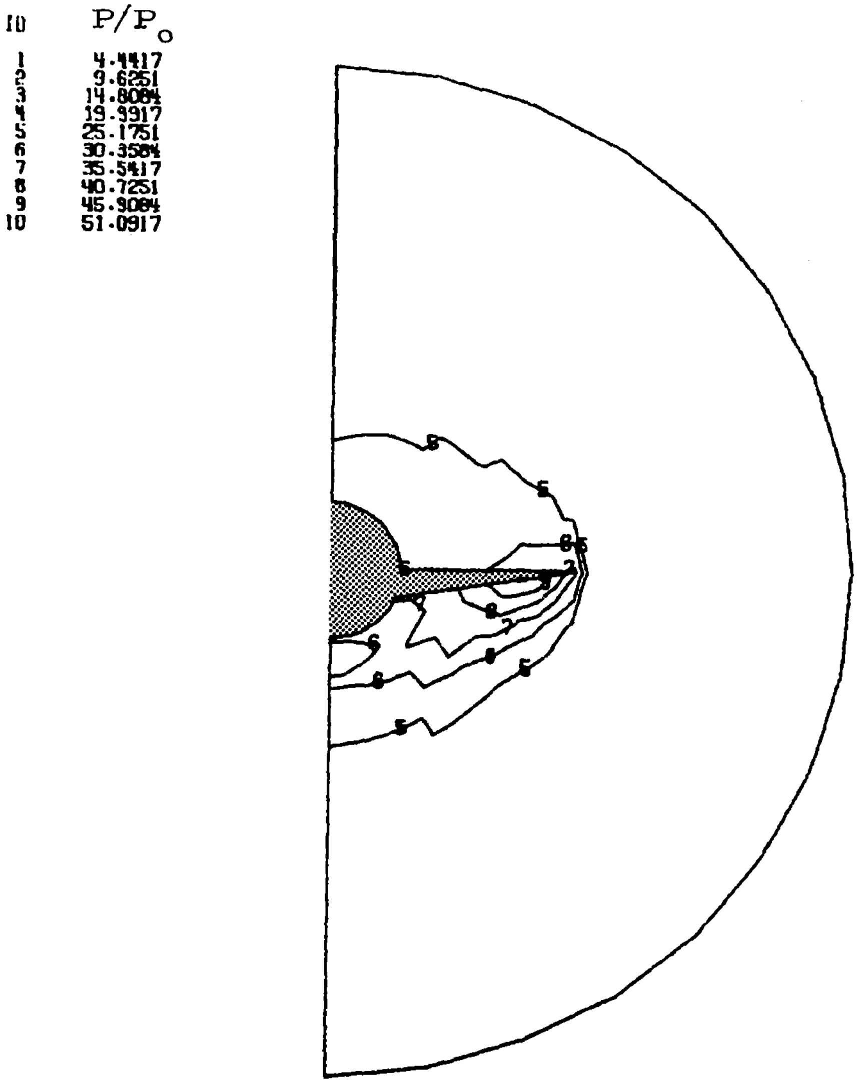


Fig. 7-34 - Static Pressure Contours for Plane 20, X = 0.2883 m
(P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

ID	P/P_o
1	4.4417
?	9.6251
3	14.8084
4	19.9917
5	25.1751
6	30.3584
7	35.5417
8	40.7251
9	45.9084
10	51.0917

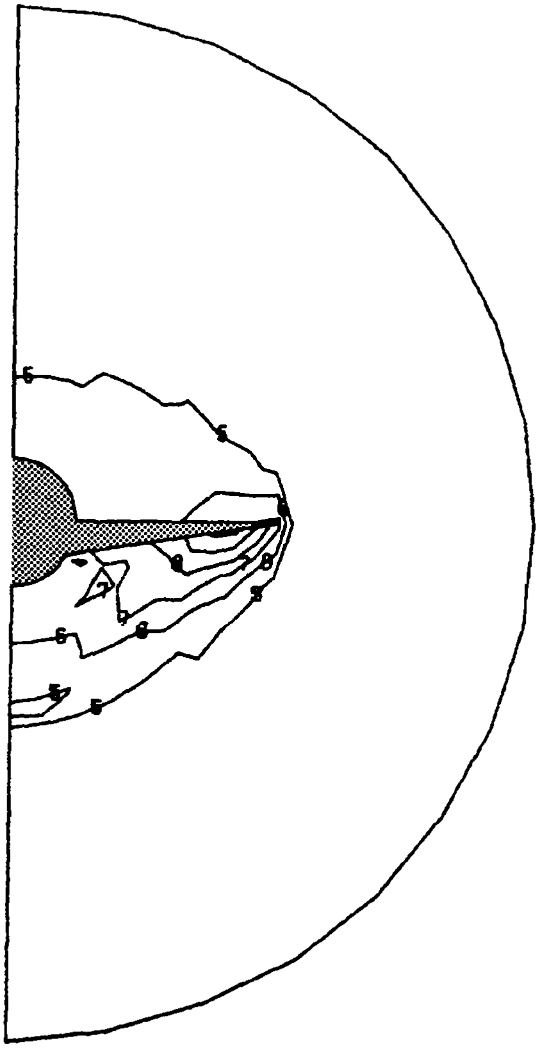


Fig. 7-35 - Static Pressure Contours for Plane 25, X = 0.3188 m
(P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

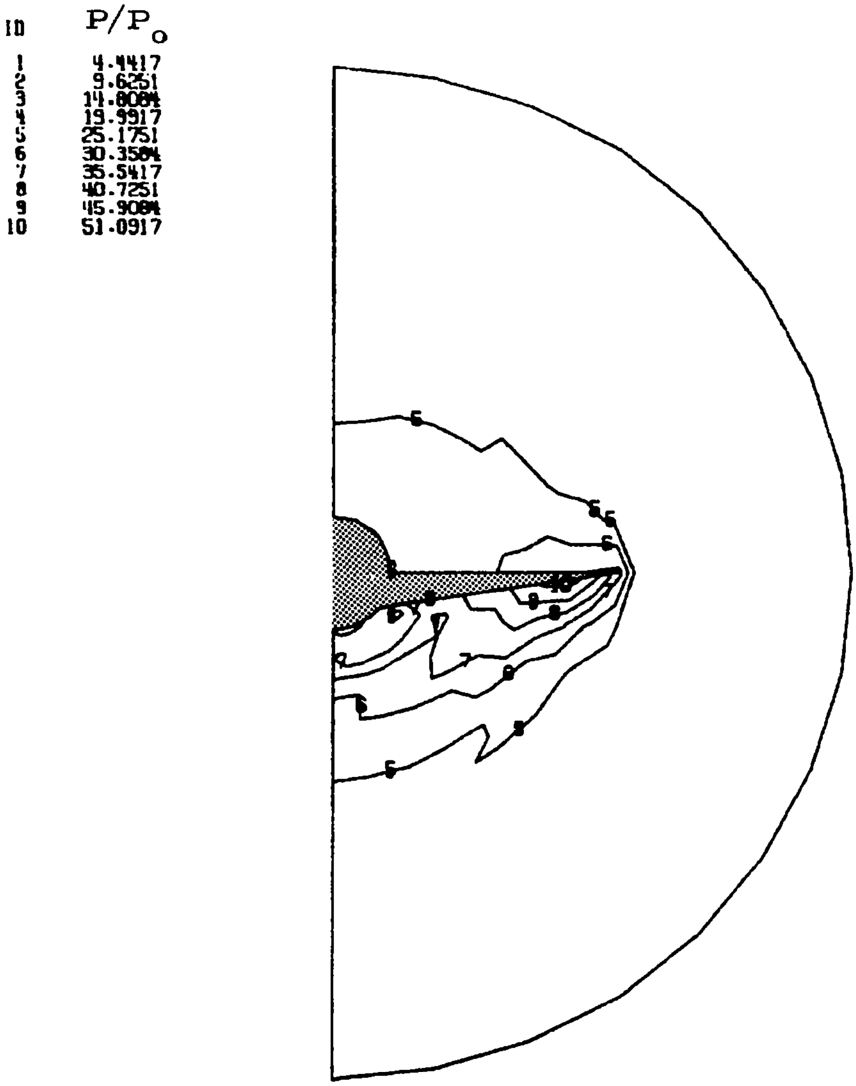


Fig. 7-36 - Static Pressure Contours for Final Plane, X = 0.3551 m
(P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

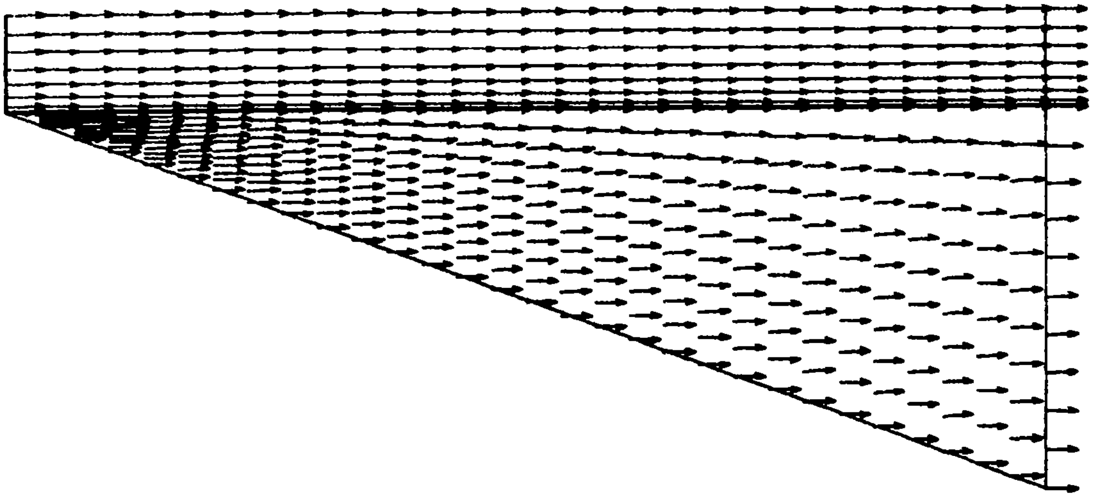


Fig. 7-37 - Upper Wing Surface Velocity Vectors

ID	P/P_o
1	4.4417
2	9.6251
3	14.8084
4	19.9917
5	25.1751
6	30.3584
7	35.5417
8	40.7251
9	45.9084
10	51.0917

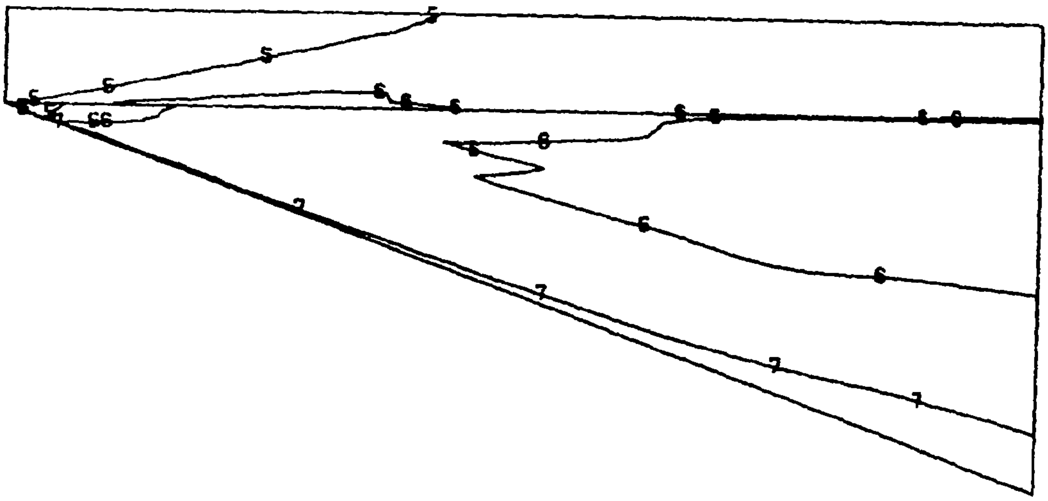


Fig. 7-38 - Upper Wing Surface Static Pressure Contours
(P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

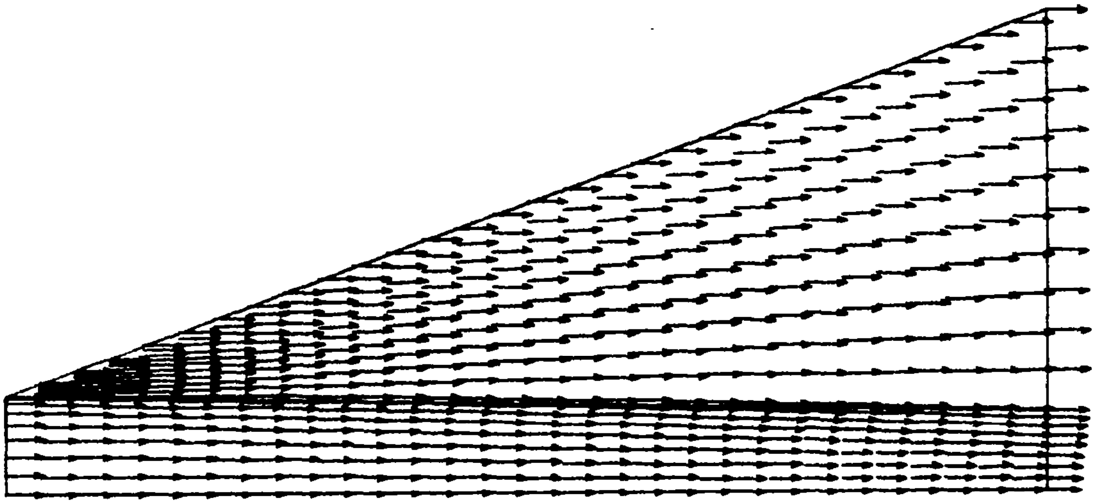


Fig. 7-39 - Lower Wing Surface Velocity Vectors

ID	P/P$_o$
1	4.4417
2	9.6251
3	14.8084
4	19.9917
5	25.1751
6	30.3584
7	35.5417
8	40.7251
9	45.9084
10	51.0917

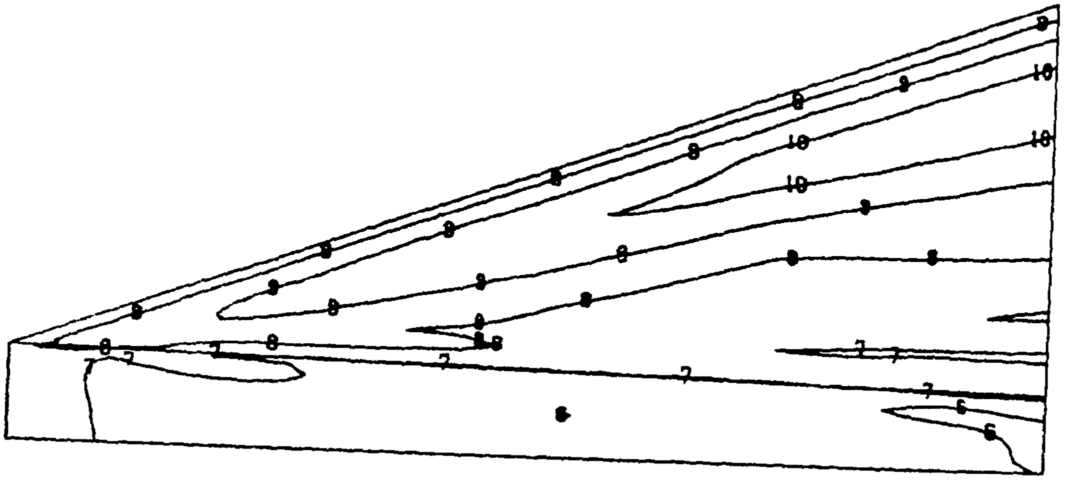


Fig. 7-40 - Lower Wing Surface Static Pressure Contours (P_o = 1.0 lbf/ft^2 = 0.04778 k Pa)

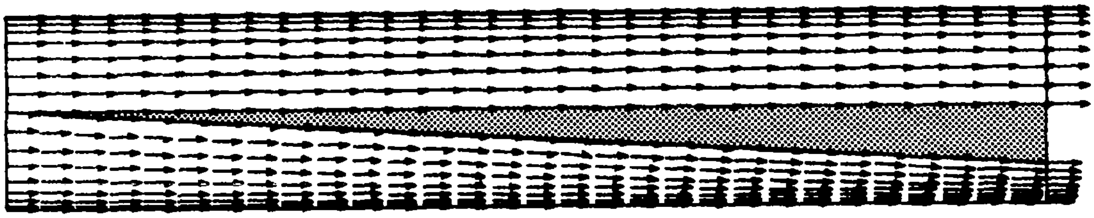


Fig. 7-41 - Body Surface Velocity Vector

ID	P/P_o
1	4.4417
2	9.6251
3	14.8084
4	19.9917
5	25.1751
6	30.3584
7	35.5417
8	40.7251
9	45.9084
10	51.0917

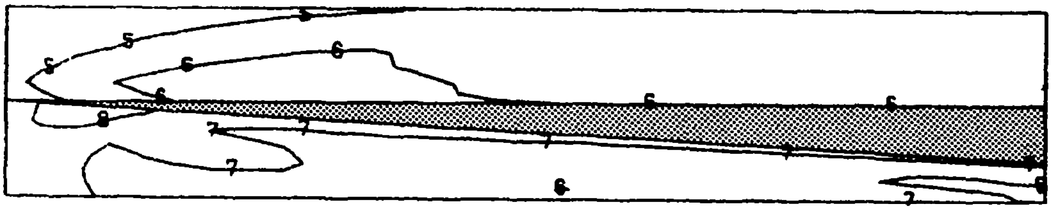


Fig. 7-42 - Body Surface Static Pressure Contours
(P_o = 1.0 lbf/ft^2 = 0.04788 k Pa)

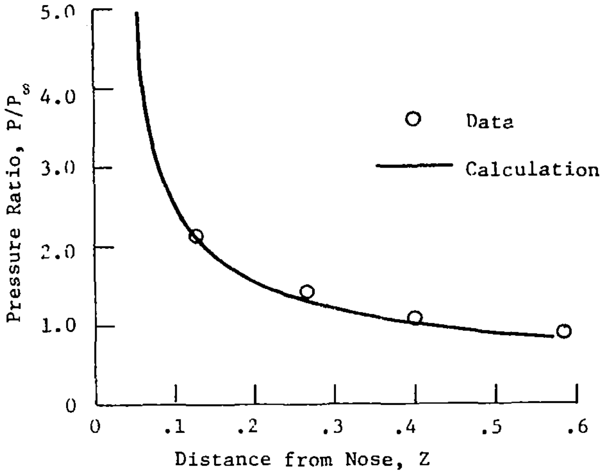


Fig. 7-43 - Surface Pressure Profile on Forebody
(P_∞ = 24.3 lbf/ft^2 = 1.1634 k Pa)
(Z_o = 1.0 ft = 0.3048 m)

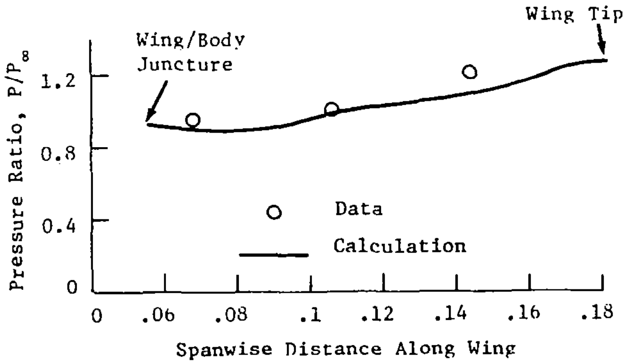


Fig. 7-44 - Spanwise Pressure Distribution
Z = 0.2804 m
(P_∞ = 24.3 lbf/ft^2 = 1.1634 k Pa)
(Y_o = 1.0 ft = 0.3048 m)

1. Report No.
NASA CR-3652

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
DEVELOPMENT AND APPLICATION OF THE GIM CODE FOR THE CYBER 203 COMPUTER

5. Report Date
December 1982

6. Performing Organization Code

7. Author(s)
J. F. Stalnaker, M. A. Robinson, E. G. Rawlinson
P. G. Anderson, A. W. Mayne, L. W. Spradley

8. Performing Organization Report No.
LMSC-HREC TR D784478

10. Work Unit No.

9. Performing Organization Name and Address
Lockheed Missiles & Space Company, Inc.
P. O. Box 1103
Huntsville, AL 35807

11. Contract or Grant No.
NAS1-15783, 15795

13. Type of Report and Period Covered
Contractor Report

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546

14. Sponsoring Agency Code

15. Supplementary Notes
Langley Technical Monitors: J. L. Hunt and J. P. Drummond
Interim Report

16. Abstract

This study is directed toward development and application of the GIM computer code for fluid dynamics research. This interim report describes a number of tasks performed as a part of this study including: enhancement of the computer code, implicit algorithm development, turbulence model implementation, chemistry model development, interactive input module coding and wing/body flowfield computation. The GIM quasi-parabolic code development was completed, and the code used to compute a number of example cases. Turbulence models, algebraic and differential equations, were added to the basic viscous code. An equilibrium reacting chemistry model and an implicit finite difference scheme were also added. Development was completed on the interactive module for generating the input data for GIM. Solutions for inviscid hypersonic flow over a wing/body configuration are also presented.

17. Key Words (Suggested by Author(s))
Computational Fluid Dynamics
Wing/Body Flows
Turbulence Modeling
Reacting Flows

18. Distribution Statement
Unclassified - Unlimited

Subject Category 34

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
182

22. Price
A09

For sale by the National Technical Information Service, Springfield, Virginia 22161