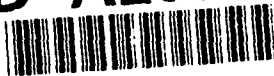


AD-A268 177



CR 93.002

NCEL

June 1993

Contract Report

An Investigation Conducted by
Adaptive Research Corporation
Huntsville, AL

STRUCTURED FINITE VOLUME MODELING OF U.S. NAVY AIRCRAFT ENGINE TEST CELLS

TASK 1: TURBOSHAFT ENGINE -

CODE DOCUMENTATION AND LISTINGS - VOLUME 2

Abstract Volume 1, Final Report, presents results of the numerical simulation of a U.S. Naval turboshaft test cell facility. The ultimate purpose of this simulation was to provide the Navy with a numerical model to be used for the evaluation of the aerothermal performance of test cells. This simulation was performed using the structured finite volume (SFV) computer code. A description of the physical model, mathematical details, boundary conditions, and results of the study are presented and covered in this report.

Volume 2, Code Documentation and Listings, provides a copy of the input files developed for the modeling of turboshaft test cells.

93-18635



13908

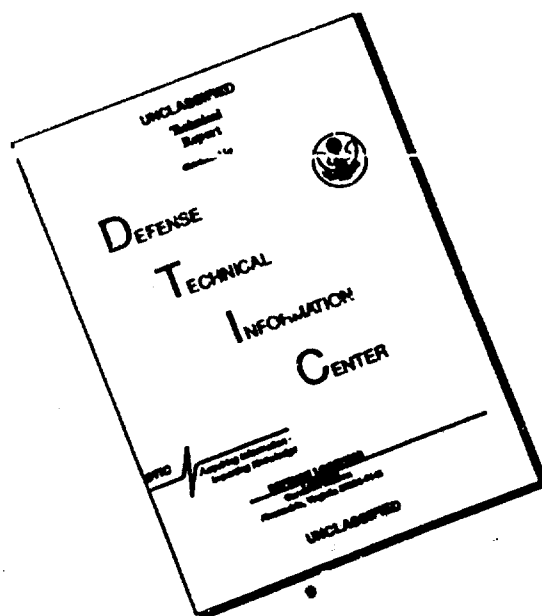
DTIC
ELECTE
AUG 12 1993
S B D

NAVAL CIVIL ENGINEERING LABORATORY PORT HUENEME CALIFORNIA 93043-4328

Approved for public release; distribution is unlimited.

00

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

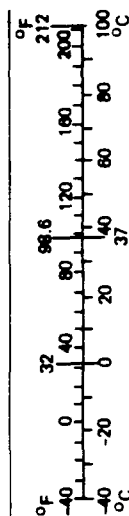
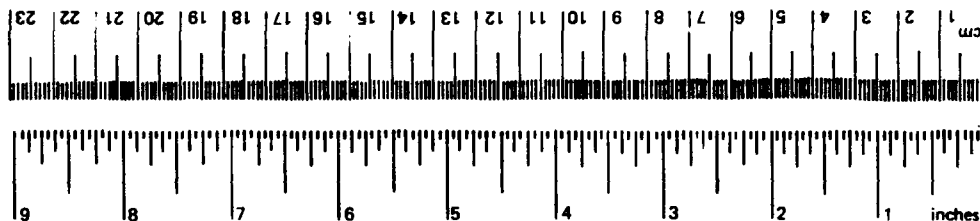
METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
in ft yd mi	inches	*2.5	centimeters	cm
	feet	30	centimeters	cm
	yards	0.9	meters	m
	miles	1.6	kilometers	km
in ² ft ² yd ² mi ²	square inches	6.5	square centimeters	cm ²
	square feet	0.09	square meters	m ²
	square yards	0.8	square meters	m ²
	square miles	2.6	square kilometers	km ²
	acres	0.4	hectares	ha
oz lb	ounces	28	grams	g
	pounds	0.45	kilograms	kg
	short tons (2,000 lb)	0.9	tonnes	t
tsp Tbsp fl oz c pt qt gal ft ³ yd ³	teaspoons	5	milliliters	ml
	tablespoons	15	milliliters	ml
	fluid ounces	30	milliliters	ml
	cups	0.24	liters	l
	pints	0.47	liters	l
	quarts	0.95	liters	l
	gallons	3.8	liters	l
	cubic feet	0.03	cubic meters	m ³
	cubic yards	0.76	cubic meters	m ³
TEMPERATURE (exact)				
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
mm cm m km	millimeters	0.04	inches	in
	centimeters	0.4	inches	in
	meters	3.3	feet	ft
	kilometers	1.1	yards	yd
		0.6	miles	mi
cm ² m ² km ² ha	square centimeters	0.16	square inches	in ²
	square meters	1.2	square yards	yd ²
	square kilometers	0.4	square miles	mi ²
	hectares (10,000 m ²)	2.5	acres	
MASS (weight)				
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1,000 kg)	1.1	short tons	
VOLUME				
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m ³	cubic meters	35	cubic feet	ft ³
m ³	cubic meters	1.3	cubic yards	yd ³
TEMPERATURE (exact)				
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



*1 in. = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price \$2.25, SD Catalog No. C13.10-286.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-018	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 1993		3. REPORT TYPE AND DATES COVERED Final; December 1990 - September 1992
4. TITLE AND SUBTITLE STRUCTURED FINITE VOLUME MODEL- ING OF U.S. NAVY AIRCRAFT ENGINE TEST CELLS TASK 1: TURBOSHAFT ENGINE - CODE DOCUMENTATION AND LISTINGS - VOLUME 2		5. FUNDING NUMBERS PR - 0604215N C - N47408-91-C-1228 DN - 661008		
6. AUTHOR(S) P. L. Daley and W. A. Mahaffey				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Adaptive Research Corporation 4960 Corporate Drive, Suite 100-A Huntsville, AL 35805		8. PERFORMING ORGANIZATION REPORT NUMBER CR 93.002		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Commander / Naval Civil Engineering Laboratory Naval Air Systems Command 560 Laboratory Drive Code 09Y Facilities Systems Division/L53 Washington, DC 20362-5101 Port Hueneme, CA 93043-4328		10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This report presents results of the numerical simulation of a U.S. Naval turboshaft test cell facility. The ultimate purpose of this simulation was to provide the Navy with a numerical model to be used for the evaluation of the aerothermal performance of test cells. This simulation was performed using the structured finite volume (SFV) computer code. A description of the physical model, mathematical details, boundary conditions, and results of the study are presented and covered in Volume 1. Volume 2, Code Documentation and Listings, provides a copy of the input files developed for the modeling of turboshaft test cells.				
14. SUBJECT TERMS Computational fluid dynamics, test cells, aviation test facilities			15. NUMBER OF PAGES 147	
			16. PRICE CODE UL	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	

TABLE OF CONTENTS

Section No.		Page No.
1.	INTRODUCTION	1
1.1	Purpose of the Report	1
1.2	The Listings Provided	1
2.	USER SECTION	2
2.1	Grid Generation	2
2.2	Other Input	7
2.3	Relaxation	7
2.4	Other Controls	8
2.5	Additional Printout	9
FIGURES		10
APPENDIX A		
APPENDIX B		
APPENDIX C		
APPENDIX D		

DTIC QUALITY INSPECTED 3

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/ov/ _____	
Availability Codes	
Dist. A-1	Avail and/or Special

LIST OF FIGURES

	Page No.
1. Regions of CS62	10
2. Grid of CS62	11
3. Regions of CS63	12
4. Initial Grid of CS63	13
5. Regions of CS61	14
6. Grid of CS61	15
7. Regions of CS64	16
8. Initial Grid of CS64	17
9. Enlargement of Initial Grid of CS64	18
10. Enlargement of Final Grid of CS64	19
11. Regions of CS65	20
12. Initial Grid of CS65	21
13. Enlargement of Initial Grid of CS65	22
14. Enlargement of Final Grid of CS65	23
15. Final Grid of CS66	24
16. Regions of CS67	25
17. Initial Grid of CS67	26
18. Final Grid of CS68	27
19. Grid of CS69	28
20. Grid of CS70	29

1. INTRODUCTION

1.1 Purpose of the Report

This report provides a copy of the input files developed for the modeling of turboshaft test cells. These copies are contained in the Appendices of this report and are described briefly below. A detailed discussion of building a computational grid for this project is provided in the second section of this report. The results of the turboshaft test cell modeling are reported in the first volume of this report.

1.2 The Listings Provided

The listings are contained in Appendices B through D. Appendix B contains the Q1 input file, Appendix C contains the FORTRAN SATELLITE program, and Appendix D contains the FORTRAN GROUND file. Sketches are provided in Appendix A.

2. USER SECTION

2.1 Grid Generation

In this section a detailed discussion for the creation of a computational grid is supplied. The bulk of the input for this model deals with producing a computational grid. The code was designed for relatively easy modifications with the flexibility to model a range of changes as called for in the scope of work.

The premise of this procedure is that a 2-dimensional package will be used to create various cross sectional planes. These planes will then be stacked, blended or rotated to create the final 3-dimensional computational grid. In general, the program works as follows: 1.) the user specifies all the inputs necessary for the creation of all the various 2-dimensional cross sectional (X-Y) planes inside the standard input files (Q1 and SATLIT), 2.) the standard input files are then executed to produce the data files needed for the 2-dimensional grid generation program (EasyMesh2D or GGP), 3.) GGP is then executed for each data plane produced, and 4.) the standard input files are re-executed to produce the final grid and the other input files needed for the solver.

The standard input files will create 5 or 6 types of X-Y planes. Each plane can have several different varieties or subsets. The first type (TYPE 1) of plane is used to describe the area around the dynamometer. The planes are broken down into various regions in the X and Y directions. The user must specify the total distance from the origin for each region, the number of cells in each region, and the clustering factor for the gridding of each region. Each of these will be detailed later in this section.

The second type (TYPE 2) is used to describe the X-Y cross section of the engine. The third type (TYPE 3) may or may not be used. This type is used if the exit of the engine falls close to the augmentor tube or even inside the tube. This type will be used if the gap distance is less than approximately 6 inches. This type of plane is used to improve the orthogonality in this region. TYPE 4 is used for X-Y cross section that across the augmentor tube. The fifth type (TYPE 5) is used to describe the front face of the chimney. The final type (TYPE 6) is used to describe the exit plane. Additional information may be supplied in the input files.

The file name nomenclature for the data files for the GGP is that the file name starts with the letter CS. Then numbers are added as suffixes starting at 61 and continuing until all planes are created. The data files are generally created in order with the exception of CS61. This file is created after CS63. For TYPE 1 there are three different subsets (CS files) created. The first (CS62) is used to describe the geometry in front of the dynamometer, the second (CS63) produces a cross section of the dynamometer, and the last (CS61) describes the inlet plane.

CS63 is an accurate representation of an X-Y plane through the dynamometer. CS62 is identical to CS63 with the exception that the circle of the dynamometer has become a square to help orthogonality. Its location is at the midpoint between the end of the inlet baffles and the start of the cart. CS61 is identical to CS62 with the exception that one line has been moved to correspond to the dimensions of the inlet opening. CS61 is located at the front of the test cell.

For TYPE 1 files there are 13 regions that are defined in the X-direction and 10 regions are used in the definition for the gridding in the Y-direction. For each region the following information is needed

- The number of cells of each region,
- The distance to the end of the region, and
- A grid clustering factor.

The nomenclature for each of these variables is given in the Q1 file. They are noted in Figure 1 of this report. In this figure the regions in both directions for CS62 are noted along with distance and clustering nomenclature. This input is used primarily for the description of lines and arcs in the data files for GGP. Figure 2 is the copy of a graphical display produced during the creation of the 2-D grid file. In this figure the full grid is displayed. Similarly plots for CS63 and CS61 are supplied. In general, the data supplied for CS62 is used for CS63 and CS61. The dimension of the dynamometer opening is used to calculate the corresponding square in CS62. This is why some of the data for distance of each region is set to 0.000000. A integer array is used as a marker to note the first region that contains an arc. The variable XINL is the X-direction length of the inlet baffles.

Note in Figure 4 that it appears that lines overlap in the circular region. This is because some lines are overwritten with arc data. If this persists after a redraw of the GGP, major problems with the grid exist. More details in regard to the execution of GGP will be given later in this section.

The coding was designed so that major changes would be fairly straight forward. The input files has slots for 14 regions in each direction so that if more regions are needed in the future the accommodations can be made. Also, the number of cells for each region in the remaining types are not required but are obtained from the number of cells supplied for each region in TYPE 1.

TYPE 2 data produces one or two CS files. The first is for the engine inlet and the last is for the engine outlet (which may be produced from TYPE 3). Since there is less complexity in this area only eight regions are needed to describe the regions in both the X- & Y-directions. Basically the area on the cart (X-direction) and the area between the cart and the top of the dynamometer exhaust shirt (Y-direction) are rearranged in to fewer regions as compared with TYPE 1 data. Also, in the circular region additional cells are picked up since the diameter of the engine inlet is larger than the dynamometer opening. These are controlled by variables NXAD and NYAD. Note that the summation of the total number of cells over each TYPE is constant.

The regions and the initial grid for CS64 is shown in Figures 7 and 8. When the initial grid is completed, the orthogonality of corner points of the circle can be improved (note Figure 9 and 10). This is done in the smoothing operations of the GGP. The number of cells affected by this is controlled by the variable ISOL located in the SATLIT file. In general these values will not need to be adjusted.

In the form delivered, TYPE 3 data is used. This data is used to produce a circle in a circle grid form. TYPE 3 data produces two or three CS files. The first is for the exit of the engine which is also the same as for the inlet of the augmeter tube. The second is for the end of the augmeter lip. A third cross section may be required if the exit of the engine falls within the lip or within the sleeve. See Appendix A for more details. The only difference in these CS files will be the diameter of the two circles. Since the diameter of augmeter tube is larger than the engine exit additional cells are needed. The number of cells is controlled by variables NXBD and NYBD. For the case delivered two CS files

(CS65 and CS66) were produced. The regions and initial grids for TYPE 3 are shown in Figure 11 through 15. In Figures 13 and 14 the initial and final (smoothed) grids are displayed.

TYPE 4 data will produce two or three CS files. If TYPE 3 data is used it will produce two files. The first file is for the augments lip (not produced when TYPE 3 is used), the second is for the augments sleeve, and the last is for the augments tube. The difference in these files are due to the different diameters. The regions and grids (CS67 and CS68) are shown in Figure 16 through 18.

TYPE 5 data produces one CS file (CS69). It is located at the front of the chimney. This is the plane that is rotated. This file has two options. The first is for a square duct and the second is for a round duct. This is the first cross section in which the first region does not start at a 0.0 X-coordinate value. A integer array element noted in the Q1 files takes this into account. A plot of the grid is shown in Figure 19.

The last grid is denoted by TYPE 6. It is located at the exit of the chimney. The input needed to produce this data file is taken from previously supplied information. The grid for CS70 is shown in Figure 20.

There is a integer array element that represents the stage of grid development. It is located in Group 6 of the Q1 file as is called IG (1). If the value of this element is set to 1, when the input files are executed, they will produce a set of data files for the GGP. If it is set to 2, then it will read the grid files produced by the GGP and create a 3-dimensional grid along with the other input files for the solver. If the grid is already created the value is set to 3 in order to bypass the grid creation coding.

In the form delivered, 10 data files for the GGP will be created during the first execution of the input files. At this time the user will then execute GGP as indicated in the documentation (probably done by entering `runezm`). The first item needed will be terminal type. Enter the appropriate value. Following this prompt, menus will appear on the screen. The following series of commands will go through these menus and produce a grid file.

<u>PROMPT</u>	<u>ENTER</u>	<u>COMMENT</u>
Model name	CS61	Use same name as file to be read in
EZ2 >	RE CS61	Reads in input file
EZ2 >	WR	Goes to menu to write grid
WRITE >	END	Writes grid
EZ2 >	END	End session

This is done when the grid to be produced is totally orthogonal (i.e. no circles). After the input file is read a redraw of the screen can be done through the REDR command. If lines cross after this point there is an error in the input file for the GGP. Looking at the grid may give clues as to the cause of the problem. If a grid needs to be smoothed (all files that contains a circle), the following commands will be needed.

<u>PROMPT</u>	<u>ENTER</u>	<u>COMMENT</u>
Model Name:	CS63	Use same name as file to be read in
EZ2 >	RE CS63	Reads in input file
EZ2 >	SM	Goes to smoothing menu
SMOOTH >	SO	Solves differential equations
SMOOTH >	REDR	Plots final grid
SMOOTH >	END	Returns to main menu
EZ2 >	WR	Goes to menu to write grid
WRITE >	END	Writes grid
EZ2 >	END	End session

After the creation of these 2-dimensional grid files, input in Q1 file is required for the formation of the final 3-dimensional grid. As in the specification of the grid in the X and Y-directions, the user must supply the number of regions, the distance to the end of the region, the number of cells, and the grid clustering factor. Allocations for 25 regions in the axial direction have been provided. As delivered, 24 have been specified.

The user must then supply the information for the building of the final grid. Four options are available 1.) Stack, 2.) Blend, 3.) Rotate, and 4.) End. Throughout the test cell the

first two options are used to stack and blend the 2-dimensional grid files as needed, while the last two options create the grid in the chimney region. This information is passed to the SATLIT from the Q1 through an integer array.

2.2 Other Input

In group 9 of the input files most of the data for the physics of the model is supplied. These deal with flow rates, temperatures, mass fractions, etc. These are documented in the input files.

2.3 Relaxation

Relaxation is a numerical technique that allows the rate of change of various solved variables to be controlled. It is generally used to dampen the amount of change computed by the various computer codes. There are many views on the optimum settings of the relaxation parameters. In a problem of this size time constraints reduce the amount of effort in optimization of these parameters. The approach used was to reduce the relaxation (base values calculated on a cell residence time) at the start of a computational run and then apply tighter relaxation after a few hundred solution sweeps through the calculation domain.

The values of the relaxation parameters is given in the following table.

Table 1. Relaxation Parameters

<u>Variable</u>	<u>Type</u>	<u>Initial Value</u>	<u>Final Value</u>
P1	LINRLX	0.15	0.10
U1	FALSDT	0.0005	0.00025
V1	FALSDT	0.0005	0.000025
W1	FALSDT	0.0005	0.00025
KE	LINRLX	0.10	0.10
EP	LINRLX	0.10	0.10
H1	FALSDT	0.001	0.0005
C1	FALSDT	0.001	0.0005
C2	FALSDT	0.001	0.0005

Note the two types of relaxations are discussed in the users guide. The final values were used after sweep 421. (See following section for procedure to change relaxation.)

2.4 Other Controls

Depending on computer systems, it may take a few weeks to obtain a fully converged solution. The code allows for restarts using previous data. For some cases this may not be the best procedure as compared to one long run. Because of this various controls were put in the GROUND coding that allows the user to vary items during one long run. This coding allows the user to:

1. Abort a run with standard output produced,
2. Modify pressure relaxation,
3. Modify turbulence relaxation,
4. Modify velocity relaxation,
5. Modify scalar relaxation,
6. Dump a restart file on demand,
7. Change frequency of monitor printout,
8. Change frequency of residual printout,
9. Change the number of variables in the monitoring values printed, and
10. Change two monitor locations.

This is accomplished by:

1. Providing a file called ABORT,
2. Providing a value in the F12.8 Format in a file called RELAXP,
3. Providing two values in the 2F12.8 Format in a file called RELAXT,
4. Providing three values in the 3F12.8 Format in a file called RELAXV,
5. Providing three values in the 3F12.8 Format in a file called RELAXS,
6. Providing a file called DUMPIT,
7. Providing a value in the I5 Format in a file called TSTMOD,
8. Providing a value in the I5 Format in a file called NPRMOD,

9. Providing four values in the 4I2 Format in a file called IGGMOD (value of 1 activates printout while a value of 0 deactivates), and
10. Provide three values in the 3I3 Format in a file called ML2MOD or ML3MOD (values are for the IX, IY, and IZ locations).

2.5 Additional Printout

In addition to the standard output the following printout is provided:

1. Ten monitoring locations,
2. The maximum and minimum values for certain variables,
3. Convergence information,
4. Pumping ratios, and
5. Heat transfer information.

Note the previous section provided some information about control of the monitoring printout. The max-min printout may give clues to problem areas. Monitoring printout can then be shifted to these locations. The convergence information gives a mass and momentum error based on mass and momentum sources. A value of under 1% for mass and 3% for momentum should be acceptable. In addition the pumping ratios are printed for the dynamometer and the engine. When these value become asymptotic, this may indicate convergence. Printout is also provided for the heat transfer through the augments tube in the building and in the chimney. Similarly asymptotic values point toward convergence.

FIGURES

Figure 1. Regions of CS62

222

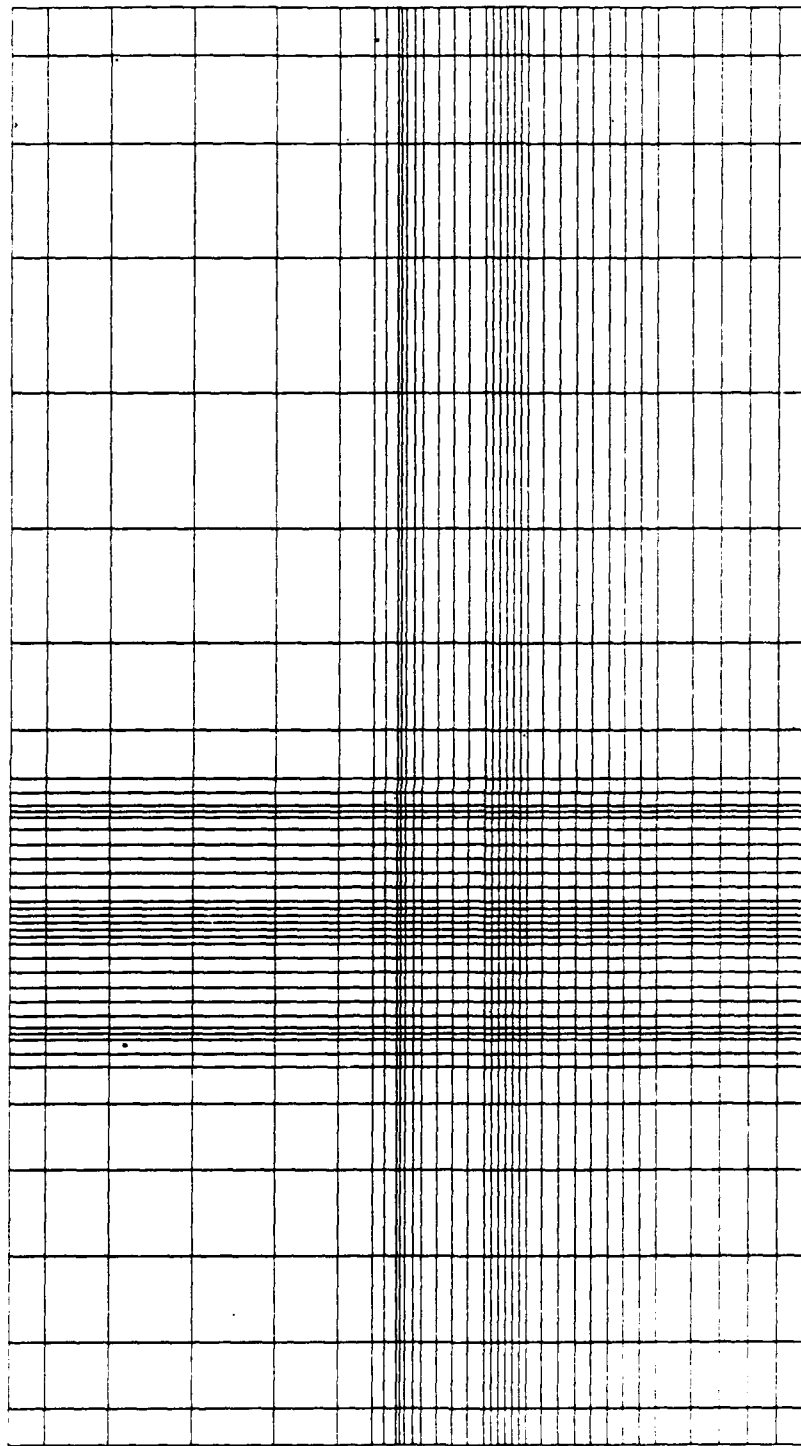


Figure 2. Grid of CS62

E22

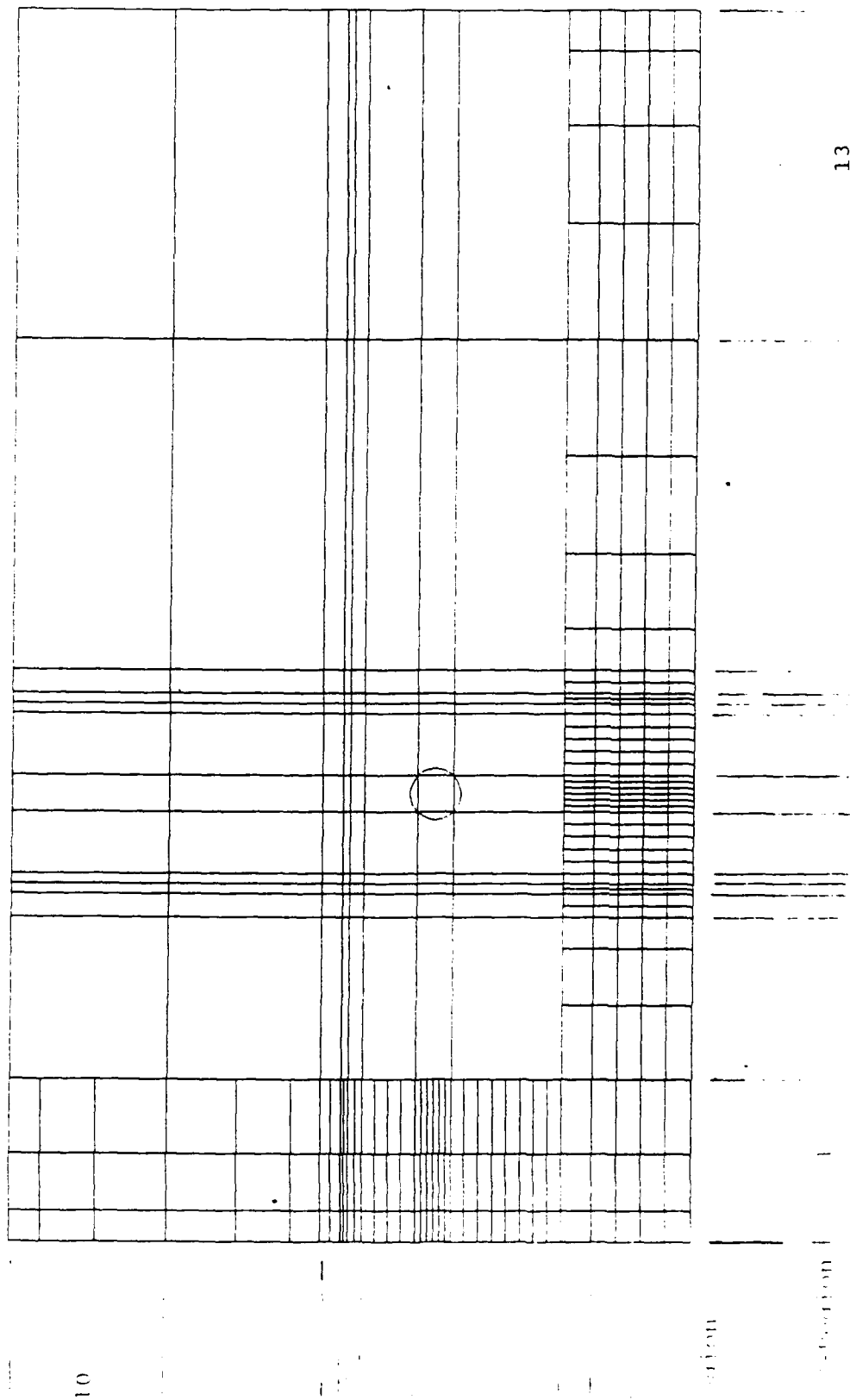


Figure 3. Regions of CS63

E22 >

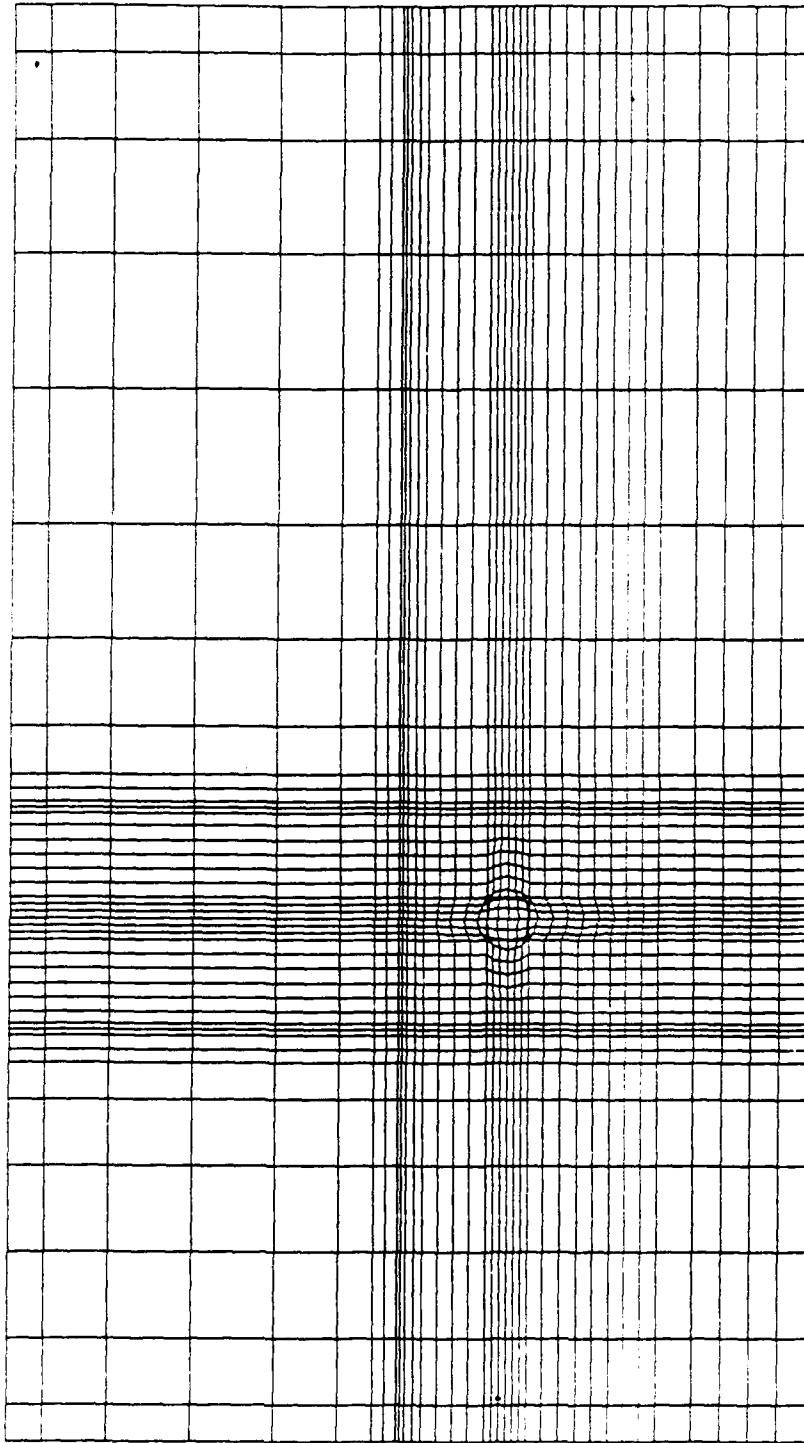


Figure 4. Initial Grid of CS63

E22 >

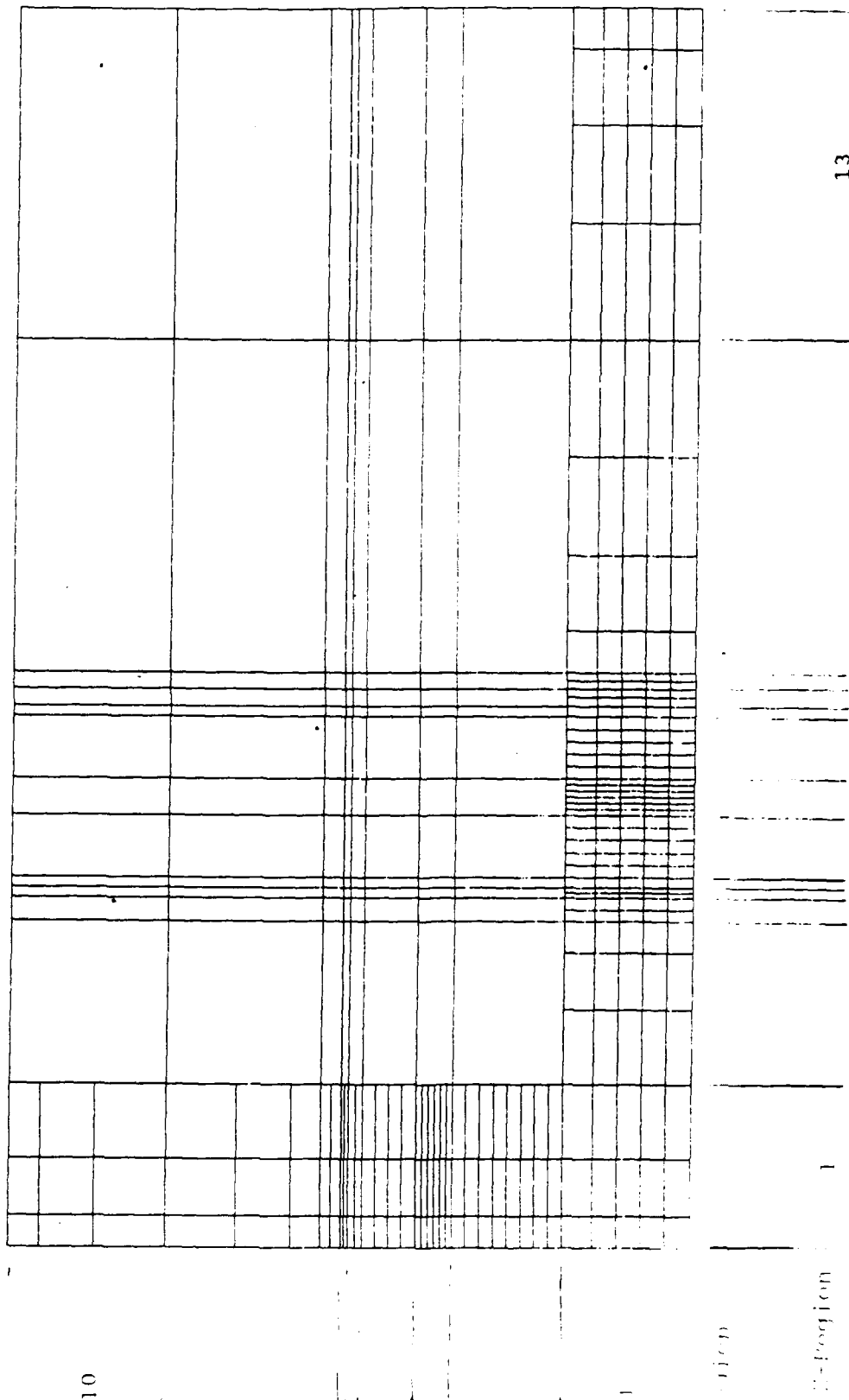


Figure 5. Regions of CS61

E22

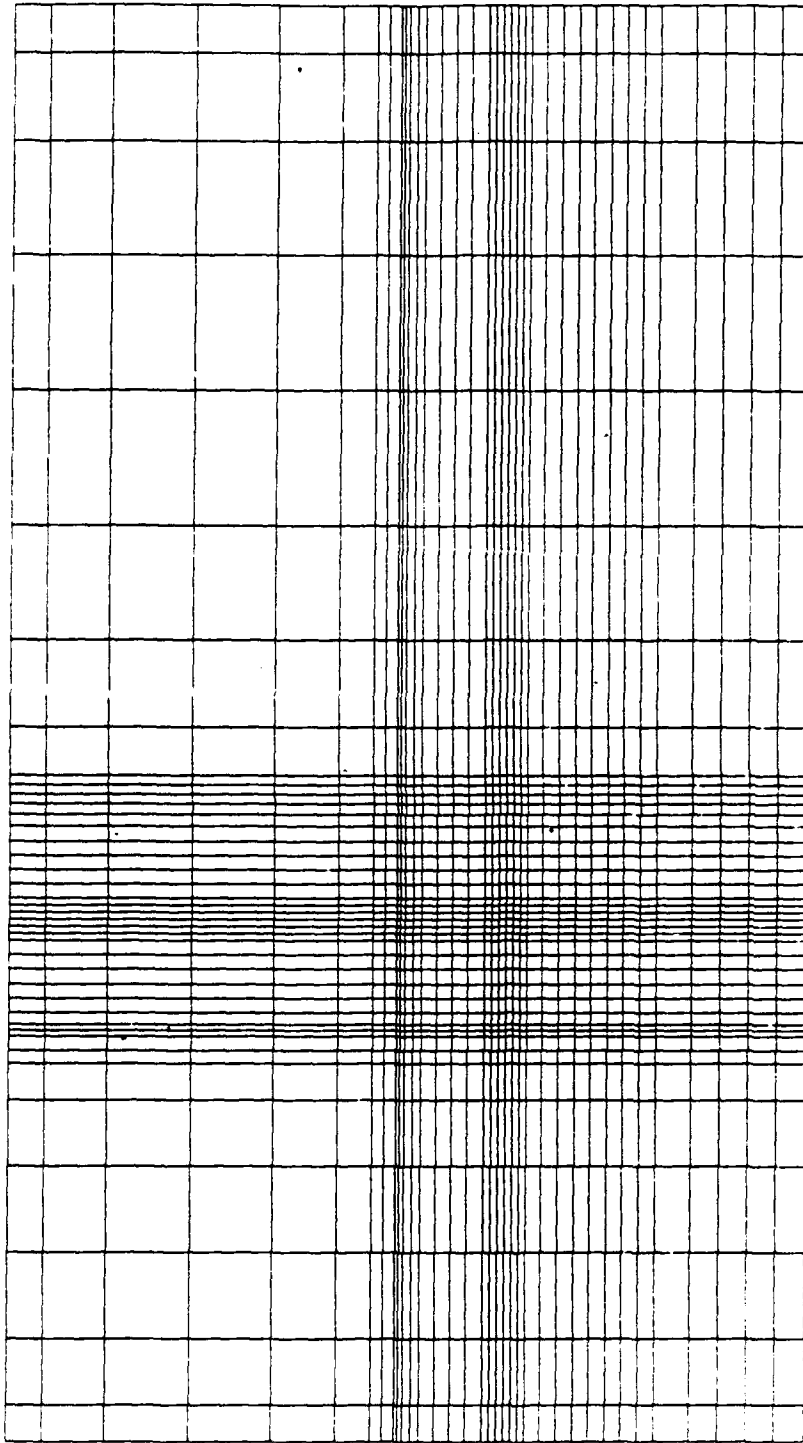


Figure 6. Grid of CS61

622

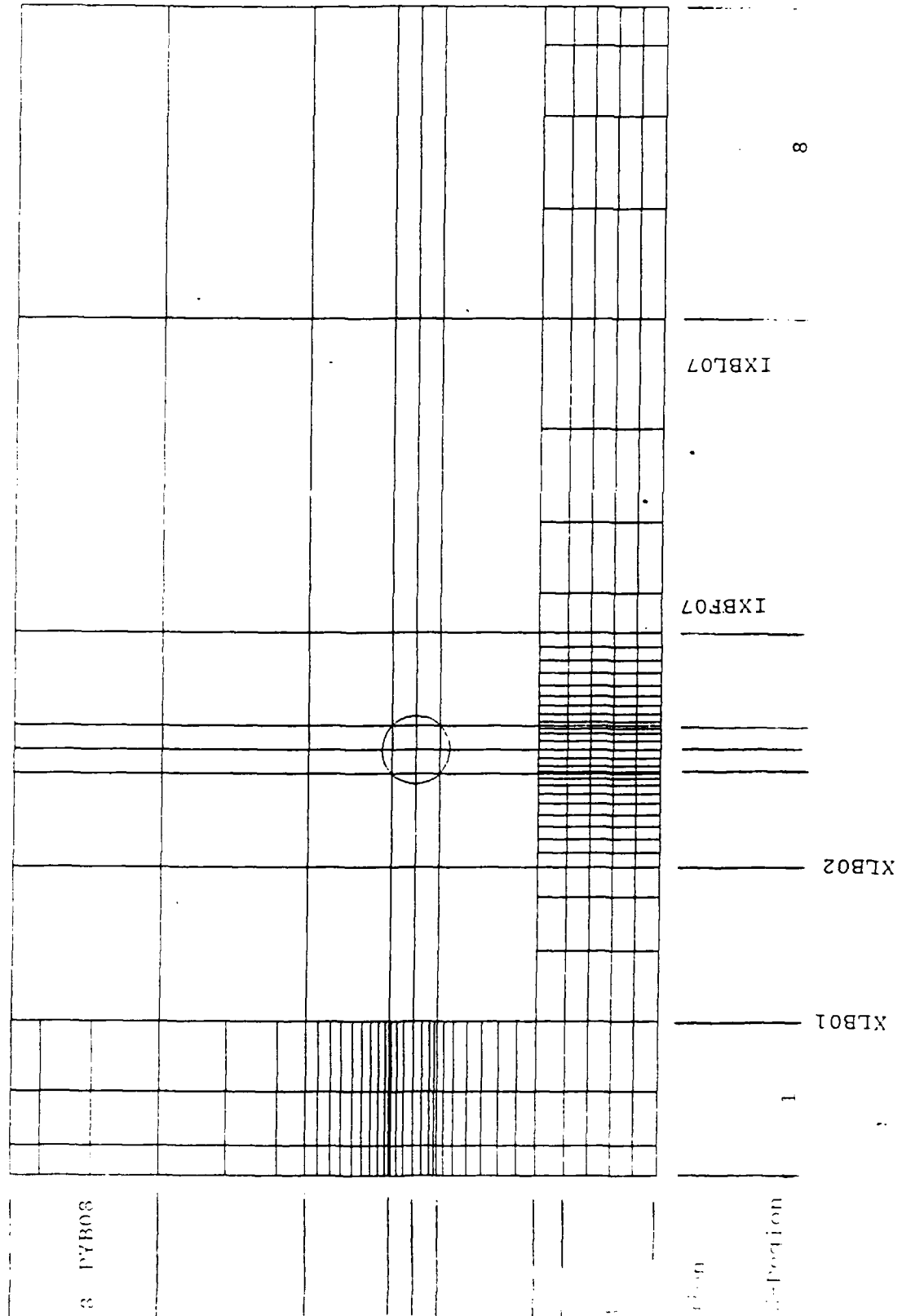


Figure 7. Regions of CS₆

622

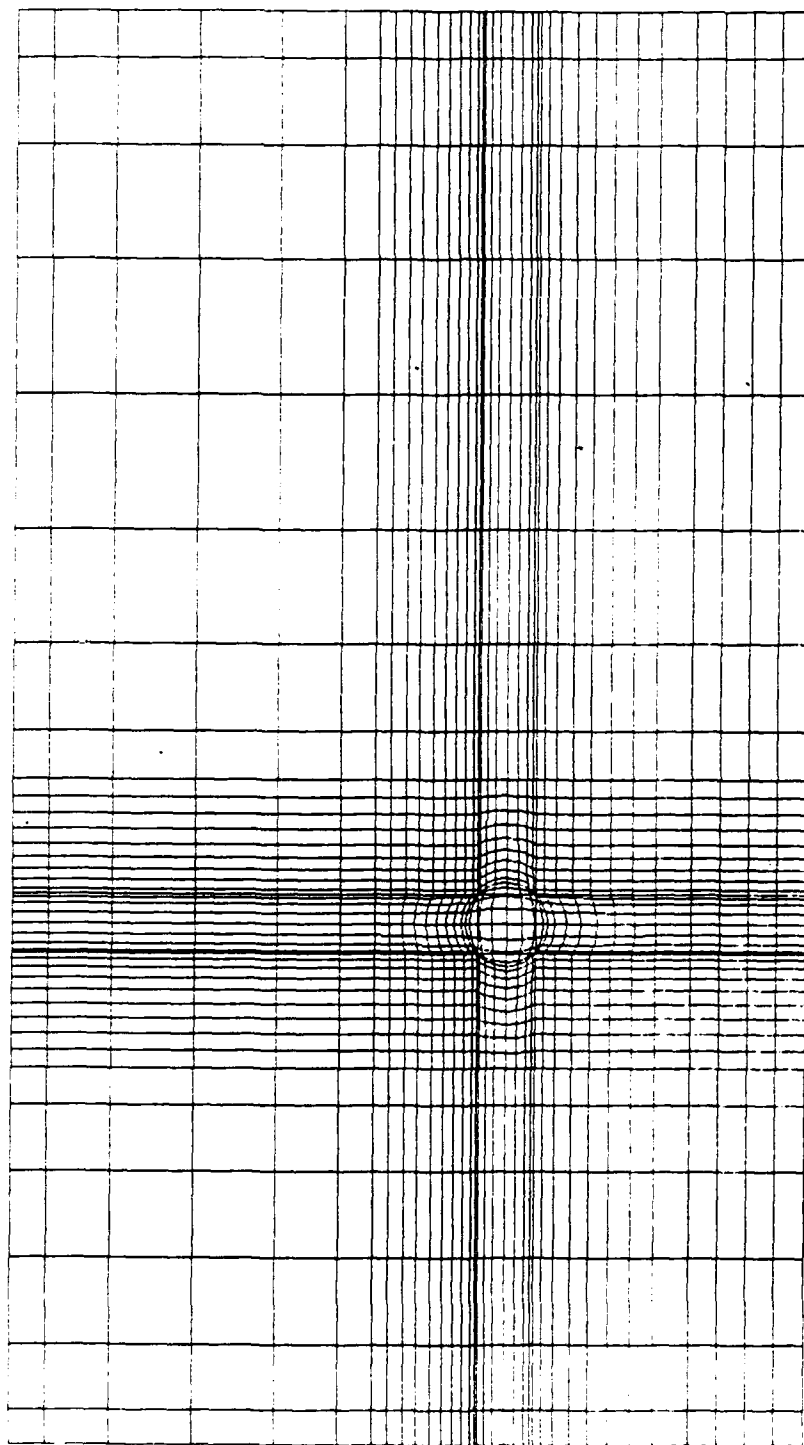


Figure 8. Initial Grid of CS64

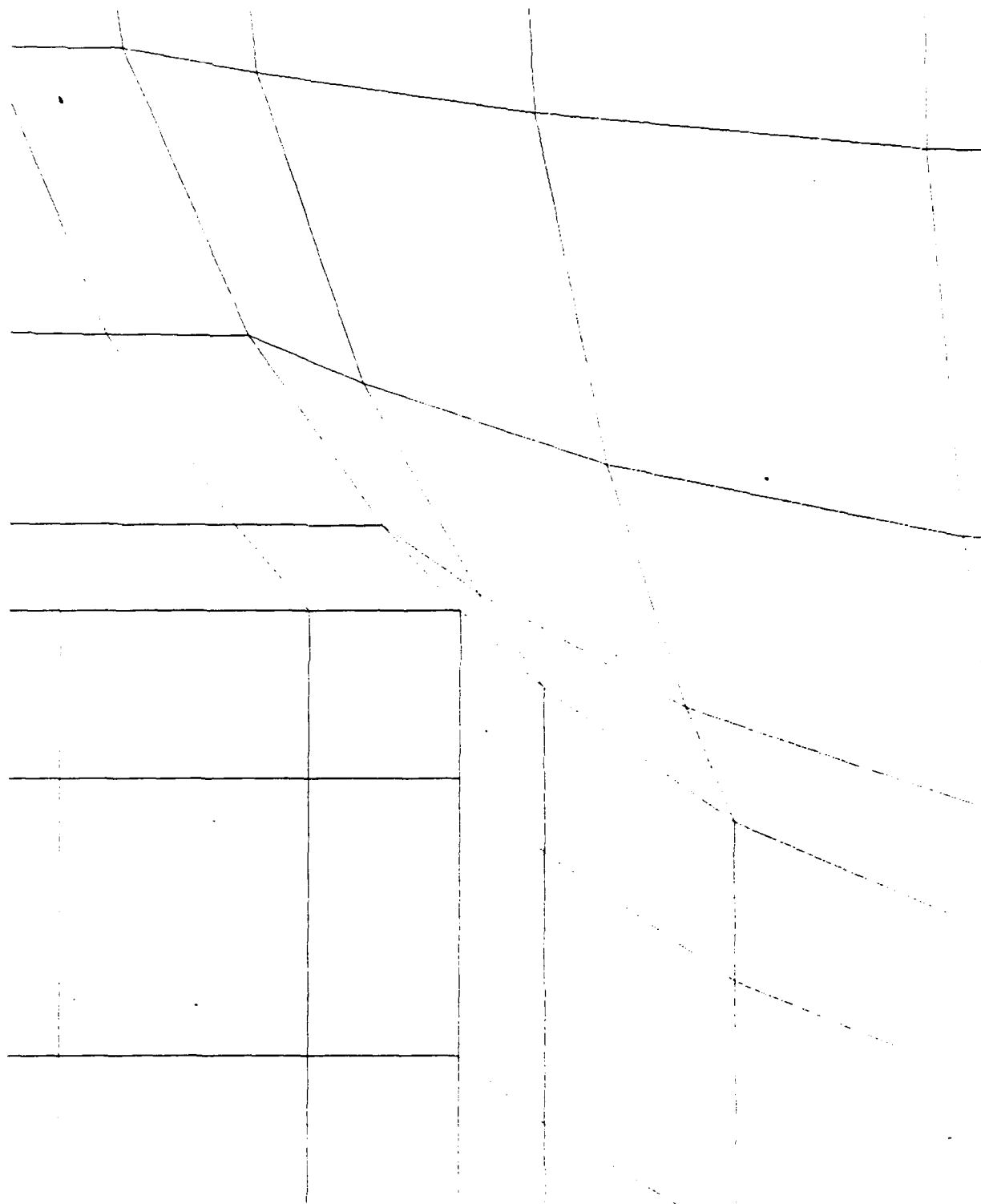


Figure 9. Enlargement of Initial Grid of CS64

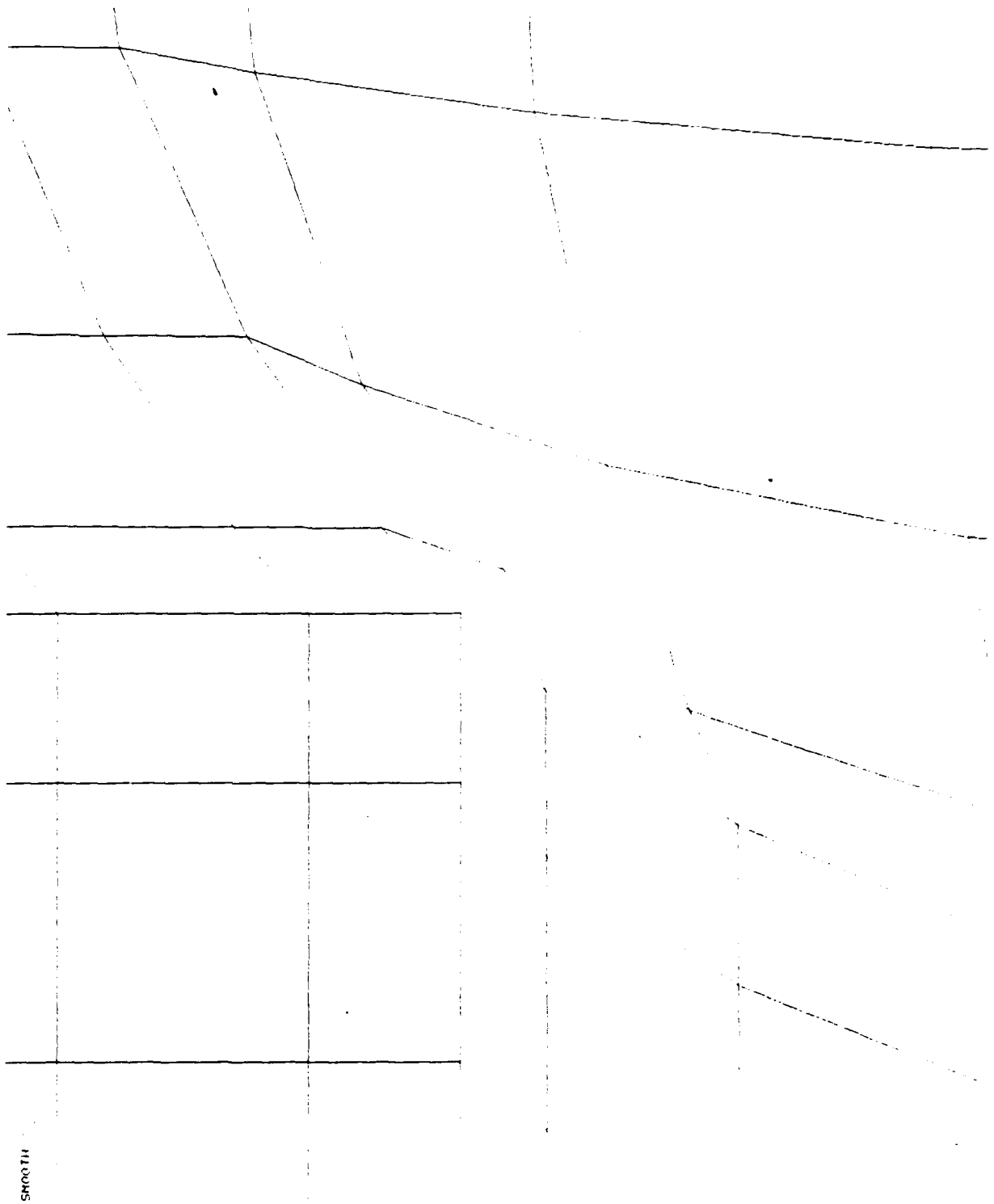
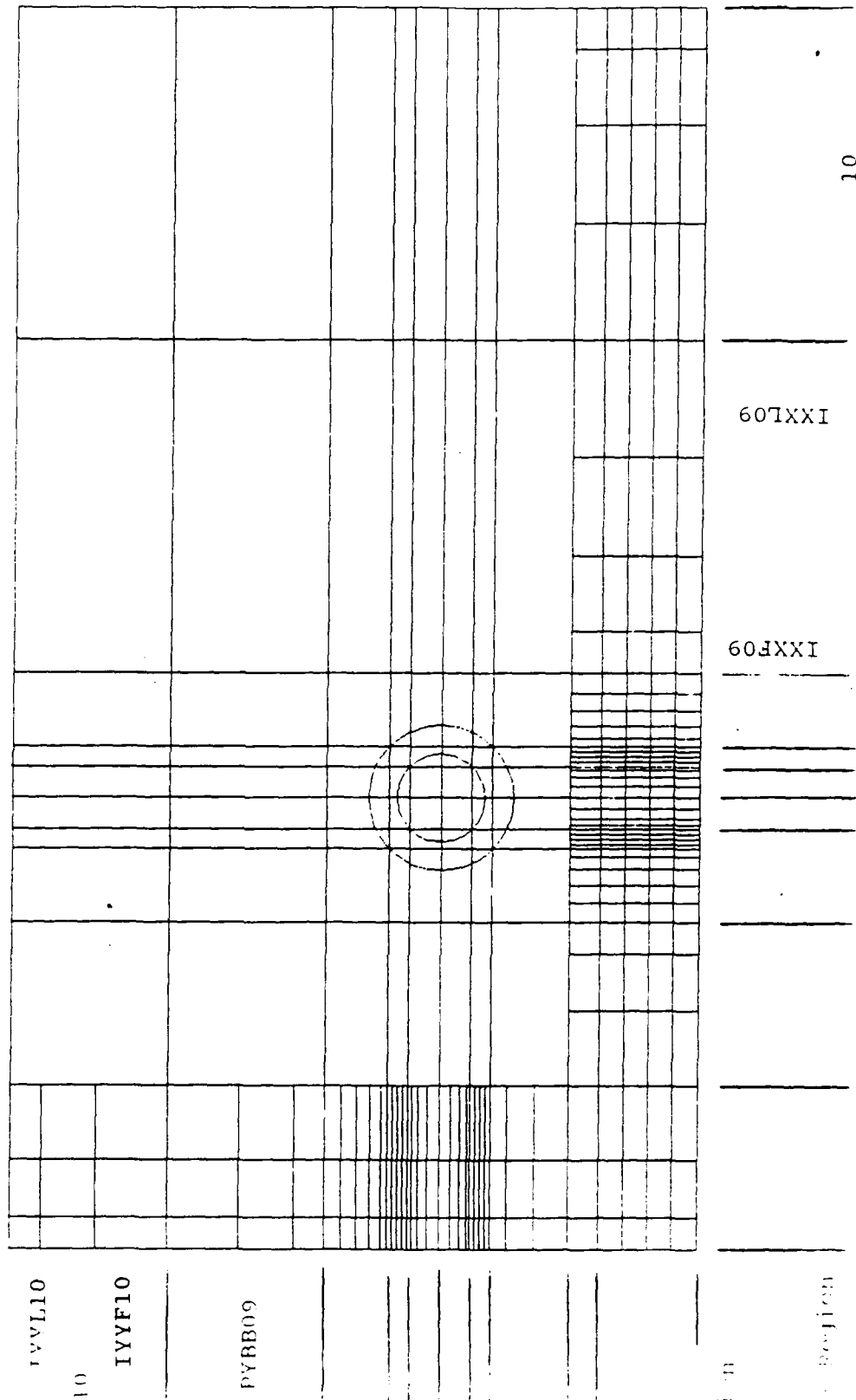


Figure 10. Enlargement of Final Grid of CS64

E22



XLBB10

10

XLBB09

IXXL09

IXXF09

Figure 11. Regions of CS65

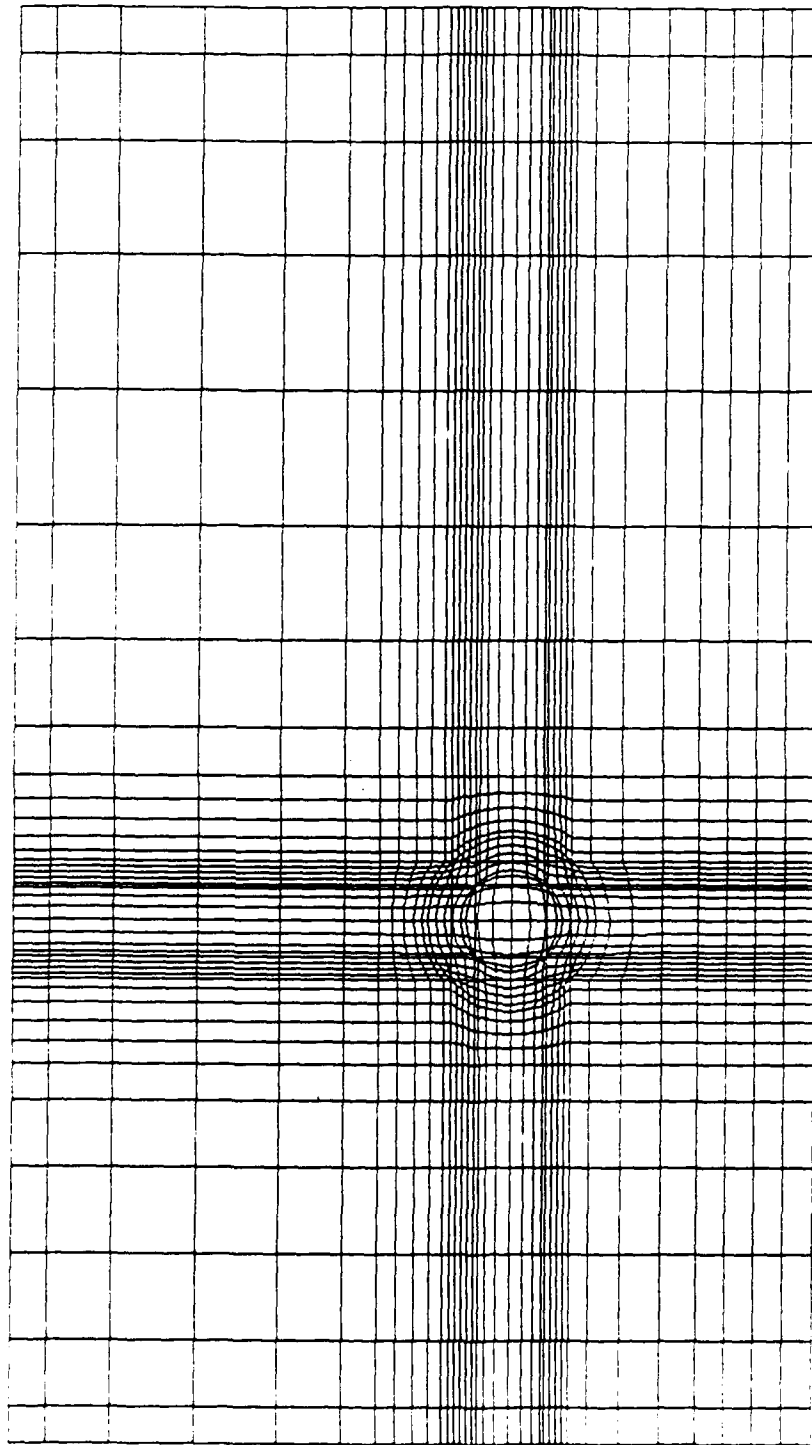


Figure 12. Initial Grid of CS65

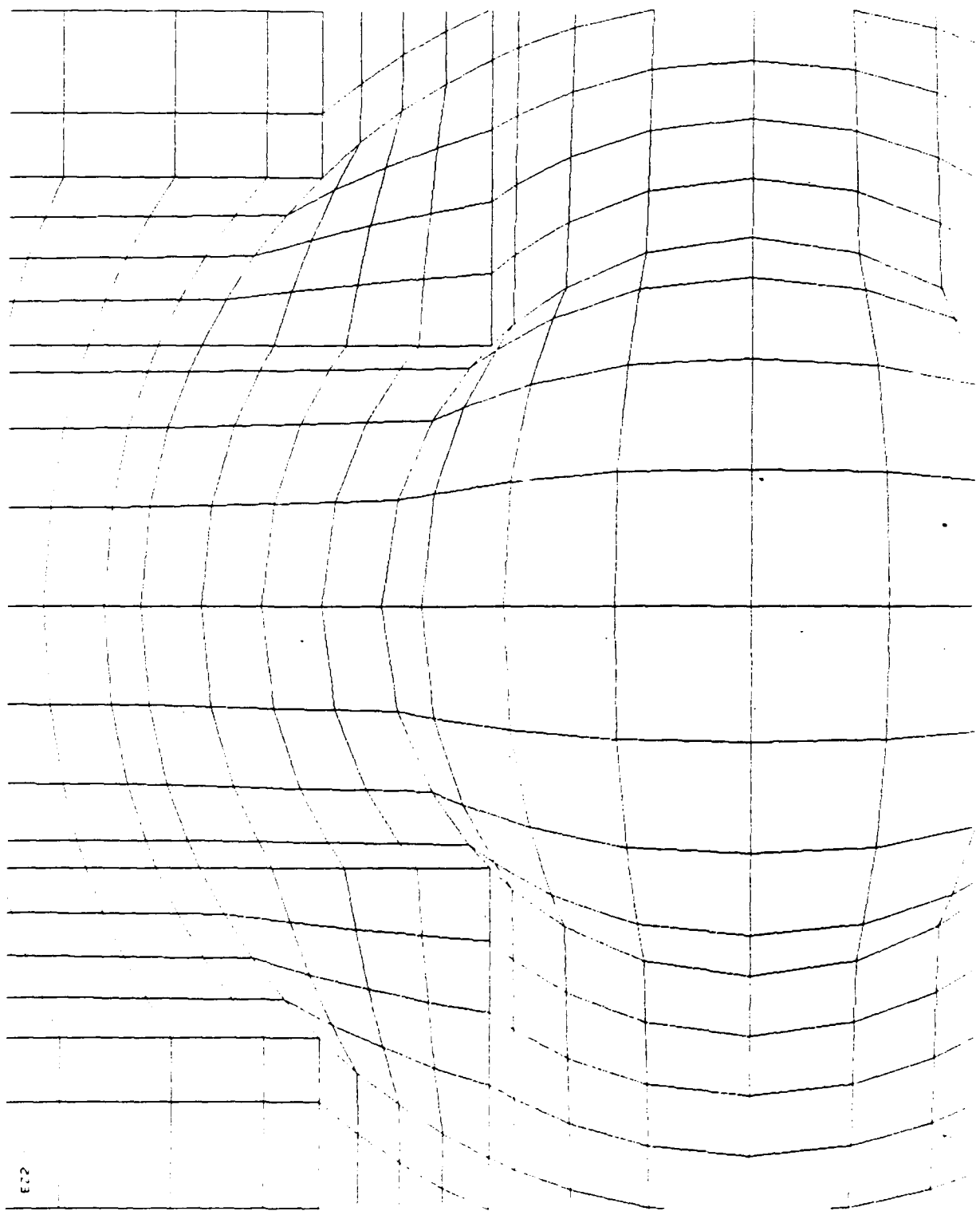


Figure 13. Enlargement of Initial Grid of CS65

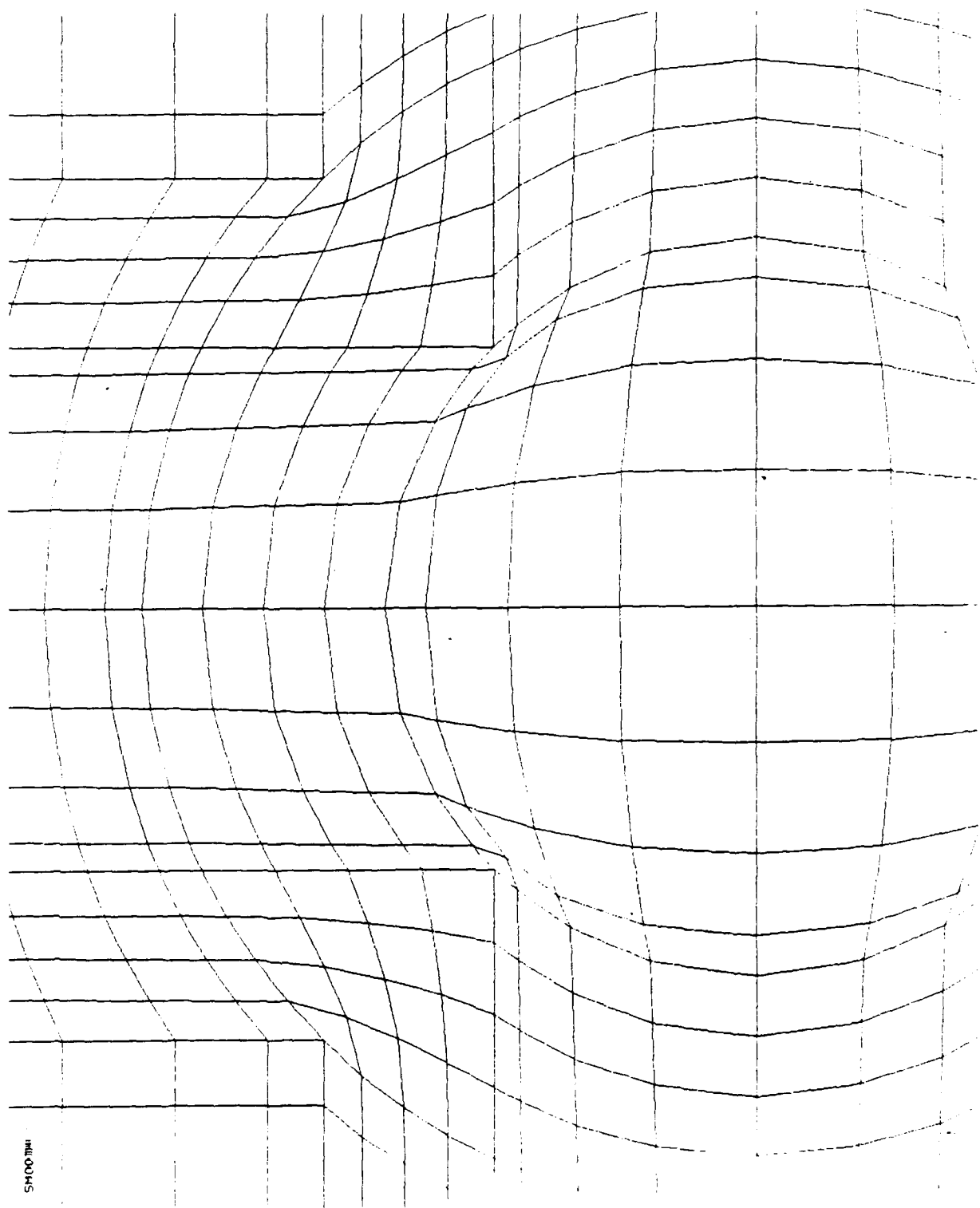


Figure 14. Enlargement of Final Grid of CS65

SHOOT

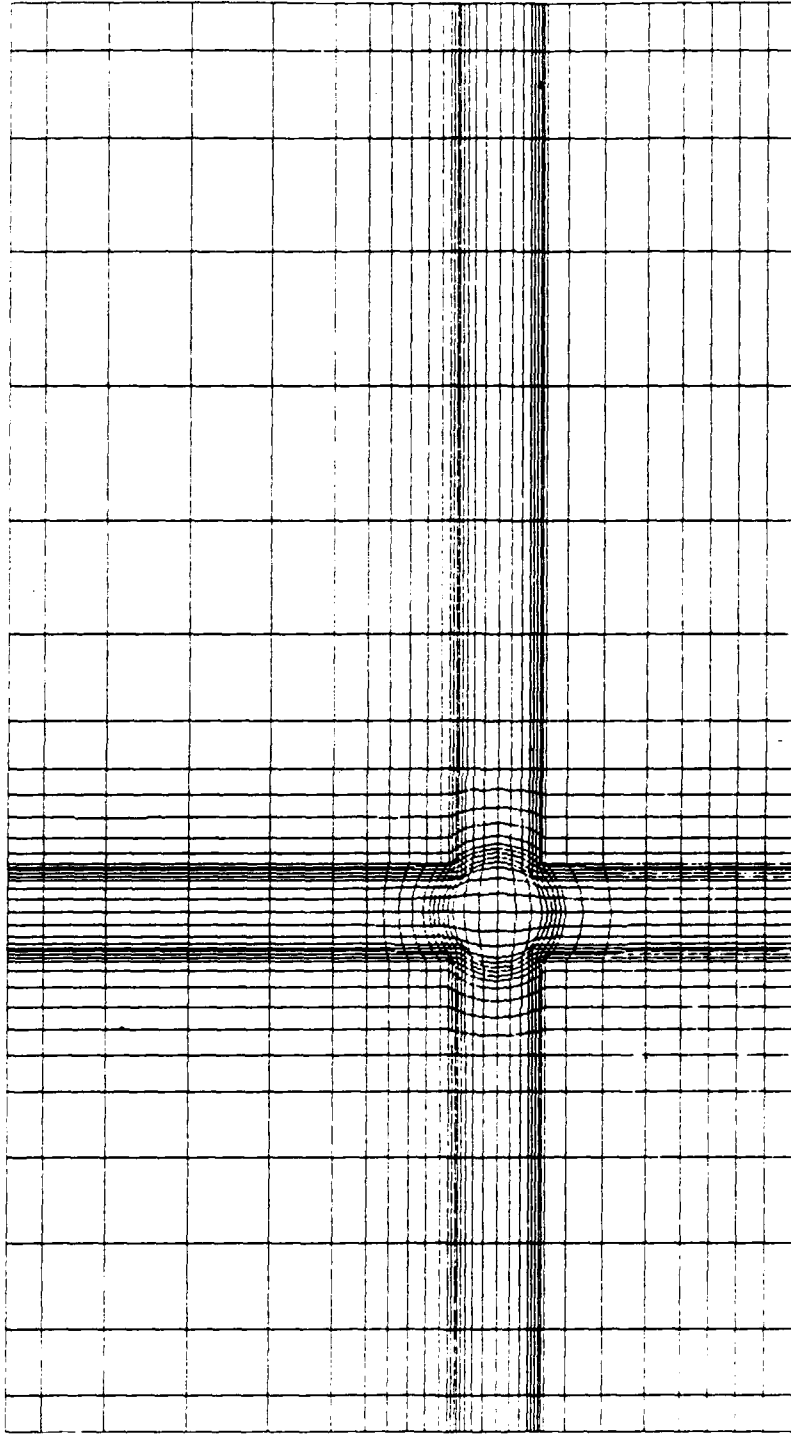
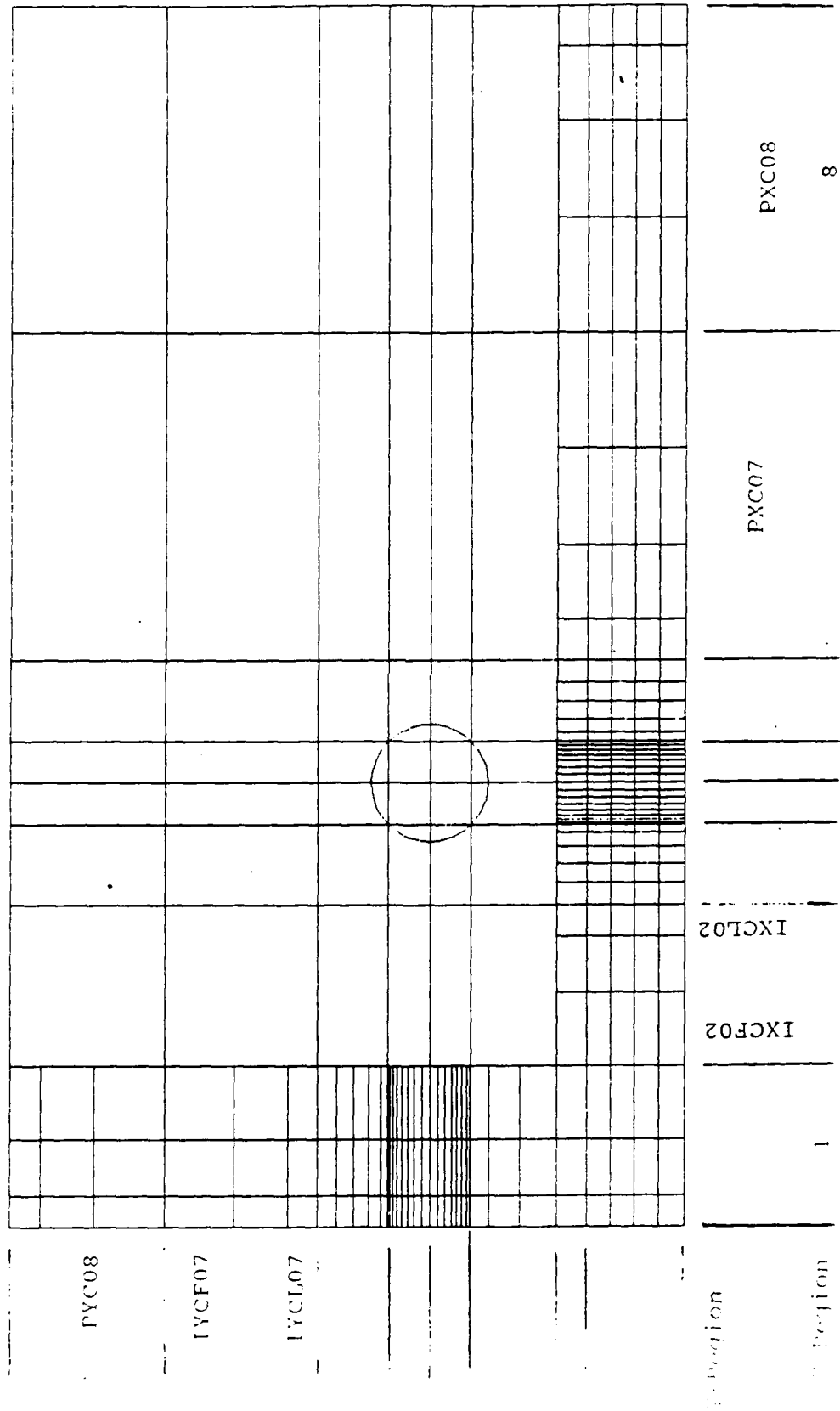


Figure 15. Final Grid of CS66

E22 >



XL08

XL07

Figure 16. Regions of CS67

E22

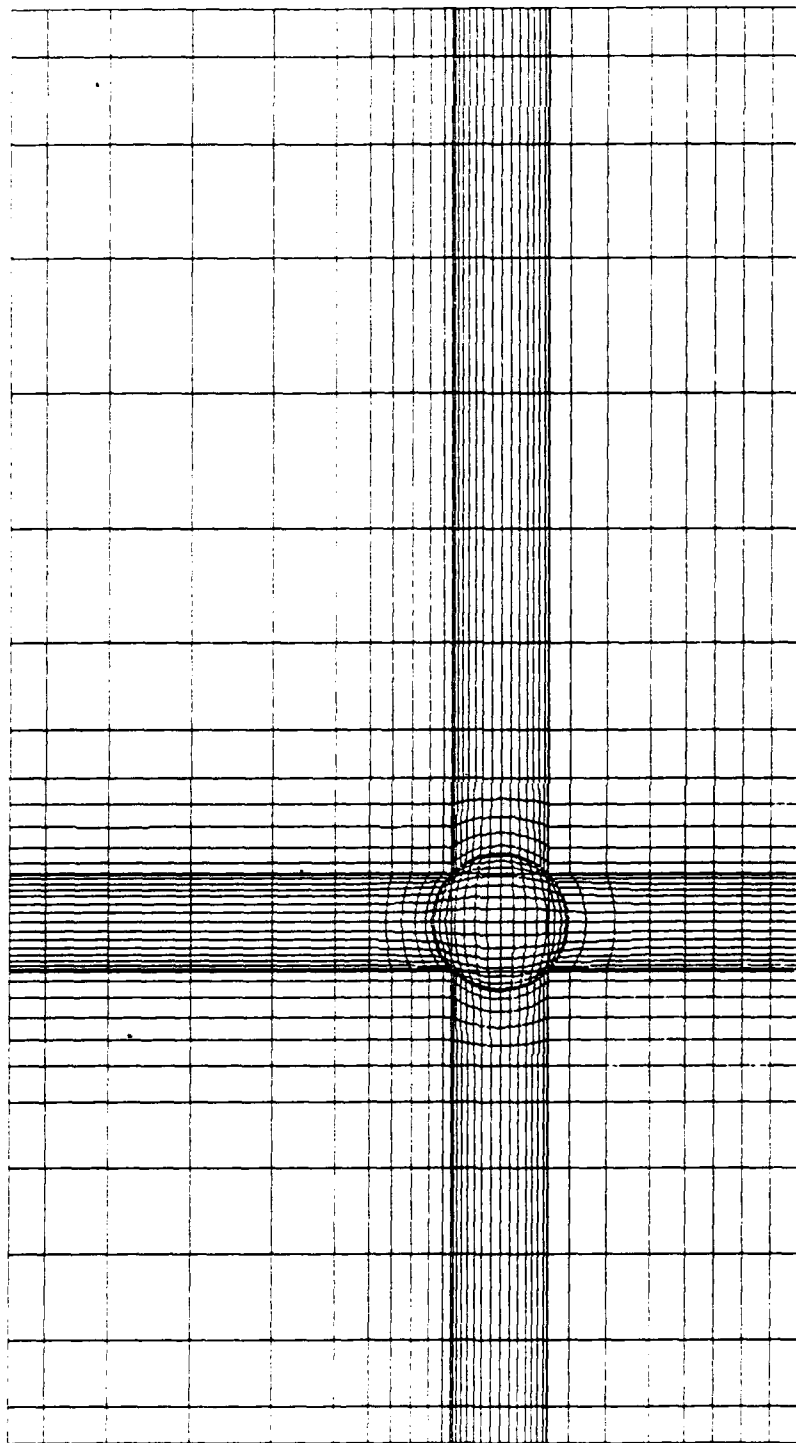


Figure 17. Initial Grid of CS67

SMOOTH

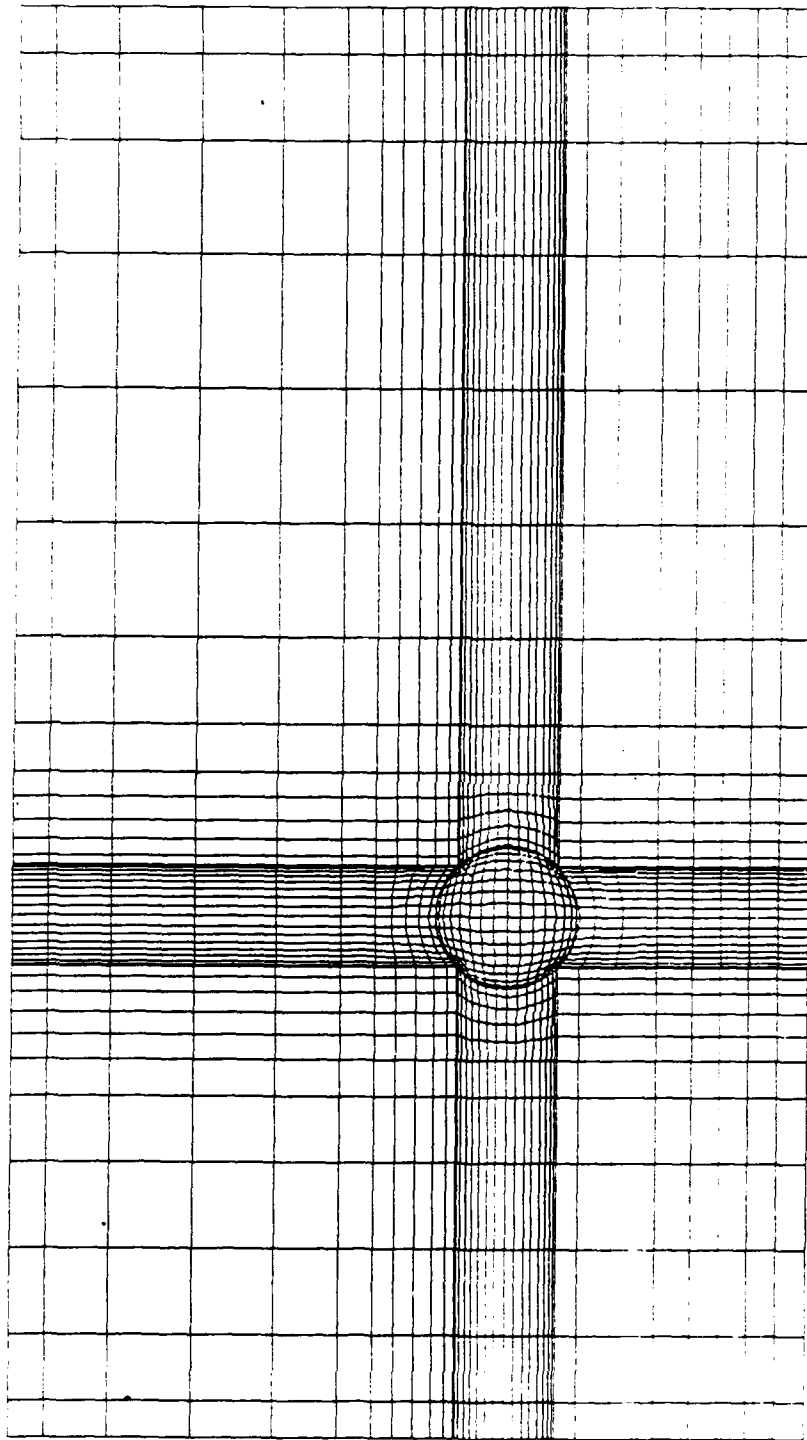


Figure 18. Final Grid of CS68

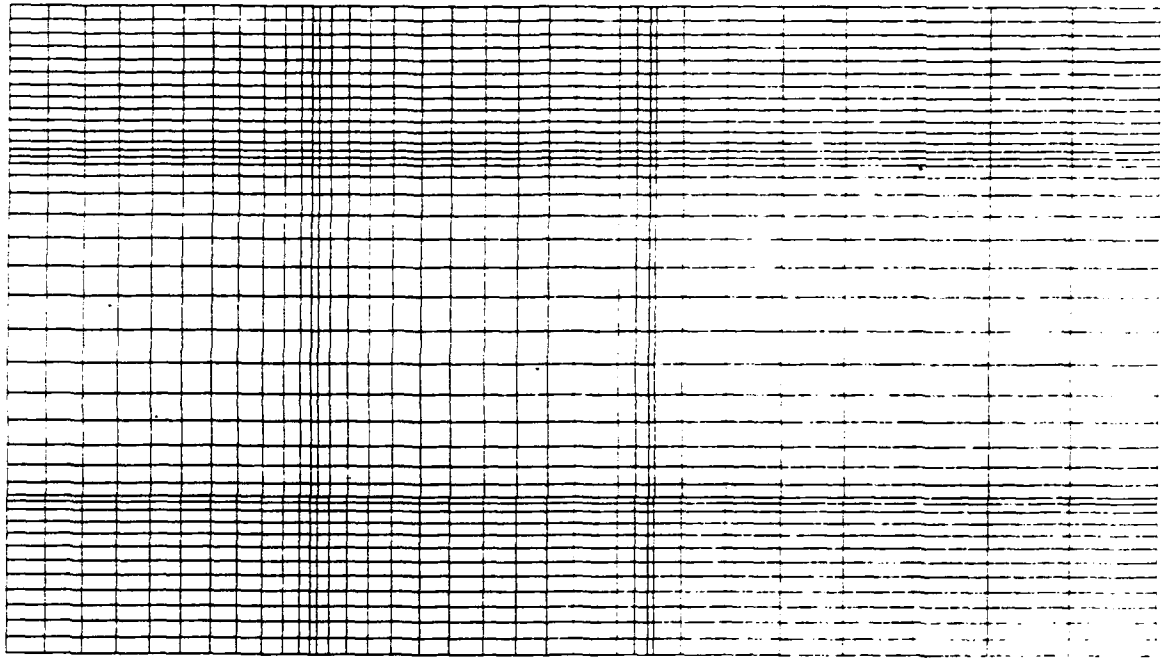


Figure 19. Grid of CS69

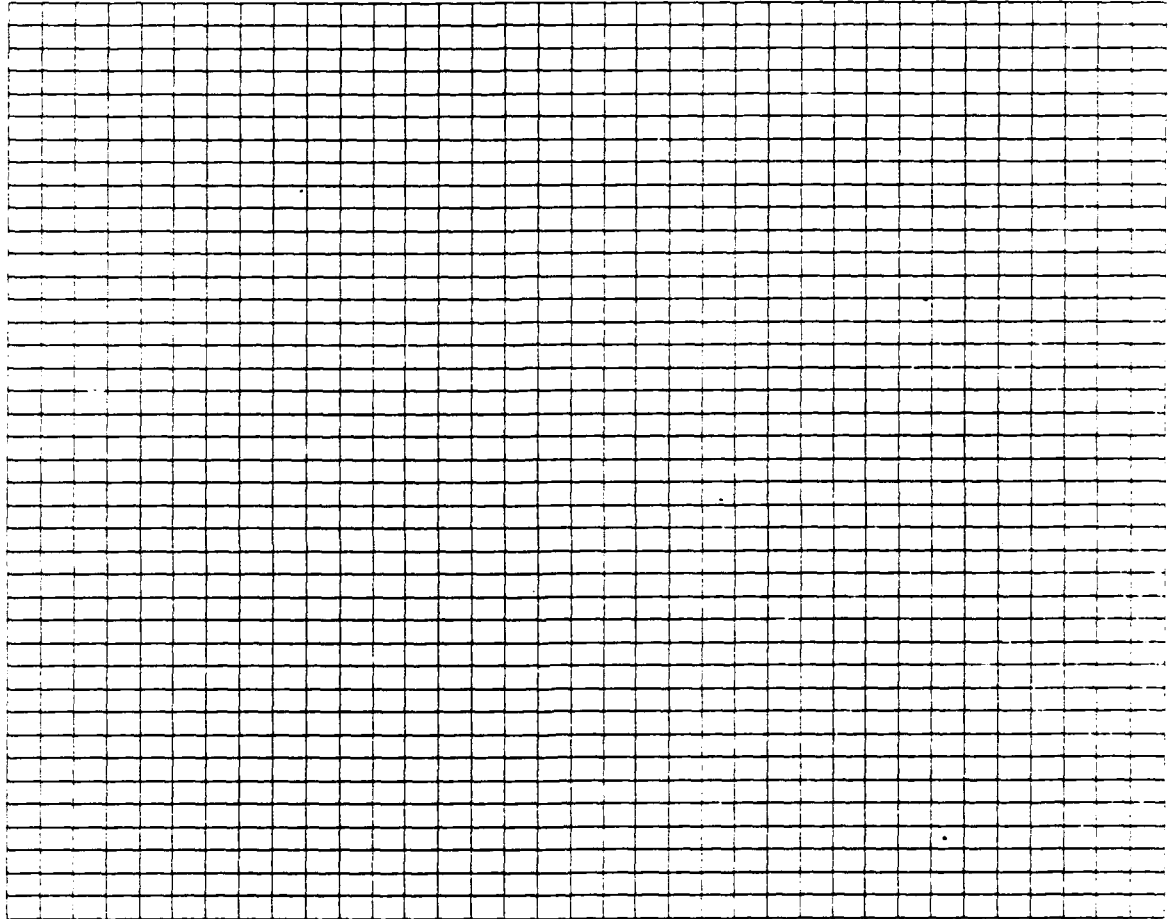
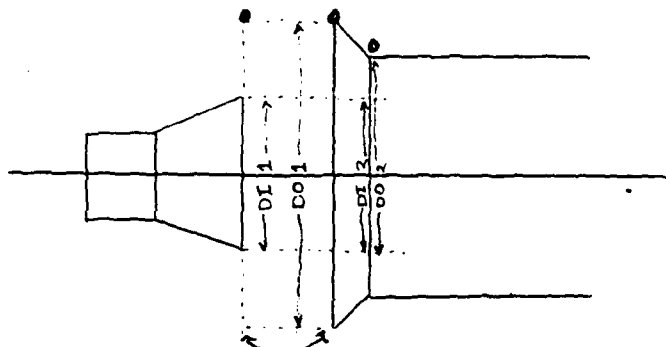


Figure 20. Grid of CS70

APPENDIX A

LOCATIONS OF DI_x + DO_x

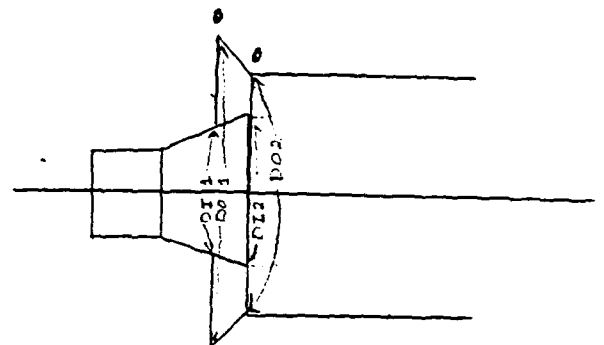
TYPE A



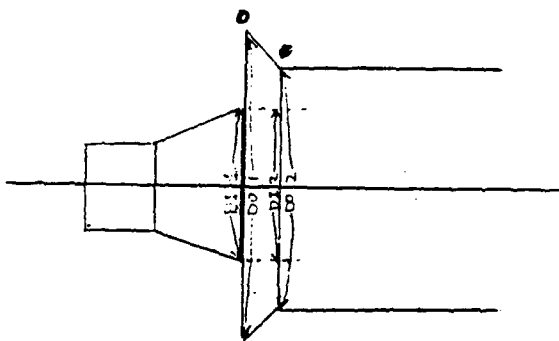
Same X-Y plane
used in these 2
locations

• X-Y plane locations

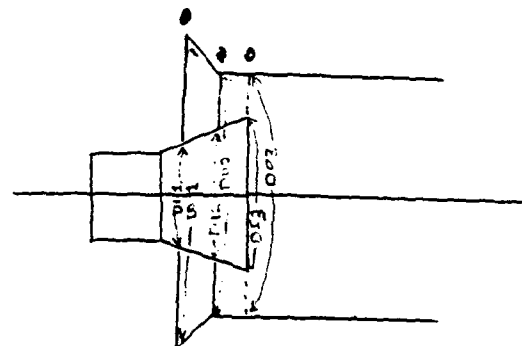
TYPE D



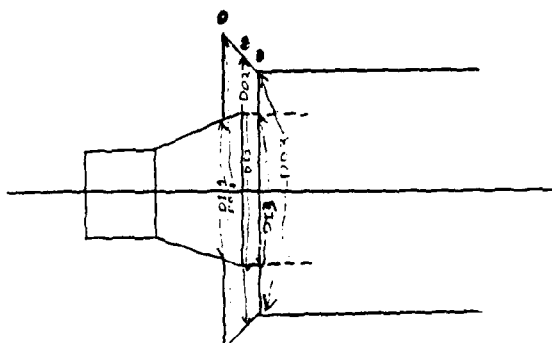
TYPE B



TYPE E



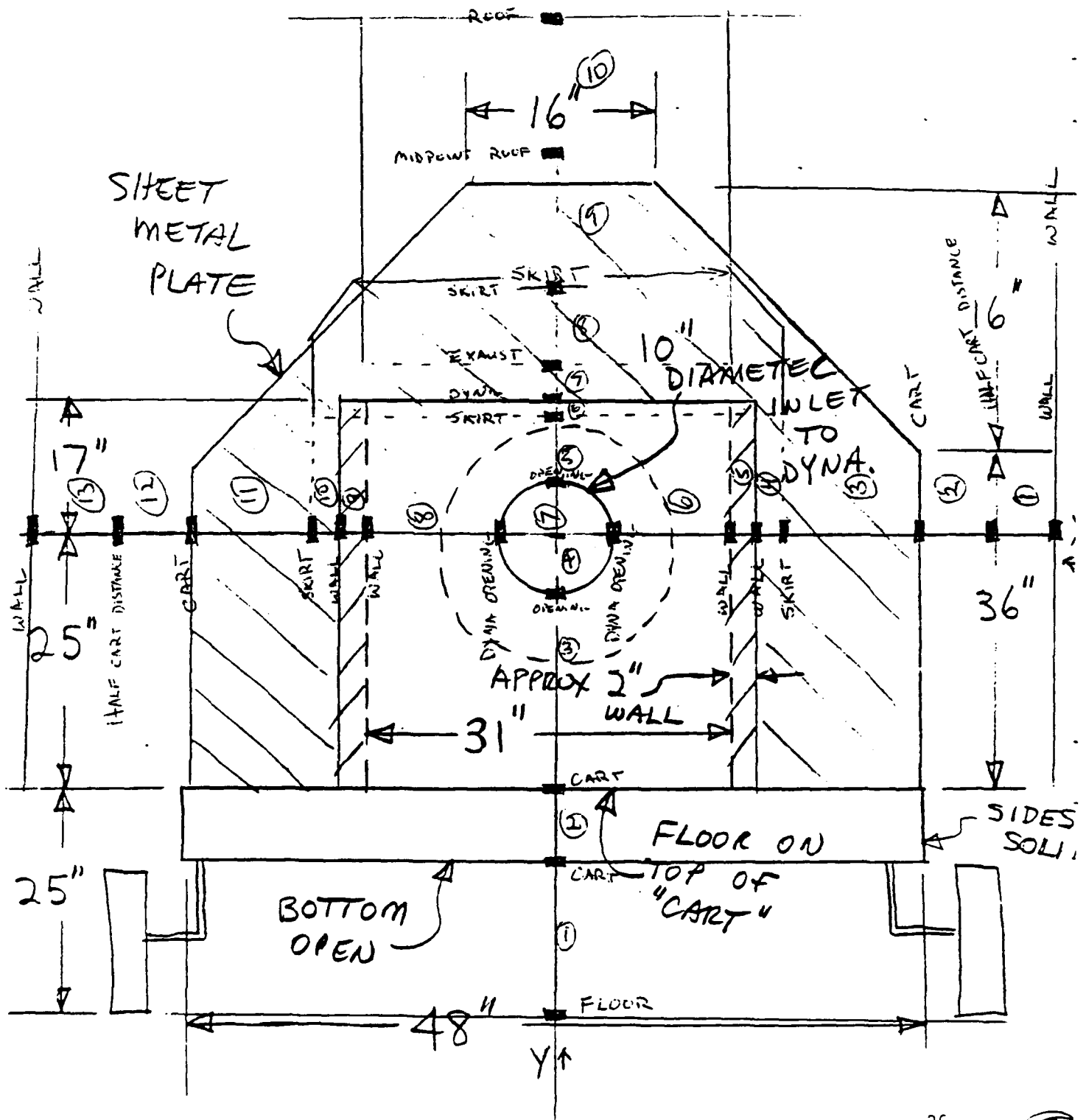
TYPE C



- D → Diameter
- I → Inner
- O → Outer
- 1 → Plane 1
- 2 → Plane 2
- 3 → Plane 3

LOCATION OF REGIONS FOR TYPE 1

DYNAMOMETER FRONT LOOKING REARWARD



NOTE: THIS IS A CUT ACROSS THE ENGINE $\pi/2$ AND AUGMENTER
NOTE: TYPE 3 CONTAINS 2 EXTRA REGIONS IN BOTH DIRECTIONS
NOTE: TYPE 4 IS LRG TYPE 2 WITH POINTS ON THE AUGMENTER

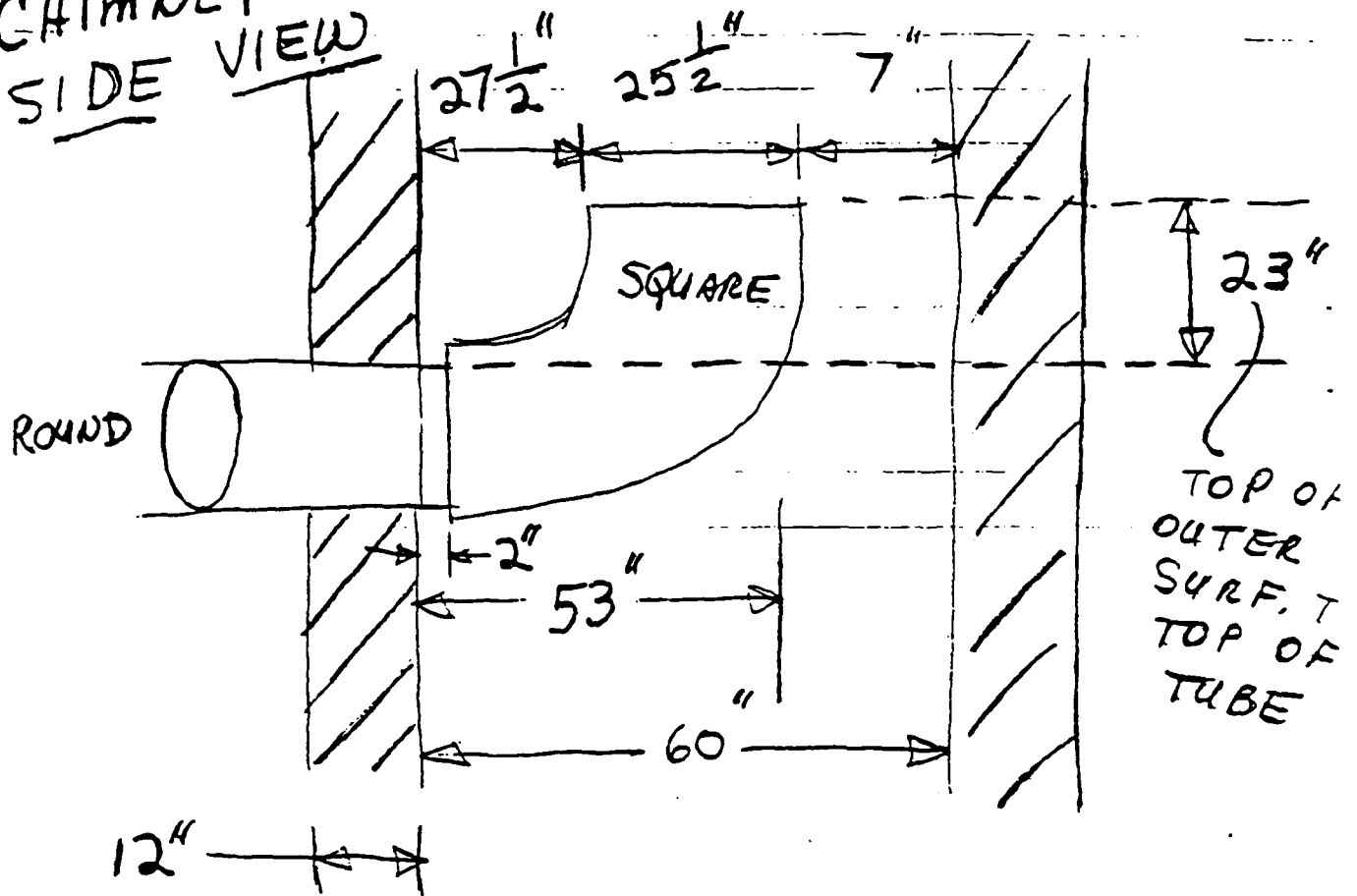
TUBE INSTEAD OF
THE ENGINE

Revisi

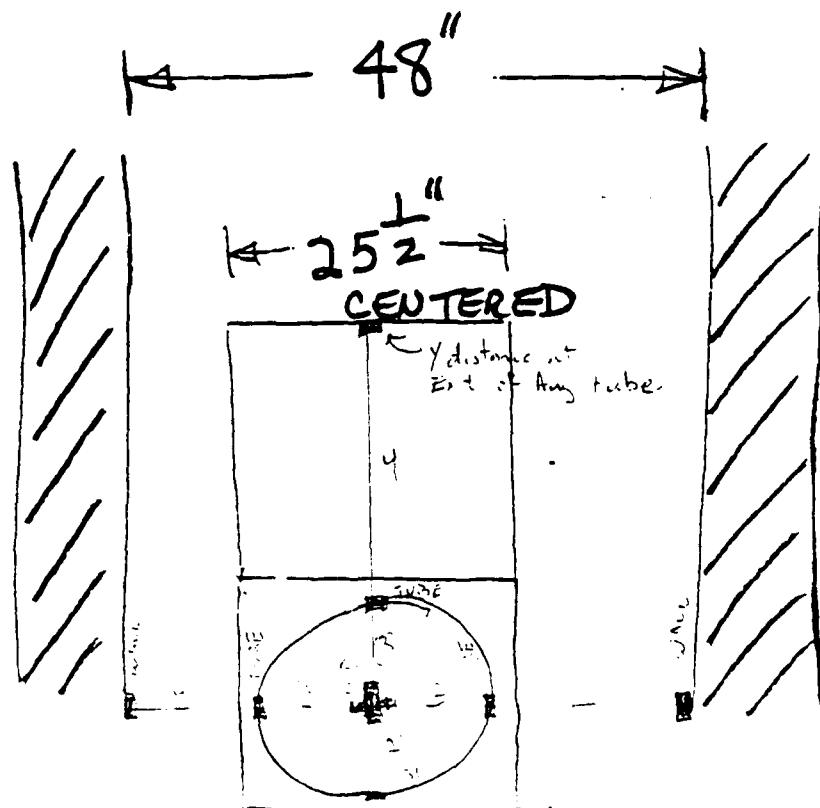


LOCATION OF REGIONS FOR TYPE 5

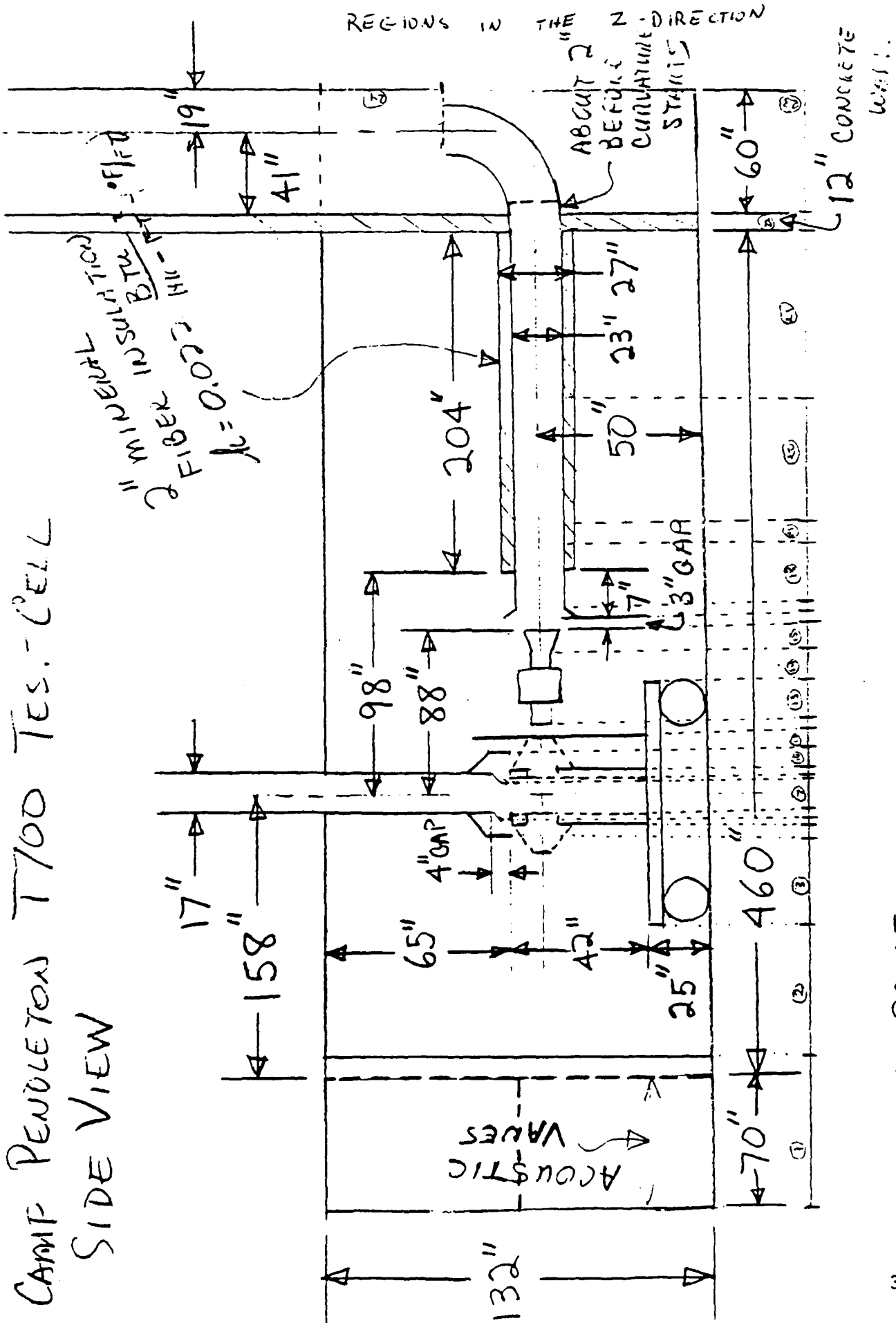
T700
CHIMNEY
SIDE VIEW



FRONT
VIEW



CAMP PENDLETON T700 TES. CELL SIDE VIEW



NOT TO SCALE
EVERYTHING IN INCHES

APPENDIX B

TALK=F;RUN(1,1)

GROUP 1. Run title and other preliminaries

TEXT(NCEL: TEST CELL--TURBOSHAFT ENGINE)

*

***** GRID SECTION *****

*** PRELIMINARY: Grid generation is an art form. This ***
*** model attempts to make this process as painless as ***
*** possible. Several assumptions will be made during ***
*** this procedure. Each will be stated at an appropriate ***
*** time. These assumptions may limit the parametric ***
*** geometrical studies that can be accomplished. ***
*** Geometric changes as called for by the contract will ***
*** be possible and fairly easy to implement. This method ***
*** will not make it easy for radical modifications to be ***
*** modeled. However, with the appropriate assistance such ***
*** changes should be possible. ***

*

*

*** PREMISE: The grid for the test cell is created from a ***
*** 2-dimensional grid generation package. This package ***
*** produces several X-Y cross sections. These sections ***
*** are then stacked, blended, or rotated to produce the ***
*** entire computational domain. In order to do this, grid ***
*** information data is supplied by the user in the Q1 ***
*** file. This information is then transferred to SATLIT ***
*** where the input files for the grid generation are ***
*** created. The user must then manually run the grid ***
*** generation program to produce a plane of X-Y grid ***
*** points for each input file. After this the user will ***
*** then rerun the preprocessor (Q1-SATLIT) at which time ***
*** the full computational grid will be produced. This ***
*** current method is not fully automated, but it requires ***
*** the user to examine each computational plane, which can ***
*** reduce grid errors. ***

*

*

*** DESCRIPTION OF PLANES: In its present form the SATLIT ***
*** will write out 5 or 6 types of X-Y planes depending on ***
*** whether a circle in circle grid is created (ie. if ***
*** nozzle is close to augments tube). Out of these ***
*** various types of planes, modifications of these types ***
*** are created (ie. the augments tube diameter changes). ***
*** For the case that is delivered, 10 planes of grids are ***
*** created. A description of each plane is now provided. ***

*** TYPE 1 -- Indicated by letter A. This type is used ***
*** for the dynamometer. There are three planes created ***
*** under this type. The first is located at the ***
*** entrance. One vertical line has been shifted to ***
*** correspond to location of the inlet. The second is ***
*** a cross section of the dynamometer with the circle ***

```

***      being a square. The third is a cross section of the      ***
***      dynamometer.                                             ***
***
***      TYPE 2 -- Indicated by letter B. This type is used      ***
***      for the engine. It creates two planes of grids, one     ***
***      for the engine inlet and one for the nozzle exit. If    ***
***      TYPE 3 is used it only creates the plane at the inlet.  ***
***
***      TYPE 3 -- Indicated by letter BB. This type is used     ***
***      only if a circle in circle grid is created. A circle   ***
***      in circle grid refers to a grid type in the X-Y plane  ***
***      that contains a circle shape in side a circle shape.   ***
***      In this case, the inner circle will represent a cross   ***
***      of the engine while the outer circle will represent a   ***
***      cross section of the augments tube. If the exit of     ***
***      the nozzle is too close to the augments tube this      ***
***      option will give better orthogonality. This type will  ***
***      produce two or three planes. If the nozzle ends before  ***
***      or at the augments lip or at the start of the constant  ***
***      cross section of the augments sleeve, two planes are    ***
***      created. The first is for the exit of the nozzle and   ***
***      the start of the lip and the second is for the start   ***
***      of the constant cross section of the sleeve. If the    ***
***      nozzle ends in the tapered (lip) section or in the     ***
***      straight section of the sleeve one additional plane    ***
***      will be needed for where the nozzle ends.              ***
***
***      TYPE 4 -- Indicated by letter C. This type is used     ***
***      to create the augments tube in the building. At the    ***
***      present time this type produces two planes. One that    ***
***      corresponds to the diameter of the sleeve and one to    ***
***      represent the diameter of the augments tube. If        ***
***      TYPE 3 is not used, the first plane is at the augments ***
***      lip.                                                     ***
***
***      TYPE 5 -- Indicated by letter D. This type is used     ***
***      to create the augments tube at the front wall of the   ***
***      chimney. The augments tube can either be a circle or   ***
***      a square. ASSUMPTION: It is assumed the center of     ***
***      curvature starts at the back wall. ASSUMPTION: If a    ***
***      square tube is used in the chimney, the circle to a    ***
***      square will be blended through the back wall of the    ***
***      building.                                               ***
***
***      TYPE 6 -- Data needed for this plane is taken from the ***
***      other types. This type creates the exit plane (top of ***
***      chimney).                                              ***
***
***      At this time each variable used in the description of ***
***      gridding in the X-direction (horizontal) will be provided. ***
***      The (0,0,0) coordinate is located (standing in front of ***
***      building) at the lower right hand corner. Parameters ***
***      are used extensively throughout this program to make ***
***      changes easier.                                         ***
***
*****
*
*
*****
*****
X-DIRECTION GRIDING
*****

```

```

*****
***
***   NRXA   --   Number of X regions for Type 1 plane   ***
***   NRXB   --   Number of X regions for Type 2 plane   ***
***   NRXBB  --   Number of X regions for Type 3 plane   ***
***   NRXC   --   Number of X regions for Type 4 plane   ***
***   NRXD   --   Number of X regions for Type 5 plane   ***
***
***   NOTE:   The number of grid cells is define for the Type 1 ***
***           plane and then redistributed for the other types. ***
***           There are fourteen available regions, some may ***
***           not be used. ***
***
***   NX01   --   Number of cells in 1st region -> Wall to ***
***           half distance cart ***
***   NX02   --   Number of cells in 2nd region -> Half ***
***           distance cart to cart ***
***   NX03   --   Number of cells in 3rd region -> Cart to ***
***           skirt ***
***   NX04   --   Number of cells in 4th region -> Skirt to ***
***           dynamometer wall ***
***   NX05   --   Number of cells in 5th region -> Dynamometer ***
***           wall to dynamometer wall ***
***   NX06   --   Number of cells in 6th region -> Dynamometer ***
***           wall to dynamometer opening ***
***   NX07   --   Number of cells in 7th region -> Dynamometer ***
***           opening to dynamometer opening ***
***   NX08   --   Number of cells in 8th region -> Dynamometer ***
***           opening to dynamometer wall ***
***   NX09   --   Number of cells in 9th region -> Dynamometer ***
***           wall to dynamometer wall ***
***   NX10   --   Number of cells in 10th region -> Dynamometer ***
***           wall to skirt ***
***   NX11   --   Number of cells in 11th region -> Skirt to ***
***           cart ***
***   NX12   --   Number of cells in 12th region -> Cart to ***
***           half distance cart ***
***   NX13   --   Number of cells in 13th region -> Half ***
***           distance cart to wall ***
***   NX14   --   Number of cells in 14th region -> Spare ***
***
***   NOTE:   The regions for the other 5 Types will now also ***
***           be defined. ***
***
***   TYPE 2 ***
***   Region 1 -- Wall to half distance cart ***
***   Region 2 -- Half distance cart to cart ***
***   Region 3 -- Cart to engine ***
***   Region 4 -- Engine to midpoint engine ***
***   Region 5 -- Midpoint engine to engine ***
***   Region 6 -- Engine to cart ***
***   Region 7 -- Cart to half distance cart ***
***   Region 8 -- Half distance cart to wall ***
***
***   TYPE 3 ***
***   Region 1 -- Wall to half distance cart ***
***   Region 2 -- Half distance cart to cart ***
***   Region 3 -- Cart to augmenter tube ***
***   Region 4 -- Augmenter tube to engine ***
***   Region 5 -- Engine to midpoint engine ***
***   Region 6 -- Midpoint engine to engine ***
***   Region 7 -- Engine to augmenter tube ***

```

```

***      Region 8  -- Augmenter tube to cart      ***
***      Region 9  -- Cart to half distance cart  ***
***      Region 10 -- Half distance cart to wall   ***
***      TYPE 4
***      Region 1  -- Wall to half distance cart  ***
***      Region 2  -- Half distance cart to cart  ***
***      Region 3  -- Cart to augmenter tube      ***
***      Region 4  -- Augmenter tube to midpoint aug tube ***
***      Region 5  -- Midpoint augmenter tube to aug tube ***
***      Region 6  -- Augmenter tube to cart      ***
***      Region 7  -- Cart to half distance cart  ***
***      Region 8  -- Half distance cart to wall   ***
***      TYPE 5
***      Region 1  -- Wall to augmenter tube      ***
***      Region 2  -- Augmenter tube to midpoint aug tube ***
***      Region 3  -- Midpoint augmenter tube to aug tube ***
***      Region 4  -- Augmenter tube to wall      ***
***      TYPE 6
***      Region 1  -- Wall to wall                ***
***
***      NXAD      -- Number of cells in X-direction picked up by ***
***                  the engine                               ***
***      NXBD      -- Number of cells in X-direction picked up by ***
***                  the augmenter tube                       ***
***      NOTE:     This last two items have corresponding parameters ***
***                  for the Y-direction. Generally their valves will ***
***                  be equal.                                  ***
***
***      IXAF**    -- First cell number of ** region Type 1      ***
***      IXAL**    -- Last cell number of ** region Type 1      ***
***      IXBF**    -- First cell number of ** region Type 2      ***
***      IXBL**    -- Last cell number of ** region Type 2      ***
***      IXXF**    -- First cell number of ** region Type 3      ***
***      IXXL**    -- Last cell number of ** region Type 3      ***
***      IXCF**    -- First cell number of ** region Type 4      ***
***      IXCL**    -- Last cell number of ** region Type 4      ***
***      IXDF**    -- First cell number of ** region Type 5      ***
***      IXDL**    -- Last cell number of ** region Type 5      ***
***
***      IXMON*    -- Location of * monitoring point (9 extra).  ***
***
***      XLA**     -- Length to end of ** region Type 1 (in)     ***
***      XLB**     -- Length to end of ** region Type 2 (in)     ***
***      XLBB**    -- Length to end of ** region Type 3 (in)     ***
***      XLC**     -- Length to end of ** region Type 4 (in)     ***
***      XLD**     -- Length to end of ** region Type 5 (in)     ***
***
***      PXA**     -- Clustering factor of ** region Type 1      ***
***      PXB**     -- Clustering factor of ** region Type 2      ***
***      PXBB**    -- Clustering factor of ** region Type 3      ***
***      PXC**     -- Clustering factor of ** region Type 4      ***
***      PXD**     -- Clustering factor of ** region Type 5      ***
***      NOTE:     Clustering factor is a number used to shift the ***
***                  cell spacing in one direction. This direction is ***
***                  controlled by setting this value to either a ***
***                  positive or negative value. The default (uniform ***
***                  spacing) is 1.0. This value may be less than or ***
***                  greater than 1.0.                               ***
***
***      NOTE:     Some Y-info defined here                    ***

```

```

*** XCENA  -- Location in the X-direction of the center of the dynamometer hole (in) ***
*** YCENA  -- Location in the Y-direction of the center of the dynamometer hole (in) ***
*** XCENB  -- Location in the X-direction of the center of the engine (in) ***
*** YCENB  -- Location in the Y-direction of the center of the engine (in) ***
*** XCENC  -- Location in the X-direction of the center of the augmeter tube in room (in) ***
*** YCENC  -- Location in the Y-direction of the center of the augmeter tube in room (in) ***
*** XCEND  -- Location in the X-direction of the center of the augmeter tube in chimney (in) ***
*** YCEND  -- Location in the Y-direction of the center of the augmeter tube in chimney (in) ***
*** DDYNA  -- Diameter of dynamometer hole (in) ***
*** DENG1  -- Diameter of engine opening (in) ***
*** DENG0  -- Diameter of engine exit (in) ***
*** DAUGL  -- Diameter of augmeter sleeve lip (in) ***
*** DAUGS  -- Diameter of augmeter sleeve (in) ***
*** DAUGT  -- Diameter of augmeter tube in room (in) ***
*** DAUGC  -- Diameter of augmeter tube in chimney (in) ***
*** NOTE:  The following input is for the circle in circle grid. The number of planes produced (either 2 or 3) is controlled by the setting of IG(60). ***
*** IG, RG, & LG are built in arrays that allow for easy transfer of integers, reals, and logicals to the various modules of the code. See report for drawings detailing these various diameters. ***
*** DI1    -- Diameter inner circle at first X-Y plane (in) ***
*** DI2    -- Diameter inner circle at second X-Y plane (in) ***
*** DI3    -- Diameter inner circle at third X-Y plane (in) ***
*** DO1    -- Diameter outer circle at first X-Y plane (in) ***
*** DO2    -- Diameter outer circle at second X-Y plane (in) ***
*** DO3    -- Diameter outer circle at third X-Y plane (in) ***
*** XINL   -- Length in X-direction of front inlet opening (in) ***
*** PI     -- PI ***
*** DYNAW  -- X-direction width of dynamometer ***
*****
*
*
*****
*** LOGICALS:  There are 4 logical flags in the Q1 file. ***
***           These are outlined below. ***
*** LG(1)    -- T if the dynamometer exhaust has a skirt ***
*** LG(2)    -- T if a circle in circle grid is used (TYPE 3) ***
*** LG(3)    -- T if augmeter tube in chimney is square ***
*** LG(4)    -- T if the inlet baffles extend into room ***
*** WARNING:  Certain lines of coding have to be activated or deactivated for certain logicals. Search ***

```

```

***          for the string &&&LG&&& to locate such coding.      ***
***          Active coding starts in the first two columns.      ***
***  NOTE:   There is certain coding that is needed for          ***
***          specific grid types.  It will be ignored if not      ***
***          needed.  Generally this type of data is indented    ***
***          by one space.                                         ***
***

```

```

*****
*
*

```

```

*****
***

```

```

***  OTHER STUFF:  Additional information is needed in the      ***
***  SATLIT to create the grid input files for the grid        ***
***  generation package.  For each type of plane in both        ***
***  the X & Y directions the user must specify what region     ***
***  the 'circle' starts on.  For instance in the              ***
***  X-direction for the Type 1 it is the sixth region,         ***
***  therefore it is passed into SATLIT in the 17 slot          ***
***  (ie IG(117)) of the last cell number.  It is assumed      ***
***  that the first X-Coordinate is 0.0.  This is the case      ***
***  in all planes except the chimney.  In this instance the    ***
***  first X-distance is passed to SATLIT through the RG        ***
***  array element that is 10 above the logical unit used       ***
***  to write out the grid data file.  For this case it is     ***
***  the 9th plane (LU=69) and RG(79) is set to 63.0 inches    ***
***

```

```

*****
*
*
*XXXXXXXXXXXXXXXXXXXXX  DECLARE X  XXXXXXXXXXXXXXXXXXXXXXXXXXXX*
*
*

```

```

INTEGER(NRXA,NRXB,NRXBB,NRXC,NRXD)
INTEGER(NX01,NX02,NX03,NX04,NX05,NX06,NX07,NX08,NX09,NX10)
INTEGER(NX11,NX12,NX13,NX14)
INTEGER(NXAD,NXBD)
INTEGER(IXAF01,IXAF02,IXAF03,IXAF04,IXAF05)
INTEGER(IXAF06,IXAF07,IXAF08,IXAF09,IXAF10)
INTEGER(IXAF11,IXAF12,IXAF13,IXAF14,IXAF15)
INTEGER(IXAL01,IXAL02,IXAL03,IXAL04,IXAL05)
INTEGER(IXAL06,IXAL07,IXAL08,IXAL09,IXAL10)
INTEGER(IXAL11,IXAL12,IXAL13,IXAL14,IXAL15)
INTEGER(IXBF01,IXBF02,IXBF03,IXBF04,IXBF05)
INTEGER(IXBF06,IXBF07,IXBF08,IXBF09,IXBF10)
INTEGER(IXBF11,IXBF12,IXBF13,IXBF14,IXBF15)
INTEGER(IXBL01,IXBL02,IXBL03,IXBL04,IXBL05)
INTEGER(IXBL06,IXBL07,IXBL08,IXBL09,IXBL10)
INTEGER(IXBL11,IXBL12,IXBL13,IXBL14,IXBL15)
INTEGER(IXXF01,IXXF02,IXXF03,IXXF04,IXXF05)
INTEGER(IXXF06,IXXF07,IXXF08,IXXF09,IXXF10)
INTEGER(IXXF11,IXXF12,IXXF13,IXXF14,IXXF15)
INTEGER(IXXL01,IXXL02,IXXL03,IXXL04,IXXL05)
INTEGER(IXXL06,IXXL07,IXXL08,IXXL09,IXXL10)
INTEGER(IXXL11,IXXL12,IXXL13,IXXL14,IXXL15)
INTEGER(IXCF01,IXCF02,IXCF03,IXCF04,IXCF05)
INTEGER(IXCF06,IXCF07,IXCF08,IXCF09,IXCF10)
INTEGER(IXCF11,IXCF12,IXCF13,IXCF14,IXCF15)
INTEGER(IXCL01,IXCL02,IXCL03,IXCL04,IXCL05)
INTEGER(IXCL06,IXCL07,IXCL08,IXCL09,IXCL10)

```

```

INTEGER (IXCL11,IXCL12,IXCL13,IXCL14,IXCL15)
INTEGER (IXDF01,IXDF02,IXDF03,IXDF04,IXDF05)
INTEGER (IXDF06,IXDF07,IXDF08,IXDF09,IXDF10)
INTEGER (IXDF11,IXDF12,IXDF13,IXDF14,IXDF15)
INTEGER (IXDL01,IXDL02,IXDL03,IXDL04,IXDL05)
INTEGER (IXDL06,IXDL07,IXDL08,IXDL09,IXDL10)
INTEGER (IXDL11,IXDL12,IXDL13,IXDL14,IXDL15)
INTEGER (IXMON1,IXMON2,IXMON3,IXMON4,IXMON5)
INTEGER (IXMON6,IXMON7,IXMON8,IXMON9,ITMP)
REAL (XLA01,XLA02,XLA03,XLA04,XLA05)
REAL (XLA06,XLA07,XLA08,XLA09,XLA10)
REAL (XLA11,XLA12,XLA13,XLA14,XLA15)
REAL (XLB01,XLB02,XLB03,XLB04,XLB05)
REAL (XLB06,XLB07,XLB08,XLB09,XLB10)
REAL (XLB11,XLB12,XLB13,XLB14,XLB15)
REAL (XLBB01,XLBB02,XLBB03,XLBB04,XLBB05)
REAL (XLBB06,XLBB07,XLBB08,XLBB09,XLBB10)
REAL (XLBB11,XLBB12,XLBB13,XLBB14,XLBB15)
REAL (XLC01,XLC02,XLC03,XLC04,XLC05)
REAL (XLC06,XLC07,XLC08,XLC09,XLC10)
REAL (XLC11,XLC12,XLC13,XLC14,XLC15)
REAL (XLD01,XLD02,XLD03,XLD04,XLD05)
REAL (XLD06,XLD07,XLD08,XLD09,XLD10)
REAL (XLD11,XLD12,XLD13,XLD14,XLD15)
REAL (PXA01,PXA02,PXA03,PXA04,PXA05)
REAL (PXA06,PXA07,PXA08,PXA09,PXA10)
REAL (PXA11,PXA12,PXA13,PXA14,PXA15)
REAL (PXB01,PXB02,PXB03,PXB04,PXB05)
REAL (PXB06,PXB07,PXB08,PXB09,PXB10)
REAL (PXB11,PXB12,PXB13,PXB14,PXB15)
REAL (PXBB01,PXBB02,PXBB03,PXBB04,PXBB05)
REAL (PXBB06,PXBB07,PXBB08,PXBB09,PXBB10)
REAL (PXBB11,PXBB12,PXBB13,PXBB14,PXBB15)
REAL (PXC01,PXC02,PXC03,PXC04,PXC05)
REAL (PXC06,PXC07,PXC08,PXC09,PXC10)
REAL (PXC11,PXC12,PXC13,PXC14,PXC15)
REAL (PXD01,PXD02,PXD03,PXD04,PXD05)
REAL (PXD06,PXD07,PXD08,PXD09,PXD10)
REAL (PXD11,PXD12,PXD13,PXD14,PXD15)
REAL (XCENA,YCENA,XCENB,YCENB,XCENC,YCENC,XCEND,YCEND)
REAL (DDYNA,DENGI,DENGO,DAUGL,DAUGS,DAUGT,DAUGC,XINL)
REAL (DI1,DI2,DI3,DO1,DO2,DO3)
REAL (PI,DYNAW)
*
*
*XXXXXXXXXXXXXXXXXXXXX      LOGICALS      XXXXXXXXXXXXXXXXXXXXXXXX*
*
*
LG(1)=T
LG(2)=T
LG(3)=T
LG(4)=T
*
*
*XXXXXXXXXXXXXXXXXXXXX      CIRCLE CENTERS & DIAMETERS      YXXXXXXXXXXXXX*
*
*
PI=3.141592654
XCENA=87.0;                RG(41)=XCENA
YCENA=50.0;                RG(42)=YCENA

```

XCENB=XCENA;	RG(43)=XCENB
YCENB=YCENA;	RG(44)=YCENB
XCENC=XCENA+0.0;	RG(45)=XCENC
YCENC=YCENA+0.0;	RG(46)=YCENC
XCEND=XCENA;	RG(47)=XCEND
YCEND=YCENA;	RG(48)=YCEND

DDYNA=10.0;	RG(50)=DDYNA
DENGI=14.0;	RG(51)=DENGI

@@@@@@ SPECIFY WHEN USING TYPE 3 @@@@@@@

IG(60)=2	
DI1=17.0;	RG(54)=DI1
DI2=17.0;	RG(55)=DI2
DI3=16.0;	RG(56)=DI3
DO1=28.0;	RG(57)=DO1
DO2=23.0;	RG(58)=DO2
DO3=22.0;	RG(59)=DO3

@@@@@@ SPECIFY WHEN NOT USING TYPE 3 @@@@@@@

DENGO=17.0;	RG(52)=DENGO
DAUGL=28.0;	RG(53)=DAUGL

DAUGS=23.0;	RG(60)=DAUGS
DAUGT=24.0;	RG(61)=DAUGT
DAUGC=25.5;	RG(62)=DAUGC

XINL=108.0;	RG(65)=XINL
-------------	-------------

*

*

XXXXXXXXXXXXXXXXXXXXX TYPE 1 DATA XXXXXXXXXXXXXXXXXXXXXXXX

*

*

NRXA=13;	IG(42)=NRXA
----------	-------------

NX01=3

NX02=3

NX03=2

NX04=2

NX05=1

NX06=5

MUST BE EVEN FOR CELLS IN CIRCLE

NX07=6

NX08=5

NX09=1

NX10=2

NX11=2

NX12=4

NX13=4

IXAF01=	1;	IXAL01=	NX01
IXAF02=IXAL01+1;		IXAL02=IXAL01+N	X02
IXAF03=IXAL02+1;		IXAL03=IXAL02+N	X03
IXAF04=IXAL03+1;		IXAL04=IXAL03+N	X04
IXAF05=IXAL04+1;		IXAL05=IXAL04+N	X05
IXAF06=IXAL05+1;		IXAL06=IXAL05+N	X06
IXAF07=IXAL06+1;		IXAL07=IXAL06+N	X07
IXAF08=IXAL07+1;		IXAL08=IXAL07+N	X08
IXAF09=IXAL08+1;		IXAL09=IXAL08+N	X09
IXAF10=IXAL09+1;		IXAL10=IXAL09+N	X10
IXAF11=IXAL10+1;		IXAL11=IXAL10+N	X11

```
IXAF12=IXAL11+1;      IXAL12=IXAL11+NX12
IXAF13=IXAL12+1;      IXAL13=IXAL12+NX13
```

```
XLA01= 31.500000;      PXA01= 1.5
XLA02= 63.000000;      PXA02=-1.5
XLA03= 67.500000;      PXA03= 1.0
XLA04= 69.500000;      PXA04= 1.0
XLA05= 71.500000;      PXA05= 1.0
XLA06=  0.000000;      PXA06= 1.0
XLA07=  0.000000;      PXA07= 1.0
XLA08=102.500000;      PXA08= 1.0
XLA09=104.500000;      PXA09= 1.0
XLA10=106.500000;      PXA10= 1.0
XLA11=111.000000;      PXA11= 1.0
XLA12=175.500000;      PXA12= 1.5
XLA13=240.000000;      PXA13=-1.5
DYNAB=XLA08-XLA05
```

```
IG(101)=IXAL01;RG(101)=XLA01;RG(121)=PXA01
IG(102)=IXAL02;RG(102)=XLA02;RG(122)=PXA02
IG(103)=IXAL03;RG(103)=XLA03;RG(123)=PXA03
IG(104)=IXAL04;RG(104)=XLA04;RG(124)=PXA04
IG(105)=IXAL05;RG(105)=XLA05;RG(125)=PXA05
IG(106)=IXAL06;RG(106)=XLA06;RG(126)=PXA06
IG(107)=IXAL07;RG(107)=XLA07;RG(127)=PXA07
IG(108)=IXAL08;RG(108)=XLA08;RG(128)=PXA08
IG(109)=IXAL09;RG(109)=XLA09;RG(129)=PXA09
IG(110)=IXAL10;RG(110)=XLA10;RG(130)=PXA10
IG(111)=IXAL11;RG(111)=XLA11;RG(131)=PXA11
IG(112)=IXAL12;RG(112)=XLA12;RG(132)=PXA12
IG(113)=IXAL13;RG(113)=XLA13;RG(133)=PXA13
IG(117)=7
```

*

*

XXXXXXXXXXXXXXXXXXXXX TYPE 2 DATA XXXXXXXXXXXXXXXXXXXXXXXX

*

*

```
NXAD=1
NRXB=8;      IG(44)=NRXB
IXBF01=IXAF01; IXBL01=IXAL01
IXBF02=IXAF02; IXBL02=IXAL02
IXBF03=IXAF03; IXBL03=IXAL06-NXAD
IXBF04=IXAF07-NXAD; IXBL04=IXAL07-NX07/2
IXBF05=IXAF08-NX07/2; IXBL05=IXAL07+NXAD
IXBF06=IXAF08+NXAD; IXBL06=IXAL11
IXBF07=IXAF12; IXBL07=IXAL12
IXBF08=IXAF13; IXBL08=IXAL13
```

```
XLB01= XLA01;      PXB01= PXA01
XLB02= XLA02;      PXB02= PXA02
XLB03=  0.000000;      PXB03=-1.4
XLB04= XCENB;      PXB04= 1.6
XLB05=  0.000000;      PXB05=-1.6
XLB06= XLA11;      PXB06= 1.4
XLB07= XLA12;      PXB07= PXA12
XLB08= XLA13;      PXB08= PXA13
```

```
IG(141)=IXBL01;RG(181)=XLB01;RG(201)=PXB01
IG(142)=IXBL02;RG(182)=XLB02;RG(202)=PXB02
IG(143)=IXBL03;RG(183)=XLB03;RG(203)=PXB03
```

```

IG(144)=IXBL04;RG(184)=XLB04;RG(204)=PXB04
IG(145)=IXBL05;RG(185)=XLB05;RG(205)=PXB05
IG(146)=IXBL06;RG(186)=XLB06;RG(206)=PXB06
IG(147)=IXBL07;RG(187)=XLB07;RG(207)=PXB07
IG(148)=IXBL08;RG(188)=XLB08;RG(208)=PXB08
IG(157)=4

```

*

*

```

*XXXXXXXXXXXXXXXXXXXXX TYPE 3 DATA XXXXXXXXXXXXXXXXXXXXXXXX*

```

*

*

NXBD=4

@@@@@@ SPECIFY WHEN USING TYPE 3 @@@@@@

```

NRXBB=10; IG(46)=NRXBB
IXXF01=IXBF01; IXXL01=IXBL01
IXXF02=IXBF02; IXXL02=IXBL02
IXXF03=IXBF03; IXXL03=IXBL03-NXBD
IXXF04=IXBF04-NXBD; IXXL04=IXBL03
IXXF05=IXBF04; IXXL05=IXBL04
IXXF06=IXBF05; IXXL06=IXBL05
IXXF07=IXBF06; IXXL07=IXBL05+NXBD
IXXF08=IXBF06+NXBD; IXXL08=IXBL06
IXXF09=IXBF07; IXXL09=IXBL07
IXXF10=IXBF08; IXXL10=IXBL08

```

```

XLBB01= XLA01; PXBB01= PXA01
XLBB02= XLA02; PXBB02= PXA02
XLBB03= 0.000000; PXBB03= PXB03
XLBB04= 0.000000; PXBB04= 1.0
XLBB05= XCENB; PXBB05= PXB04
XLBB06= 0.000000; PXBB06= PXB05
XLBB07= 0.000000; PXBB07= 1.0
XLBB08= XLA11; PXBB08= PXB06
XLBB09= XLA12; PXBB09= PXA12
XLBB10= XLA13; PXBB10= PXA13

```

```

IG(181)=IXXL01;RG(261)=XLBB01;RG(281)=PXBB01
IG(182)=IXXL02;RG(262)=XLBB02;RG(282)=PXBB02
IG(183)=IXXL03;RG(263)=XLBB03;RG(283)=PXBB03
IG(184)=IXXL04;RG(264)=XLBB04;RG(284)=PXBB04
IG(185)=IXXL05;RG(265)=XLBB05;RG(285)=PXBB05
IG(186)=IXXL06;RG(266)=XLBB06;RG(286)=PXBB06
IG(187)=IXXL07;RG(267)=XLBB07;RG(287)=PXBB07
IG(188)=IXXL08;RG(268)=XLBB08;RG(288)=PXBB08
IG(189)=IXXL09;RG(269)=XLBB09;RG(289)=PXBB09
IG(190)=IXXL10;RG(270)=XLBB10;RG(290)=PXBB10
IG(197)=4

```

*

*

```

*XXXXXXXXXXXXXXXXXXXXX TYPE 4 DATA XXXXXXXXXXXXXXXXXXXXXXXX*

```

*

*

```

NRXC=8; IG(48)=NRXC
IXCF01=IXBF01; IXCL01=IXBL01
IXCF02=IXBF02; IXCL02=IXBL02
IXCF03=IXBF03; IXCL03=IXBL03-NXBD
IXCF04=IXBF04-NXBD; IXCL04=IXBL04
IXCF05=IXBF05; IXCL05=IXBL05+NXBD
IXCF06=IXBF06+NXBD; IXCL06=IXBL06

```

IXCF07=IXBF07; IXCL07=IXBL07
IXCF08=IXBF08; IXCL08=IXBL08

XLC01= XLA01; PXC01= PXA01
XLC02= XLA02; PXC02= PXA02
XLC03= 0.000000; PXC03= PXB03
XLC04= XCENC; PXC04= PXB04
XLC05= 0.000000; PXC05= PXB05
XLC06= XLA11; PXC06= PXB06
XLC07= XLA12; PXC07= PXA12
XLC08= XLA13; PXC08= PXA13

IG(221)=IXCL01;RG(341)=XLC01;RG(361)=PXC01
IG(222)=IXCL02;RG(342)=XLC02;RG(362)=PXC02
IG(223)=IXCL03;RG(343)=XLC03;RG(363)=PXC03
IG(224)=IXCL04;RG(344)=XLC04;RG(364)=PXC04
IG(225)=IXCL05;RG(345)=XLC05;RG(365)=PXC05
IG(226)=IXCL06;RG(346)=XLC06;RG(366)=PXC06
IG(227)=IXCL07;RG(347)=XLC07;RG(367)=PXC07
IG(228)=IXCL08;RG(348)=XLC08;RG(368)=PXC08
IG(237)=4

*

*

XXXXXXXXXXXXXXXXXXXXX TYPE 5 DATA XXXXXXXXXXXXXXXXXXXXXXXX

*

*

NRXD=4; IG(50)=NRXD
IXDF01=IXCF01; IXDL01=IXCL03
IXDF02=IXCF04; IXDL02=IXCL04
IXDF03=IXCF05; IXDL03=IXCL05
IXDF04=IXCF06; IXDL04=IXCL08

ITMP=77+IG(60)

RG(ITMP)=63.0

XLD01= 0.000000; PXD01=-1.2
XLD02= XCEND; PXD02= PXB04
XLD03= 0.000000; PXD03= PXB05
XLD04=111.000000; PXD04= 1.2

IG(261)=IXDL01;RG(421)=XLD01;RG(441)=PXD01
IG(262)=IXDL02;RG(422)=XLD02;RG(442)=PXD02
IG(263)=IXDL03;RG(423)=XLD03;RG(443)=PXD03
IG(264)=IXDL04;RG(424)=XLD04;RG(444)=PXD04
IG(277)=2

*

***** Y-DIRECTION GRIDING *****

*** NRYA -- Number of Y regions for Type 1 plane ***

*** NRYB -- Number of Y regions for Type 2 plane ***

*** NRYBB -- Number of Y regions for Type 3 plane ***

*** NRYC -- Number of Y regions for Type 4 plane ***

*** NRYD -- Number of Y regions for Type 5 plane ***

*** NOTE: The number of grid cells is define for the Type 1 ***
plane and then redistributed for the other types. ***

*** There are fourteen available regions, some may ***

*** not be used. ***

```

*** NY01 -- Number of cells in 1st region -> Floor to ***
*** bottom of cart ***
*** NY02 -- Number of cells in 2nd region -> Bottom of ***
*** cart to top of cart ***
*** NY03 -- Number of cells in 3rd region -> Top of cart ***
*** to bottom of dynamometer opening ***
*** NY04 -- Number of cells in 4th region -> Bottom of ***
*** dynamometer opening to top dyna opening ***
*** NY05 -- Number of cells in 5th region -> Top of ***
*** dynamometer opening to bottom dyna skirt ***
*** NY06 -- Number of cells in 6th region -> Bottom of ***
*** dynamometer skirt to top of dynamometer ***
*** NY07 -- Number of cells in 7th region -> Top of ***
*** dynamometer to bottom dynamometer exhaust ***
*** NY08 -- Number of cells in 8th region -> Bottom of ***
*** dynamometer exhaust to top of dyna sleeve ***
*** NY09 -- Number of cells in 9th region -> Top of ***
*** dynamometer sleeve to midpoint roof ***
*** NY10 -- Number of cells in 10th region -> Midpoint ***
*** roof to roof ***
*** NY11 -- Number of cells in 11th region -> Spare ***
*** NY12 -- Number of cells in 12th region -> Spare ***
*** NY13 -- Number of cells in 13th region -> Spare ***
*** NY14 -- Number of cells in 14th region -> Spare ***
***
*** NOTE: The regions for the other 5 Types will now also ***
*** be defined. ***
***
*** TYPE 2 ***
*** Region 1 -- Floor to bottom of cart ***
*** Region 2 -- Bottom of cart to top of cart ***
*** Region 3 -- Top of cart to engine ***
*** Region 4 -- Engine to midpoint engine ***
*** Region 5 -- Midpoint engine to engine ***
*** Region 6 -- Engine to top of skirt ***
*** Region 7 -- Top of skirt to midpoint roof ***
*** Region 8 -- Midpoint roof to roof ***
***
*** TYPE 3 ***
*** Region 1 -- Floor to bottom of cart ***
*** Region 2 -- Bottom of cart to top of cart ***
*** Region 3 -- Top of cart to augments tube ***
*** Region 4 -- Augments tube to engine ***
*** Region 5 -- Engine to midpoint engine ***
*** Region 6 -- Midpoint engine to engine ***
*** Region 7 -- Engine to augments tube ***
*** Region 8 -- Augments tube to top of skirt ***
*** Region 9 -- Top of skirt to midpoint roof ***
*** Region 10 -- Midpoint roof to roof ***
***
*** TYPE 4 ***
*** Region 1 -- Floor to bottom of cart ***
*** Region 2 -- Bottom of cart to top of cart ***
*** Region 3 -- Top of cart to augments tube ***
*** Region 4 -- Augments tube to midpoint aug tube ***
*** Region 5 -- Midpoint augments tube to aug tube ***
*** Region 6 -- Augments tube to top of skirt ***
*** Region 7 -- Top of skirt to midpoint roof ***
*** Region 8 -- Midpoint roof to roof ***
***
*** TYPE 5 ***
*** Region 1 -- Floor to augments tube ***
*** Region 2 -- Augments tube to midpoint aug tube ***
*** Region 3 -- Midpoint augments tube to aug tube ***

```

```

***      Region 4  -- Augmenter tube to center of curvature      ***
***      TYPE 6                                     ***
***      Region 1  -- Wall to wall                               ***
***
***      NYAD      -- Number of cells in Y-direction picked up by ***
***                  the engine                                     ***
***      NYBD      -- Number of cells in Y-directions picked up by ***
***                  the augmenter tube                             ***
***      NOTE:     This last two items have corresponding parameters ***
***                  for the X-direction. Generally they will be the ***
***                  the same.                                     ***
***
***      IYAF**     -- First cell number of ** region Type 1      ***
***      IYAL**     -- Last cell number of ** region Type 1      ***
***      IYBF**     -- First cell number of ** region Type 2      ***
***      IYBL**     -- Last cell number of ** region Type 2      ***
***      IYF**      -- First cell number of ** region Type 3      ***
***      IYL**      -- Last cell number of ** region Type 3      ***
***      IYCF**     -- First cell number of ** region Type 4      ***
***      IYCL**     -- Last cell number of ** region Type 4      ***
***      IYDF**     -- First cell number of ** region Type 5      ***
***      IYDL**     -- Last cell number of ** region Type 5      ***
***
***      IYMON*     -- Location of * monitoring point (9 extra)   ***
***
***      YLA**      -- Length to end of ** region Type 1 (in)     ***
***      YLB**      -- Length to end of ** region Type 2 (in)     ***
***      YLBB**     -- Length to end of ** region Type 3 (in)     ***
***      YLC**      -- Length to end of ** region Type 4 (in)     ***
***      YLD**      -- Length to end of ** region Type 5 (in)     ***
***
***      PYA**      -- Clustering factor of ** region Type 1      ***
***      PYB**      -- Clustering factor of ** region Type 2      ***
***      PYBB**     -- Clustering factor of ** region Type 3      ***
***      PYC**      -- Clustering factor of ** region Type 4      ***
***      PYD**      -- Clustering factor of ** region Type 5      ***
***
*****
*
*
*YYYYYYYYYYYYYYYYYYYYY  DECLARE Y  YYYYYYYYYYYYYYYYYYYYYYYYYYYYYY*
*
*

```

```

INTEGER(NRYA,NRYB,NRYBB,NRYC,NRYD)
INTEGER(NYAD,NYBD)
INTEGER(NY01,NY02,NY03,NY04,NY05,NY06,NY07,NY08,NY09,NY10)
INTEGER(NY11,NY12,NY13,NY14)
INTEGER(IYAF01,IYAF02,IYAF03,IYAF04,IYAF05)
INTEGER(IYAF06,IYAF07,IYAF08,IYAF09,IYAF10)
INTEGER(IYAF11,IYAF12,IYAF13,IYAF14,IYAF15)
INTEGER(IYAL01,IYAL02,IYAL03,IYAL04,IYAL05)
INTEGER(IYAL06,IYAL07,IYAL08,IYAL09,IYAL10)
INTEGER(IYAL11,IYAL12,IYAL13,IYAL14,IYAL15)
INTEGER(IYBF01,IYBF02,IYBF03,IYBF04,IYBF05)
INTEGER(IYBF06,IYBF07,IYBF08,IYBF09,IYBF10)
INTEGER(IYBF11,IYBF12,IYBF13,IYBF14,IYBF15)
INTEGER(IYBL01,IYBL02,IYBL03,IYBL04,IYBL05)
INTEGER(IYBL06,IYBL07,IYBL08,IYBL09,IYBL10)
INTEGER(IYBL11,IYBL12,IYBL13,IYBL14,IYBL15)
INTEGER(IYF01,IYF02,IYF03,IYF04,IYF05)

```

```

INTEGER(IYYF06,IYYF07,IYYF08,IYYF09,IYYF10)
INTEGER(IYYF11,IYYF12,IYYF13,IYYF14,IYYF15)
INTEGER(IYYL01,IYYL02,IYYL03,IYYL04,IYYL05)
INTEGER(IYYL06,IYYL07,IYYL08,IYYL09,IYYL10)
INTEGER(IYYL11,IYYL12,IYYL13,IYYL14,IYYL15)
INTEGER(IYCF01,IYCF02,IYCF03,IYCF04,IYCF05)
INTEGER(IYCF06,IYCF07,IYCF08,IYCF09,IYCF10)
INTEGER(IYCF11,IYCF12,IYCF13,IYCF14,IYCF15)
INTEGER(IYCL01,IYCL02,IYCL03,IYCL04,IYCL05)
INTEGER(IYCL06,IYCL07,IYCL08,IYCL09,IYCL10)
INTEGER(IYCL11,IYCL12,IYCL13,IYCL14,IYCL15)
INTEGER(IYDF01,IYDF02,IYDF03,IYDF04,IYDF05)
INTEGER(IYDF06,IYDF07,IYDF08,IYDF09,IYDF10)
INTEGER(IYDF11,IYDF12,IYDF13,IYDF14,IYDF15)
INTEGER(IYDL01,IYDL02,IYDL03,IYDL04,IYDL05)
INTEGER(IYDL06,IYDL07,IYDL08,IYDL09,IYDL10)
INTEGER(IYDL11,IYDL12,IYDL13,IYDL14,IYDL15)
INTEGER(IYMON1,IYMON2,IYMON3,IYMON4,IYMON5)
INTEGER(IYMON6,IYMON7,IYMON8,IYMON9)
REAL(YLA01,YLA02,YLA03,YLA04,YLA05)
REAL(YLA06,YLA07,YLA08,YLA09,YLA10)
REAL(YLA11,YLA12,YLA13,YLA14,YLA15)
REAL(YLB01,YLB02,YLB03,YLB04,YLB05)
REAL(YLB06,YLB07,YLB08,YLB09,YLB10)
REAL(YLB11,YLB12,YLB13,YLB14,YLB15)
REAL(YLBB01,YLBB02,YLBB03,YLBB04,YLBB05)
REAL(YLBB06,YLBB07,YLBB08,YLBB09,YLBB10)
REAL(YLBB11,YLBB12,YLBB13,YLBB14,YLBB15)
REAL(YLC01,YLC02,YLC03,YLC04,YLC05)
REAL(YLC06,YLC07,YLC08,YLC09,YLC10)
REAL(YLC11,YLC12,YLC13,YLC14,YLC15)
REAL(YLD01,YLD02,YLD03,YLD04,YLD05)
REAL(YLD06,YLD07,YLD08,YLD09,YLD10)
REAL(YLD11,YLD12,YLD13,YLD14,YLD15)
REAL(PYA01,PYA02,PYA03,PYA04,PYA05)
REAL(PYA06,PYA07,PYA08,PYA09,PYA10)
REAL(PYA11,PYA12,PYA13,PYA14,PYA15)
REAL(PYB01,PYB02,PYB03,PYB04,PYB05)
REAL(PYB06,PYB07,PYB08,PYB09,PYB10)
REAL(PYB11,PYB12,PYB13,PYB14,PYB15)
REAL(PYBB01,PYBB02,PYBB03,PYBB04,PYBB05)
REAL(PYBB06,PYBB07,PYBB08,PYBB09,PYBB10)
REAL(PYBB11,PYBB12,PYBB13,PYBB14,PYBB15)
REAL(PYC01,PYC02,PYC03,PYC04,PYC05)
REAL(PYC06,PYC07,PYC08,PYC09,PYC10)
REAL(PYC11,PYC12,PYC13,PYC14,PYC15)
REAL(PYD01,PYD02,PYD03,PYD04,PYD05)
REAL(PYD06,PYD07,PYD08,PYD09,PYD10)
REAL(PYD11,PYD12,PYD13,PYD14,PYD15)
*
*
*YYYYYYYYYYYYYYYYYYYY TYPE 1 DATA YYYYYYYYYYYYYYYYYYYYYYYYYY*
*
*
NRYA=10; IG(43)=NRYA
NY01=4
NY02=1
NY03=8
##### MUST BE EVEN FOR CELLS IN CIRCLE #####
NY04=6

```

NY05=4
NY06=2
NY07=2
NY08=2
NY09=3
NY10=3

IYAF01=	1;	IYAL01=	NY01
IYAF02=	IYAL01+1;	IYAL02=	IYAL01+NY02
IYAF03=	IYAL02+1;	IYAL03=	IYAL02+NY03
IYAF04=	IYAL03+1;	IYAL04=	IYAL03+NY04
IYAF05=	IYAL04+1;	IYAL05=	IYAL04+NY05
IYAF06=	IYAL05+1;	IYAL06=	IYAL05+NY06
IYAF07=	IYAL06+1;	IYAL07=	IYAL06+NY07
IYAF08=	IYAL07+1;	IYAL08=	IYAL07+NY08
IYAF09=	IYAL08+1;	IYAL09=	IYAL08+NY09
IYAF10=	IYAL09+1;	IYAL10=	IYAL09+NY10

YLA01=	19.000000;	PYA01=	1.0
YLA02=	25.000000;	PYA02=	1.0
YLA03=	0.000000;	PYA03=	1.0
YLA04=	0.000000;	PYA04=	1.0
YLA05=	64.000000;	PYA05=	1.0
YLA06=	66.750000;	PYA06=	1.0
YLA07=	68.000000;	PYA07=	1.0
YLA08=	72.000000;	PYA08=	1.0
YLA09=	102.000000;	PYA09=	1.5
YLA10=	132.000000;	PYA10=	-1.5

IG(121)=IYAL01;RG(141)=YLA01;RG(161)=PYA01
IG(122)=IYAL02;RG(142)=YLA02;RG(162)=PYA02
IG(123)=IYAL03;RG(143)=YLA03;RG(163)=PYA03
IG(124)=IYAL04;RG(144)=YLA04;RG(164)=PYA04
IG(125)=IYAL05;RG(145)=YLA05;RG(165)=PYA05
IG(126)=IYAL06;RG(146)=YLA06;RG(166)=PYA06
IG(127)=IYAL07;RG(147)=YLA07;RG(167)=PYA07
IG(128)=IYAL08;RG(148)=YLA08;RG(168)=PYA08
IG(129)=IYAL09;RG(149)=YLA09;RG(169)=PYA09
IG(130)=IYAL10;RG(150)=YLA10;RG(170)=PYA10
IG(137)=4

*

*

YYYYYYYYYYYYYYYYYYYY TYPE 2 DATA YYYYYYYYYYYYYYYYYYYYYYYYYYYYYY

*

*

NYAD=1	
NRYB=8;	IG(45)=NRYB
IYBF01=IYAF01;	IYBL01=IYAL01
IYBF02=IYAF02;	IYBL02=IYAL02
IYBF03=IYAF03;	IYBL03=IYAL03-NYAD
IYBF04=IYAF04-NYAD;	IYBL04=IYAL04-NY04/2
IYBF05=IYAF05-NY04/2;	IYBL05=IYAL04+NYAD
IYBF06=IYAF05+NYAD;	IYBL06=IYAL08
IYBF07=IYAF09;	IYBL07=IYAL09
IYBF08=IYAF10;	IYBL08=IYAL10

YLB01=	YLA01;	PYB01=	PYA01
YLB02=	YLA02;	PYB02=	PYA02
YLB03=	0.000000;	PYB03=	-1.4
YLB04=	YCENB;	PYB04=	1.6

YLB05= 0.000000;	PYB05=-1.6
YLB06= YLA08;	PYB06= 1.4
YLB07= YLA09;	PYB07= PYA09
YLB08= YLA10;	PYB08= PYA10

IG(161)=IYBL01;RG(221)=YLB01;RG(241)=PYB01
 IG(162)=IYBL02;RG(222)=YLB02;RG(242)=PYB02
 IG(163)=IYBL03;RG(223)=YLB03;RG(243)=PYB03
 IG(164)=IYBL04;RG(224)=YLB04;RG(244)=PYB04
 IG(165)=IYBL05;RG(225)=YLB05;RG(245)=PYB05
 IG(166)=IYBL06;RG(226)=YLB06;RG(246)=PYB06
 IG(167)=IYBL07;RG(227)=YLB07;RG(247)=PYB07
 IG(168)=IYBL08;RG(228)=YLB08;RG(248)=PYB08
 IG(177)=4

*

*

YYYYYYYYYYYYYYYYYYYY TYPE 3 DATA YYYYYYYYYYYYYYYYYYYYYYYYYYYYYY

*

*

NYBD=4

@@@@@@ SPECIFY WHEN USING TYPE 3 @@@@@@@

NRYBB=10;	IG(47)=NRYBB
IYYF01=IYBF01;	IYYL01=IYBL01
IYYF02=IYBF02;	IYYL02=IYBL02
IYYF03=IYBF03;	IYYL03=IYBL03-NYBD
IYYF04=IYBF04-NYBD;	IYYL04=IYBL03
IYYF05=IYBF04;	IYYL05=IYBL04
IYYF06=IYBF05;	IYYL06=IYBL05
IYYF07=IYBF06;	IYYL07=IYBL05+NYBD
IYYF08=IYBF06+NYBD;	IYYL08=IYBL06
IYYF09=IYBF07;	IYYL09=IYBL07
IYYF10=IYBF08;	IYYL10=IYBL08

YLBB01= YLA01;	PYBB01= PYA01
YLBB02= YLA02;	PYBB02= PYA02
YLBB03= 0.000000;	PYBB03= PYB03
YLBB04= 0.000000;	PYBB04= 1.0
YLBB05= YCENB;	PYBB05= PYB04
YLBB06= 0.000000;	PYBB06= PYB05
YLBB07= 0.000000;	PYBB07= 1.0
YLBB08= YLA08;	PYBB08= PYB06
YLBB09= YLA09;	PYBB09= PYA09
YLBB10= YLA10;	PYBB10= PYA10

IG(201)=IYYL01;RG(301)=YLBB01;RG(321)=PYBB01
 IG(202)=IYYL02;RG(302)=YLBB02;RG(322)=PYBB02
 IG(203)=IYYL03;RG(303)=YLBB03;RG(323)=PYBB03
 IG(204)=IYYL04;RG(304)=YLBB04;RG(324)=PYBB04
 IG(205)=IYYL05;RG(305)=YLBB05;RG(325)=PYBB05
 IG(206)=IYYL06;RG(306)=YLBB06;RG(326)=PYBB06
 IG(207)=IYYL07;RG(307)=YLBB07;RG(327)=PYBB07
 IG(208)=IYYL08;RG(308)=YLBB08;RG(328)=PYBB08
 IG(209)=IYYL09;RG(309)=YLBB09;RG(329)=PYBB09
 IG(210)=IYYL10;RG(310)=YLBB10;RG(330)=PYBB10
 IG(217)=4

*

*

YYYYYYYYYYYYYYYYYYYY TYPE 4 DATA YYYYYYYYYYYYYYYYYYYYYYYYYYYYYY

*

*

```

NRYC=8;          IG(49)=NRYC
IYCF01=IYBF01;   IYCL01=IYBL01
IYCF02=IYBF02;   IYCL02=IYBL02
IYCF03=IYBF03;   IYCL03=IYBL03-NYBD
IYCF04=IYBF04-NYBD; IYCL04=IYBL04
IYCF05=IYBF05;   IYCL05=IYBL05+NYBD
IYCF06=IYBF06+NYBD; IYCL06=IYBL06
IYCF07=IYBF07;   IYCL07=IYBL07
IYCF08=IYBF08;   IYCL08=IYBL08

```

```

YLC01= YLA01;     PYC01= PYA01
YLC02= YLA02;     PYC02= PYA02
YLC03= 0.000000;  PYC03= PYB03
YLC04= YCENC;     PYC04= PYB04
YLC05= 0.000000;  PYC05= PYB05
YLC06= YLA08;     PYC06= PYB06
YLC07= YLA09;     PYC07= PYA09
YLC08= YLA10;     PYC08= PYA10

```

```

IG(241)=IYCL01;RG(381)=YLC01;RG(401)=PYC01
IG(242)=IYCL02;RG(382)=YLC02;RG(402)=PYC02
IG(243)=IYCL03;RG(383)=YLC03;RG(403)=PYC03
IG(244)=IYCL04;RG(384)=YLC04;RG(404)=PYC04
IG(245)=IYCL05;RG(385)=YLC05;RG(405)=PYC05
IG(246)=IYCL06;RG(386)=YLC06;RG(406)=PYC06
IG(247)=IYCL07;RG(387)=YLC07;RG(407)=PYC07
IG(248)=IYCL08;RG(388)=YLC08;RG(408)=PYC08
IG(257)=4

```

*

*

YYYYYYYYYYYYYYYYYYYY TYPE 5 DATA YYYYYYYYYYYYYYYYYYYYYYYYYYYYYY

*

*

```

NRYD=4;          IG(51)=NRYD
IYDF01=IYCF01;   IYDL01=IYCL03
IYDF02=IYCF04;   IYDL02=IYCL04
IYDF03=IYCF05;   IYDL03=IYCL05
IYDF04=IYCF06;   IYDL04=IYCL08

```

```

YLD01= 0.000000;  PYD01=-1.4
YLD02= YCEND;     PYD02= PYB04
YLD03= 0.000000;  PYD03= PYB05
YLD04= 85.000000; PYD04= 1.4

```

```

IG(281)=IYDL01;RG(461)=YLD01;RG(481)=PYD01
IG(282)=IYDL02;RG(462)=YLD02;RG(482)=PYD02
IG(283)=IYDL03;RG(463)=YLD03;RG(483)=PYD03
IG(284)=IYDL04;RG(464)=YLD04;RG(484)=PYD04
IG(297)=2

```

*

 ***** Z-DIRECTION GRIDING *****

*** NOTE: With the X-Y grid information, several planes of
 *** grid points will be produced in the SATLIT. In
 *** this section the user must specify how these
 *** planes are then stacked, blended, or rotated.
 *** There will be a plane of data for the front face
 *** of each of the following regions.

```

***
*** NCS -- Number of regions in Z-direction ***
***
***
*** NZ01 -- Number of cells in 1st region -> Front wall ***
*** to end of wall extension (or midpoint to cart) ***
*** NZ02 -- Number of cells in 2nd region -> End of wall ***
*** extension to front of cart ***
*** NZ03 -- Number of cells in 3rd region -> Front of ***
*** cart to front of dynamometer skirt ***
*** NZ04 -- Number of cells in 4th region -> Front of ***
*** dynamometer skirt to outside front of dyna ***
*** NZ05 -- Number of cells in 5th region -> Outside ***
*** front of dynamometer to front of dyna exhaust ***
*** NZ06 -- Number of cells in 6th region -> Front of ***
*** dynamometer exhaust to inside front of dyna ***
*** NZ07 -- Number of cells in 7th region -> Inside ***
*** front of dynamometer to back of dyna exhaust ***
*** NZ08 -- Number of cells in 8th region -> Back of ***
*** dynamometer exhaust to inside back of dyna ***
*** NZ09 -- Number of cells in 9th region -> Inside back ***
*** of dynamometer to outside back of dyna ***
*** NZ10 -- Number of cells in 10th region -> Outside ***
*** back of dynamometer to back of dyna skirt ***
*** NZ11 -- Number of cells in 11th region -> Back of ***
*** dynamometer skirt to plate ***
*** NZ12 -- Number of cells in 12th region -> Plate to ***
*** engine inlet ***
*** NZ13 -- Number of cells in 13th region -> Engine ***
*** inlet to end of cart ***
*** NZ14 -- Number of cells in 14th region -> End of ***
*** cart to start of nozzle ***
*** NZ15 -- Number of cells in 15th region -> Start of ***
*** nozzle to end of nozzle ***
*** NZ16 -- Number of cells in 16th region -> End of ***
*** nozzle to augments lip ***
*** NZ17 -- Number of cells in 17th region -> Augments ***
*** lip to start of augments sleeve ***
*** NZ18 -- Number of cells in 18th region -> Start of ***
*** augments sleeve to end augments sleeve ***
*** NZ19 -- Number of cells in 19th region -> End of ***
*** augments sleeve to augments tube ***
*** ASSUMPTION: This is an arbitrary region to ***
*** make up for the difference in diameter. ***
*** NZ20 -- Number of cells in 20th region -> Augments ***
*** tube to midpoint of wall ***
*** NZ21 -- Number of cells in 21th region -> Midpoint ***
*** of wall to front of wall ***
*** NZ22 -- Number of cells in 22th region -> Front of ***
*** wall to back of wall ***
*** NZ23 -- Number of cells in 23th region -> Back of ***
*** wall to end of augments tube ***
*** NZ24 -- Number of cells in 24th region -> End of ***
*** augments tube to end of domain ***
*** NZ25 -- Number of cells in 25th region -> Spare ***
***
*** IZF** -- First cell number of ** region ***
*** IZL** -- Last cell number of ** region ***
***
*** IZMON* -- Location of * monitoring point (9 extra) ***

```

```
***  
***      ZL**      -- Length to end of ** region (in) ***  
***  
***      PZ**      -- Clustering factor of ** region ***  
***  
***      DYNAL     --. Z-direction width of dynamometer ***  
*****  
*  
*  
*ZZZZZZZZZZZZZZZZZZZZZZZ DECLARE Z ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ*  
*  
*  
  
INTEGER(NCS)  
INTEGER(NZ01,NZ02,NZ03,NZ04,NZ05,NZ06,NZ07,NZ08,NZ09,NZ10)  
INTEGER(NZ11,NZ12,NZ13,NZ14,NZ15,NZ16,NZ17,NZ18,NZ19,NZ20)  
INTEGER(NZ21,NZ22,NZ23,NZ24,NZ25)  
INTEGER(IZF01,IZF02,IZF03,IZF04,IZF05)  
INTEGER(IZF06,IZF07,IZF08,IZF09,IZF10)  
INTEGER(IZF11,IZF12,IZF13,IZF14,IZF15)  
INTEGER(IZF16,IZF17,IZF18,IZF19,IZF20)  
INTEGER(IZF21,IZF22,IZF23,IZF24,IZF25)  
INTEGER(IZL01,IZL02,IZL03,IZL04,IZL05)  
INTEGER(IZL06,IZL07,IZL08,IZL09,IZL10)  
INTEGER(IZL11,IZL12,IZL13,IZL14,IZL15)  
INTEGER(IZL16,IZL17,IZL18,IZL19,IZL20)  
INTEGER(IZL21,IZL22,IZL23,IZL24,IZL25)  
INTEGER(IZMON1,IZMON2,IZMON3,IZMON4,IZMON5)  
INTEGER(IZMON6,IZMON7,IZMON8,IZMON9)  
REAL(ZL01,ZL02,ZL03,ZL04,ZL05)  
REAL(ZL06,ZL07,ZL08,ZL09,ZL10)  
REAL(ZL11,ZL12,ZL13,ZL14,ZL15)  
REAL(ZL16,ZL17,ZL18,ZL19,ZL20)  
REAL(ZL21,ZL22,ZL23,ZL24,ZL25)  
REAL(PZ01,PZ02,PZ03,PZ04,PZ05)  
REAL(PZ06,PZ07,PZ08,PZ09,PZ10)  
REAL(PZ11,PZ12,PZ13,PZ14,PZ15)  
REAL(PZ16,PZ17,PZ18,PZ19,PZ20)  
REAL(PZ21,PZ22,PZ23,PZ24,PZ25)  
REAL(DYNAL)  
*  
*  
*ZZZZZZZZZZZZZZZZZZZZZZZ GEOMETRY & STACKING INFO ZZZZZZZZZZZZZZZZZ*  
*  
*  
  
NCS=24;  
NZ01=3  
NZ02=3  
NZ03=4  
NZ04=2  
NZ05=1  
NZ06=1  
NZ07=4  
NZ08=1  
NZ09=1  
NZ10=2  
NZ11=2  
NZ12=4  
NZ13=3  
NZ14=2
```

NZ15=4
NZ16=2
NZ17=2
NZ18=4
NZ19=2
NZ20=6
NZ21=6
NZ22=2

MUST BE EVEN FOR CELLS IN BEND

NZ23=10
NZ24=6

IZF01= 1;
IZF02=IZL01+1;
IZF03=IZL02+1;
IZF04=IZL03+1;
IZF05=IZL04+1;
IZF06=IZL05+1;
IZF07=IZL06+1;
IZF08=IZL07+1;
IZF09=IZL08+1;
IZF10=IZL09+1;
IZF11=IZL10+1;
IZF12=IZL11+1;
IZF13=IZL12+1;
IZF14=IZL13+1;
IZF15=IZL14+1;
IZF16=IZL15+1;
IZF17=IZL16+1;
IZF18=IZL17+1;
IZF19=IZL18+1;
IZF20=IZL19+1;
IZF21=IZL20+1;
IZF22=IZL21+1;
IZF23=IZL22+1;
IZF24=IZL23+1;

IZL01= NZ01
IZL02=IZL01+NZ02
IZL03=IZL02+NZ03
IZL04=IZL03+NZ04
IZL05=IZL04+NZ05
IZL06=IZL05+NZ06
IZL07=IZL06+NZ07
IZL08=IZL07+NZ08
IZL09=IZL08+NZ09
IZL10=IZL09+NZ10
IZL11=IZL10+NZ11
IZL12=IZL11+NZ12
IZL13=IZL12+NZ13
IZL14=IZL13+NZ14
IZL15=IZL14+NZ15
IZL16=IZL15+NZ16
IZL17=IZL16+NZ17
IZL18=IZL17+NZ18
IZL19=IZL18+NZ19
IZL20=IZL19+NZ20
IZL21=IZL20+NZ21
IZL22=IZL21+NZ22
IZL23=IZL22+NZ23
IZL24=IZL23+NZ24

ZL01= 48.0;
ZL02=107.0;
ZL03=181.25;
ZL04=184.5;
ZL05=185.5;
ZL06=186.5;
ZL07=202.5;
ZL08=203.5;
ZL09=205.5;
ZL10=206.75;
ZL11=211.5;
ZL12=222.0;
ZL13=255.0;
ZL14=267.0;
ZL15=283.0;
ZL16=285.5;
ZL17=288.0;
ZL18=323.0;
ZL19=332.0;
ZL20=414.0;
ZL21=496.0;
ZL22=508.0;

PZ01= 1.0
PZ02= 1.3
PZ03=-1.6
PZ04= 1.0
PZ05= 1.0
PZ06= 1.0
PZ07= 1.0
PZ08= 1.0
PZ09= 1.0
PZ10= 1.0
PZ11= 1.0
PZ12= 1.0
PZ13= 1.5
PZ14= 1.0
PZ15=-1.4
PZ16= 1.0
PZ17= 1.0
PZ18= 1.4
PZ19= 1.0
PZ20= 1.6
PZ21=-1.4
PZ22= 1.0

ZL23=568.0; PZ23= 1.0
ZL24=568.0; PZ24= 1.2
DYNAL=ZL08-ZL06

IG(511)=NZ01;RG(511)=ZL01;RG(541)=PZ01
IG(512)=NZ02;RG(512)=ZL02;RG(542)=PZ02
IG(513)=NZ03;RG(513)=ZL03;RG(543)=PZ03
IG(514)=NZ04;RG(514)=ZL04;RG(544)=PZ04
IG(515)=NZ05;RG(515)=ZL05;RG(545)=PZ05
IG(516)=NZ06;RG(516)=ZL06;RG(546)=PZ06
IG(517)=NZ07;RG(517)=ZL07;RG(547)=PZ07
IG(518)=NZ08;RG(518)=ZL08;RG(548)=PZ08
IG(519)=NZ09;RG(519)=ZL09;RG(549)=PZ09
IG(520)=NZ10;RG(520)=ZL10;RG(550)=PZ10
IG(521)=NZ11;RG(521)=ZL11;RG(551)=PZ11
IG(522)=NZ12;RG(522)=ZL12;RG(552)=PZ12
IG(523)=NZ13;RG(523)=ZL13;RG(553)=PZ13
IG(524)=NZ14;RG(524)=ZL14;RG(554)=PZ14
IG(525)=NZ15;RG(525)=ZL15;RG(555)=PZ15
IG(526)=NZ16;RG(526)=ZL16;RG(556)=PZ16
IG(527)=NZ17;RG(527)=ZL17;RG(557)=PZ17
IG(528)=NZ18;RG(528)=ZL18;RG(558)=PZ18
IG(529)=NZ19;RG(529)=ZL19;RG(559)=PZ19
IG(530)=NZ20;RG(530)=ZL20;RG(560)=PZ20
IG(531)=NZ21;RG(531)=ZL21;RG(561)=PZ21
IG(532)=NZ22;RG(532)=ZL22;RG(562)=PZ22
IG(533)=NZ23;RG(533)=ZL23;RG(563)=PZ23
IG(534)=NZ24;RG(534)=ZL24;RG(564)=PZ24

IG WHERE CHIMNEY STARTS -- RG HEIGHT TO BAFFLES #####
IG(537)=23;RG(537)=YLA10

IG(541)=1;IG(571)=61;IG(601)=61
IG(542)=2;IG(572)=61;IG(602)=62
IG(543)=1;IG(573)=62;IG(603)=62
IG(544)=2;IG(574)=62;IG(604)=63
IG(545)=1;IG(575)=63;IG(605)=63
IG(546)=1;IG(576)=63;IG(606)=63
IG(547)=1;IG(577)=63;IG(607)=63
IG(548)=1;IG(578)=63;IG(608)=63
IG(549)=1;IG(579)=63;IG(609)=63
IG(550)=1;IG(580)=63;IG(610)=63
IG(551)=1;IG(581)=63;IG(611)=63
IG(552)=2;IG(582)=63;IG(612)=64
IG(553)=1;IG(583)=64;IG(613)=64
IG(554)=1;IG(584)=64;IG(614)=64
IG(555)=2;IG(585)=64;IG(615)=65
IG(556)=1;IG(586)=65;IG(616)=65
IG(557)=2;IG(587)=65;IG(617)=66
IG(558)=2;IG(588)=66;IG(618)=67
IG(559)=2;IG(589)=67;IG(619)=68
IG(560)=1;IG(590)=68;IG(620)=68
IG(561)=1;IG(591)=68;IG(621)=68
IG(562)=2;IG(592)=68;IG(622)=69
IG(563)=3;IG(593)=69;IG(623)=69
IG(564)=4;IG(594)=69;IG(624)=70

GROUP 2. Transience; time-step specification

GROUP 3. X-direction grid specification

NX=Nx01+Nx02+Nx03+Nx04+Nx05+Nx06+Nx07+Nx08+Nx09

NX=Nx10+Nx11+Nx12+Nx13+Nx14

GROUP 4. Y-direction grid specification

NY=NY01+NY02+NY03+NY04+NY05+NY06+NY07+NY08+NY09
NY=NY+NY10+NY11+NY12+NY13+NY14

GROUP 5. Z-direction grid specification

NZ=NZ01+NZ02+NZ03+NZ04+NZ05+NZ06+NZ07+NZ08+NZ09+NZ10+NZ11
NZ=NZ+NZ12+NZ13+NZ14+NZ15+NZ16+NZ17+NZ18+NZ19+NZ20+NZ21
NZ=NZ+NZ22+NZ23+NZ24+NZ25

GROUP 6. Body-fitted coordinates or grid distortion

BFC=T;NONORT=T
IG(1)=0
SATRUN(NECL)
READCO(GRID)

GROUP 7. Variables stored, solved & named

SOLUTN(P1,Y,Y,Y,N,N,N)
SOLVE(U1,V1,W1)
SOLUTN(U1,Y,Y,N,Y,N,N)
SOLUTN(V1,Y,Y,N,Y,N,N)
SOLUTN(W1,Y,Y,N,Y,N,N)
SOLVE(H1,C1)
SOLVE(C2)
STORE(RHO1)
STORE(C3,C4,C5,C6,C7)
STORE(U2,V2,W2,C8,C9,C10,C11)
NAME(C4)=TEMP
NAME(C5)=CP
NAME(C8)=PH2O
NAME(C9)=TFAR
NAME(C10)=RHOE
NAME(C11)=SPAR
TURMOD(KEMODL)
STORE(ENUT)
KELIN=1

GROUP 8. Terms (in differential equations) & devices

TERMS(H1,N,P,P,P,P,P)

GROUP 9. Properties of the medium (or media)

```
*
*****
*****                      USER DEFINED VARIABLES                      *****
*****
***                                                                    ***
***  NOTE:  These are the variables used to define this                ***
***                  problem.                                           ***
***                                                                    ***
***  COND1    --  k for mineral fiber (BTU/hr/ft/F)                    ***
***  COND2    --  k for steel (BTU/hr/ft/F)                            ***
***  THICK1   --  Thickness of mineral fiber (in)                      ***
***  THICK2   --  Thickness of steel (in)                              ***
***  TAMB     --  Temperature ambient (F)                              ***
***  TDYN     --  Temperature dynamometer (F)                         ***
***  TENG     --  Temperature engine (F)                               ***
***  EMDOT    --  Engine flow rate (lb/s)                              ***
***  DMDOT    --  Dynamometer flow rate (lb/s)                         ***
***  FMDOT    --  Fuel flow rate (lb/s)                                ***
***  PAMB     --  Pressure ambient (mm Hg)                             ***
***  TIG      --  Turbulence intensity inlet (-)                       ***
***  TID      --  Turbulence intensity dynamometer (-)                ***
***  TIE      --  Turbulence intensity engine (-)                      ***
***  XKFACT1  --  K-loss factor inlet (-)                              ***
***  XKFACT2  --  K-loss factor dynamometer stack (-)                 ***
***  XKFACT3  --  K-loss factor chimney (-)                            ***
```

```

*** AMF1 -- N2 mass fraction ambient (-) ***
*** AMF2 -- O2 mass fraction ambient (-) ***
*** AMF3 -- CO2 mass fraction ambient (-) ***
*** AMF4 -- H2O mass fraction ambient (-) ***
*** DMF1 -- N2 mass fraction dynamometer (-) ***
*** DMF2 -- O2 mass fraction dynamometer (-) ***
*** DMF3 -- CO2 mass fraction dynamometer (-) ***
*** DMF4 -- H2O mass fraction dynamometer (-) ***
*** EMF1 -- N2 mass fraction engine (-) ***
*** EMF2 -- O2 mass fraction engine (-) ***
*** EMF3 -- CO2 mass fraction engine (-) ***
*** EMF4 -- H2O mass fraction engine (-) ***
*** VCOM -- Angle of momentum at dyna plate (deg) ***
***          (90 deg for straight up) ***
***

```

```

*****

```

```

*
*

```

```

*****
*****              OTHER VARIABLES              *****
*****

```

```

***
***

```

```

*** NOTE: These are the variables used to define this ***
*** problem. ***
***

```

```

*** XCON01 -- Converts in to m ***
*** XCON02 -- Converts F to R ***
*** XCON03 -- Converts R to K ***
*** XCON04 -- Converts BTU/ft/h/R to J/s/m/K ***
*** XCON05 -- Converts lb to kg ***
*** XCON06 -- Converts N/sq m to in H2O ***
*** XCON07 -- Converts m/s to ft/s ***
*** XCON08 -- Converts kg/cu m to lb/cu ft ***
*** XCON09 -- Converts in Hg to N/sq m ***
*** XCON10 -- Spare ***
*** XCON11 -- Spare ***
*** PTRAP -- Pressure trap (N/sq m) ***
*** RGAS -- Gas constant (N-m/K/kgmol) ***
*** XMW1 -- Molecular weight N2 (kg/kgmol) ***
*** XMW2 -- Molecular weight O2 (kg/kgmol) ***
*** XMW3 -- Molecular weight CO2 (kg/kgmol) ***
*** XMW4 -- Molecular weight H2O (kg/kgmol) ***
*** DARSOR -- Area of dynamometer plate source side (sq m) ***
*** EARSOR -- Area of engine plate source side (sq m) ***
*** RHOAMB -- Density ambient (kg/cu m) ***
*** RHODYN -- Density dynamometer (kg/cu m) ***
*** RHOENG -- Density engine (kg/cu m) ***
*** ENTHA -- Enthalpy ambient (J/kg) ***
*** ENTHD -- Enthalpy dynamometer (J/kg) ***
*** ENTHE -- Enthalpy engine (J/kg) ***
*** GAPIN -- Gap between inlet baffles (m) ***
*** GKE -- Inlet KE (sq m/sq s) ***
*** GEP -- Inlet EP (sq m/cu s) ***
*** DKE -- Dynamometer KE (sq m/sq s) ***
*** DEP -- Dynamometer EP (sq m/cu s) ***
*** EKE -- Engine KE (sq m/sq s) ***
*** EEP -- Engine EP (sq m/cu s) ***

```

```

*****
*

```

REAL(TAMB, RGAS, TDYN, TENG)

```

REAL(DARSOR,EARSOR,RHOAMB,RHODYN,RHOENG)
REAL(AMF1,AMF2,AMF3,AMF4,DMF1,DMF2,DMF3,DMF4,EMF1,EMF2,EMF3,EMF4)
REAL(ENTHA,ENTHD,ENTHE,XMW1,XMW2,XMW3,XMW4,XMWA,XMWE)
REAL(COND1,THICK1,COND2,THICK2,PTRAP)
REAL(XCON01,XCON02,XCON03,XCON04,XCON05,XCON06)
REAL(XCON07,XCON08,XCON09,XCON10,XCON11)
REAL(EMDOT,DMDOT,FMDOT,PAMB)
REAL(GAPIN,TIG,TID,TIE,GKE,GEP,DKE,DEP,EKE,EEP)
REAL(XKFCT1,XKFCT2,XKFCT3,VCOM,SINO,COSO)

```

```

*
*****
*****              USER SECTION              *****
*****
*

```

```

COND1=0.022
COND2=26.0
THICK1=2.0
THICK2=0.25
TAMB=77.0
TDYN=413.0
TENG=1200.0
EMDOT=11.2
DMDOT=14.0
FMDOT=0.2222222
PAMB=29.91
TIG=0.02
TID=0.10
TIE=0.15
XKFCT1=0.1
XKFCT2=0.1
XKFCT3=0.1
AMF1=0.7683
AMF2=0.2317
AMF3=0.0
AMF4=0.0
DMF1=0.7683
DMF2=0.2317
DMF3=0.0
DMF4=0.0
EMF1=0.7512
EMF2=0.1548
EMF3=0.069
EMF4=0.025
VCOM=45.0

```

```

*
*****
*

```

conversions

```

XCON01=0.0254;      RG(31)=XCON01
XCON02=459.67;      RG(32)=XCON02
XCON03=5.0/9.0;      RG(33)=XCON03
XCON04=1.73073;      RG(34)=XCON04
XCON05=0.45359;      RG(35)=XCON05
XCON06=407.16/101325.0; RG(36)=XCON06
XCON07=3.2802;      RG(37)=XCON07
XCON08=0.062428;     RG(38)=XCON08
XCON09=101325.0/29.92; RG(39)=XCON09

```

```

COND1=COND1*XCON04
COND2=COND2*XCON04

```

```

THICK1=THICK1*XCON01
THICK2=THICK2*XCON01
TAMB=(TAMB+XCON02)*XCON03
TDYN=(TDYN+XCON02)*XCON03
TENG=(TENG+XCON02)*XCON03
EMDOT=EMDOT*XCON05
DMDOT=DMDOT*XCON05
FMDOT=FMDOT*XCON05
PAMB=PAMB*XCON09
    density info
PTRAP=0.05
RG(29)=PTRAP
RGAS=8314.32
RG( 1)=AMF1
RG( 2)=AMF2
RG( 3)=AMF3
RG( 4)=AMF4
RG( 5)=DMF1
RG( 6)=DMF2
RG( 7)=DMF3
RG( 8)=DMF4
RG( 9)=EMF1
RG(10)=EMF2
RG(11)=EMF3
RG(12)=EMF4
RG(13)=TAMB
RG(14)=TDYN
RG(15)=TENG
XMW1=28.1608;          RG(21)=XMW1
XMW2=31.9988;          RG(22)=XMW2
XMW3=44.0100;          RG(23)=XMW3
XMW4=18.0152;          RG(24)=XMW4
RG(25)=RGAS
    area calculation dyna
DARSOR=DYNAW*DYNAL*XCON01*XCON01
    area calculation engine
EARSOR=PI*(DENG1/2.*XCON01)*(DENG1/2.*XCON01)
RG(801)=DARSOR
RG(802)=EARSOR
    density calculation
XMWA=1.0/(AMF1/XMW1+AMF2/XMW2+AMF3/XMW3+AMF4/XMW4)
XMWE=1.0/(EMF1/XMW1+EMF2/XMW2+EMF3/XMW3+EMF4/XMW4)
RHOAMB=PAMB*XMWA/(RGAS*TAMB)
RG(701)=RHOAMB
RHODYN=PAMB*XMWA/(RGAS*TDYN)
RHOENG=PAMB*XMWE/(RGAS*TENG)
    run satlit for enthalpy calculation
IG(1)=3
SATRUN(NECL)
    other stuff
ENTHA=RG(16)
ENTHD=RG(17)
ENTHE=RG(18)
PRESS0=PAMB
RHO1=GRND
DRH1DP=GRND
    turbulence (assume 1 m/s velocity)
GAPIN=XINL/14.0*XCON01
GKE=0.5*(1.0*TIG)**2
GEP=0.164*GKE**1.5/(0.09*GAPIN)

```

```

DKE=0.5*((DMDOT/DARSOR/RHODYN)*TID)**2
DEP=0.164*DKE**1.5/(0.09*DARSOR**0.5)
EKE=0.5*((EMDOT/EARSOR/RHOENG)*TIE)**2
EEP=0.164*EKE**1.5/(0.09*EARSOR**0.5)
    angle calculation
VCOM=VCOM*PI/180.
SINO=VCOM
SINO=SINO-(VCOM**3)/(3.*2.)
SINO=SINO+(VCOM**5)/(5.*4.*3.*2.)
SINO=SINO-(VCOM**7)/(7.*6.*5.*4.*3.*2.)
COSO=1.-(VCOM**2)/2.
COSO=COSO+(VCOM**4)/(4.*3.*2.)
COSO=COSO-(VCOM**6)/(6.*5.*4.*3.*2.)
COSO=COSO+(VCOM**8)/(8.*7.*6.*5.*4.*3.*2.)

```

GROUP 10. Inter-phase-transfer processes and properties

```

*
*****
*****                               INDEX                               *****
*****
***
*** The following variables are used as an index to define ***
*** the extent of blockages in the X, Y, & Z directions. ***
*** This was done because a user may change the number of ***
*** regions in each direction. The user will then make the ***
*** appropriate changes in this section and then no further ***
*** changes will be required below this section. The ***
*** nomenclature for the variables below is as follows: ***
*** 1.) The first letter represents direction (ie I for X), ***
*** 2.) Middle two letters represents the blockage name, & ***
*** 3.) Last letter represents first or last. ***
***
*****
*

```

```

INTEGER(IFWF,IFWL,JFWF,JFWL,KFWF,KFWL)
INTEGER(ICAF,ICAL,JCAF,JCAL,KCAF,KCAL)
INTEGER(ID1F,ID1L,JD1F,JD1L,KD1F,KD1L)
INTEGER(ID2F,ID2L,JD2F,JD2L,KD2F,KD2L)
INTEGER(ID3F,ID3L,JD3F,JD3L,KD3F,KD3L)
INTEGER(ID4F,ID4L,JD4F,JD4L,KD4F,KD4L)
INTEGER(ID5F,ID5L,JD5F,JD5L,KD5F,KD5L)
INTEGER(ID6F,ID6L,JD6F,JD6L,KD6F,KD6L)
INTEGER(ID7F,ID7L,JD7F,JD7L,KD7F,KD7L)
INTEGER(ID8F,ID8L,JD8F,JD8L,KD8F,KD8L)
INTEGER(ID9F,ID9L,JD9F,JD9L,KD9F,KD9L)
INTEGER(ID0F,ID0L,JD0F,JD0L,KD0F,KD0L)
INTEGER(IESF,IESL,JESF,JESL,KESF,KESL)
INTEGER(IDEF,IDEL,JDEF,JDEL,KDEF,KDEL)
INTEGER(IPLF,IPLL,JPLF,JPLL,KPLF,KPLL)
INTEGER(IEGF,IEGL,JEGF,JEGL,KEGF,KEGL)
INTEGER(IA1F,IA1L,JA1F,JA1L,KA1F,KA1L)
INTEGER(IA2F,IA2L,JA2F,JA2L,KA2F,KA2L)
INTEGER(IW1F,IW1L,JW1F,JW1L,KW1F,KW1L)
INTEGER(IW2F,IW2L,JW2F,JW2L,KW2F,KW2L)
INTEGER(IW3F,IW3L,JW3F,JW3L,KW3F,KW3L)
INTEGER(IW4F,IW4L,JW4F,JW4L,KW4F,KW4L)
INTEGER(IIN,JDP,KEP)
INTEGER(IDPF,IDPL,KDPF,KDPL)
INTEGER(IEPF,IEPL,JEPF,JEPL)

```

```

inlet cutoff
IIN=IG(61)
@@@@@@ SPECIFY WHEN BAFFLES EXTEND INTO DOMAIN @@@@@@
IFWF=IXAF01;          IFWL=IIN
JFWF=IYAF01;          JFWL=IYAL10
KFWF=IZF01;          KFWL=IZL01-1
@@@@@@ SPECIFY WHEN BAFFLES DO NOT EXTEND INTO DOMAIN @@@@@@
IFWL=IIN

```

```

cart
ICAF=IXAF03;          ICAL=IXAL11
JCAF=IYAF02;          JCAL=IYAL02
KCAF=IZF03;          KCAL=IZL13

```

```

dynamometer front wall (lower section)
ID1F=IXAF05;          ID1L=IXAL09
JD1F=IYAF03;          JD1L=IYAL03
KD1F=IZF05;          KD1L=IZL06
dynamometer front wall (mid-right section)
ID2F=IXAF05;          ID2L=IXAL06
JD2F=IYAF04;          JD2L=IYAL04
KD2F=IZF05;          KD2L=IZL06
dynamometer front wall (mid-left section)
ID3F=IXAF08;          ID3L=IXAL09
JD3F=IYAF04;          JD3L=IYAL04
KD3F=IZF05;          KD3L=IZL06
dynamometer front wall (top section)
ID4F=IXAF05;          ID4L=IXAL09
JD4F=IYAF05;          JD4L=IYAL06
KD4F=IZF05;          KD4L=IZL06
dynamometer back wall (lower section)
ID5F=IXAF05;          ID5L=IXAL09
JD5F=IYAF03;          JD5L=IYAL03
KD5F=IZF09;          KD5L=IZL09
dynamometer back wall (mid-right section)
ID6F=IXAF05;          ID6L=IXAL06
JD6F=IYAF04;          JD6L=IYAL04
KD6F=IZF09;          KD6L=IZL09
dynamometer back wall (mid-left section)
ID7F=IXAF08;          ID7L=IXAL09
JD7F=IYAF04;          JD7L=IYAL04
KD7F=IZF09;          KD7L=IZL09
dynamometer back wall (top section)
ID8F=IXAF05;          ID8L=IXAL09
JD8F=IYAF05;          JD8L=IYAL06
KD8F=IZF09;          KD8L=IZL09
dynamometer side wall (right)
ID9F=IXAF05;          ID9L=IXAL05
JD9F=IYAF03;          JD9L=IYAL06
KD9F=IZF07;          KD9L=IZL08
dynamometer side wall (left)
ID0F=IXAF09;          ID0L=IXAL09
JD0F=IYAF03;          JD0L=IYAL06
KD0F=IZF07;          KD0L=IZL08
dynamometer plate
IDPF=IXAF06;          IDPL=IXAL08
JDP=IYAL04+2
KDPF=IZF07;          KDPL=IZL08

```

```

IG(701)=JDP

```

IG(702)=IDPF; IG(703)=IDPL
IG(704)=KDPF; IG(705)=KDPL

exhaust skirt

@@@@@@ SPECIFY FOR EXHAUST SKIRT @@@@@@

IESF=IXAF04; IESL=IXAL10
JESF=IYAF06; JESL=IYAL08
KESF=IZF04; KESL=IZL10

dynamometer exhaust

IDEF=IXAF06; IDEL=IXAL08
JDEF=IYAF08; JDEL=IYAL10
KDEF=IZF06; KDEL=IZL07

plate (WARNING: conpor sec does NONgeneralized stair step)

IPLF=IXAF03; IPLL=IXAL11
JPLF=IYAF03; JPLL=IYAL06
KPLF=IZF12; KPLL=IZF12

engine

IEGF=IXBF04; IEGL=IXBL05
JEGF=IYBF04; JEGl=IYBL05
KEGF=IZF13; KEGL=IZL15

engine plate

IEPF=IEGF; IEPL=IEGL
JEPF=JEGF; JEPL=JEGL
KEP=IZF14

IG(711)=KEP
IG(712)=IEPF;IG(713)=IEPL
IG(714)=JEPF;IG(715)=JEPL

augmenter tube (in building)

IA1F=IXCF04; IA1L=IXCL05
JA1F=IYCF04; JA1L=IYCL05
KA1F=IZF17; KA1L=IZL21

augmenter tube (in chimney)

IA2F=IXDF02; IA2L=IXDL03
JA2F=IYDF02; JA2L=IYDL03
KA2F=IZF23; KA2L=IZL23

wall (lower section)

IW1F=IXDF01; IW1L=IXDL04
JW1F=IYDF01; JW1L=IYDL01
KW1F=IZF22; KW1L=IZL22

wall (mid-right section)

IW2F=IXDF01; IW2L=IXDL01
JW2F=IYDF02; JW2L=IYDL03
KW2F=IZF22; KW2L=IZL22

wall (mid-left section)

IW3F=IXDF04; IW3L=IXDL04
JW3F=IYDF02; JW3L=IYDL03
KW3F=IZF22; KW3L=IZL22

wall (top section)

IW4F=IXDF01; IW4L=IXDL04
JW4F=IYDF04; JW4L=IYDL04
KW4F=IZF22; KW4L=IZL22

GROUP 11. Initialization of variable or porosity fields

inlet wall protrusion
 &&&LG&&& ACTIVATE FOR BAFFLES EXTENDING INTO DOMAIN &&&4T&&&
 CONPOR(0.0,CELL, IFWF,-IFWL, JFWF, JFWL, KFWF, KFWL)
 CONPOR(0.0,EAST, -IFWL,-IFWL, JFWF, JFWL, KFWL, KFWL+1)

cart
 CONPOR(0.0,CELL, ICAF, ICAL,-JCAF,-JCAL, KCAF, KCAL)

dynamometer exhaust
 CONPOR(0.0,LOW, IDEF, IDEL, JDEF, JDEL,-KDEF,-KDEF)
 CONPOR(0.0,HIGH, IDEF, IDEL, JDEF, JDEL,-KDEL,-KDEL)
 CONPOR(0.0,WEST, -IDEF,-IDEF, JDEF, JDEL, KDEF, KDEL)
 CONPOR(0.0,EAST, -IDEL,-IDEL, JDEF, JDEL, KDEF, KDEL)

exhaust skirt
 &&&LG&&& ACTIVATE FOR EXHAUST SKIRT &&&1T&&&
 CONPOR(0.0,LOW, IESF, IESL, JESF, JESL,-KESF,-KESF)
 CONPOR(0.0,HIGH, IESF, IESL, JESF, JESL,-KESL,-KESL)
 CONPOR(0.0,WEST, -IESF,-IESF, JESF, JESL, KESF, KESL)
 CONPOR(0.0,EAST, -IESL,-IESL, JESF, JESL, KESF, KESL)

CONPOR(0.0,NORTH, IESF, IDEF-1,-JESL,-JESL, KESF, KESL)
 CONPOR(0.0,NORTH, IDEL+1, IESL, -JESL,-JESL, KESF, KESL)
 CONPOR(0.0,NORTH, IDEF, IDEL, -JESL,-JESL, KESF, KDEF-1)
 CONPOR(0.0,NORTH, IDEF, IDEL, -JESL,-JESL, KDEL+1, KESL)

dynamometer
 CONPOR(0.0,CELL, -ID1F,-ID1L, JD1F, JD1L,-KD1F,-KD1L)
 CONPOR(0.0,CELL, -ID2F, ID2L, JD2F, JD2L,-KD2F,-KD2L)
 CONPOR(0.0,CELL, ID3F,-ID3L, JD3F, JD3L,-KD3F,-KD3L)
 CONPOR(0.0,CELL, -ID4F,-ID4L, JD4F, JD4L,-KD4F,-KD4L)
 CONPOR(0.0,CELL, -ID5F,-ID5L, JD5F, JD5L,-KD5F,-KD5L)
 CONPOR(0.0,CELL, -ID6F, ID6L, JD6F, JD6L,-KD6F,-KD6L)
 CONPOR(0.0,CELL, ID7F,-ID7L, JD7F, JD7L,-KD7F,-KD7L)
 CONPOR(0.0,CELL, -ID8F,-ID8L, JD8F, JD8L,-KD8F,-KD8L)
 CONPOR(0.0,CELL, -ID9F,-ID9F, JD9F, JD9L, KD9F, KD9L)
 CONPOR(0.0,CELL, -ID0L,-ID0L, JD0F, JD0L, KD0F, KD0L)

dynamometer plate
 CONPOR(0.0,NORTH, IDPF, IDPL, JDP, JDP, KDPF, KDPL)

plate (WARNING: last 4 hardwired)
 CONPOR(0.0,LOW, IPLF, IPLL, JPLF, JPLL,-KPLF,-KPLF)
 CONPOR(0.0,LOW, 11, 28, 26, 27,-KPLF,-KPLF)
 CONPOR(0.0,LOW, 12, 27, 28, 28,-KPLF,-KPLF)
 CONPOR(0.0,LOW, 13, 26, 29, 29,-KPLF,-KPLF)
 CONPOR(0.0,LOW, 14, 25, 30, 30,-KPLF,-KPLF)

engine
 CONPOR(0.0,SOUTH, IEGF, IEGL,-JEGF,-JEGF, KEGF, KEGL)
 CONPOR(0.0,NORTH, IEGF, IEGL,-JEGF,-JEGF, KEGF, KEGL)
 CONPOR(0.0,WEST, -IEGF,-IEGF, JEGF, JEGF, KEGF, KEGL)
 CONPOR(0.0,EAST, -IEGL,-IEGL, JEGF, JEGF, KEGF, KEGL)

engine plate
 CONPOR(0.0,HIGH, IEPF, IEPL, JEPF, JEPL, KEP, KEP)

augmenter tube (in building)
 CONPOR(0.0,SOUTH, IA1F, IA1L,-JA1F,-JA1F, KA1F, KA1L)
 CONPOR(0.0,NORTH, IA1F, IA1L,-JA1L,-JA1L, KA1F, KA1L)

```
CONPOR(0.0, WEST, -IA1F, -IA1F, JA1F, JA1L, KA1F, KA1L)
CONPOR(0.0, EAST, -IA1L, -IA1L, JA1F, JA1L, KA1F, KA1L)
```

end wall

```
CONPOR(0.0, CELL, IW1F, IW1L, JW1F, -JW1L, -KW1F, -KW1L)
CONPOR(0.0, CELL, IW2F, -IW2L, JW2F, JW2L, -KW2F, -KW2L)
CONPOR(0.0, CELL, -IW3F, IW3L, JW3F, JW3L, -KW3F, -KW3L)
CONPOR(0.0, CELL, IW4F, IW4L, -JW4F, JW4L, -KW4F, -KW4L)
```

augmenter tube (in chimney)

```
CONPOR(0.0, SOUTH, IA2F, IA2L, -JA2F, -JA2F, KA2F, KA2L)
CONPOR(0.0, NORTH, IA2F, IA2L, -JA2L, -JA2L, KA2F, KA2L)
CONPOR(0.0, WEST, -IA2F, -IA2F, JA2F, JA2L, KA2F, KA2L)
CONPOR(0.0, EAST, -IA2L, -IA2L, JA2F, JA2L, KA2F, KA2L)
```

init all

```
FIINIT(H1)=ENTHA
FIINIT(TEMP)=TAMB
FIINIT(RHO1)=RHOAMB
FIINIT(C3)=1.0
FIINIT(KE)=GKE
FIINIT(EP)=GEP
```

init duct

```
PATCH(INITA, INIVAL, IDPF, IDPL, JDP+1, JD4L, KDPF, KDPL, 1, 1)
INIT (INITA, H1, 0.0, ENTHD)
INIT (INITA, TEMP, 0.0, TDYN)
INIT (INITA, RHO1, 0.0, RHODYN)
INIT (INITA, C2, 0.0, 1.0)
INIT (INITA, KE, 0.0, DKE)
INIT (INITA, EP, 0.0, DEP)
INIT (INITA, V1, 0.0, DMDOT/DARSOR/RHODYN)
PATCH(INITB, INIVAL, IDEF, IDEL, JD4L+1, JDEL, KDEF, KDEL, 1, 1)
INIT (INITB, H1, 0.0, ENTHD)
INIT (INITB, TEMP, 0.0, TDYN)
INIT (INITB, RHO1, 0.0, RHODYN)
INIT (INITB, C2, 0.0, 1.0)
INIT (INITB, KE, 0.0, DKE)
INIT (INITB, EP, 0.0, DEP)
INIT (INITB, V1, 0.0, DMDOT/DARSOR/RHODYN)
```

init eng

```
PATCH(INITC, INIVAL, IEGF, IEGL, JEGF, JEGL, KEP+1, KA1F-1, 1, 1)
INIT (INITC, H1, 0.0, ENTHE)
INIT (INITC, TEMP, 0.0, TENG)
INIT (INITC, RHO1, 0.0, RHOENG)
INIT (INITC, C1, 0.0, 1.0)
INIT (INITC, KE, 0.0, EKE)
INIT (INITC, EP, 0.0, EEP)
INIT (INITC, W1, 0.0, EMDOT/EARSOR/RHOENG)
```

init aug

```
PATCH(INITD, INIVAL, IA1F, IA1L, JA1F, JA1L, KA1F, KA2L, 1, 1)
INIT (INITD, H1, 0.0, ENTHE)
INIT (INITD, TEMP, 0.0, TENG)
INIT (INITD, C1, 0.0, 1.0)
INIT (INITD, KE, 0.0, EKE)
INIT (INITD, EP, 0.0, EEP)
INIT (INITD, W1, 0.0, EMDOT/EARSOR/RHOENG)
```

GROUP 12. Convection and diffusion adjustments

GROUP 13. Boundary conditions and special sources
front wall

```

PATCH(XWALL01,LWALL,IFWL+1,NX,1,NY,1,1,1,1)
COVAL(XWALL01,U1,GRND2,0.0)
COVAL(XWALL01,V1,GRND2,0.0)
COVAL(XWALL01,KE,GRND2,GRND2)
COVAL(XWALL01,EP,GRND2,GRND2)
    top wall
PATCH(XWALL02,NWALL,1,NX,NY,NY,IZF01,KDEF-1,1,1)
COVAL(XWALL02,U1,GRND2,0.0)
COVAL(XWALL02,W1,GRND2,0.0)
COVAL(XWALL02,KE,GRND2,GRND2)
COVAL(XWALL02,EP,GRND2,GRND2)
PATCH(XWALL03,NWALL,1,IDEF-1,NY,NY,KDEF,KDEL,1,1)
COVAL(XWALL03,U1,GRND2,0.0)
COVAL(XWALL03,W1,GRND2,0.0)
COVAL(XWALL03,KE,GRND2,GRND2)
COVAL(XWALL03,EP,GRND2,GRND2)
PATCH(XWALL04,NWALL,IDEL+1,NX,NY,NY,KDEF,KDEL,1,1)
COVAL(XWALL04,U1,GRND2,0.0)
COVAL(XWALL04,W1,GRND2,0.0)
COVAL(XWALL04,KE,GRND2,GRND2)
COVAL(XWALL04,EP,GRND2,GRND2)
PATCH(XWALL05,NWALL,1,NX,NY,NY,KDEL+1,KW1F-1,1,1)
COVAL(XWALL05,U1,GRND2,0.0)
COVAL(XWALL05,W1,GRND2,0.0)
COVAL(XWALL05,KE,GRND2,GRND2)
COVAL(XWALL05,EP,GRND2,GRND2)
    bottom wall
PATCH(XWALL06,SWALL,1,NX,1,1,1,KW1F-1,1,1)
COVAL(XWALL06,U1,GRND2,0.0)
COVAL(XWALL06,W1,GRND2,0.0)
COVAL(XWALL06,KE,GRND2,GRND2)
COVAL(XWALL06,EP,GRND2,GRND2)
    side to block wall
PATCH(XWALL07,WWALL,1,1,1,NY,1,KW1F-1,1,1)
COVAL(XWALL07,V1,GRND2,0.0)
COVAL(XWALL07,W1,GRND2,0.0)
COVAL(XWALL07,KE,GRND2,GRND2)
COVAL(XWALL07,EP,GRND2,GRND2)
PATCH(XWALL08,EWALL,NX,NX,1,NY,1,KW1F-1,1,1)
COVAL(XWALL08,V1,GRND2,0.0)
COVAL(XWALL08,W1,GRND2,0.0)
COVAL(XWALL08,KE,GRND2,GRND2)
COVAL(XWALL08,EP,GRND2,GRND2)
    chimney wall
PATCH(XWALL09,NWALL,1,NX,NY,NY,KA2L+1,NZ,1,1)
COVAL(XWALL09,U1,GRND2,0.0)
COVAL(XWALL09,W1,GRND2,0.0)
COVAL(XWALL09,KE,GRND2,GRND2)
COVAL(XWALL09,EP,GRND2,GRND2)
PATCH(XWALL10,SWALL,1,NX,1,1,KW1L+1,NZ,1,1)
COVAL(XWALL10,U1,GRND2,0.0)
COVAL(XWALL10,W1,GRND2,0.0)
COVAL(XWALL10,KE,GRND2,GRND2)
COVAL(XWALL10,EP,GRND2,GRND2)
PATCH(XWALL11,WWALL,1,1,1,NY,KW1L+1,NZ,1,1)
COVAL(XWALL11,V1,GRND2,0.0)
COVAL(XWALL11,W1,GRND2,0.0)
COVAL(XWALL11,KE,GRND2,GRND2)
COVAL(XWALL11,EP,GRND2,GRND2)
PATCH(XWALL12,EWALL,NX,NX,1,NY,KW1L+1,NZ,1,1)

```

```

COVAL(XWALL12,V1,GRND2,0.0)
COVAL(XWALL12,W1,GRND2,0.0)
COVAL(XWALL12,KE,GRND2,GRND2)
COVAL(XWALL12,EP,GRND2,GRND2)
    front opening
    &&&LG&&&  ACTIVATE FOR BAFFLES EXTENDING INTO DOMAIN  &&&4T&&&
    PATCH(XOPEN1,LOW,1,IFWL,1,NY,KFWL+1,KFWL+1,1,1)
    &&&LG&&&  ACTIVATE FOR BAFFLES NOT EXTENDING INTO DOMAIN  &&&4F&&&
    PATCH(XOPEN1,LOW,1,IFWL,1,NY,1,1,1,1)
COVAL(XOPEN1,P1,GRND7,0.0)
COVAL(XOPEN1,W1,ONLYMS,GRND7)
COVAL(XOPEN1,H1,ONLYMS,ENTHA)
COVAL(XOPEN1,KE,ONLYMS,GKE)
COVAL(XOPEN1,EP,ONLYMS,GEP)
COVAL(XOPEN1,UCRT,ONLYMS,XKFCT1)
    dynamometer exhaust
    PATCH(XOPEN2,NORTH,IDEF,IDEL,NY,NY,KDEF,KDEL,1,1)
COVAL(XOPEN2,P1,GRND7,0.0)
COVAL(XOPEN2,V1,ONLYMS,GRND7)
COVAL(XOPEN2,H1,ONLYMS,ENTHA)
COVAL(XOPEN2,UCRT,ONLYMS,XKFCT2)
    chimney exhaust
    PATCH(XOPEN3,HIGH,1,NX,1,NY,NZ,NZ,1,1)
COVAL(XOPEN3,P1,GRND7,0.0)
COVAL(XOPEN3,W1,ONLYMS,SAME)
COVAL(XOPEN3,H1,ONLYMS,ENTHA)
COVAL(XOPEN3,UCRT,ONLYMS,XKFCT3)
    dynamometer mdot sink
    PATCH(XDYNIN,NORTH,IDPF,IDPL,JDP,JDP,KDPF,KDPL,1,1)
COVAL(XDYNIN,P1,FIXFLU,GRND9)
    dynamometer mdot source
    PATCH(XDYNOUT,SOUTH,IDPF,IDPL,JDP+1,JDP+1,KDPF,KDPL,1,1)
COVAL(XDYNOUT,P1,FIXFLU,GRND9)
COVAL(XDYNOUT,V1,ONLYMS,DMDOT/DARSOR/RHODYN*SINO)
COVAL(XDYNOUT,U1,ONLYMS,DMDOT/DARSOR/RHODYN*COSO)
COVAL(XDYNOUT,H1,ONLYMS,ENTHD)
COVAL(XDYNOUT,C2,ONLYMS,1.0)
COVAL(XDYNOUT,KE,ONLYMS,DKE)
COVAL(XDYNOUT,EP,ONLYMS,DEP)
RG(803)=DMDOT/DARSOR
    engine mdot sink
    PATCH(XENGIN,HIGH,IEGF,IEGL,JEGF,JEGL,KEP,KEP,1,1)
COVAL(XENGIN,P1,FIXFLU,GRND10)
RG(804)=(EMDOT-FMDOT)/EARSOR
    engine mdot source
    PATCH(XENGOUT,LOW,IEGF,IEGL,JEGF,JEGL,KEP+1,KEP+1,1,1)
COVAL(XENGOUT,P1,FIXFLU,GRND10)
COVAL(XENGOUT,W1,ONLYMS,EMDOT/EARSOR/RHOENG)
COVAL(XENGOUT,H1,ONLYMS,ENTHE)
COVAL(XENGOUT,C1,ONLYMS,1.0)
COVAL(XENGOUT,KE,ONLYMS,EKE)
COVAL(XENGOUT,EP,ONLYMS,EPE)
RG(805)=EMDOT/EARSOR
    heat transfer augments tube (in building)
    PATCH(HEATTR1E,EWALL,IA1F-1,IA1F,JA1F,JA1L,KA1F,KA1L,1,1)
COVAL(HEATTR1E,H1,GRND8,GRND8); COVAL(HEATTR1E,UCRT,COND1,THICK1)
    PATCH(HEATTR1W,WALL,IA1L,IA1L+1,JA1F,JA1L,KA1F,KA1L,1,1)
COVAL(HEATTR1W,H1,GRND8,GRND8); COVAL(HEATTR1W,UCRT,COND1,THICK1)
    PATCH(HEATTR1N,NWALL,IA1F,IA1L,JA1F-1,JA1F,KA1F,KA1L,1,1)
COVAL(HEATTR1N,H1,GRND8,GRND8); COVAL(HEATTR1N,UCRT,COND1,THICK1)

```

```

PATCH(HEATTR1S,SWALL,IA1F,IA1L,JA1L,JA1L+1,KA1F,KA1L,1,1)
COVAL(HEATTR1S,H1,GRND8,GRND8);COVAL(HEATTR1S,UCRT,COND1,THICK1)
    heat transfer augmenter tube (in chimney)
PATCH(HEATTR2E,EWALL,IA2F-1,IA2F,JA2F,JA2L,KA2F,KA2L,1,1)
COVAL(HEATTR2E,H1,GRND8,GRND8);COVAL(HEATTR2E,UCRT,COND2,THICK2)
PATCH(HEATTR2W,WWALL,IA2L,IA2L+1,JA2F,JA2L,KA2F,KA2L,1,1)
COVAL(HEATTR2W,H1,GRND8,GRND8);COVAL(HEATTR2W,UCRT,COND2,THICK2)
PATCH(HEATTR2N,NWALL,IA2F,IA2L,JA2F-1,JA2F,KA2F,KA2L,1,1)
COVAL(HEATTR2N,H1,GRND8,GRND8);COVAL(HEATTR2N,UCRT,COND2,THICK2)
PATCH(HEATTR2S,SWALL,IA2F,IA2L,JA2L,JA2L+1,KA2F,KA2L,1,1)
COVAL(HEATTR2S,H1,GRND8,GRND8);COVAL(HEATTR2S,UCRT,COND2,THICK2)

```

GROUP 14. Downstream pressure for PARAB=.TRUE.

GROUP 15. Termination of sweeps

FSWEEP=2

LSWEEP=2500

*

```

*****
*****                      USER CONTROLS                      *****
*****

```

```

***                                                                    ***

```

```

***   The following integer arrays are described below.               ***

```

```

***                                                                    ***

```

```

***   IG(901)  --  Frequency of ground printout on wall heat          ***

```

```

***                      transfer.                                     ***

```

```

***   IG(902)  --  Frequency of restart files and English unit        ***

```

```

***                      calculation.                                   ***

```

```

***   IG(999)  --  Set to 1 to stop run on first sweep.               ***

```

```

***   IG( 38)  --  Set to 1 for first set of spot value info.         ***

```

```

***   IG( 39)  --  Set to 1 for second set of spot value info.        ***

```

```

***   IG( 40)  --  Set to 1 for third set of spot value info.         ***

```

```

***   IG( 41)  --  Set to 1 for additional heat transfer info.        ***

```

```

***                                                                    ***

```

IG(901)=50

IG(902)=100

IG(999)=0

GROUP 16. Termination of iterations

LITER(P1)=30

ENDIT(P1)=1.0E-3

ENDIT(H1)=1.0E-2

RESREF(P1)=1.0E-8

RESREF(U1)=1.0E-8

RESREF(V1)=1.0E-8

RESREF(W1)=1.0E-8

RESREF(H1)=1.0E-8

RESREF(C1)=1.0E-8

RESREF(C2)=1.0E-8

RESREF(KE)=1.0E-8

RESREF(EP)=1.0E-8

GROUP 17. Under-relaxation devices

RELAX(P1,LINRLX,0.1)

RELAX(KE,LINRLX,0.1)

RELAX(EP,LINRLX,0.1)

RELAX(U1,FALSDT,0.00025)

RELAX(V1,FALSDT,0.00025)

RELAX(W1,FALSDT,0.00025)

RELAX(H1,FALSDT,0.0005)

RELAX(C1,FALSDT,0.0005)

RELAX(C2,FALS DT,0.0005)

GROUP 18. Limits on variables or increments to them

VARMAX(C1)=1.00;VARMIN(C1)=1.0E-10

VARMAX(C2)=1.00;VARMIN(C2)=1.0E-10

VARMAX(ENUT)=10000000.*ENUL

GROUP 19. Data communicated by satellite to GROUND

GROUP 20. Preliminary print-out

GROUP 21. Print-out of variables

OUTPUT(P1,Y,Y,N,Y,Y,Y)

OUTPUT(U1,Y,N,N,Y,Y,Y)

OUTPUT(V1,Y,N,N,Y,Y,Y)

OUTPUT(W1,Y,N,N,Y,Y,Y)

OUTPUT(KE,N,N,N,Y,Y,Y)

OUTPUT(EP,N,N,N,Y,Y,Y)

OUTPUT(H1,N,N,N,Y,Y,Y)

OUTPUT(C1,N,N,N,Y,Y,Y)

OUTPUT(C2,N,N,N,Y,Y,Y)

OUTPUT(C3,N,N,N,N,N,N)

OUTPUT(TEMP,Y,N,N,N,N,N)

OUTPUT(CP,N,N,N,N,N,N)

OUTPUT(C6,N,N,N,N,N,N)

OUTPUT(C7,N,N,N,N,N,N)

OUTPUT(U2,N,N,N,N,N,N)

OUTPUT(V2,N,N,N,N,N,N)

OUTPUT(W2,N,N,N,N,N,N)

OUTPUT(PH2O,N,N,N,N,N,N)

OUTPUT(TFAR,N,N,N,N,N,N)

OUTPUT(RHOE,N,N,N,N,N,N)

OUTPUT(SPAR,N,N,N,N,N,N)

OUTPUT(RHO1,Y,N,N,N,N,N)

OUTPUT(UCRT,N,N,N,N,N,N)

OUTPUT(VCRT,N,N,N,N,N,N)

OUTPUT(WCRT,N,N,N,N,N,N)

GROUP 22. Spot-value print-out

IXMON =19;IYMON = 6;IZMON =11

IXMON1=13;IYMON1= 8;IZMON1=15

IXMON2=19;IYMON2=16;IZMON2=26

IXMON3=16;IYMON3=13;IZMON3=39

IXMON4=19;IYMON4=22;IZMON4=44

IXMON5=10;IYMON5=11;IZMON5=59

IXMON6=17;IYMON6=13;IZMON6=71

IXMON7=17;IYMON7=17;IZMON7=71

IXMON8=17;IYMON8=20;IZMON8=71

IXMON9=19;IYMON9=16;IZMON9=73

IG(11)=IXMON1;IG(12)=IYMON1;IG(13)=IZMON1

IG(14)=IXMON2;IG(15)=IYMON2;IG(16)=IZMON2

IG(17)=IXMON3;IG(18)=IYMON3;IG(19)=IZMON3

IG(20)=IXMON4;IG(21)=IYMON4;IG(22)=IZMON4

IG(23)=IXMON5;IG(24)=IYMON5;IG(25)=IZMON5

IG(26)=IXMON6;IG(27)=IYMON6;IG(28)=IZMON6

IG(29)=IXMON7;IG(30)=IYMON7;IG(31)=IZMON7

IG(32)=IXMON8;IG(33)=IYMON8;IG(34)=IZMON8

IG(35)=IXMON9;IG(36)=IYMON9;IG(37)=IZMON9

IG(38)=1

IG(39)=1

IG(40)=0

IG(41)=0

GROUP 23. Field print-out and plot control

YZPR=T;IXPRF=19;IXPRL=19

TSTSWP=5; NPRMON=5

NPRINT=LSWEEP; IPLTL=LSWEEP; ITABL=3
ABSIZ=.8; ORSIZ=.8; NUMCLS=10
NPLT=25

GROUP 24. Dumps for restarts

RESTRT(ALL);NAMFI=INXS

RESTRT(ALL);NAMFI=IIII

STOP

APPENDIX C

```

C THIS IS THE MAIN PROGRAM OF THE SATELLITE
  PROGRAM MAIN
C FILE NAME satlit.f 09/27/87
C
C (C) COPYRIGHT 1984, LAST REVISION 1987.
C CONCENTRATION HEAT AND MOMENTUM LTD. ALL RIGHTS RESERVED.
C This subroutine and the remainder of the PHOENICS code are
C proprietary software owned by Concentration Heat and Momentum
C Limited, 40 High Street, Wimbledon, London SW19 5AU, England.
C
C LOGICAL TALK,RUN,LVAL
C EXTERNAL WAYOUT
C
C 1 Set dimensions of blank-COMMON arrays here. WARNING: the
C corresponding blank-COMMON arrays in subroutine SATLIT must
C have the same dimensions.
C PARAMETER (NXFD=1000,NYFD=1000,NZFD=1000,NTFD=10000)
C PARAMETER (NTCVD=25000,NBFCD=500000)
C COMMON TCVDA(NTCVD),XFRAC(NXFD),YFRAC(NYFD),ZFRAC(NZFD),
C 1TFRAC(NTFD),BFCS(NBFCD)
C
C 2 Set dimensions of PATCH-name array and the instruction-stack
C array here. The dimension of the array NLN must be the same
C as that of STACK. WARNING: the array NAMPAT in the MAIN
C program of EARTH (see GROUND) must have the same dimension.
C These are specified by the parameters npatd and nld, set below.
C PARAMETER (NPATD=1000,NLD=2000)
C COMMON/NPAT/NAMPAT(NPATD)/NSTCK/STACK(NLD)/LINENO/NLN(NLD)
C CHARACTER NAMPAT*8,STACK*72
C COMMON/CNFG/CNFIG
C CHARACTER CNFIG*48
C
C 3 Set dimension of run array to MAXRUN.
C PARAMETER (NRUND=500)
C COMMON/RUNS/RUN(NRUND)
C
C 4 Set dimensions of data-for-GROUND arrays here. WARNING: the
C corresponding arrays in the MAIN program of EARTH (see
C GROUND) must have the same dimensions.
C PARAMETER (NLGD=1000,NIGD=1000,NRGD=10000,NCGD=1000)
C COMMON/LGRND/LG(NLGD)/IGRND/IG(NIGD)/RGRND/RG(NRGD)
C COMMON/CGRND/CG(NCGD)
C LOGICAL LG
C CHARACTER*4 CG
C
C 5 Set dimensions of data-for-GREX1 arrays here. WARNING: the
C corresponding arrays in the MAIN program of EARTH (see
C GROUND) must have the same dimensions.
C COMMON/LSG/LSGD(20)/ISG/ISGD(20)/RSG/RSGD(100)/CSG/CSGD(10)
C LOGICAL LSGD
C CHARACTER*4 CSGD
C
C 6 Set dimensions for user-declared PIL variables here.
C PARAMETER (NIPD=1000,NRPD=1000)
C COMMON/NIDEC/INDEC(NIPD)/IDEC/INVAL(NIPD)
C COMMON/NRDEC/REDEC(NRPD)/RDEC/REVAL(NRPD)
C CHARACTER REDEC*6,INDEC*6
C
C 7 For more than the default of 80 variables increase nvd.
C WARNING: the corresponding parameter nvd in the MAIN program of

```

```

C      EARTH (see ground.f) must be the same.
      PARAMETER (NVD=80)
      COMMON/LDB1/DBGPHI(NVD)/IDA1/ITERMS(NVD)/IDA2/LITER(NVD)
      1/IDA3/IORCVF(NVD)/IDA4/IORCVL(NVD)/IDA5/ISLN(NVD)/IDA6/IPRN(NVD)
      1/HDA1/NAME(NVD)/RDA1/DTFALS(NVD)/RDA2/RESREF(NVD)
      1/RDA3/PRNDTL(NVD)/RDA4/PRT(NVD)/RDA5/ENDIT(NVD)/RDA6/VARMIN(NVD)
      1/RDA7/VARMAX(NVD)/RDA8/FIINIT(NVD)/RDA9/PHINT(NVD)
      1/RDA10/CINT(NVD)/RDA11/EX(NVD)
      1/IPIP1/IP1(NVD)/HPIP2/IHP2(NVD)/RPIP1/RVAL(NVD)
      1/LPIP1/LVAL(NVD)

C
C 8      Set dimension indicators to correspond with above dimensions.
      CALL SUB4(MAXTCV,NTCVD,MAXRUN,NRUN,NBFC,NBFC,NUMPHI,NVD)
      CALL SUB4(NLG,NLGD,NIG,NIGD,NRG,NRGD,NCG,NCGD)
      CALL SUB4(NLSG,20,NISG,20,NRSG,100,NCSG,10)
      CALL SUB4(NIPIL,NIPD,NRPIL,NRPD,NPNAM,NPATD,NSTACK,NLD)
      CALL SUB4(NXFR,NXFD,NYFR,NYFD,NZFR,NZFD,NTFR,NTFD)

C
C 9      Logical unit numbers & file names.
      CALL CNFGZZ(1)
      CALL OPENFL(6)
      CALL OPENFL(5)
      CALL READQ1(TALK,RUN,MAXRUN)

C
      CALL SMAIN1(TALK,MAXTCV,MAXRUN,NBFC,NUMPHI,NLG,NIG,NRG,NCG,
      1NLSG,NISG,NRSG,NCSG,NIPIL,NRPIL,NPNAM,NSTACK,NXFR,NYFR,NZFR,
      1NTFR)
      CALL WAYOUT(0)
      END

C*****
      SUBROUTINE SAT
C
      #include "satear"
      #include "satloc"
C---- Call satellite used in BFC test-battery.
      CALL BFCTST
C---- the users USERST subroutine.
      IF(NAMSAT.EQ.'USER') CALL USERST
C---- Call the SATLIT subroutine.
      CALL SATLIT
      RETURN
      END

C*****
      SUBROUTINE BFCTST
C
      #include "satear"
      #include "satloc"
      PARAMETER (NLGD=1000,NIGD=1000,NRGD=10000,NCGD=1000)
      COMMON/LGRND/LG(NLGD)/IGRND/IG(NIGD)/RGRND/RG(NRGD)
      COMMON/CGRND/CG(NCGD)
      LOGICAL LG
C
C---- Special sequence for BFC test battery : IG(1)=28
C
      IF(.NOT.(BFC.AND.IG(1).EQ.28.AND.IGR.EQ.1)) RETURN
      L1=MIN0(IG(2),NZ)
      IF(L1.LT.1) GO TO 2
      DO 1 IZ=1,L1
1 CALL XCYIZ(IZ,LG(10))
2 L2=MAX0(1,IG(3))

```

```

        IF(L2.GT.NZ) RETURN
        DO 3 IZ=L2,NZ
3      CALL XCYIZ(IZ,LG(10))
        RETURN
        END
        SUBROUTINE USERST
        CALL WRIT40('DUMMY SUBROUTINE USERST CALLED.      ')
        RETURN
        END
C*****
SUBROUTINE SATLIT
C
#include "satear"
#include "satloc"
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1  Set dimensions of blank-COMMON arrays here to the
C     dimensions of the same arrays in the MAIN program of the
C     satellite.
        PARAMETER (NXFD=1000,NYFD=1000,NZFD=1000,NTFD=10000)
        PARAMETER (NTCVD=25000,NBFCD=500000)
        COMMON TCVDA(NTCVD),XFRAC(NXFD),YFRAC(NYFD),ZFRAC(NZFD),
1TFRAC(NTFD),BFCS(NBFCD)
C
C 2  Set dimensions of data-for-GROUND arrays here. WARNING: the
C     corresponding arrays in the MAIN program of the
C     satellite program and the EARTH program must have the same
C     dimensions.
        PARAMETER (NLGD=1000,NIGD=1000,NRGD=10000,NCGD=1000)
        COMMON/LGRND/LG(NLGD)/IGRND/IG(NIGD)/RGRND/RG(NRGD)
        COMMON/CGRND/CG(NCGD)
        LOGICAL LG
        CHARACTER*4 CG
C
C 3  Introduce SATLIT-only commons, arrays, equivalences.
C
        DIMENSION SC(4),IX(16),XL(16),XP(16),IY(16),YL(16),YP(16),
&                NZC(26),ZL(26),ZP(26),IZT(26),IZF1(26),IZF2(26),
&                XAS(2500),YAS(2500),ZAS(2500),XAS1(2500),YAS1(2500),
&                ZAS1(2500),XAS2(2500),YAS2(2500),ZAS2(2500),ZASL(100)
C
C 4  User places his data statements here.
C
        GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
122,23,24),IGR
C
C--- GROUP 1. Run title and other preliminaries
1  CONTINUE
    WRITE(6,*)'                               IN SATLIT '
    RETURN
C
C--- GROUP 2. Transience; time-step specification
2  CONTINUE
    RETURN
C
C--- GROUP 3. X-direction grid specification
3  CONTINUE
    RETURN
C
C--- GROUP 4. Y-direction grid specification

```

```

4 CONTINUE
  RETURN
C
C--- GROUP 5. Z-direction grid specification
5 CONTINUE
C
  IF(IG(1).GT.2) RETURN
  IF(IG(1).EQ.0) WRITE(6,*)' CREATING GRID INPUT FILES'
  IF(IG(1).GE.1) WRITE(6,*)' CALCULATING INLET LOCATION'
C
C*****
C-pd---First grid input file located at some distance between the----
C--- inlet and the dynamometer. There are no arcs in this-----
C--- section.-----
C
  NI=14
  IX(1)=1
  CALL SETIV(IX,IG,100,1,NI)
C
  XCENA=RG(41)
  YCENA=RG(42)
  RAD1=RG(50)/2.
  DXI=(RAD1*RAD1/2.)**0.5
  IFST=IG(117)
  JFST=IG(137)
  ILST=IFST+1
  JLST=JFST+1
C
  XL(1 )=0.0
  CALL SETRV(XL,RG,100,1,NI)
  XL(IFST )=XCENA-DXI
  XL(IFST+1)=XCENA+DXI
C
  CALL SETRV(XP,RG,120,2,NI)
C
  IY(1 )=1
  CALL SETIV(IY,IG,120,1,NI)
C
  YL(1 )=0.0
  CALL SETRV(YL,RG,140,1,NI)
  YL(JFST )=YCENA-DXI
  YL(JFST+1)=YCENA+DXI
C
  CALL SETRV(YP,RG,160,2,NI)
C
  LU=62
  CG(LU)='CS62'
  OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
  IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
  IRX=IG(42)
  IRY=IG(43)
  CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
  CALL WRTFI(LU,IRX,IRY,IX,IY)
  WRITE(LU,105) IX(1),IX(IRX+1),IY(1),IY(IRY+1)
C
C
C*****
C-pd---Second grid input file located at a dynamometer cross-----
C--- section. This input file contains the arcs for the-----
C--- inlet. Unifrom grid spacing across the arc is assumed.-----

```

```

C
  LU=63
  CG(LU)='CS63'
  OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
  IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
  CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C
C-pd---Overwrite line info with arc data-----
C
  ANG1= 45.0
  ANG2=135.0
  ANG3=225.0
  ANG4=315.0
  WRITE(LU,*)
  WRITE(LU,104)IX(IFST),IX(ILST),IY(JFST),IY(JFST),
&          XCENA,YCENA,RAD1,ANG3,ANG4,XP(IFST)
  WRITE(LU,104)IX(IFST),IX(ILST),IY(JLST),IY(JLST),
&          XCENA,YCENA,RAD1,ANG2,ANG1,XP(IFST)
  WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST),IY(JLST),
&          XCENA,YCENA,RAD1,ANG3,ANG2,YP(JFST)
  WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST),IY(JLST),
&          XCENA,YCENA,RAD1,ANG4,ANG1,YP(JFST)
C
  CALL WRTFI(LU,IRX,IRY,IX,IY)
C
C-pd---Fix points around circle and certain ones inside-----
C
  WRITE(LU,*)
  WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(JFST)
  WRITE(LU,105)IX(1),IX(IFST),IY(JFST),IY(JLST)
  WRITE(LU,105)IX(ILST),IX(IRX+1),IY(JFST),IY(JLST)
  WRITE(LU,105)IX(1),IX(IRX+1),IY(JLST),IY(IRY+1)
C
  ISOL=0
  WRITE(LU,105)IX(IFST)+ISOL,IX(IFST+1)-ISOL,IY(JFST),IY(JFST+1)
  WRITE(LU,105)IX(IFST),IX(IFST+1),IY(JFST)+ISOL,IY(JFST+1)-ISOL
C
C
C*****
C-pd---Third grid input file located at a inlet of calculation-----
C--- domain. This input file uses the data from the first input---
C--- file and overwrites one of the vertical lines in order to-----
C--- correspond to the inlet opening.-----
C
  DELO=50000.
  DO 510 I=1,IRX+1
    DEL=ABS(XL(I)-RG(65))
    IF (DEL.LT.DELO) THEN
      IPT=I
      DELO=DEL
    ENDIF
  510 CONTINUE
  XL(IPT)=RG(65)
C
C-pd---Send back to g1 for boundary condition-----
  IG(61)=IX(IPT)-1
  IF(IG(1).EQ.2) RETURN
C
  LU=61
  CG(LU)='CS61'

```

```

OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
CALL WRTFI(LU,IRX,IRY,IX,IY)
WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(IRY+1)
C
C
C*****
C-pd---Fourth grid input file located at a inlet of engine.-----
C
      IX(1)=1
      CALL SETIV(IX,IG,140,1,NI)
C
      XCENB=RG(43)
      YCENB=RG(44)
      RAD1=RG(51)/2.
      DXI=(RAD1*RAD1/2.)**0.5
      IFST=IG(157)
      JFST=IG(177)
      IMID=IFST+1
      JMID=JFST+1
      ILST=IFST+2
      JLST=JFST+2
      IBEF=IFST-1
      JBEF=JFST-1
      IAFI=IFST+3
      JAFI=JFST+3
C
      XL(1)=0.0
      CALL SETRV(XL,RG,180,1,NI)
      XL(IFST)=XCENB-DXI
      XL(IFST+2)=XCENB+DXI
C
      CALL SETRV(XP,RG,200,2,NI)
C
      IY(1)=1
      CALL SETIV(IY,IG,160,1,NI)
C
      YL(1)=0.0
      CALL SETRV(YL,RG,220,1,NI)
      YL(JFST)=YCENB-DXI
      YL(JFST+2)=YCENB+DXI
C
      CALL SETRV(YP,RG,240,2,NI)
C
      LU=64
      CG(LU)='CS64'
      OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
      IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
      IRX=IG(44)
      IRY=IG(45)
      CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C
C-pd---Overwrite line info with arc data-----
C
      ANG1= 0.0
      ANG2= 45.0
      ANG3= 90.0
      ANG4=135.0
      ANG5=180.0

```

```

ANG6=225.0
ANG7=270.0
ANG8=315.0
WRITE(LU,*)
WRITE(LU,104)IX(IFST),IX(IMID),IY(JFST),IY(JLST),
&          XCENB,YCENB,RAD1,ANG6,ANG7,XP(IFST)
WRITE(LU,104)IX(IMID),IX(ILST),IY(JFST),IY(JLST),
&          XCENB,YCENB,RAD1,ANG7,ANG8,XP(IMID)
WRITE(LU,104)IX(IFST),IX(IMID),IY(JLST),IY(JLST),
&          XCENB,YCENB,RAD1,ANG4,ANG3,XP(IFST)
WRITE(LU,104)IX(IMID),IX(ILST),IY(JLST),IY(JLST),
&          XCENB,YCENB,RAD1,ANG3,ANG2,XP(IMID)
WRITE(LU,104)IX(IFST),IX(IFST),IY(JFST),IY(JMID),
&          XCENB,YCENB,RAD1,ANG6,ANG5,YP(JFST)
WRITE(LU,104)IX(IFST),IX(IFST),IY(JMID),IY(JLST),
&          XCENB,YCENB,RAD1,ANG5,ANG4,YP(JMID)
WRITE(LU,104)IX(ILST),IX(ILST),IY(JFST),IY(JMID),
&          XCENB,YCENB,RAD1,ANG8,ANG1,YP(JFST)
WRITE(LU,104)IX(ILST),IX(ILST),IY(JMID),IY(JLST),
&          XCENB,YCENB,RAD1,ANG1,ANG2,YP(JMID)

```

C

C-pd---Shuffle lines-----

C

```

WRITE(LU,102)IX(IMID),IX(IMID),IY(JBEF),IY(JFST),
&          XL(IMID),YL(JBEF),XL(IMID),YCENB-RAD1,YP(JBEF)
WRITE(LU,102)IX(IMID),IX(IMID),IY(JFST),IY(JMID),
&          XL(IMID),YCENB-RAD1,XL(IMID),YCENB,YP(JFST)
WRITE(LU,102)IX(IMID),IX(IMID),IY(JMID),IY(JLST),
&          XL(IMID),YCENB,XL(IMID),YCENB+RAD1,YP(JMID)
WRITE(LU,102)IX(IMID),IX(IMID),IY(JLST),IY(JAFT),
&          XL(IMID),YCENB+RAD1,XL(IMID),YL(JAFT),YP(JLST)
WRITE(LU,102)IX(IBEf),IX(IFST),IY(JMID),IY(JMID),
&          XL(IBEf),YL(JMID),XCENB-RAD1,YL(JMID),XP(IBEf)
WRITE(LU,102)IX(IFST),IX(IMID),IY(JMID),IY(JMID),
&          XCENB-RAD1,YL(JMID),XCENB,YL(JMID),XP(IFST)
WRITE(LU,102)IX(IMID),IX(ILST),IY(JMID),IY(JMID),
&          XCENB,YL(JMID),XCENB+RAD1,YL(JMID),XP(IMID)
WRITE(LU,102)IX(ILST),IX(IAFT),IY(JMID),IY(JMID),
&          XCENB+RAD1,YL(JMID),XL(IAFT),YL(JMID),XP(ILST)

```

C

CALL WRTFI(LU,IRX,IRY,IX,IY)

C

C-pd---Fix points around circle and certain ones inside-----

C

```

WRITE(LU,*)
WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(JFST)
WRITE(LU,105)IX(1),IX(IFST),IY(JFST),IY(JLST)
WRITE(LU,105)IX(ILST),IX(IRX+1),IY(JFST),IY(JLST)
WRITE(LU,105)IX(1),IX(IRX+1),IY(JLST),IY(IRY+1)

```

C

ISOL=2

```

WRITE(LU,105)IX(IFST)+ISOL,IX(ILST)-ISOL,IY(JFST),IY(JLST)
WRITE(LU,105)IX(IFST),IX(ILST),IY(JFST)+ISOL,IY(JLST)-ISOL

```

C

C

C*****

C-pd---Fifth grid input file located at a exit of engine. There-----

C--- are two options for this location. This one is for a gap of---

C--- approximately 3 or more inches between the exit of the -----

C--- engine and the augments tube. The other option is for a-----

```

C--- circle inside a circle. Note: This case uses the same-----
C--- spacing as for the engine inlet.-----
C
      IF(LG(2)) GOTO 560
C
      RAD1=RG(52)/2.
      DXI=(RAD1*RAD1/2.)*0.5
      IFST=IG(157)
      JFST=IG(177)
      IMID=IFST+1
      JMID=JFST+1
      ILST=IFST+2
      JLST=JFST+2
      IBEF=IFST-1
      JBEF=JFST-1
      IAFT=IFST+3
      JAFT=JFST+3
C
      XL(IFST )=XCENB-DXI
      XL(IFST+2)=XCENB+DXI
C
      YL(JFST )=YCENB-DXI
      YL(JFST+2)=YCENB+DXI
C
      LU=65
      CG(LU)='CS65'
      OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
      IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
      CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C
C-pd---Overwrite line info with arc data-----
C
      ANG1= 0.0
      ANG2= 45.0
      ANG3= 90.0
      ANG4=135.0
      ANG5=180.0
      ANG6=225.0
      ANG7=270.0
      ANG8=315.0
      WRITE(LU,*)
      WRITE(LU,104) IX(IFST),IX(IMID),IY(JFST),IY(JFST),
&                XCENB,YCENB,RAD1,ANG6,ANG7,XP(IFST)
      WRITE(LU,104) IX(IMID),IX(ILST),IY(JFST),IY(JFST),
&                XCENB,YCENB,RAD1,ANG7,ANG8,XP(IMID)
      WRITE(LU,104) IX(IFST),IX(IMID),IY(JLST),IY(JLST),
&                XCENB,YCENB,RAD1,ANG4,ANG3,XP(IFST)
      WRITE(LU,104) IX(IMID),IX(ILST),IY(JLST),IY(JLST),
&                XCENB,YCENB,RAD1,ANG3,ANG2,XP(IMID)
      WRITE(LU,104) IX(IFST),IX(IFST),IY(JFST),IY(JMID),
&                XCENB,YCENB,RAD1,ANG6,ANG5,YP(JFST)
      WRITE(LU,104) IX(IFST),IX(IFST),IY(JMID),IY(JLST),
&                XCENB,YCENB,RAD1,ANG5,ANG4,YP(JMID)
      WRITE(LU,104) IX(ILST),IX(ILST),IY(JFST),IY(JMID),
&                XCENB,YCENB,RAD1,ANG8,ANG1,YP(JFST)
      WRITE(LU,104) IX(ILST),IX(ILST),IY(JMID),IY(JLST),
&                XCENB,YCENB,RAD1,ANG1,ANG2,YP(JMID)
C
C-pd---Shuffle lines-----
C

```

```

WRITE(LU,102) IX(IMID), IX(IMID), IY(JBEF), IY(JFST),
& XL(IMID), YL(JBEF), XL(IMID), YCENB-RAD1, YP(JBEF)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JFST), IY(JMID),
& XL(IMID), YCENB-RAD1, XL(IMID), YCENB, YP(JFST)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JMID), IY(JLST),
& XL(IMID), YCENB, XL(IMID), YCENB+RAD1, YP(JMID)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JLST), IY(JAFT),
& XL(IMID), YCENB+RAD1, XL(IMID), YL(JAFT), YP(JLST)
WRITE(LU,102) IX(IBEf), IX(IFST), IY(JMID), IY(JMID),
& XL(IBEf), YL(JMID), XCENB-RAD1, YL(JMID), XP(IBEf)
WRITE(LU,102) IX(IFST), IX(IMID), IY(JMID), IY(JMID),
& XCENB-RAD1, YL(JMID), XCENB, YL(JMID), XP(IFST)
WRITE(LU,102) IX(IMID), IX(ILST), IY(JMID), IY(JMID),
& XCENB, YL(JMID), XCENB+RAD1, YL(JMID), XP(IMID)
WRITE(LU,102) IX(ILST), IX(IAFT), IY(JMID), IY(JMID),
& XCENB+RAD1, YL(JMID), XL(IAFT), YL(JMID), XP(ILST)

```

```

C
CALL WRTFI(LU,IRX,IRY,IX,IY)

```

```

C
C-pd---Fix points around circle and certain ones inside-----
C

```

```

WRITE(LU,*)
WRITE(LU,105) IX(1), IX(IRX+1), IY(1), IY(JFST)
WRITE(LU,105) IX(1), IX(IFST), IY(JFST), IY(JLST)
WRITE(LU,105) IX(ILST), IX(IRX+1), IY(JFST), IY(JLST)
WRITE(LU,105) IX(1), IX(IRX+1), IY(JLST), IY(IRY+1)

```

```

C
ISOL=2
WRITE(LU,105) IX(IFST)+ISOL, IX(ILST)-ISOL, IY(JFST), IY(JLST)
WRITE(LU,105) IX(IFST), IX(ILST), IY(JFST)+ISOL, IY(JLST)-ISOL

```

```

C
C
C*****
C-pd---This is the second option for exit of the engine. There will--
C--- will be either 2 or 3 cross sections written here depending---
C--- on the location exit. It will be 2 if it ends before the-----
C--- augmeter tube or if it ends at the start of the tapered-----
C--- section or at the start of straight section. It will be 3-----
C--- if it falls in the tapered section or after the start of-----
C--- the straight section.-----
C

```

```

560 CONTINUE
DO 561 I=1,IG(60)

```

```

C
IX(1)=1
CALL SETIV(IX,IG,180,1,NI)

```

```

C
XCENB=RG(43)
YCENB=RG(44)
XCENC=RG(45)
YCENC=RG(46)
RAD1=RG(53+I)/2.
DXI=(RAD1*RAD1/2.)**0.5
RAD2=RG(56+I)/2.
DXII=(RAD2*RAD2/2.)**0.5
IFST=IG(197)
JFST=IG(217)

```

```

C
C-pd---Do trig-----
C

```

```

DXI02=DXI+(YCENB-YCENC)
TETT02=ASIN(DXI02/RAD2)*180./3.141592654
DXI16=DXI-(YCENB-YCENC)
TETT16=ASIN(DXI16/RAD2)*180./3.141592654
DXI04=DXI+(XCENB-XCENC)
TETT04=ASIN(DXI04/RAD2)*180./3.141592654
DXI06=DXI-(XCENB-XCENC)
TETT06=ASIN(DXI06/RAD2)*180./3.141592654
DXI08=DXI+(YCENB-YCENC)
TETT08=ASIN(DXI08/RAD2)*180./3.141592654
DXI10=DXI-(YCENB-YCENC)
TETT10=ASIN(DXI10/RAD2)*180./3.141592654
DXI12=DXI-(XCENB-XCENC)
TETT12=ASIN(DXI12/RAD2)*180./3.141592654
DXI14=DXI+(XCENB-XCENC)
TETT14=ASIN(DXI14/RAD2)*180./3.141592654

```

C

```

XL(1)=0.0
CALL SETRV(XL, RG, 260, 1, NI)
XL(IFST+1)=XCENB-DXI
XL(IFST+3)=XCENB+DXI
XL(IFST)=XCENC-DXII
XL(IFST+4)=XCENC+DXII

```

C

```

CALL SETRV(XP, RG, 280, 2, NI)

```

C

```

IY(1)=1
CALL SETIV(IY, IG, 200, 1, NI)

```

C

```

YL(1)=0.0
CALL SETRV(YL, RG, 300, 1, NI)
YL(JFST+1)=YCENB-DXI
YL(JFST+3)=YCENB+DXI
YL(JFST)=YCENC-DXII
YL(JFST+4)=YCENC+DXII

```

C

```

CALL SETRV(YP, RG, 320, 2, NI)

```

C

```

LU=64+I
CG(LU)='CS'
I10=LU/10
I1=LU-I10*10
WRITE(CG(LU)(3:3), '(I1)') I10
WRITE(CG(LU)(4:4), '(I1)') I1
OPEN(LU, FILE=CG(LU), FORM='FORMATTED', STATUS='UNKNOWN')
IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
IRX=IG(46)
IRY=IG(47)
CALL WRTSQ(LU, NX, NY, IRX, IRY, IX, IY, XL, YL, XP, YP)

```

C

C-pd---Overwrite line info with arc data-----

C-pd---Inner circle-----

C

```

ANG1= 0.0
ANG2= 45.0
ANG3= 90.0
ANG4=135.0
ANG5=180.0
ANG6=225.0
ANG7=270.0

```

```

ANG8=315.0
IFST=IG(197)+1
JFST=IG(217)+1
IMID=IFST+1
JMID=JFST+1
ILST=IFST+2
JLST=JFST+2
IBEF=IFST-1
JBEF=JFST-1
IAFT=IFST+3
JAFT=JFST+3
WRITE(LU,*)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JFST), IY(JFST),
& XCENB, YCENB, RAD1, ANG6, ANG7, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JFST), IY(JFST),
& XCENB, YCENB, RAD1, ANG7, ANG8, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JLST), IY(JLST),
& XCENB, YCENB, RAD1, ANG4, ANG3, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JLST), IY(JLST),
& XCENB, YCENB, RAD1, ANG3, ANG2, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JFST), IY(JMID),
& XCENB, YCENB, RAD1, ANG6, ANG5, YP(JFST)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JMID), IY(JLST),
& XCENB, YCENB, RAD1, ANG5, ANG4, YP(JMID)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JFST), IY(JMID),
& XCENB, YCENB, RAD1, ANG8, ANG1, YP(JFST)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JMID), IY(JLST),
& XCENB, YCENB, RAD1, ANG1, ANG2, YP(JMID)

```

C
C-pd---Shuffle lines-----
C

```

WRITE(LU,102) IX(IMID), IX(IMID), IY(JFST), IY(JMID),
& XCENB, YCENB-RAD1, XCENB, YCENB, YP(JFST)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JMID), IY(JLST),
& XCENB, YCENB, XCENB, YCENB+RAD1, YP(JMID)
WRITE(LU,102) IX(IFST), IX(IMID), IY(JMID), IY(JMID),
& XCENB-RAD1, YCENB, XCENB, YCENB, XP(IFST)
WRITE(LU,102) IX(IMID), IX(ILST), IY(JMID), IY(JMID),
& XCENB, YCENB, XCENB+RAD1, YCENB, XP(IMID)

```

C
C-pd---Outer circle-----
C

```

ANG01= 0.0
ANG02= 0.0+TETT02
ANG03= 45.0
ANG04= 90.0-TETT04
ANG05= 90.0
ANG06= 90.0+TETT06
ANG07=135.0
ANG08=180.0-TETT08
ANG09=180.0
ANG10=180.0+TETT10
ANG11=225.0
ANG12=270.0-TETT12
ANG13=270.0
ANG14=270.0+TETT14
ANG15=315.0
ANG16=360.0-TETT16
IFST=IG(197)
JFST=IG(217)

```

```

IMID=IFST+2
JMID=JFST+2
ILST=IFST+4
JLST=JFST+4
IBEF=IFST-1
JBEF=JFST-1
IAFT=IFST+5
JAFT=JFST+5
WRITE(LU,*)
WRITE(LU,104) IX(IFST), IX(IFST+1), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD2, ANG11, ANG12, XP(IFST)
WRITE(LU,104) IX(IFST+1), IX(IMID), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD2, ANG12, ANG13, XP(IFST+1)
WRITE(LU,104) IX(IMID), IX(ILST-1), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD2, ANG13, ANG14, XP(IMID)
WRITE(LU,104) IX(ILST-1), IX(ILST), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD2, ANG14, ANG15, XP(ILST-1)
WRITE(LU,104) IX(IFST), IX(IFST+1), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD2, ANG07, ANG06, XP(IFST)
WRITE(LU,104) IX(IFST+1), IX(IMID), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD2, ANG06, ANG05, XP(IFST+1)
WRITE(LU,104) IX(IMID), IX(ILST-1), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD2, ANG05, ANG04, XP(IMID)
WRITE(LU,104) IX(ILST-1), IX(ILST), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD2, ANG04, ANG03, XP(ILST-1)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JFST), IY(JFST+1),
& XCENC, YCENC, RAD2, ANG11, ANG10, YP(JFST)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JFST+1), IY(JMID),
& XCENC, YCENC, RAD2, ANG10, ANG09, YP(JFST+1)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JMID), IY(JLST-1),
& XCENC, YCENC, RAD2, ANG09, ANG08, YP(JMID)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JLST-1), IY(JLST),
& XCENC, YCENC, RAD2, ANG08, ANG07, YP(JLST-1)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JFST), IY(JFST+1),
& XCENC, YCENC, RAD2, ANG15, ANG16, YP(JFST)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JFST+1), IY(JMID),
& XCENC, YCENC, RAD2, ANG16, ANG01, YP(JFST+1)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JMID), IY(JLST-1),
& XCENC, YCENC, RAD2, ANG01, ANG02, YP(JMID)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JLST-1), IY(JLST),
& XCENC, YCENC, RAD2, ANG02, ANG03, YP(JLST-1)

```

C

C-pd---Shuffle lines-----

C

```

WRITE(LU,102) IX(IMID), IX(IMID), IY(JBEF), IY(JFST),
& XL(IMID), YL(JBEF), XCENC, YCENC-RAD2, YP(JBEF)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JFST), IY(JFST+1),
& XCENC, YCENC-RAD2, XCENB, YCENB-RAD1, YP(JFST)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JLST-1), IY(JLST),
& XCENB, YCENB+RAD1, XCENC, YCENC+RAD2, YP(JLST-1)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JLST), IY(JAFT),
& XCENC, YCENC+RAD2, XL(IMID), YL(JAFT), YP(JLST)
WRITE(LU,102) IX(IBEF), IX(IFST), IY(JMID), IY(JMID),
& XL(IBEF), YL(JMID), XCENC-RAD2, YCENC, XP(IBEF)
WRITE(LU,102) IX(IFST), IX(IFST+1), IY(JMID), IY(JMID),
& XCENC-RAD2, YCENC, XCENB-RAD1, YCENB, XP(IFST)
WRITE(LU,102) IX(ILST-1), IX(ILST), IY(JMID), IY(JMID),
& XCENB+RAD1, YCENB, XCENC+RAD2, YCENC, XP(ILST-1)
WRITE(LU,102) IX(ILST), IX(IAFT), IY(JMID), IY(JMID),
& XCENC+RAD2, YCENC, XL(IAFT), YL(JMID), XP(ILST)

```

C
C-pd---More trig-----
C

```
DELL02=(RAD2*RAD2-DXI02*DXI02)**0.5  
DELL04=(RAD2*RAD2-DXI04*DXI04)**0.5  
DELL06=(RAD2*RAD2-DXI06*DXI06)**0.5  
DELL08=(RAD2*RAD2-DXI08*DXI08)**0.5  
DELL10=(RAD2*RAD2-DXI10*DXI10)**0.5  
DELL12=(RAD2*RAD2-DXI12*DXI12)**0.5  
DELL14=(RAD2*RAD2-DXI14*DXI14)**0.5  
DELL16=(RAD2*RAD2-DXI16*DXI16)**0.5
```

C

```
WRITE(LU,*)  
WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JBEF),IY(JFST),  
& XL(IFST+1),YL(JBEF),XL(IFST+1),YCENC-DELL12,YP(JBEF)  
WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JFST),IY(JFST+1),  
& XL(IFST+1),YCENC-DELL12,XL(IFST+1),YL(JFST+1),YP(JFST)  
WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JLST-1),IY(JLST),  
& XL(IFST+1),YL(JLST-1),XL(IFST+1),YCENC+DELL06,YP(JLST-1)  
WRITE(LU,102)IX(IFST+1),IX(IFST+1),IY(JLST),IY(JAFT),  
& XL(IFST+1),YCENC+DELL06,XL(IFST+1),YL(JAFT),YP(JLST)  
WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JBEF),IY(JFST),  
& XL(ILST-1),YL(JBEF),XL(ILST-1),YCENC-DELL14,YP(JBEF)  
WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JFST),IY(JFST+1),  
& XL(ILST-1),YCENC-DELL14,XL(ILST-1),YL(JFST+1),YP(JFST)  
WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JLST-1),IY(JLST),  
& XL(ILST-1),YL(JLST-1),XL(ILST-1),YCENC+DELL04,YP(JLST-1)  
WRITE(LU,102)IX(ILST-1),IX(ILST-1),IY(JLST),IY(JAFT),  
& XL(ILST-1),YCENC+DELL04,XL(ILST-1),YL(JAFT),YP(JLST)
```

C

```
WRITE(LU,102)IX(IBEf),IX(IFST),IY(JFST+1),IY(JFST+1),  
& XL(IBEf),YL(JFST+1),XCENC-DELL10,YL(JFST+1),XP(IBEf)  
WRITE(LU,102)IX(IFST),IX(IFST+1),IY(JFST+1),IY(JFST+1),  
& XCENC-DELL10,YL(JFST+1),XL(IFST+1),YL(JFST+1),XP(IFST)  
WRITE(LU,102)IX(ILST-1),IX(ILST),IY(JFST+1),IY(JFST+1),  
& XL(ILST-1),YL(JFST+1),XCENC+DELL16,YL(JFST+1),XP(ILST-1)  
WRITE(LU,102)IX(ILST),IX(IAFT),IY(JFST+1),IY(JFST+1),  
& XCENC+DELL16,YL(JFST+1),XL(IAFT),YL(JFST+1),XP(ILST)  
WRITE(LU,102)IX(IBEf),IX(IFST),IY(JLST-1),IY(JLST-1),  
& XL(IBEf),YL(JLST-1),XCENC-DELL08,YL(JLST-1),XP(IBEf)  
WRITE(LU,102)IX(IFST),IX(IFST+1),IY(JLST-1),IY(JLST-1),  
& XCENC-DELL08,YL(JLST-1),XL(IFST+1),YL(JLST-1),XP(IFST)  
WRITE(LU,102)IX(ILST-1),IX(ILST),IY(JLST-1),IY(JLST-1),  
& XL(ILST-1),YL(JLST-1),XCENC+DELL02,YL(JLST-1),XP(ILST-1)  
WRITE(LU,102)IX(ILST),IX(IAFT),IY(JLST-1),IY(JLST-1),  
& XCENC+DELL02,YL(JLST-1),XL(IAFT),YL(JLST-1),XP(ILST)
```

C
CALL WRTFI(LU,IRX,IRY,IX,IY)

C
C-pd---Fix points around circle and certain ones inside-----
C

```
WRITE(LU,*)  
WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(JFST)  
WRITE(LU,105)IX(1),IX(IFST),IY(JFST),IY(JLST)  
WRITE(LU,105)IX(ILST),IX(IRX+1),IY(JFST),IY(JLST)  
WRITE(LU,105)IX(1),IX(IRX+1),IY(JLST),IY(IRY+1)
```

C

```
ISOL=2  
WRITE(LU,105)IX(IFST+1)+ISOL,IX(ILST-1)-ISOL,IY(JFST+1),IY(JLST-1)  
WRITE(LU,105)IX(IFST+1),IX(ILST-1),IY(JFST+1)+ISOL,IY(JLST-1)-ISOL
```

```

C      WRITE(LU,105)IX(IFST+1),IX(ILST-1),IY(JFST),IY(JFST+1)
      WRITE(LU,105)IX(IFST),IX(IFST+1),IY(JFST+1),IY(JLST-1)
      WRITE(LU,105)IX(ILST-1),IX(ILST),IY(JFST+1),IY(JLST-1)
      WRITE(LU,105)IX(IFST+1),IX(ILST-1),IY(JLST-1),IY(JLST)
C
C      561 CONTINUE
C
C
C*****
C-pd---This is the second option for the lip of the augments tube.---
C---   This section is used if enough gap between the exit of the-----
C---   engine exists.-----
C
      IF(LG(2)) GOTO 580
C
      IX(1)=1
      CALL SETIV(IX,IG,220,1,NI)
C
      XCENC=RG(45)
      YCENC=RG(46)
      RAD1=RG(53)/2.
      DXI=(RAD1*RAD1/2.)**0.5
      IFST=IG(237)
      JFST=IG(257)
      IMID=IFST+1
      JMID=JFST+1
      ILST=IFST+2
      JLST=JFST+2
      IBEF=IFST-1
      JBEF=JFST-1
      IAFI=IFST+3
      JAFI=JFST+3
C
      XL(1)=0.0
      CALL SETRV(XL,RG,340,1,NI)
      XL(IFST)=XCENC-DXI
      XL(IFST+2)=XCENC+DXI
C
      CALL SETRV(XP,RG,360,2,NI)
C
      IY(1)=1
      CALL SETIV(IY,IG,240,1,NI)
C
      YL(1)=0.0
      CALL SETRV(YL,RG,380,1,NI)
      YL(JFST)=YCENC-DXI
      YL(JFST+2)=YCENC+DXI
C
      CALL SETRV(YP,RG,400,2,NI)
C
      LU=66
      CG(LU)='CS66'
      OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
      IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
      IRX=IG(48)
      IRY=IG(49)
      CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C
C-pd---Overwrite line info with arc data-----

```

C

```

ANG1= 0.0
ANG2= 45.0
ANG3= 90.0
ANG4=135.0
ANG5=180.0
ANG6=225.0
ANG7=270.0
ANG8=315.0
WRITE(LU,*)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD1, ANG6, ANG7, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD1, ANG7, ANG8, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD1, ANG4, ANG3, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD1, ANG3, ANG2, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JFST), IY(JMID),
& XCENC, YCENC, RAD1, ANG6, ANG5, YP(JFST)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JMID), IY(JLST),
& XCENC, YCENC, RAD1, ANG5, ANG4, YP(JMID)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JFST), IY(JMID),
& XCENC, YCENC, RAD1, ANG8, ANG1, YP(JFST)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JMID), IY(JLST),
& XCENC, YCENC, RAD1, ANG1, ANG2, YP(JMID)

```

C

C-pd---Shuffle lines-----

C

```

WRITE(LU,102) IX(IMID), IX(IMID), IY(JBEF), IY(JFST),
& XL(IMID), YL(JBEF), XL(IMID), YCENC-RAD1, YP(JBEF)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JFST), IY(JMID),
& XL(IMID), YCENC-RAD1, XL(IMID), YCENC, YP(JFST)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JMID), IY(JLST),
& XL(IMID), YCENC, XL(IMID), YCENC+RAD1, YP(JMID)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JLST), IY(JAFT),
& XL(IMID), YCENC+RAD1, XL(IMID), YL(JAFT), YP(JLST)
WRITE(LU,102) IX(IBEf), IX(IFST), IY(JMID), IY(JMID),
& XL(IBEf), YL(JMID), XCENC-RAD1, YL(JMID), XP(IBEf)
WRITE(LU,102) IX(IFST), IX(IMID), IY(JMID), IY(JMID),
& XCENC-RAD1, YL(JMID), XCENC, YL(JMID), XP(IFST)
WRITE(LU,102) IX(IMID), IX(ILST), IY(JMID), IY(JMID),
& XCENC, YL(JMID), XCENC+RAD1, YL(JMID), XP(IMID)
WRITE(LU,102) IX(ILST), IX(IAFT), IY(JMID), IY(JMID),
& XCENC+RAD1, YL(JMID), XL(IAFT), YL(JMID), XP(ILST)

```

C

CALL WRTFI(LU,IRX,IRY,IX,IY)

C

C-pd---Fix points around circle and certain ones inside-----

C

```

WRITE(LU,*)
WRITE(LU,105) IX(1), IX(IRX+1), IY(1), IY(JFST)
WRITE(LU,105) IX(1), IX(IFST), IY(JFST), IY(JLST)
WRITE(LU,105) IX(ILST), IX(IRX+1), IY(JFST), IY(JLST)
WRITE(LU,105) IX(1), IX(IRX+1), IY(JLST), IY(IRY+1)

```

C

ISOL=4

```

WRITE(LU,105) IX(IFST)+ISOL, IX(ILST)-ISOL, IY(JFST)+ISOL, IY(JLST)-ISOL
WRITE(LU,105) IX(IFST), IX(ILST), IY(JFST)+ISOL, IY(JLST)-ISOL

```

C

```

C
C*****
C-pd---This section is for the constant cross sectional area of the---
C---  augments sleeve.-----
C
  580 CONTINUE
C
  IX(1 )=1
  CALL SETIV(IX,IG,220,1,NI)
C
  XCENC=RG(45)
  YCENC=RG(46)
  RAD1=RG(60)/2.
  DXI=(RAD1*RAD1/2.)**0.5
  IFST=IG(237)
  JFST=IG(257)
  IMID=IFST+1
  JMID=JFST+1
  ILST=IFST+2
  JLST=JFST+2
  IBEF=IFST-1
  JBEF=JFST-1
  IAFT=IFST+3
  JAFT=JFST+3
C
  XL(1 )=0.0
  CALL SETRV(XL,RG,340,1,NI)
  XL(IFST )=XCENC-DXI
  XL(IFST+2)=XCENC+DXI
C
  CALL SETRV(XP,RG,360,2,NI)
C
  IY(1 )=1
  CALL SETIV(IY,IG,240,1,NI)
C
  YL(1 )=0.0
  CALL SETRV(YL,RG,380,1,NI)
  YL(JFST )=YCENC-DXI
  YL(JFST+2)=YCENC+DXI
C
  CALL SETRV(YP,RG,400,2,NI)
C
  IF(LG(2)) THEN
    LU=65+IG(60)
  ELSE
    LU=67
  ENDIF
  CG(LU)='CS '
  I10=LU/10
  I1=LU-I10*10
  WRITE(CG(LU)(3:3),'(I1)') I10
  WRITE(CG(LU)(4:4),'(I1)') I1
  OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
  IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
  IRX=IG(48)
  IRY=IG(49)
  CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C
C-pd---Overwrite line info with arc data-----
C

```

```

ANG1= 0.0
ANG2= 45.0
ANG3= 90.0
ANG4=135.0
ANG5=180.0
ANG6=225.0
ANG7=270.0
ANG8=315.0
WRITE(LU,*)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD1, ANG6, ANG7, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD1, ANG7, ANG8, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD1, ANG4, ANG3, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD1, ANG3, ANG2, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JFST), IY(JMID),
& XCENC, YCENC, RAD1, ANG6, ANG5, YP(JFST)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JMID), IY(JLST),
& XCENC, YCENC, RAD1, ANG5, ANG4, YP(JMID)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JFST), IY(JMID),
& XCENC, YCENC, RAD1, ANG8, ANG1, YP(JFST)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JMID), IY(JLST),
& XCENC, YCENC, RAD1, ANG1, ANG2, YP(JMID)

```

C

C-pd---Shuffle lines-----

C

```

WRITE(LU,102) IX(IMID), IX(IMID), IY(JBEF), IY(JFST),
& XL(IMID), YL(JBEF), XL(IMID), YCENC-RAD1, YP(JBEF)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JFST), IY(JMID),
& XL(IMID), YCENC-RAD1, XL(IMID), YCENC, YP(JFST)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JMID), IY(JLST),
& XL(IMID), YCENC, XL(IMID), YCENC+RAD1, YP(JMID)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JLST), IY(JAFT),
& XL(IMID), YCENC+RAD1, XL(IMID), YL(JAFT), YP(JLST)
WRITE(LU,102) IX(IBEf), IX(IFST), IY(JMID), IY(JMID),
& XL(IBEf), YL(JMID), XCENC-RAD1, YL(JMID), XP(IBEf)
WRITE(LU,102) IX(IFST), IX(IMID), IY(JMID), IY(JMID),
& XCENC-RAD1, YL(JMID), XCENC, YL(JMID), XP(IFST)
WRITE(LU,102) IX(IMID), IX(ILST), IY(JMID), IY(JMID),
& XCENC, YL(JMID), XCENC+RAD1, YL(JMID), XP(IMID)
WRITE(LU,102) IX(ILST), IX(IAFT), IY(JMID), IY(JMID),
& XCENC+RAD1, YL(JMID), XL(IAFT), YL(JMID), XP(ILST)

```

C

CALL WRTFI(LU,IRX,IRY,IX,IY)

C

C-pd---Fix points around circle and certain ones inside-----

C

```

WRITE(LU,*)
WRITE(LU,105) IX(1), IX(IRX+1), IY(1), IY(JFST)
WRITE(LU,105) IX(1), IX(IFST), IY(JFST), IY(JLST)
WRITE(LU,105) IX(ILST), IX(IRX+1), IY(JFST), IY(JLST)
WRITE(LU,105) IX(1), IX(IRX+1), IY(JLST), IY(IRY+1)

```

C

ISOL=4

```

WRITE(LU,105) IX(IFST)+ISOL, IX(ILST)-ISOL, IY(JFST), IY(JLST)
WRITE(LU,105) IX(IFST), IX(ILST), IY(JFST)+ISOL, IY(JLST)-ISOL

```

C

C

```

C*****
C-pd---This section is for the constant cross sectional area of the---
C--- augmeter tube.-----
C
      IX(1 )=1
      CALL SETIV(IX,IG,220,1,NI)
C
      XCENC=RG(45)
      YCENC=RG(46)
      RAD1=RG(61)/2.
      DXI=(RAD1*RAD1/2.)*0.5
      IFST=IG(237)
      JFST=IG(257)
      IMID=IFST+1
      JMID=JFST+1
      ILST=IFST+2
      JLST=JFST+2
      IBEF=IFST-1
      JBEF=JFST-1
      IAFT=IFST+3
      JAFT=JFST+3
C
      XL(1 )=0.0
      CALL SETRV(XL,RG,340,1,NI)
      XL(IFST )=XCENC-DXI
      XL(IFST+2)=XCENC+DXI
C
      CALL SETRV(XP,RG,360,2,NI)
C
      IY(1 )=1
      CALL SETIV(IY,IG,240,1,NI)
C
      YL(1 )=0.0
      CALL SETRV(YL,RG,380,1,NI)
      YL(JFST )=YCENC-DXI
      YL(JFST+2)=YCENC+DXI
C
      CALL SETRV(YP,RG,400,2,NI)
C
      IF(LG(2)) THEN
        LU=66+IG(60)
      ELSE
        LU=68
      ENDIF
      CG(LU)='CS '
      I10=LU/10
      I1=LU-I10*10
      WRITE(CG(LU)(3:3),'(I1)') I10
      WRITE(CG(LU)(4:4),'(I1)') I1
      OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
      IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
      IRX=IG(48)
      IRY=IG(49)
      CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C
C-pd---Overwrite line info with arc data-----
C
      ANG1= 0.0
      ANG2= 45.0
      ANG3= 90.0

```

```

ANG4=135.0
ANG5=180.0
ANG6=225.0
ANG7=270.0
ANG8=315.0
WRITE(LU,*)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD1, ANG6, ANG7, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JFST), IY(JFST),
& XCENC, YCENC, RAD1, ANG7, ANG8, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD1, ANG4, ANG3, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JLST), IY(JLST),
& XCENC, YCENC, RAD1, ANG3, ANG2, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JFST), IY(JMID),
& XCENC, YCENC, RAD1, ANG6, ANG5, YP(JFST)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JMID), IY(JLST),
& XCENC, YCENC, RAD1, ANG5, ANG4, YP(JMID)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JFST), IY(JMID),
& XCENC, YCENC, RAD1, ANG8, ANG1, YP(JFST)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JMID), IY(JLST),
& XCENC, YCENC, RAD1, ANG1, ANG2, YP(JMID)

```

C
C-pd---Shuffle lines-----
C

```

WRITE(LU,102) IX(IMID), IX(IMID), IY(JBEF), IY(JFST),
& XL(IMID), YL(JBEF), XL(IMID), YCENC-RAD1, YP(JBEF)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JFST), IY(JMID),
& XL(IMID), YCENC-RAD1, XL(IMID), YCENC, YP(JFST)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JMID), IY(JLST),
& XL(IMID), YCENC, XL(IMID), YCENC+RAD1, YP(JMID)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JLST), IY(JAFT),
& XL(IMID), YCENC+RAD1, XL(IMID), YL(JAFT), YP(JLST)
WRITE(LU,102) IX(IBEf), IX(IFST), IY(JMID), IY(JMID),
& XL(IBEf), YL(JMID), XCENC-RAD1, YL(JMID), XP(IBEf)
WRITE(LU,102) IX(IFST), IX(IMID), IY(JMID), IY(JMID),
& XCENC-RAD1, YL(JMID), XCENC, YL(JMID), XP(IFST)
WRITE(LU,102) IX(IMID), IX(ILST), IY(JMID), IY(JMID),
& XCENC, YL(JMID), XCENC+RAD1, YL(JMID), XP(IMID)
WRITE(LU,102) IX(ILST), IX(IAFT), IY(JMID), IY(JMID),
& XCENC+RAD1, YL(JMID), XL(IAFT), YL(JMID), XP(ILST)

```

C
CALL WRTFI(LU,IRX,IRY,IX,IY)

C
C-pd---Fix points around circle and certain ones inside-----
C

```

WRITE(LU,*)
WRITE(LU,105) IX(1), IX(IRX+1), IY(1), IY(JFST)
WRITE(LU,105) IX(1), IX(IFST), IY(JFST), IY(JLST)
WRITE(LU,105) IX(ILST), IX(IRX+1), IY(JFST), IY(JLST)
WRITE(LU,105) IX(1), IX(IRX+1), IY(JLST), IY(IRY+1)

```

C
ISOL=4
WRITE(LU,105) IX(IFST)+ISOL, IX(ILST)-ISOL, IY(JFST), IY(JLST)
WRITE(LU,105) IX(IFST), IX(ILST), IY(JFST)+ISOL, IY(JLST)-ISOL

C
C
C*****
C-pd---This section is for the constant cross sectional area of the---
C--- augmeter tube. This cross section is located at the back----

C--- side of the end wall. Two options exist, one for a circle-----
C--- and one for a square.-----

```
C
IX(1 )=1
CALL SETIV(IX,IG,260,1,NI)

C
XCEND=RG(47)
YCEND=RG(48)
RAD1=RG(62)/2.
DXI=(RAD1*RAD1/2.)**0.5
IF(LG(3)) DXI=RAD1
IFST=IG(277)
JFST=IG(297)
IMID=IFST+1
JMID=JFST+1
ILST=IFST+2
JLST=JFST+2
IBEF=IFST-1
JBEF=JFST-1
IAFT=IFST+3
JAFT=JFST+3

C
XL(1 )=0.0
CALL SETRV(XL,RG,420,1,NI)
XL(IFST )=XCEND-DXI
XL(IFST+2)=XCEND+DXI

C
CALL SETRV(XP,RG,440,2,NI)

C
IY(1 )=1
CALL SETIV(IY,IG,280,1,NI)

C
YL(1 )=0.0
CALL SETRV(YL,RG,460,1,NI)
YL(JFST )=YCEND-DXI
YL(JFST+2)=YCEND+DXI

C
CALL SETRV(YP,RG,480,2,NI)

C
IF(LG(2)) THEN
  LU=67+IG(60)
ELSE
  LU=69
ENDIF
CG(LU)='CS '
I10=LU/10
I1=LU-I10*10
WRITE(CG(LU)(3:3),'(I1)') I10
WRITE(CG(LU)(4:4),'(I1)') I1
OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')
IF(RG(LU+10).NE.0.0) XL(1)=RG(LU+10)
IRX=IG(50)
IRY=IG(51)
CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
IF(LG(3)) GOTO 590

C
C-pd---Overwrite line info with arc data-----
C
ANG1= 0.0
ANG2= 45.0
```

```

ANG3= 90.0
ANG4=135.0
ANG5=180.0
ANG6=225.0
ANG7=270.0
ANG8=315.0
WRITE(LU,*)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JFST), IY(JFST),
& XCEND, YCEND, RAD1, ANG6, ANG7, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JFST), IY(JFST),
& XCEND, YCEND, RAD1, ANG7, ANG8, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IMID), IY(JLST), IY(JLST),
& XCEND, YCEND, RAD1, ANG4, ANG3, XP(IFST)
WRITE(LU,104) IX(IMID), IX(ILST), IY(JLST), IY(JLST),
& XCEND, YCEND, RAD1, ANG3, ANG2, XP(IMID)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JFST), IY(JMID),
& XCEND, YCEND, RAD1, ANG6, ANG5, YP(JFST)
WRITE(LU,104) IX(IFST), IX(IFST), IY(JMID), IY(JLST),
& XCEND, YCEND, RAD1, ANG5, ANG4, YP(JMID)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JFST), IY(JMID),
& XCEND, YCEND, RAD1, ANG8, ANG1, YP(JFST)
WRITE(LU,104) IX(ILST), IX(ILST), IY(JMID), IY(JLST),
& XCEND, YCEND, RAD1, ANG1, ANG2, YP(JMID)

```

C
C
C-pd---Shuffle lines-----
C

```

WRITE(LU,102) IX(IMID), IX(IMID), IY(JBEF), IY(JFST),
& XL(IMID), YL(JBEF), XL(IMID), YCEND-RAD1, YP(JBEF)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JFST), IY(JMID),
& XL(IMID), YCEND-RAD1, XL(IMID), YCEND, YP(JFST)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JMID), IY(JLST),
& XL(IMID), YCEND, XL(IMID), YCEND+RAD1, YP(JMID)
WRITE(LU,102) IX(IMID), IX(IMID), IY(JLST), IY(JAFT),
& XL(IMID), YCEND+RAD1, XL(IMID), YL(JAFT), YP(JLST)
WRITE(LU,102) IX(IBEf), IX(IFST), IY(JMID), IY(JMID),
& XL(IBEf), YL(JMID), XCEND-RAD1, YL(JMID), XP(IBEf)
WRITE(LU,102) IX(IFST), IX(IMID), IY(JMID), IY(JMID),
& XCEND-RAD1, YL(JMID), XCEND, YL(JMID), XP(IFST)
WRITE(LU,102) IX(IMID), IX(ILST), IY(JMID), IY(JMID),
& XCEND, YL(JMID), XCEND+RAD1, YL(JMID), XP(IMID)
WRITE(LU,102) IX(ILST), IX(IAFT), IY(JMID), IY(JMID),
& XCEND+RAD1, YL(JMID), XL(IAFT), YL(JMID), XP(ILST)

```

C
590 CALL WRTFI(LU,IRX,IRY,IX,IY)
WRITE(LU,105) IX(1), IX(IRX+1), IY(1), IY(IRY+1)
IF(LG(3)) GOTO 591

C
C-pd---Fix points around circle and certain ones inside-----
C

```

WRITE(LU,*)
WRITE(LU,105) IX(1), IX(IRX+1), IY(1), IY(JFST)
WRITE(LU,105) IX(1), IX(IFST), IY(JFST), IY(JLST)
WRITE(LU,105) IX(ILST), IX(IRX+1), IY(JFST), IY(JLST)
WRITE(LU,105) IX(1), IX(IRX+1), IY(JLST), IY(IRY+1)

```

C
ISOL=4
WRITE(LU,105) IX(IFST)+ISOL, IX(ILST)-ISOL, IY(JFST), IY(JLST)
WRITE(LU,105) IX(IFST), IX(ILST), IY(JFST)+ISOL, IY(JLST)-ISOL
C

591 CONTINUE

J1TMP=IY(JFST)

J2TMP=IY(JLST)

YDTMP=YL(IRY+1)

C

C

C*****

C-pd---This section is for the exit of the chimney. Unifrom spacing--

C--- in each direction is assumed.-----

IX(1)=1

IX(2)=NX+1

C

XL(1)=XL(1)

XL(2)=XL(IRX+1)

C

XP(1)=1.0

C

IY(1)=1

IY(2)=NY+1

C

YL(1)=RG(510+IG(537))

YL(2)=RG(510+IG(537)-1)

C

YP(1)=1.0

C

IF(LG(2)) THEN

LU=68+IG(60)

ELSE

LU=70

ENDIF

CG(LU)='CS'

I10=LU/10

I1=LU-I10*10

WRITE(CG(LU)(3:3),'(I1)') I10

WRITE(CG(LU)(4:4),'(I1)') I1

OPEN(LU,FILE=CG(LU),FORM='FORMATTED',STATUS='UNKNOWN')

IRX=1

IRY=1

CALL WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)

CALL WRTFI(LU,IRX,IRY,IX,IY)

WRITE(LU,105)IX(1),IX(IRX+1),IY(1),IY(IRY+1)

C

IF(IG(1).EQ.0) THEN

WRITE(6,*)' TOTAL NUMBER OF GRID INPUT FILES CREATED = ',LU-60

WRITE(6,*)' AT THIS POINT USE GGP TO CREATE GRID PLANES'

RETURN

ENDIF

C

C

C*****

C*****

C-pd---Call ggp-----

C

INACTIVE

C

C

C

C

C*****

C*****

C

```

WRITE(6,*)' CREATING READCO FILE'
C
C-pd---Stack grids (NOTE: SFAC hardwired in - SATLIT call before-----
C--- conversions set in Q1)-----
C
SFAC=0.0254
NI=25
LMX=(NX+1)*(NY+1)
CALL SETIV(NZC,IG,510,3,NI)
ZL(1)=0.0
CALL SETRV(ZL,RG,510,1,NI)
CALL SETRV(ZP,RG,540,2,NI)
CALL SETIV(IZT,IG,540,3,NI)
CALL SETIV(IZF1,IG,570,3,NI)
CALL SETIV(IZF2,IG,600,3,NI)
C
LUW1=88
OPEN(LUW1,FILE='grid',FORM='FORMATTED',STATUS='UNKNOWN')
WRITE(LUW1,366)NX+1,NY+1,NZ+1
C
DO 5005 I=1,IG(501)
IF(IZT(I).EQ.1) THEN
CALL XSTACK(CG(IZF1(I)),LMX,NZC(I),ZL(I),ZL(I+1),ZP(I),
& XAS1,YAS1,ZASL,SFAC,LUW1)
ELSEIF (IZT(I).EQ.2) THEN
CALL XBLEND(CG(IZF1(I)),CG(IZF2(I)),LMX,NZC(I),ZL(I),ZL(I+1),
& ZP(I),XAS,YAS,XAS1,YAS1,XAS2,YAS2,ZASL,SFAC,LUW1)
ELSEIF (IZT(I).EQ.3) THEN
CALL XCURVE(CG(IZF1(I)),LMX,NZC(I),ZL(I),ZL(I+1),ZP(I),YDTMP,
& J1TMP,J2TMP,YAS,ZAS,XAS1,YAS1,SFAC,LUW1)
ELSEIF (IZT(I).EQ.4) THEN
CALL XLASTS(CG(IZF2(I)),LMX,NZC(I),YDTMP,RG(537),ZP(I),
& XAS,ZAS,XAS1,ZAS1,XAS2,ZAS2,ZL,SFAC,LUW1)
ELSE
WRITE(6,*)' ERROR IN STACKING TYPE '
ENDIF
5005 CONTINUE
CLOSE(LUW1,STATUS='KEEP')
C
RETURN
C
102 FORMAT('LI',4I3,F12.6,3F11.6,F7.2)
104 FORMAT('AR',4I3,F12.6,4F11.6,F7.2)
105 FORMAT('FXY'4I3)
366 FORMAT(3I5)
C
C--- GROUP 6. Body-fitted coordinates or grid distortion
6 CONTINUE
RETURN
C
C--- GROUP 7. Variables stored, solved & named
7 CONTINUE
RETURN
C
C--- GROUP 8. Terms (in differential equations) & devices
8 CONTINUE
RETURN
C
C--- GROUP 9. Properties of the medium (or media)
9 CONTINUE

```

```

C      IF(IG(1).NE.3) RETURN
      WRITE(6,*)' CALCULATING BOUNDARY CONDITIONS'
C
C-pd---Ambient-----
      RGAS=RG(25)
      SC(1)=RG(1)/RG(21)
      SC(2)=RG(2)/RG(22)
      SC(3)=RG(3)/RG(23)
      SC(4)=RG(4)/RG(24)
      TEMP=RG(13)
      CALL ENTHAL(TEMP,HSUM,CPSUM,SC,4,0)
      RG(16)=CPSUM*RGAS*TEMP
C-pd---Dynamometer-----
      SC(1)=RG(5)/RG(21)
      SC(2)=RG(6)/RG(22)
      SC(3)=RG(7)/RG(23)
      SC(4)=RG(8)/RG(24)
      TEMP=RG(14)
      CALL ENTHAL(TEMP,HSUM,CPSUM,SC,4,0)
      RG(17)=CPSUM*RGAS*TEMP
C-pd---Engine-----
      SC(1)=RG(9)/RG(21)
      SC(2)=RG(10)/RG(22)
      SC(3)=RG(11)/RG(23)
      SC(4)=RG(12)/RG(24)
      TEMP=RG(15)
      CALL ENTHAL(TEMP,HSUM,CPSUM,SC,4,0)
      RG(18)=CPSUM*RGAS*TEMP
      RETURN
C
C--- GROUP 10. Inter-phase-transfer processes and properties
10 CONTINUE
   RETURN
C
C--- GROUP 11. Initialization of variable or porosity fields
11 CONTINUE
   RETURN
C
C--- GROUP 12. Convection and diffusion adjustments
12 CONTINUE
   RETURN
C
C--- GROUP 13. Boundary conditions and special sources
13 CONTINUE
   RETURN
C
C--- GROUP 14. Downstream pressure for PARAB=.TRUE.
14 CONTINUE
   RETURN
C
C--- GROUP 15. Termination of sweeps
15 CONTINUE
   RETURN
C
C--- GROUP 16. Termination of iterations
16 CONTINUE
   RETURN
C
C--- GROUP 17. Under-relaxation devices

```

```

17 CONTINUE
   RETURN
C
C--- GROUP 18. Limits on variables or increments to them
18 CONTINUE
   RETURN
C
C--- GROUP 19. Data communicated by satellite to GROUND
19 CONTINUE
   RETURN
C
C--- GROUP 20. Preliminary print-out
20 CONTINUE
   RETURN
C
C--- GROUP 21. Print-out of variables
21 CONTINUE
   RETURN
C
C--- GROUP 22. Spot-value print-out
22 CONTINUE
   RETURN
C
C--- GROUP 23. Field print-out and plot control
23 CONTINUE
   RETURN
C
C--- GROUP 24. Dumps for restarts
24 CONTINUE
   WRITE(6,*) '          OUT OF IT '
   RETURN
   END
C*****
C      SUBROUTINE GSCALE(GFACT)
C*****
C  GSCALE gets information needed to scale grid points.
C-----
C
C      INCLUDE 'satear'
C      INCLUDE 'satloc'
C      INCLUDE 'bfcsat'
C      COMMON F(1)
C
C      NI=NX+1
C      NJ=NY+1
C      NK=NZ+1
C      JNNN=NI*NJ*NK
C      CALL SCALEW(F(KXC+1),F(KYC+1),F(KZC+1),GFACT,JNNN)
C
C      RETURN
C      END
C
C*****
C      SUBROUTINE SCALEW(X,Y,Z,F,N)
C*****
C  GCALEW converts grid nodes to the proper units (m).
C-----
C
C      DIMENSION X(*),Y(*),Z(*)
C

```

```

DO 1 I=1,N
X(I)=X(I)*F
Y(I)=Y(I)*F
1 Z(I)=Z(I)*F

```

C

```

RETURN
END

```

C

C*****

```

SUBROUTINE ENTHAL(TEMP,HSUM,CPSUM,SC,NS,NFO)

```

C*****

```

C ENTHAL calculates H/RT from JANNAF data. The order of
C species is N O C H.

```

C

C

```

DIMENSION SC(4),ZS(7,2,4)

```

```

DATA ZS/ 0.28532899E+01, 0.16022128E-02, -0.62936893E-06,
& 0.11441022E-09, -0.78057465E-14, -0.89008093E+03,
& 0.63964897E+01, 0.37044177E+01, -0.14218753E-02,
& 0.28670392E-05, -0.12028885E-08, -0.13954677E-13,
& -0.10640795E+04, 0.22336285E+01,
& 0.36122139E+01, 0.74853166E-03, -0.19820647E-06,
& 0.33749008E-10, -0.23907374E-14, -0.11978151E+04,
& 0.36703307E+01, 0.37837135E+01, -0.30233634E-02,
& 0.99492751E-05, -0.98189101E-08, 0.33031825E-11,
& -0.10638107E+04, 0.36416345E+01,
& 0.44608041E+01, 0.30981719E-02, -0.12392571E-05,
& 0.22741325E-09, -0.15525954E-13, -0.48961442E+05,
& -0.98635982E+00, 0.24007797E+01, 0.87350957E-02,
& -0.66070878E-05, 0.20021861E-08, 0.63274039E-15,
& -0.48377527E+05, 0.96951457E+01,
& 0.27167633E+01, 0.29451374E-02, -0.80224374E-06,
& 0.10226682E-09, -0.48472145E-14, -0.29905826E 05,
& 0.66305671E+01, 0.40701275E+01, -0.11084499E-02,
& 0.41521180E-05, -0.29637404E-08, 0.80702103E-12,
& -0.30279722E+05, -0.32270046E+00 /

```

C

```

K=1
IF(TEMP.LT.1000.) K=2
TEMP2=TEMP*TEMP
HSUM=0.
CPSUM=0.
DO 100 IS=1,NS
CP1=ZS(1,K,IS)
CP2=ZS(2,K,IS)*TEMP
CP3=ZS(3,K,IS)*TEMP2
CP4=ZS(4,K,IS)*TEMP2*TEMP
CP5=ZS(5,K,IS)*TEMP2*TEMP2
CPSUM=CPSUM+SC(IS)*(CP1+CP2+CP3+CP4+CP5)
100 HSUM =HSUM+
1 SC(IS)*(CP1+.5*CP2+.33333*CP3+.25*CP4+.2*CP5+ZS(6,K,IS)/TEMP)

```

C

```

RETURN
END

```

C

C*****

```

SUBROUTINE SETIV(IA,IG,IFST,ITY,NI)

```

C*****

```

C SETIV places interger values from the IG array into the
C proper local array.

```

```

C-----
C
C      DIMENSION IA(*),IG(*)
C
C      IF(ITY.EQ.1) THEN
C        DO 1 I=1,NI
1       IA(I+1)=IG(IFST+I)+1
C      ELSEIF (ITY.EQ.2) THEN
C        DO 2 I=1,NI
2       IA(I)=IG(IFST+I)+1
C      ELSEIF (ITY.EQ.3) THEN
C        DO 3 I=1,NI
3       IA(I)=IG(IFST+I)
C      ELSE
C        WRITE(6,*)' ERROR SETIV --- INVALID TYPE '
C      ENDIF
C
C      RETURN
C      END
C
C*****
C      SUBROUTINE SETRV(RA,RG,IFST,ITY,NI)
C*****
C      SETRV places real values from the RG array into the proper
C      local array.
C-----
C
C      DIMENSION RA(*),RG(*)
C
C      IF(ITY.EQ.1) THEN
C        DO 1 I=1,NI
1       RA(I+1)=RG(IFST+I)
C      ELSEIF (ITY.EQ.2) THEN
C        DO 2 I=1,NI
2       RA(I)=RG(IFST+I)
C      ELSE
C        WRITE(6,*)' ERROR SETRV --- INVALID TYPE '
C      ENDIF
C
C      RETURN
C      END
C
C*****
C      SUBROUTINE WRTSQ(LU,NX,NY,IRX,IRY,IX,IY,XL,YL,XP,YP)
C*****
C      WRTSQ writes input grid file assuming all straight lines.
C-----
C
C      DIMENSION IX(*),IY(*),XL(*),YL(*),XP(*),YP(*)
C
C      WRITE(LU,100) NX+1
C      WRITE(LU,101) NY+1
C      DO 10 I=1,IRY+1
C        WRITE(LU,*)
C        DO 10 J=1,IRX
10     WRITE(LU,102)
C      &      IX(J),IX(J+1),IY(I),IY(I),XL(J),YL(I),XL(J+1),YL(I),XP(J)
C        DO 20 I=1,IRX+1
C        WRITE(LU,*)
C        DO 20 J=1,IRY

```

```

20 WRITE(LU,102)
&      IX(I),IX(I),IY(J),IY(J+1),XL(I),YL(J),XL(I),YL(J+1),YP(J)
C
100 FORMAT('IMAX',I3)
101 FORMAT('JMAX',I3)
102 FORMAT('LI',4I3,F12.6,3F11.6,F7.2)
C
      RETURN
      END
C
C*****
      SUBROUTINE WRTFI(LU,IRX,IRY,IX,IY)
C*****
C  WRTFI writes commands needed to fill subsections.
C-----
C
      DIMENSION IX(*),IY(*)
C
      DO 10 I=1,IRY
        WRITE(LU,*)
        DO 10 J=1,IRX
          10 WRITE(LU,103)IX(J),IX(J+1),IY(I),IY(I+1)
C
103 FORMAT('FI',4I3)
C
      RETURN
      END
C
C*****
      SUBROUTINE XSTACK(F1PRE,LMX,NZC,ZFST,ZLST,ZP,X1,Y1,ZL,CV,LUW1)
C*****
C  XSTACK repeats one computational grid file
C-----
C
      CHARACTER*4 F1PRE,FEXT
      CHARACTER*8 F1NAME
      DIMENSION X1(*),Y1(*),ZL(*)
C
      FEXT='.GRD'
      F1NAME=F1PRE//FEXT
      LUR1=80
      OPEN(LUR1,FILE=F1NAME,FORM='FORMATTED',STATUS='OLD')
C
      READ(LUR1,366)LP1,MP1,NTP1
      READ(LUR1,333)((X1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((Y1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((ZTEMP,IJ=I,LMX,LP1),I=1,LP1)
C
      CALL ZLSET(ZL,1,NZC+1,ZFST,ZLST,ZP)
C
      DO 10 K=1,NZC
        WRITE(LUW1,333)((X1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
        WRITE(LUW1,333)((Y1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
        WRITE(LUW1,333)((ZL(K)*CV,IJ=I,LMX,LP1),I=1,LP1)
      10 CONTINUE
      CLOSE(LUR1,STATUS='KEEP')
C
      RETURN
333 FORMAT(5(1P,E13.6))
366 FORMAT(3I5)

```

END

```
C
C*****
      SUBROUTINE XBLEND(F1PRE,F2PRE,LMX,NZC,ZFST,ZLST,ZP,X,Y,X1,Y1,
        &                X2,Y2,ZL,CV,LUW1)
C*****
C  XBLEND blends two computational grids files
C-----
C
      CHARACTER*4 F1PRE,F2PRE,FEXT
      CHARACTER*8 F1NAME,F2NAME
      DIMENSION X(2500),Y(2500),X1(2500),Y1(2500),X2(2500),Y2(2500),
        &          ZL(100)
C
      FEXT='.GRD'
      F1NAME=F1PRE//FEXT
      F2NAME=F2PRE//FEXT
      LUR1=80
      LUR2=81
      OPEN(LUR1,FILE=F1NAME,FORM='FORMATTED',STATUS='OLD')
      OPEN(LUR2,FILE=F2NAME,FORM='FORMATTED',STATUS='OLD')
C
      READ(LUR1,366) LP1,MP1,NTP1
      READ(LUR1,333) ((X1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333) ((Y1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333) ((ZTEMP,IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR2,366) LP1,MP1,NTP1
      READ(LUR2,333) ((X2(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR2,333) ((Y2(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR2,333) ((ZTEMP,IJ=I,LMX,LP1),I=1,LP1)
C
      CALL ZLSET(ZL,1,NZC+1,ZFST,ZLST,ZP)
C
      DO 20 K=1,NZC
      DO 21 I=1,LMX
      IF(NZC.EQ.1) THEN
        X(I)=X1(I)
        Y(I)=Y1(I)
      ELSE
        X(I)=X1(I)*FLOAT(NZC-K+1)/FLOAT(NZC)+
        &        X2(I)*FLOAT(K-1)/FLOAT(NZC)
        Y(I)=Y1(I)*FLOAT(NZC-K+1)/FLOAT(NZC)+
        &        Y2(I)*FLOAT(K-1)/FLOAT(NZC)
      ENDIF
21 CONTINUE
      WRITE(LUW1,333) ((X(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333) ((Y(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333) ((ZL(K)*CV,IJ=I,LMX,LP1),I=1,LP1)
20 CONTINUE
      CLOSE(LUR1,STATUS='KEEP')
      CLOSE(LUR2,STATUS='KEEP')
C
      RETURN
333 FORMAT(5(1P,E13.6))
366 FORMAT(3I5)
      END
C
C*****
      SUBROUTINE XCURVE(F1PRE,LMX,NZC,ZFST,ZLST,ZP,CENC,NY2,NY3,
        &                Y,Z,X1,Y1,CV,LUW1)
```

```

C*****
C  XCURVE creates the grid in the augmentor tube bend section
C-----
C
      CHARACTER*4 F1PRE,FEXT
      CHARACTER*8 F1NAME
      DIMENSION Y(*),Z(*),X1(*),Y1(*)
C
      FEXT='.GRD'
      F1NAME=F1PRE//FEXT
      LUR1=80
C
C-pd---NZC number of cells in bend  (WARNING: Must be even)-----
C---  NY1 lower Y line-----
C---  NY2 lower Y circle line-----
C---  NY3 upper Y circle line-----
C---  NY4 upper Y line-----
C
      OPEN(LUR1,FILE=F1NAME,FORM='FORMATTED',STATUS='OLD')
      READ(LUR1,366)LP1,MP1,NTP1
      READ(LUR1,333)((X1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((Y1(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((ZTEMP ,IJ=I,LMX,LP1),I=1,LP1)
C
      NY1=1
      NY4=MP1
      ZLEN=ZLST-ZFST
C
C-pd---Do Boundary-----
      WRITE(LUW1,333)((X1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Y1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((ZFST*CV, IJ=I,LMX,LP1),I=1,LP1)
C
      DO 400 IP=1,NZC
      ANG=90.0/FLOAT(NZC)*FLOAT(IP)
      PI=3.141592654
      RAD=ANG/360.*2.*PI
      YFAC=COS(RAD)
C
C-pd---Lower Y row-----
      DO 205 I=1,LP1
      IF(IP.LE.NZC/2) THEN
        Y(I)=0.0
      ELSE
        Y(I)=FLOAT(IP-(NZC/2))/FLOAT(NZC/2)*CENC
      ENDIF
      205 CONTINUE
C
C-pd---Lower Y circle row-----
      IAD=(NY2-1)*LP1
      DO 210 I=1,LP1
      Y(IAD+I)=Y1(IAD+I)+(1.0-YFAC)*(CENC-Y1(IAD+I))
      210 CONTINUE
C
C-pd---Upper Y circle row-----
      IAD=(NY3-1)*LP1
      DO 215 I=1,LP1
      Y(IAD+I)=Y1(IAD+I)+(1.0-YFAC)*(CENC-Y1(IAD+I))
      215 CONTINUE
C

```

```

C-pd---Upper Y row-----
      IAD=(MP1-1)*LP1
      DO 220 I=1,LP1
C-pd---add fact to give a north cell area-----
      Y(IAD+I)=Y1(IAD+I)+((FLOAT(IP)/FLOAT(NZC))*0.01)
      220 CONTINUE
C
C-pd---Fill first section-----
      DO 250 J=2,NY2-1
      DO 250 I=1,LP1
      LOC=(J-1)*LP1+I
      IAD1=0
      IAD2=(NY2-1)*LP1
      Y(LOC)=Y(IAD1+I)+((Y1(LOC)-Y1(IAD1+I))/
+      (Y1(IAD2+I)-Y1(IAD1+I))*(Y(IAD2+I)-Y(IAD1+I)))
      250 CONTINUE
C
C-pd---Fill circle section-----
      DO 260 J=NY2+1,NY3-1
      DO 260 I=1,LP1
      LOC=(J-1)*LP1+I
      IAD1=(NY2-1)*LP1
      IAD2=(NY3-1)*LP1
      Y(LOC)=Y(IAD1+I)+((Y1(LOC)-Y1(IAD1+I))/
+      (Y1(IAD2+I)-Y1(IAD1+I))*(Y(IAD2+I)-Y(IAD1+I)))
      260 CONTINUE
C
C-pd---Fill top section-----
      DO 270 J=NY3+1,MP1-1
      DO 270 I=1,LP1
      LOC=(J-1)*LP1+I
      IAD1=(NY3-1)*LP1
      IAD2=(MP1-1)*LP1
      Y(LOC)=Y(IAD1+I)+((Y1(LOC)-Y1(IAD1+I))/
+      (Y1(IAD2+I)-Y1(IAD1+I))*(Y(IAD2+I)-Y(IAD1+I)))
      270 CONTINUE
C
C
      ZD4=0.0
C
C-pd---Lower Z row-----
      ZFAC=SIN(RAD)
      DO 305 I=1,LP1
      IF(IP.LE.NZC/2) THEN
        Z(I)=FLOAT(IP)/FLOAT(NZC/2)*ZLEN+ZFST
      ELSE
        Z(I)=ZLEN+ZFST
      ENDIF
      305 CONTINUE
C
C-pd---Lower Z circle row-----
      IAD=(NY2-1)*LP1
      DO 310 I=1,LP1
      Z(IAD+I)=SIN(RAD)*(CENC-Y1(IAD+I))+ZFST
      310 CONTINUE
C
C-pd---Upper Z circle row-----
      IAD=(NY3-1)*LP1
      DO 315 I=1,LP1
      Z(IAD+I)=SIN(RAD)*(CENC-Y1(IAD+I))+ZFST

```

```

315 CONTINUE
C
C-pd---Upper Z row-----
      IAD=(MP1-1)*LP1
      DO 320 I=1,LP1
        Z(IAD+I)=ZD4+ZFST
320 CONTINUE
C
C-pd---Fill first section-----
      DO 350 J=2,NY2-1
        DO 350 I=1,LP1
          LOC=(J-1)*LP1+I
          IAD1=0
          IAD2=(NY2-1)*LP1
          Z(LOC)=Z(IAD1+I)-((Y1(LOC)-Y1(IAD1+I))/
+          (Y1(IAD2+I)-Y1(IAD1+I))*(Z(IAD1+I)-Z(IAD2+I)))
350 CONTINUE
C
C-pd---Fill circle section-----
      DO 360 J=NY2+1,NY3-1
        DO 360 I=1,LP1
          LOC=(J-1)*LP1+I
          IAD1=(NY2-1)*LP1
          IAD2=(NY3-1)*LP1
          Z(LOC)=Z(IAD1+I)-((Y1(LOC)-Y1(IAD1+I))/
+          (Y1(IAD2+I)-Y1(IAD1+I))*(Z(IAD1+I)-Z(IAD2+I)))
360 CONTINUE
C
C-pd---Fill top section-----
      DO 370 J=NY3+1,MP1-1
        DO 370 I=1,LP1
          LOC=(J-1)*LP1+I
          IAD1=(NY3-1)*LP1
          IAD2=(MP1-1)*LP1
          Z(LOC)=Z(IAD1+I)-((Y1(LOC)-Y1(IAD1+I))/
+          (Y1(IAD2+I)-Y1(IAD1+I))*(Z(IAD1+I)-Z(IAD2+I)))
370 CONTINUE
C
C-pd---Write data-----
      WRITE(LUW1,333)((X1(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Y(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Z(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
400 CONTINUE
C
      CLOSE(LUR1,STATUS='KEEP')
C
      RETURN
C
333 FORMAT(5(1F,E13.6))
366 FORMAT(3I5)
C
      END
C
C*****
      SUBROUTINE XLASTS(F1PRE,LMX,NZC,YFST,YLST,ZP,
&          X,Z1,X1,Z,X2,Z2,YL,CV,LUW1)
C*****
C  XLASTS creates the grid in the last section
C-----
C

```

```

      CHARACTER*4 F1PRE,F2PRE,FEXT
      CHARACTER*8 F1NAME
      DIMENSION X(*),Z(*),X1(*),Z1(*),X2(*),Z2(*),YL(*)

C
      FEXT='.GRD'
      F1NAME=F1PRE//FEXT
      LUR1=80

C
C-pd---Do last section (blend)-----
      OPEN(LUR1,FILE=F1NAME,FORM='FORMATTED',STATUS='OLD')
      READ(LUR1,366) LP1,MP1,NTP1
      READ(LUR1,333)((X2(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((Z2(IJ),IJ=I,LMX,LP1),I=1,LP1)
      READ(LUR1,333)((ZTEMP ,IJ=I,LMX,LP1),I=1,LP1)
      K1=1
      KL=NZC+1
      CALL ZLSET(YL,K1,KL,YFST,YLST,ZP)
      DO 440 K=2,NZC+1
      DO 441 I=1,LMX
      IF(NZC.EQ.1) THEN
        X(I)=X1(I)
        Z(I)=Z1(I)
      ELSE
        X(I)=X1(I)*FLOAT(NZC-K+1)/FLOAT(NZC)+
&        X2(I)*FLOAT(K-1)/FLOAT(NZC)
        Z(I)=Z1(I)*FLOAT(NZC-K+1)/FLOAT(NZC)+
&        Z2(I)*FLOAT(K-1)/FLOAT(NZC)
      ENDIF
441 CONTINUE
      WRITE(LUW1,333)((X(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((YL(K)*CV,IJ=I,LMX,LP1),I=1,LP1)
      WRITE(LUW1,333)((Z(IJ)*CV,IJ=I,LMX,LP1),I=1,LP1)
440 CONTINUE
      CLOSE(LUR1,STATUS='KEEP')

C
      RETURN
333 FORMAT(5(1P,E13.6))
366 FORMAT(3I5)
      END

C
C*****
      SUBROUTINE ZLSET(ZBND,INDEX1,INDEXL,Z1,ZL,PWR)
C*****
C
C (C) COPYRIGHT 1991 DOC D of North America, Inc.  ALL RIGHTS RESERVED
C
C Read input parameters to distribute a number of points along a
C line segment.
C Syntax is : LINE K1 KL Z1 ZL [PWR]
C-----
      DIMENSION ZBND(*)

C
      IF(PWR.GT.0) THEN
        K1=INDEX1
        KL=INDEXL
        INC=1
        DELZ = ZL-Z1
        ZF = Z1
      ELSE
        K1=INDEXL

```

```
KL=INDEX1
INC=-1
DELZ = Z1-ZL
ZF = ZL
PWR=ABS(PWR)
ENDIF
DO 10 I = K1,KL,INC
RAT = (FLOAT(I-K1)/FLOAT(KL-K1))**PWR
ZBND(I) = ZF + DELZ*RAT
10 CONTINUE
```

C

```
RETURN
END
```

APPENDIX D

```

C FILE NAME GROUND.FTN-----22 April 87
C THIS IS THE MAIN PROGRAM OF EARTH
C
C (C) COPYRIGHT 1984, LAST REVISION 1987.
C CONCENTRATION HEAT AND MOMENTUM LTD. ALL RIGHTS RESERVED.
C This subroutine and the remainder of the PHOENICS code are
C proprietary software owned by Concentration Heat and Momentum
C Limited, 40 High Street, Wimbledon, London SW19 5AU, England.
C
C
C PROGRAM MAIN
C
C 1 The following two COMMON's, which appear identically in the
C satellite MAIN program, allow up to 80 dependent variables to
C be solved for (or their storage spaces to be occupied by
C other variables, such as density). If a larger number is
C required increase the parameter nvd. Less than 50 for nvd is not
C permitted.
C
C If more patches are required increase npatd.
C
C If a larger F-array is needed increase nfd.
C
C PARAMETER (NVD=80,NFD=18000000,NPATD=1000)
C
C COMMON/LGE4/L4(NVD)
C 1/LDB1/L5(NVD)/IDA1/I1(NVD)/IDA2/I2(NVD)/IDA3/I3(NVD)/IDA4/I4(NVD)
C 1/IDA5/I5(NVD)/IDA6/I6(NVD)/GI1/I7(NVD)/GI2/I8(NVD)/HDA1/IH1(NVD)
C 1/GH1/IH2(NVD)/RDA1/R1(NVD)/RDA2/R2(NVD)/RDA3/R3(NVD)/RDA4/R4(NVD)
C 1/RDA5/R5(NVD)/RDA6/R6(NVD)/RDA7/R7(NVD)/RDA8/R8(NVD)/RDA9/R9(NVD)
C 1/RDA10/R10(NVD)/RDA11/R11(NVD)
C 1/GR1/R12(NVD)/GR2/R13(NVD)/GR3/R14(NVD)/GR4/R15(NVD)
C 1/IPIP1/IP1(NVD)/HPIP2/IHP2(NVD)/RPIP1/RVAL(NVD)/LPIP1/LVAL(NVD)
C 1/IFPL/IPLO(NVD)/RFPL1/ORPRIN(NVD)/RFPL2/ORMAX(NVD)
C 1/RFPL3/ORMIN(NVD)
C LOGICAL L1,L2,L3,L4,L5,DBGFIL,LVAL
C CHARACTER*4 IH1,IH2,IHP2,NSDA
C
C COMMON/F01/I9(4*NVD)
C COMMON/DISC/DBGFIL
C COMMON/LUNITS/LUNIT(60)
C EXTERNAL WAYOUT
C
C 2 Set dimensions of data-for-GROUND arrays here. WARNING: the
C corresponding arrays in the MAIN program of the satellite
C (see SATLIT) must have the same dimensions.
C COMMON/LGRND/LG(1000)/IGRND/IG(1000)/RGRND/RG(10000)
C COMMON/CGRND/CG(1000)
C LOGICAL LG
C CHARACTER*4 CG
C
C 3 Set dimensions of data-for-GREX2 arrays here. WARNING: the
C corresponding arrays in the MAIN program of the satellite
C (see SATLIT) must have the same dimensions.
C COMMON/LSG/LSGD(20)/ISG/ISGD(20)/RSG/RSGD(100)/CSG/CSGD(10)
C LOGICAL LSGD
C CHARACTER*4 CSGD
C
C 4 Set dimension of patch-name array here. WARNING: the array
C NAMPAT in the MAIN program of the satellite must have the

```

```

C      dimension.
C      COMMON/NPAT/NAMPAT(NPATD)
C      CHARACTER*8 NAMPAT
C
C      CONFIG FILE name declaration.
C      COMMON/CNFG/CNFIG
C      CHARACTER CNFIG*48
C
C 5    The numbers in the next two statements (which must be ident-
C      ical) indicate how much computer memory is to be set aside
C      for storing the main and auxiliary variables. The user may
C      alter them if he wishes, to accord with the number of
C      grid nodes and dependent variables he is concerned with.
C      COMMON F(NFD)
C      NFDIM=NFD
C
C 6    Logical-unit numbers and file names, not to be changed.
C      CALL CNFGZZ(2)
C      CALL EARSET(1)
C      CALL OPENFL(6)
C
C      User may here change message transmitted to logical unit
C      LUPR3
C      CALL WRIT40('Ground-Station is ground.f, 09/25/87.  ')
C      CALL MAIN1(NFDIM)
C      CALL WAYOUT(0)
C      STOP
C      END
C*****
C      SUBROUTINE GROSTA
C
C      #include "satear"
C      #include "grdloc"
C      #include "grdear"
C.... This subroutine directs control to the GROUNDS selected by
C      the satellite settings of USEGRX, NAMGRD & USEGRD.
C
C      Subroutine GREX2 contains options for fluid properties,
C      turbulence models, wall functions, chemical reaction etc. It
C      was introduced in version 1.4 of PHOENICS.
C
C      IF(USEGRX) CALL GREX2
C
C.... BTSTGR contains the sequences used in conjunction with
C      the BFC test battery.
C
C      IF(NAMGRD.EQ.'BTST') CALL BTSTGR
C
C.... TESTGR contains test battery sequences used in conjunction
C      with the test-battery SATLIT subroutine, TESTST.
C
C      IF(NAMGRD.EQ.'TEST') CALL TESTGR
C
C.... SPECGR is a generic "special" GROUND the name of which can
C      be used by anyone for their own purposes. SPC1GR, SPC2GR and
C      SPC3GR permit the user to attach his own library of special
C      GROUNDS selected according to the prescription of NAMGRD.
C
C      IF(NAMGRD.EQ.'SPEC') CALL SPECGR
C

```



```

C      COMMONS /RSG/, /ISG/, /LSG/ and /CSG/ (which are now automatically
C      included in grdloc) can be used but the user must check GREX2 for any
C      conflicting uses. The same comment applies to the EARTH-spare working
C      arrays EASP1, EASP2,...EASP10. If the call to GREX2 has been
C      deactivated then they can all be used without reservation.
C

```

```

      IXL=IABS(IXL)
      IF(IGR.EQ.13) GO TO 13
      IF(IGR.EQ.19) GO TO 19
      GO TO (1,2,3,4,5,6,24,8,9,10,11,12,13,14,24,24,24,24,19,20,24,
124,23,24),IGR

```

```

C*****

```

```

C
C--- GROUP 1. Run title and other preliminaries
C

```

```

      1 GO TO (1001,1002),ISC
1001 CONTINUE

```

```

C
      NSC=4
      NFO=0
      TNY=1.E-15
      RGAS=RG(25)
      JSWPRN=TSTSWP
      PTRAP=RG(29)

```

```

C
      RETURN
1002 CONTINUE
      RETURN

```

```

C*****

```

```

C
C--- GROUP 2. Transience; time-step specification
C

```

```

      2 CONTINUE
      RETURN

```

```

C*****

```

```

C
C--- GROUP 3. X-direction grid specification
C

```

```

      3 CONTINUE
      RETURN

```

```

C*****

```

```

C
C--- GROUP 4. Y-direction grid specification
C

```

```

      4 CONTINUE
      RETURN

```

```

C*****

```

```

C
C--- GROUP 5. Z-direction grid specification
C

```

```

      5 CONTINUE
      RETURN

```

```

C*****

```

```

C
C--- GROUP 6. Body-fitted coordinates or grid distortion
C

```

```

      6 CONTINUE
      RETURN

```

```

C*****

```

```

C      * Make changes for this group only in group 19.

```

```

C--- GROUP 7. Variables stored, solved & named
C*****
C
C--- GROUP 8. Terms (in differential equations) & devices
C
      8 GO TO (81,82,83,84,85,86,87,88,89,810,811,812,813,814,815)
      1,ISC
      81 CONTINUE
C   * ----- SECTION 1 -----
C   For U1AD.LE.GRND--- phase 1 additional velocity (VELAD).
      RETURN
      82 CONTINUE
C   * ----- SECTION 2 -----
C   For U2AD.LE.GRND--- phase 2 additional velocity (VELAD).
      RETURN
      83 CONTINUE
C   * ----- SECTION 3 -----
C   For V1AD.LE.GRND--- phase 1 additional velocity (VELAD).
      RETURN
      84 CONTINUE
C   * ----- SECTION 4 -----
C   For V2AD.LE.GRND--- phase 2 additional velocity (VELAD).
      RETURN
      85 CONTINUE
C   * ----- SECTION 5 -----
C   For W1AD.LE.GRND--- phase 1 additional velocity (VELAD).
      RETURN
      86 CONTINUE
C   * ----- SECTION 6 -----
C   For W2AD.LE.GRND--- phase 2 additional velocity (VELAD).
      RETURN
      87 CONTINUE
C   * ----- SECTION 7 ---- VOLUMETRIC SOURCE FOR GALA
      RETURN
      88 CONTINUE
C   * ----- SECTION 8 --- CONVECTION FLUXES
      RETURN
      89 CONTINUE
C   * ----- SECTION 9 --- DIFFUSION COEFFICIENTS
      RETURN
      810 CONTINUE
C   * ----- SECTION 10 --- CONVECTION NEIGHBOURS
      RETURN
      811 CONTINUE
C   * ----- SECTION 11 --- DIFFUSION NEIGHBOURS
      RETURN
      812 CONTINUE
C   * ----- SECTION 12 --- LINEARISED SOURCES
      RETURN
      813 CONTINUE
C   * ----- SECTION 13 --- CORRECTION COEFFICIENTS
      RETURN
      814 CONTINUE
C   * ----- SECTION 14 --- USER'S SOLVER
      RETURN
      815 CONTINUE
C   * ----- SECTION 15 --- CHANGE SOLUTION
      RETURN
C   * Make all other group-8 changes in group 19.
C*****

```

```

C
C--- GROUP 9. Properties of the medium (or media)
C
C   The sections in this group are arranged sequentially in their
C   order of calling from EARTH. Thus, as can be seen from below,
C   the temperature sections (10 and 11) precede the density
C   sections (1 and 3); so, density formulae can refer to
C   temperature stores already set.
C   9 GO TO (91,92,93,94,95,96,97,98,99,900,901,902,903),ISC
C*****
900 CONTINUE
C * ----- SECTION 10 -----
C   For TMP1.LE.GRND----- phase-1 temperature Index AUX(TEMP1)
C   RETURN
901 CONTINUE
C * ----- SECTION 11 -----
C   For TMP2.LE.GRND----- phase-2 temperature Index AUX(TEMP2)
C   RETURN
902 CONTINUE
C * ----- SECTION 12 -----
C   For EL1.LE.GRND----- phase-1 length scale Index AUX(LEN1)
C   RETURN
903 CONTINUE
C * ----- SECTION 13 -----
C   For EL2.LE.GRND----- phase-2 length scale Index AUX(LEN2)
C   RETURN
91 CONTINUE
C * ----- SECTION 1 -----
C   For RHO1.LE.GRND--- density for phase 1 Index AUX(DEN1).
C
C   CALL GETYX (P1,GP1,JNY,JNX)
C   CALL GETYX (H1,GH1,JNY,JNX)
C   CALL GETYX (C1,GC1,JNY,JNX)
C   CALL GETYX (TEMP,GTMP,JNY,JNX)
C   CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
C   DO 9101 IX=1,NX
C   DO 9101 IY=1,NY
C   IF (GVPR(IY,IX).LE.1.E-4) THEN
C       GC3(IY,IX)=0.0
C       GTMP(IY,IX)=300.
C       PHI(IY,IX)=0.0
C       GRH(IY,IX)=1.
C       GOTO 9101
C   ENDIF
C-pd---Calculate mass fractions-----
C   GC3(IY,IX)=1.0-GC1(IY,IX)
C   SC(1)=(GC3(IY,IX)*RG(1)+GC1(IY,IX)*RG(9))/RG(21)
C   SC(2)=(GC3(IY,IX)*RG(2)+GC1(IY,IX)*RG(10))/RG(22)
C   SC(3)=(GC3(IY,IX)*RG(3)+GC1(IY,IX)*RG(11))/RG(23)
C   SC(4)=(GC3(IY,IX)*RG(4)+GC1(IY,IX)*RG(12))/RG(24)
C   SC(1)=AMAX1(1.E-10,SC(1))
C   SC(2)=AMAX1(1.E-10,SC(2))
C   SC(3)=AMAX1(1.E-10,SC(3))
C   SC(4)=AMAX1(1.E-10,SC(4))
C   TGUS=GTMP(IY,IX)
C   HSTAT=GH1(IY,IX)
C   CALL TEMPER(HSTAT,TGUS,TCELL,CPDR,RGAS,SC,NSC,NFO)
C
C   TCELL=AMAX1(273.,TCELL)
C   TCELL=AMIN1(950.,TCELL)

```

```

C      GP=PRESSO+GP1(IY,IX)
      PHI(IY,IX)=1.0/(GP+TNY)
      XMWA=1.0/(SC(1)+SC(2)+SC(3)+SC(4))
      GRH(IY,IX)=GP*XMWA/(RGAS*TCELL+TNY)
      GTMP(IY,IX)=TCELL
      GCP(IY,IX)=CPDR*RGAS
9101  CONTINUE
C
      CALL SETYX(AUX(DEN1),GRH,JNY,JNX)
      CALL SETYX(C3,GC3,JNY,JNX)
      CALL SETYX(TEMP,GTMP,JNY,JNX)
      CALL SETYX(CP,GCP,JNY,JNX)
C
      RETURN
92  CONTINUE
C      * ----- SECTION 2 -----
C      For DRH1DP.LE.GRND--- D(LN(DEN))/DP for phase 1 (D1DP).
      CALL SETYX(D1DP,PHI,JNY,JNX)
      RETURN
93  CONTINUE
C      * ----- SECTION 3 -----
C      For RHO2.LE.GRND--- density for phase 2 Index AUX(DEN2).
      RETURN
94  CONTINUE
C      * ----- SECTION 4 -----
C      For DRH2DP.LE.GRND--- D(LN(DEN))/DP for phase 2 (D2DP).
      RETURN
95  CONTINUE
C      * ----- SECTION 5 -----
C      For ENUT.LE.GRND--- reference turbulent kinematic viscosity.
      RETURN
96  CONTINUE
C      * ----- SECTION 6 -----
C      For ENUL.LE.GRND--- reference laminar kinematic viscosity.
      RETURN
97  CONTINUE
C      * ----- SECTION 7 -----
C      For PRNDTL( ).LE.GRND--- laminar PRANDTL nos., or diffusivity.
      RETURN
98  CONTINUE
C      * ----- SECTION 8 -----
C      For PHINT( ).LE.GRND--- interface value of first phase(FII1).
      RETURN
99  CONTINUE
C      * ----- SECTION 9 -----
C      For PHINT( ).LE.GRND--- interface value of second phase(FII2)
      RETURN
C*****
C
C--- GROUP 10. Inter-phase-transfer processes and properties
C
      10 GO TO (101,102,103,104),ISC
101  CONTINUE
C      * ----- SECTION 1 -----
C      For CFIPS.LE.GRND--- inter-phase friction coeff. AUX(INTFRC).
      RETURN
102  CONTINUE
C      * ----- SECTION 2 -----
C      For CMDOT.EQ.GRND- inter-phase mass transfer Index AUX(INTMDT)

```

```

      RETURN
103  CONTINUE
C   * ----- SECTION 3 -----
C   For CINT( ).EQ.GRND--- phasel-to-interface transfer
C                               coefficients (COI1)
      RETURN
104  CONTINUE
C   * ----- SECTION 4 -----
C   For CINT( ).EQ.GRND--- phase2-to-interface transfer
C                               coefficients (COI2)
      RETURN
C*****
C
C--- GROUP 11. Initialization of variable or porosity fields
C
      11  CONTINUE
      RETURN
C*****
C
C--- GROUP 12. Convection and diffusion adjustments
C
      12  CONTINUE
      RETURN
C*****
C
C--- GROUP 13. Boundary conditions and special sources
C
      13  CONTINUE
      GO TO (130,131,132,133,134,135,136,137,138,139,1310,
      11311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321),ISC
130  CONTINUE
C----- SECTION 1 ----- coefficient = GRND
      RETURN
131  CONTINUE
C----- SECTION 2 ----- coefficient = GRND1
      RETURN
132  CONTINUE
C----- SECTION 3 ----- coefficient = GRND2
      RETURN
133  CONTINUE
C----- SECTION 4 ----- coefficient = GRND3
      RETURN
134  CONTINUE
C----- SECTION 5 ----- coefficient = GRND4
      RETURN
135  CONTINUE
C----- SECTION 6 ----- coefficient = GRND5
      RETURN
136  CONTINUE
C----- SECTION 7 ----- coefficient = GRND6
      RETURN
137  CONTINUE
C----- SECTION 8 ----- coefficient = GRND7
      IF(INDVAR.GT.P1) GO TO 13799
      CALL GETYX(AUX(DEN1),A1,NY,NX)
      CALL GETYX(P1,A2,NY,NX)
      CALL GETCOV(NPATCH,INAME('UCRT'),COEFF,GKLOSS)
      CALL GETCOV(NPATCH, P1 ,COEFF,GPBV )
      I=(IXF-2)*NY
      DO 13701 II=IXF,IXL

```

```

      I=I + NY
      DO 13702 J=IYF,IYL
        IJ=I + J
        DELTAP=AMAX1(ABS(A2(IJ)-GPBV),PTRAP)
        RHO =A1(IJ)
        COEFF =SQRT(2.*RHO/(GKLOSS*DELTAP))
        A1(IJ)=COEFF
13702   CONTINUE
13701   CONTINUE
      CALL SETYX(CO,A1,NY,NX)
      RETURN
13799   CALL WRIT40('CO = GRND7 FOR VARIABLE BESIDES P1 !!!! ')
      CALL WAYOUT(1)
      RETURN
138   CONTINUE
C----- SECTION 9 ----- coefficient = GRND8
C
C ... GENERATE WALL SHEAR COEFFICIENTS ....
C
      CALL FNLGLW(CO,CO,AK,1.0001,EWAL,4)
C
C ... NOW CONVERT TO Stanton #'s ....
C
      CALL GETYX(CO,A1,NY,NX)
      RPRL=1./PRNDTL(H1)
      RPRT=1./PRT(H1)
      P =9.*(RPRT/RPRL - 1.)*(RPRL/RPRT)**0.25
      I=(IXF-2)*NY
      DO 13801 II=IXF,IXL
        I=I + NY
        DO 13802 J=IYF,IYL
          IJ=I + J
          S=A1(IJ)
          STL=S*RPRL
          STT=S*RPRT/(1. + P*SQRT(S))
          A1(IJ)=AMAX1(STL,STT)
13802   CONTINUE
13801   CONTINUE
C
C ... NOW ASSEMBLE COMPOSITE HEAT TRANSFER COEFFICIENTS ....
C
      CALL GETYX(AUX(DEN1),A2,NY,NX)
      CALL GETYX(LD7,A3,NY,NX)
      CALL GETYX(CP,A4,NY,NX)
      CALL GETCOV(NPATCH,INAME('UCRT'),COND,THICK)
      CWALL=COND/(THICK+TINY)
C
      CALL SUB4(I1,IXF,I2,IXL,J1,IYF,J2,IYL)
      READ(NPATCH(8:8),'(A1)') ADIR
      NDIREC=0
      IF(ADIR.EQ.'E' .OR. ADIR.EQ.'e') NDIREC= 1
      IF(ADIR.EQ.'W' .OR. ADIR.EQ.'w') NDIREC=-1
      IF(ADIR.EQ.'N' .OR. ADIR.EQ.'n') NDIREC= 2
      IF(ADIR.EQ.'S' .OR. ADIR.EQ.'s') NDIREC=-2
      IDIR=IABS(NDIREC)
      IF(IDIR.EQ.1) THEN
        KAREA=5
        KADD=NY
        I2=I1
      ELSEIF(IDIR.EQ.2) THEN

```

```

KAREA=7
KADD=1
J2=J1
ELSE
  CALL WRIT40('PATCH NAME PROTOCOL VIOLATED FOR GRNDS ')
  CALL WRIT40('COEFFICIENT OF CONJUGATE HEAT TRANSFER ')
  CALL WRIT40('MODEL. TSK TSK TSK ')
  CALL WAYOUT(1)
ENDIF

```

C

```

I=(I1-2)*NY
DO 13811 II=I1,I2
  I=I + NY
  DO 13812 J=J1,J2
    IJ1=I + J
    IJ2=IJ1 + KADD
    ST1=A1(IJ1)
    ST2=A1(IJ2)
    RO1=A2(IJ1)
    RO2=A2(IJ2)
    VW1=A3(IJ1)
    VW2=A3(IJ2)
    CP1=A4(IJ1)
    CP2=A4(IJ2)
    CO1=RO1*VW1*CP1*ST1
    CO2=RO2*VW2*CP2*ST2
    COEFF=CO1*CWALL*CO2/(CO1*CWALL + CO1*CO2 + CWALL*CO2 + TINY)
    A5(IJ1)=COEFF/CP1
    A5(IJ2)=COEFF/CP2
    A6(IJ1)=COEFF
    A6(IJ2)=COEFF
  13812 CONTINUE
13811 CONTINUE

```

13812 CONTINUE

13811 CONTINUE

C

```

CALL SETYX(C6,A5,NY,NX)
CALL SETYX(C7,A6,NY,NX)

```

C

C ... NOW MULTIPLY BY CORRECT AREA'S & DIVIDE BY PATGEO,RHO & Vwall

C

```

CALL GTIZYX(KAREA,IZ,A1,NY,NX)
I=(I1-2)*NY
DO 13821 II=I1,I2
  I=I + NY
  DO 13822 J=J1,J2
    IJ1=I + J
    IJ2=IJ1 + KADD
    AREA=A1(IJ1)
    A5(IJ1)=A5(IJ1)*AREA
    A5(IJ2)=A5(IJ2)*AREA
  13822 CONTINUE
13821 CONTINUE

```

13822 CONTINUE

13821 CONTINUE

C

```

CALL GETYX(PATGEO,A1,NY,NX)
I=(IXF-2)*NY
DO 13831 II=IXF,IXL
  I=I + NY
  DO 13832 J=IYF,IYL
    IJ=I + J
    A5(IJ)=A5(IJ)/(A1(IJ)*A2(IJ)*A3(IJ) + TINY)
  13832 CONTINUE

```

13832 CONTINUE

```

13831 CONTINUE
C
      CALL SETYX(CO,A5,NY,NX)
      CALL FN1(LGEN1,0.0)
C
C ... ADD UP TOTAL HEAT TRANSFERRED ....
C
      IF(ISWEEP.LT.LSWEEP-1.AND.MOD(ISWEEP,IG(901)).NE.0) RETURN
C
      CALL WRITBL
      CALL WRIT40('ADDING UP TOTAL Qdot FROM DUCT TO AIR. ')
      CALL WRIT2I('SWEEP # ',ISWEEP,',SLAB # ',IZSTEP)
      CALL GETYX(H1,A4,NY,NX)
      CALL GETYX(CP,A2,NY,NX)
      CALL SUB4(I1,IXF,I2,IXL,J1,IYF,J2,IYL)
      IF (NDIREC.EQ. 1) THEN
        I1=I2
        KADD=-NY
      ELSEIF(NDIREC.EQ.-1) THEN
        I2=I1
        KADD= NY
      ELSEIF(NDIREC.EQ. 2) THEN
        J1=J2
        KADD=-1
      ELSEIF(NDIREC.EQ.-2) THEN
        J2=J1
        KADD= 1
      ENDIF
C
      READ(NPATCH(7:7),'(A1)') ANUX
      I=(I1-2)*NY
      DO 13841 II=I1,I2
        I=I + NY
        DO 13842 J=J1,J2
          IJ1=I + J
          IJ2=IJ1 + KADD
          H11=A4(IJ1)
          H12=A4(IJ2)
          CP1=A2(IJ1)
          CP2=A2(IJ2)
          CO1=A5(IJ1)*A1(IJ1)*A3(IJ1)
          VA1=H12*CP1/CP2
          QDTTOT=QDTTOT + CO1*(VA1-H11)
          IF(ANUX.EQ.'1') QDOT01=QDOT01 + CO1*(VA1-H11)
          IF(ANUX.EQ.'2') QDOT02=QDOT02 + CO1*(VA1-H11)
          IF(ANUX.EQ.'3') QDOT03=QDOT03 + CO1*(VA1-H11)
          IF(ANUX.EQ.'4') QDOT04=QDOT04 + CO1*(VA1-H11)
        13842 CONTINUE
      13841 CONTINUE
C
      RETURN
139 CONTINUE
C----- SECTION 10 ----- coefficient = GRND9
      RETURN
1310 CONTINUE
C----- SECTION 11 ----- coefficient = GRND10
      RETURN
1311 CONTINUE
C----- SECTION 12 ----- value = GRND
      RETURN

```

```

1312 CONTINUE
C----- SECTION 13 ----- value = GRND1
      RETURN
1313 CONTINUE
C----- SECTION 14 ----- value = GRND2
      RETURN
1314 CONTINUE
C----- SECTION 15 ----- value = GRND3
      RETURN
1315 CONTINUE
C----- SECTION 16 ----- value = GRND4
      RETURN
1316 CONTINUE
C----- SECTION 17 ----- value = GRND5
      RETURN
1317 CONTINUE
C----- SECTION 18 ----- value = GRND6
      RETURN
1318 CONTINUE
C----- SECTION 19 ----- value = GRND7
      IF(INDVAR.LT.U1 .OR. INDVAR.GT.W2) GO TO 13189
      CALL GETYX(AUX(DEN1),A1,NY,NX)
      CALL GETYX(P1      ,A2,NY,NX)
      CALL GETCOV(NPATCH,INAME('UCRT'),COEFF,GKLOSS)
      CALL GETCOV(NPATCH, P1 ,COEFF,GPBV )
      I=(IXF-2)*NY
      DO 13181 II=IXF,IXL
        I=I + NY
        DO 13182 J=IYF,IYL
          IJ=I + J
          DELTAP= A2(IJ)-GPBV
          ABSDP = ABS(DELTAP)
          RHO   = A1(IJ)
          VMAG  = SQRT(2.*ABSDP/(GKLOSS*RHO))
          A1(IJ)=-SIGN(VMAG,DELTAP)
13182 CONTINUE
13181 CONTINUE
      CALL SETYX(VAL,A1,NY,NX)
      RETURN
13189 CALL WRIT40('VAL = GRND7 FOR VARBLE BESIDES [U,V,W]1.')
      CALL WAYOUT(1)
      RETURN
1319 CONTINUE
C----- SECTION 20 ----- value = GRND8
      CALL GETYX(H1,A1,NY,NX)
      CALL GETYX(CP,A2,NY,NX)
C
      I=(I1-2)*NY
      DO 13191 II=I1,I2
        I=I + NY
        DO 13192 J=J1,J2
          IJ1=I + J
          IJ2=IJ1 + KADD
          H11=A1(IJ1)
          H12=A1(IJ2)
          CP1=A2(IJ1)
          CP2=A2(IJ2)
          VA1=H12*CP1/CP2
          VA2=H11*CP2/CP1
          A3(IJ1)=VA1

```

```

      A3(IJ2)=VA2
13192  CONTINUE
13191  CONTINUE
C
      CALL SETYX(VAL,A3,NY,NX)
      RETURN
1320  CONTINUE
C----- SECTION 21 ----- value = GRND9
      DO 13201 IX=1,NX
      DO 13201 IY=1,NY
      IF(NPATCH.EQ.'XDYNIN') PHI(IY,IX)=-RG(803)*XFCTD
      IF(NPATCH.EQ.'XDYNOUT') PHI(IY,IX)=RG(803)*XFCTD
13201  CONTINUE
      CALL SETYX(VAL,PHI,JNY,JNX)
      RETURN
1321  CONTINUE
C----- SECTION 22 ----- value = GRND10
      DO 13211 IX=1,NX
      DO 13211 IY=1,NY
      PHI(IY,IX)=RG(804)*XFCTE
      IF(NPATCH.EQ.'XENGIN') PHI(IY,IX)=-RG(804)*XFCTE
      IF(NPATCH.EQ.'XENGOUT') PHI(IY,IX)=RG(805)*XFCTE
13211  CONTINUE
      CALL SETYX(VAL,PHI,JNY,JNX)
      RETURN
C*****
C
C--- GROUP 14. Downstream pressure for PARAB=.TRUE.
C
      14 CONTINUE
      RETURN
C*****
C * Make changes for this group only in group 19.
C--- GROUP 15. Termination of sweeps
C--- GROUP 16. Termination of iterations
C--- GROUP 17. Under-relaxation devices
C--- GROUP 18. Limits on variables or increments to them
C*****
C
C--- GROUP 19. Special calls to GROUND from EARTH
C
      19 GO TO (191,192,193,194,195,196,197,198),ISC
191  CONTINUE
C * ----- SECTION 1 ----- START OF TIME STEP.
C
C-pd---Misc-----
C
      IF(IG(999).EQ.1) STOP
      QDTTOT=0.0
      QDOTT1=0.0
      QDOTT2=0.0
      QDOTT3=0.0
      QDOTT4=0.0
      IPASS=0
      IRAXV=0
      IRAXT=0
      IRAXS=0
      XFCTD=1.0
      XFCTE=1.0
      ITST=TSTSWP

```

INPR=NPRMON
NPRMON=1

```
C
C-pd---Assign monitoring locations-----
C
      IXMON1 =IXMON
      IYMON1 =IYMON
      IZMON1 =IZMON
C
      IXMON2 =IG(11)
      IYMON2 =IG(12)
      IZMON2 =IG(13)
C
      IXMON3 =IG(14)
      IYMON3 =IG(15)
      IZMON3 =IG(16)
C
      IXMON4 =IG(17)
      IYMON4 =IG(18)
      IZMON4 =IG(19)
C
      IXMON5 =IG(20)
      IYMON5 =IG(21)
      IZMON5 =IG(22)
C
      IXMON6 =IG(23)
      IYMON6 =IG(24)
      IZMON6 =IG(25)
C
      IXMON7 =IG(26)
      IYMON7 =IG(27)
      IZMON7 =IG(28)
C
      IXMON8 =IG(29)
      IYMON8 =IG(30)
      IZMON8 =IG(31)
C
      IXMON9 =IG(32)
      IYMON9 =IG(33)
      IZMON9 =IG(34)
C
      IXMON10=IG(35)
      IYMON10=IG(36)
      IZMON10=IG(37)
C
      RETURN
192 CONTINUE
C  * ----- SECTION 2 ----- START OF SWEEP.
C
C-pd---WARNING:  machine dependent-----
C
      call flush(6)
C
C-pd---Init-----
C
      IF(ISWEEP.EQ.FSWEEP) SUMA=0.0
C
C-pd---Check to reset tstswp-----
C
      IOPEN=0
```

```
IF(ITST.NE.TSTSWP) IPASS=IPASS+1
IF(IPASS.GT.10) THEN
  IPASS=0
  TSTSWP=ITST
ENDIF
```

```
C
C-pd---Init stuff for printout of max and min-----
C
```

```
XP1MIN= 1000000.0
XP1MAX=-1000000.0
XU1MIN= 1000000.0
XU1MAX=-1000000.0
XV1MIN= 1000000.0
XV1MAX=-1000000.0
XW1MIN= 1000000.0
XW1MAX=-1000000.0
XKEMIN= 1000000.0
XKEMAX=-1000000.0
XEPMIN= 1000000.0
XEPMAX=-1000000.0
XH1MIN= 1000000.0
XH1MAX=-1000000.0
XT1MIN= 1000000.0
XT1MAX=-1000000.0
XETMIN= 1000000.0
XETMAX=-1000000.0
IXPMAX=0
IYPMAX=0
IZPMAX=0
IXPMIN=0
IYPMIN=0
IZPMIN=0
IXUMAX=0
IYUMAX=0
IZUMAX=0
IXUMIN=0
IYUMIN=0
IZUMIN=0
IXVMAX=0
IYVMAX=0
IZVMAX=0
IXVMIN=0
IYVMIN=0
IZVMIN=0
IXWMAX=0
IYWMAX=0
IZWMAX=0
IXWMIN=0
IYWMIN=0
IZWMIN=0
IXKMAX=0
IYKMAX=0
IZKMAX=0
IXKMIN=0
IYKMIN=0
IZKMIN=0
IXEMAX=0
IYEMAX=0
IZEMAX=0
IXEMIN=0
```

```

IYEMIN=0
IZEMIN=0
IXHMAX=0
IYHMAX=0
IZHMAX=0
IXHMIN=0
IYHMIN=0
IZHMIN=0
IXTMAX=0
IYTMAX=0
IZTMAX=0
IXTMIN=0
IYTMIN=0
IZTMIN=0
IXXMAX=0
IYXMAX=0
IZXMAX=0
IXXMIN=0
IYXMIN=0
IZXMIN=0

```

```

C
RETURN
193 CONTINUE
C * ----- SECTION 3 ---- START OF IZ SLAB.
RETURN
194 CONTINUE
C * ----- SECTION 4 ---- START OF ITERATION.
C IF(IRAXV.EQ.1) THEN
C CALL XSETCV('RAX1', U1,XCOF,XVEL,RAXFTV,1.0)
C CALL XSETCV('RAX1', V1,XCOF,XVEL,RAXFTV,1.0)
C CALL XSETCV('RAX1', W1,XCOF,XVEL,RAXFTV,1.0)
C WRITE(6,*)' CO FROM SETCV VEL -> ',XCOF
C IRAXV=0
C ENDIF
C IF(IRAXT.EQ.1) THEN
C CALL XSETCV('RAX1', KE,XCOF,XVEL,RAXFTT,1.0)
C CALL XSETCV('RAX1', EP,XCOF,XVEL,RAXFTT,1.0)
C WRITE(6,*)' CO FROM SETCV TUR -> ',XCOF
C IRAXT=0
C ENDIF
C IF(IRAXS.EQ.1) THEN
C CALL XSETCV('RAX1', H1,XCOF,XVEL,RAXFTS,1.0)
C CALL XSETCV('RAX1', C1,XCOF,XVEL,RAXFTS,1.0)
C CALL XSETCV('RAX1', C2,XCOF,XVEL,RAXFTS,1.0)
C WRITE(6,*)' CO FROM SETCV SCA -> ',XCOF
C IRAXS=0
C ENDIF
C
C-pd---Modify inlet areas-----
C
IF(ISWEEP.NE.FSWEEP) RETURN
IF(IZ.GE.IG(704).AND.IZ.LE.IG(705)) THEN
CALL GTIZYX(7,IZ,GAH,JNY,JNX)
DO 19301 IX=IG(702),IG(703)
SUMA=SUMA+GAH(IG(701),IX)
19301 CONTINUE
ENDIF
C
IF(IZ.EQ.IG(711)) THEN
CALL GTIZYX(9,IZ,GAH,JNY,JNX)

```

```

        SUMB=0.0
        DO 19302 IX=IG(712),IG(713)
        DO 19302 IY=IG(714),IG(715)
        SUMB=SUMB+GAH(IY,IX)
19302    CONTINUE
        .ENDIF
C
        IF(IZ.EQ.NZ) THEN
            XFCTD=RG(801)/SUMA
            XFCTE=RG(802)/SUMB
C            CALL XSETCV('XDYNOUT',P1,XCOF,XVEL,1.0,XFCTD)
C            CALL XSETCV('XENGOUT',P1,XCOF,XVEL,1.0,XFCTE)
C            CALL XSETCV('XDYNOIN',P1,XCOF,XVEL,1.0,XFCTD)
C            CALL XSETCV('XENGIN',P1,XCOF,XVEL,1.0,XFCTE)
        ENDIF
C
        RETURN
195 CONTINUE
C * ----- SECTION 5 ---- FINISH OF ITERATION.
        RETURN
196 CONTINUE
C * ----- SECTION 6 ---- FINISH OF IZ SLAB.
        IF (MOD(ISWEEP,IG(902)).NE.0.AND.ISWEEP.NE.LSWEEP-1) GOTO 1961
        IF(IZ.EQ.1) WRITE(6,*)' ==> CALCULATING ENGLISH UNITS '
C-pd---Dispensed by DBS for unknown reasons????????????????????????????????
        CALL GETCAR
C        CALL BCARTC(1,1)
C
        CALL GETYX(P1,PHI,JNY,JNX)
        DO 19611 IX=1,NX
        DO 19611 IY=1,NY
19611 PHI(IY,IX)=PHI(IY,IX)*RG(36)
        CALL SETYX(PH20,PHI,JNY,JNX)
C
        CALL GETYX(INAME('UCRT'),PHI,JNY,JNX)
        DO 19612 IX=1,NX
        DO 19612 IY=1,NY
19612 PHI(IY,IX)=PHI(IY,IX)*RG(37)
        CALL SETYX(U2,PHI,JNY,JNX)
C
        CALL GETYX(INAME('VCRT'),PHI,JNY,JNX)
        DO 19613 IX=1,NX
        DO 19613 IY=1,NY
19613 PHI(IY,IX)=PHI(IY,IX)*RG(37)
        CALL SETYX(V2,PHI,JNY,JNX)
C
        CALL GETYX(INAME('WCRT'),PHI,JNY,JNX)
        DO 19614 IX=1,NX
        DO 19614 IY=1,NY
19614 PHI(IY,IX)=PHI(IY,IX)*RG(37)
        CALL SETYX(W2,PHI,JNY,JNX)
C
        CALL GETYX(TEMP,PHI,JNY,JNX)
        DO 19615 IX=1,NX
        DO 19615 IY=1,NY
19615 PHI(IY,IX)=PHI(IY,IX)/RG(33)-RG(32)
        CALL SETYX(TFAR,PHI,JNY,JNX)
C
        CALL GETYX(AUX(DEN1),PHI,JNY,JNX)
        DO 19616 IX=1,NX

```

```
DO 19616 IY=1,NY
19616 PHI(IY,IX)=PHI(IY,IX)*RG(38)
CALL SETYX(RHOE,PHI,JNY,JNX)
```

C

C-pd---Find max and min-----

C

```
1961 IF(MOD(ISWEEP,NPRMON).EQ.0) THEN
CALL GETYX(P1,PHI,JNY,JNX)
CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
DO 19617 IX=1,NX
DO 19617 IY=1,NY
IF (GVPR(IY,IX).LE.1.E-4) GOTO 19617
IF(PHI(IY,IX).GT.XP1MAX) THEN
XP1MAX=PHI(IY,IX)
IXP1MAX=IX
IYP1MAX=IY
IZP1MAX=IZ
ENDIF
IF(PHI(IY,IX).LT.XP1MIN) THEN
XP1MIN=PHI(IY,IX)
IXP1MIN=IX
IYP1MIN=IY
IZP1MIN=IZ
```

ENDIF

```
19617 CONTINUE
```

C

```
CALL GETYX(U1,PHI,JNY,JNX)
CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
DO 19618 IX=1,NX
DO 19618 IY=1,NY
IF (GVPR(IY,IX).LE.1.E-4) GOTO 19618
IF(PHI(IY,IX).GT.XU1MAX) THEN
XU1MAX=PHI(IY,IX)
IXU1MAX=IX
IYU1MAX=IY
IZU1MAX=IZ
ENDIF
IF(PHI(IY,IX).LT.XU1MIN) THEN
XU1MIN=PHI(IY,IX)
IXU1MIN=IX
IYU1MIN=IY
IZU1MIN=IZ
```

ENDIF

```
19618 CONTINUE
```

C

```
CALL GETYX(V1,PHI,JNY,JNX)
CALL GETYX (INAME('VPOR'),GVPR,JNY,JNX)
DO 19619 IX=1,NX
DO 19619 IY=1,NY
IF (GVPR(IY,IX).LE.1.E-4) GOTO 19619
IF(PHI(IY,IX).GT.XV1MAX) THEN
XV1MAX=PHI(IY,IX)
IXV1MAX=IX
IYV1MAX=IY
IZV1MAX=IZ
ENDIF
IF(PHI(IY,IX).LT.XV1MIN) THEN
XV1MIN=PHI(IY,IX)
IXV1MIN=IX
IYV1MIN=IY
```

```

        IZVMIN=IZ
        ENDIF
19619   CONTINUE
C
        CALL GETYX(W1, PHI, JNY, JNX)
        CALL GETYX (INAME('VPOR'), GVPR, JNY, JNX)
        DO 19620 IX=1, NX
        DO 19620 IY=1, NY
        IF (GVPR(IY, IX).LE.1.E-4) GOTO 19620
        IF (PHI(IY, IX).GT.XW1MAX) THEN
            XW1MAX=PHI(IY, IX)
            IXWMAX=IX
            IYWMAX=IY
            IZWMAX=IZ
        ENDIF
        IF (PHI(IY, IX).LT.XW1MIN) THEN
            XW1MIN=PHI(IY, IX)
            IXWMIN=IX
            IYWMIN=IY
            IZWMIN=IZ
        ENDIF
19620   CONTINUE
C
        CALL GETYX(KE, PHI, JNY, JNX)
        CALL GETYX (INAME('VPOR'), GVPR, JNY, JNX)
        DO 19621 IX=1, NX
        DO 19621 IY=1, NY
        IF (GVPR(IY, IX).LE.1.E-4) GOTO 19621
        IF (PHI(IY, IX).GT.XKEMAX) THEN
            XKEMAX=PHI(IY, IX)
            IXKMAX=IX
            IYKMAX=IY
            IZKMAX=IZ
        ENDIF
        IF (PHI(IY, IX).LT.XKEMIN) THEN
            XKEMIN=PHI(IY, IX)
            IXKMIN=IX
            IYKMIN=IY
            IZKMIN=IZ
        ENDIF
19621   CONTINUE
C
        CALL GETYX(EP, PHI, JNY, JNX)
        CALL GETYX (INAME('VPOR'), GVPR, JNY, JNX)
        DO 19622 IX=1, NX
        DO 19622 IY=1, NY
        IF (GVPR(IY, IX).LE.1.E-4) GOTO 19622
        IF (PHI(IY, IX).GT.XEPMAX) THEN
            XEPMAX=PHI(IY, IX)
            IXEMAX=IX
            IYEMAX=IY
            IZEMAX=IZ
        ENDIF
        IF (PHI(IY, IX).LT.XEPMIN) THEN
            XEPMIN=PHI(IY, IX)
            IXEMIN=IX
            IYEMIN=IY
            IZEMIN=IZ
        ENDIF
19622   CONTINUE

```

C

```

CALL GETYX(H1, PHI, JNY, JNX)
CALL GETYX (INAME('VPOR'), GVPR, JNY, JNX)
DO 19623 IX=1, NX
DO 19623 IY=1, NY
IF (GVPR(IY, IX).LE.1.E-4) GOTO 19623
IF (PHI(IY, IX).GT.XH1MAX) THEN
    XH1MAX=PHI(IY, IX)
    IXHMAX=IX
    IYHMAX=IY
    IZHMAX=IZ
ENDIF
IF (PHI(IY, IX).LT.XH1MIN) THEN
    XH1MIN=PHI(IY, IX)
    IXHMIN=IX
    IYHMIN=IY
    IZHMIN=IZ
ENDIF

```

19623

C

```

CONTINUE
CALL GETYX(TEMP, PHI, JNY, JNX)
CALL GETYX (INAME('VPOR'), GVPR, JNY, JNX)
DO 19624 IX=1, NX
DO 19624 IY=1, NY
IF (GVPR(IY, IX).LE.1.E-4) GOTO 19624
IF (PHI(IY, IX).GT.XT1MAX) THEN
    XT1MAX=PHI(IY, IX)
    IXTMAX=IX
    IYTMAX=IY
    IZTMAX=IZ
ENDIF
IF (PHI(IY, IX).LT.XT1MIN) THEN
    XT1MIN=PHI(IY, IX)
    IXTMIN=IX
    IYTMIN=IY
    IZTMIN=IZ
ENDIF

```

19624

C

```

CONTINUE
CALL GETYX(AUX(VIST), PHI, JNY, JNX)
CALL GETYX (INAME('VPOR'), GVPR, JNY, JNX)
DO 19625 IX=1, NX
DO 19625 IY=1, NY
IF (GVPR(IY, IX).LE.1.E-4) GOTO 19625
IF (PHI(IY, IX).GT.XETMAX) THEN
    XETMAX=PHI(IY, IX)
    IXXMAX=IX
    IYXMAX=IY
    IZXMAX=IZ
ENDIF
IF (PHI(IY, IX).LT.XETMIN) THEN
    XETMIN=PHI(IY, IX)
    IXXMIN=IX
    IYXMIN=IY
    IZXMIN=IZ
ENDIF

```

19625

C

C-pd---Get monitoring values-----

C

```

IF(MOD(ISWEEP,TSTSWP).NE.0) GOTO 19692
IF(IZ.NE.IZMON1) GOTO 1962
  CALL GETONE(P1,PP1,IYMON1,IXMON1)
  CALL GETONE(U1,UU1,IYMON1,IXMON1)
  CALL GETONE(V1,VV1,IYMON1,IXMON1)
  CALL GETONE(W1,WW1,IYMON1,IXMON1)
  CALL GETONE(AUX(DEN1),DD1,IYMON1,IXMON1)
  IF(STORE(KE)) CALL GETONE(KE,KE1,IYMON1,IXMON1)
  IF(STORE(EP)) CALL GETONE(EP,EP1,IYMON1,IXMON1)
  IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET1,IYMON1,IXMON1)
  CALL GETONE(C1,C1C1,IYMON1,IXMON1)
  IF(STORE(C2)) CALL GETONE(C2,C2C1,IYMON1,IXMON1)
  IF(STORE(C3)) CALL GETONE(C3,C3C1,IYMON1,IXMON1)
  IF(STORE(CP)) CALL GETONE(CP,CPC1,IYMON1,IXMON1)
  IF(STORE(C11)) CALL GETONE(C11,CXC1,IYMON1,IXMON1)
  CALL GETONE(TEMP,C4C1,IYMON1,IXMON1)
  CALL GETONE(H1,H1H1,IYMON1,IXMON1)
1962 IF(IZ.NE.IZMON2) GOTO 1963
  CALL GETONE(P1,PP2,IYMON2,IXMON2)
  CALL GETONE(U1,UU2,IYMON2,IXMON2)
  CALL GETONE(V1,VV2,IYMON2,IXMON2)
  CALL GETONE(W1,WW2,IYMON2,IXMON2)
  CALL GETONE(AUX(DEN1),DD2,IYMON2,IXMON2)
  IF(STORE(KE)) CALL GETONE(KE,KE2,IYMON2,IXMON2)
  IF(STORE(EP)) CALL GETONE(EP,EP2,IYMON2,IXMON2)
  IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET2,IYMON2,IXMON2)
  CALL GETONE(C1,C1C2,IYMON2,IXMON2)
  IF(STORE(C2)) CALL GETONE(C2,C2C2,IYMON2,IXMON2)
  IF(STORE(C3)) CALL GETONE(C3,C3C2,IYMON2,IXMON2)
  IF(STORE(CP)) CALL GETONE(CP,CPC2,IYMON2,IXMON2)
  IF(STORE(C11)) CALL GETONE(C11,CXC2,IYMON2,IXMON2)
  CALL GETONE(TEMP,C4C2,IYMON2,IXMON2)
  CALL GETONE(H1,H1H2,IYMON2,IXMON2)
1963 IF(IZ.NE.IZMON3) GOTO 1964
  CALL GETONE(P1,PP3,IYMON3,IXMON3)
  CALL GETONE(U1,UU3,IYMON3,IXMON3)
  CALL GETONE(V1,VV3,IYMON3,IXMON3)
  CALL GETONE(W1,WW3,IYMON3,IXMON3)
  CALL GETONE(AUX(DEN1),DD3,IYMON3,IXMON3)
  IF(STORE(KE)) CALL GETONE(KE,KE3,IYMON3,IXMON3)
  IF(STORE(EP)) CALL GETONE(EP,EP3,IYMON3,IXMON3)
  IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET3,IYMON3,IXMON3)
  CALL GETONE(C1,C1C3,IYMON3,IXMON3)
  IF(STORE(C2)) CALL GETONE(C2,C2C3,IYMON3,IXMON3)
  IF(STORE(C3)) CALL GETONE(C3,C3C3,IYMON3,IXMON3)
  IF(STORE(CP)) CALL GETONE(CP,CPC3,IYMON3,IXMON3)
  IF(STORE(C11)) CALL GETONE(C11,CXC3,IYMON3,IXMON3)
  CALL GETONE(TEMP,C4C3,IYMON3,IXMON3)
  CALL GETONE(H1,H1H3,IYMON3,IXMON3)
1964 IF(IZ.NE.IZMON4) GOTO 1965
  CALL GETONE(P1,PP4,IYMON4,IXMON4)
  CALL GETONE(U1,UU4,IYMON4,IXMON4)
  CALL GETONE(V1,VV4,IYMON4,IXMON4)
  CALL GETONE(W1,WW4,IYMON4,IXMON4)
  CALL GETONE(AUX(DEN1),DD4,IYMON4,IXMON4)
  IF(STORE(KE)) CALL GETONE(KE,KE4,IYMON4,IXMON4)
  IF(STORE(EP)) CALL GETONE(EP,EP4,IYMON4,IXMON4)
  IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET4,IYMON4,IXMON4)
  CALL GETONE(C1,C1C4,IYMON4,IXMON4)

```

```

IF(STORE(C2)) CALL GETONE(C2,C2C4,IYMON4,IXMON4)
IF(STORE(C3)) CALL GETONE(C3,C3C4,IYMON4,IXMON4)
IF(STORE(CP)) CALL GETONE(CP,CPC4,IYMON4,IXMON4)
IF(STORE(C11)) CALL GETONE(C11,CXC4,IYMON4,IXMON4)
CALL GETONE(TEMP,C4C4,IYMON4,IXMON4)
CALL GETONE(H1,H1H4,IYMON4,IXMON4)
1965 IF(IZ.NE.IZMON5) GOTO 1966
CALL GETONE(P1,PP5,IYMON5,IXMON5)
CALL GETONE(U1,UU5,IYMON5,IXMON5)
CALL GETONE(V1,VV5,IYMON5,IXMON5)
CALL GETONE(W1,WW5,IYMON5,IXMON5)
CALL GETONE(AUX(DEN1),DD5,IYMON5,IXMON5)
IF(STORE(KE)) CALL GETONE(KE,KE5,IYMON5,IXMON5)
IF(STORE(EP)) CALL GETONE(EP,EP5,IYMON5,IXMON5)
IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET5,IYMON5,IXMON5)
CALL GETONE(C1,C1C5,IYMON5,IXMON5)
IF(STORE(C2)) CALL GETONE(C2,C2C5,IYMON5,IXMON5)
IF(STORE(C3)) CALL GETONE(C3,C3C5,IYMON5,IXMON5)
IF(STORE(CP)) CALL GETONE(CP,CPC5,IYMON5,IXMON5)
IF(STORE(C11)) CALL GETONE(C11,CXC5,IYMON5,IXMON5)
CALL GETONE(TEMP,C4C5,IYMON5,IXMON5)
CALL GETONE(H1,H1H5,IYMON5,IXMON5)
1966 IF(IZ.NE.IZMON6) GOTO 1967
CALL GETONE(P1,PP6,IYMON6,IXMON6)
CALL GETONE(U1,UU6,IYMON6,IXMON6)
CALL GETONE(V1,VV6,IYMON6,IXMON6)
CALL GETONE(W1,WW6,IYMON6,IXMON6)
CALL GETONE(AUX(DEN1),DD6,IYMON6,IXMON6)
IF(STORE(KE)) CALL GETONE(KE,KE6,IYMON6,IXMON6)
IF(STORE(EP)) CALL GETONE(EP,EP6,IYMON6,IXMON6)
IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET6,IYMON6,IXMON6)
CALL GETONE(C1,C1C6,IYMON6,IXMON6)
IF(STORE(C2)) CALL GETONE(C2,C2C6,IYMON6,IXMON6)
IF(STORE(C3)) CALL GETONE(C3,C3C6,IYMON6,IXMON6)
IF(STORE(CP)) CALL GETONE(CP,CPC6,IYMON6,IXMON6)
IF(STORE(C11)) CALL GETONE(C11,CXC6,IYMON6,IXMON6)
CALL GETONE(TEMP,C4C6,IYMON6,IXMON6)
CALL GETONE(H1,H1H6,IYMON6,IXMON6)
1967 IF(IZ.NE.IZMON7) GOTO 1968
CALL GETONE(P1,PP7,IYMON7,IXMON7)
CALL GETONE(U1,UU7,IYMON7,IXMON7)
CALL GETONE(V1,VV7,IYMON7,IXMON7)
CALL GETONE(W1,WW7,IYMON7,IXMON7)
CALL GETONE(AUX(DEN1),DD7,IYMON7,IXMON7)
IF(STORE(KE)) CALL GETONE(KE,KE7,IYMON7,IXMON7)
IF(STORE(EP)) CALL GETONE(EP,EP7,IYMON7,IXMON7)
IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET7,IYMON7,IXMON7)
CALL GETONE(C1,C1C7,IYMON7,IXMON7)
IF(STORE(C2)) CALL GETONE(C2,C2C7,IYMON7,IXMON7)
IF(STORE(C3)) CALL GETONE(C3,C3C7,IYMON7,IXMON7)
IF(STORE(CP)) CALL GETONE(CP,CPC7,IYMON7,IXMON7)
IF(STORE(C11)) CALL GETONE(C11,CXC7,IYMON7,IXMON7)
CALL GETONE(TEMP,C4C7,IYMON7,IXMON7)
CALL GETONE(H1,H1H7,IYMON7,IXMON7)
1968 IF(IZ.NE.IZMON8) GOTO 1969
CALL GETONE(P1,PP8,IYMON8,IXMON8)
CALL GETONE(U1,UU8,IYMON8,IXMON8)
CALL GETONE(V1,VV8,IYMON8,IXMON8)
CALL GETONE(W1,WW8,IYMON8,IXMON8)
CALL GETONE(AUX(DEN1),DD8,IYMON8,IXMON8)

```

```

IF(STORE(KE)) CALL GETONE(KE,KE8,IYMON8,IXMON8)
IF(STORE(EP)) CALL GETONE(EP,EP8,IYMON8,IXMON8)
IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET8,IYMON8,IXMON8)
CALL GETONE(C1,C1C8,IYMON8,IXMON8)
IF(STORE(C2)) CALL GETONE(C2,C2C8,IYMON8,IXMON8)
IF(STORE(C3)) CALL GETONE(C3,C3C8,IYMON8,IXMON8)
IF(STORE(CP)) CALL GETONE(CP,CPC8,IYMON8,IXMON8)
IF(STORE(C11)) CALL GETONE(C11,CXC8,IYMON8,IXMON8)
CALL GETONE(TEMP,C4C8,IYMON8,IXMON8)
CALL GETONE(H1,H1H8,IYMON8,IXMON8)
1969 IF(IZ.NE.IZMON9) GOTO 19691
CALL GETONE(P1,PP9,IYMON9,IXMON9)
CALL GETONE(U1,UU9,IYMON9,IXMON9)
CALL GETONE(V1,VV9,IYMON9,IXMON9)
CALL GETONE(W1,WW9,IYMON9,IXMON9)
CALL GETONE(AUX(DEN1),DD9,IYMON9,IXMON9)
IF(STORE(KE)) CALL GETONE(KE,KE9,IYMON9,IXMON9)
IF(STORE(EP)) CALL GETONE(EP,EP9,IYMON9,IXMON9)
IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET9,IYMON9,IXMON9)
CALL GETONE(C1,C1C9,IYMON9,IXMON9)
IF(STORE(C2)) CALL GETONE(C2,C2C9,IYMON9,IXMON9)
IF(STORE(C3)) CALL GETONE(C3,C3C9,IYMON9,IXMON9)
IF(STORE(CP)) CALL GETONE(CP,CPC9,IYMON9,IXMON9)
IF(STORE(C11)) CALL GETONE(C11,CXC9,IYMON9,IXMON9)
CALL GETONE(TEMP,C4C9,IYMON9,IXMON9)
CALL GETONE(H1,H1H9,IYMON9,IXMON9)
19691 IF(IZ.NE.IZMON10) GOTO 19692
CALL GETONE(P1,PP10,IYMON10,IXMON10)
CALL GETONE(U1,UU10,IYMON10,IXMON10)
CALL GETONE(V1,VV10,IYMON10,IXMON10)
CALL GETONE(W1,WW10,IYMON10,IXMON10)
CALL GETONE(AUX(DEN1),DD10,IYMON10,IXMON10)
IF(STORE(KE)) CALL GETONE(KE,KE10,IYMON10,IXMON10)
IF(STORE(EP)) CALL GETONE(EP,EP10,IYMON10,IXMON10)
IF(STORE(AUX(VIST))) CALL GETONE(AUX(VIST),ET10,IYMON10,IXMON10)
CALL GETONE(C1,C1C10,IYMON10,IXMON10)
IF(STORE(C2)) CALL GETONE(C2,C2C10,IYMON10,IXMON10)
IF(STORE(C3)) CALL GETONE(C3,C3C10,IYMON10,IXMON10)
IF(STORE(CP)) CALL GETONE(CP,CPC10,IYMON10,IXMON10)
IF(STORE(C11)) CALL GETONE(C11,CXC10,IYMON10,IXMON10)
CALL GETONE(TEMP,C4C10,IYMON10,IXMON10)
CALL GETONE(H1,H1H10,IYMON10,IXMON10)
19692 CONTINUE
C
RETURN
197 CONTINUE
C * ----- SECTION 7 ---- FINISH OF SWEEP.
C
C-pd---Printout of monitoring locations-----
C
IF(MOD(ISWEEP,TSTSWP).EQ.0.AND.IG(38).EQ.1) WRITE(6,1977)
& IXMON1,IYMON1,IZMON1,PP1,UU1,VV1,WW1,DD1,
& IXMON2,IYMON2,IZMON2,PP2,UU2,VV2,WW2,DD2,
& IXMON3,IYMON3,IZMON3,PP3,UU3,VV3,WW3,DD3,
& IXMON4,IYMON4,IZMON4,PP4,UU4,VV4,WW4,DD4,
& IXMON5,IYMON5,IZMON5,PP5,UU5,VV5,WW5,DD5,
& IXMON6,IYMON6,IZMON6,PP6,UU6,VV6,WW6,DD6,
& IXMON7,IYMON7,IZMON7,PP7,UU7,VV7,WW7,DD7,
& IXMON8,IYMON8,IZMON8,PP8,UU8,VV8,WW8,DD8,
& IXMON9,IYMON9,IZMON9,PP9,UU9,VV9,WW9,DD9,

```

```

& IXMON10,IYMON10,IZMON10,PP10,UU10,VV10,WW10,DD10
1977 FORMAT(1X,'MONITORING VALUES : '2X,'P1',11X,'U1',11X,'V1',11X,
& 'W1',10X,'RHO1'/,10(1X,'AT(',I2,',',I2,',',I2,')':1P,5E13.5:/))
IF(MOD(ISWEEP,TSTSWP).EQ.0.AND.IG(39).EQ.1) WRITE(6,1978)
& IXMON1,IYMON1,IZMON1,KE1,EP1,C1C1,ET1,C4C1,
& IXMON2,IYMON2,IZMON2,KE2,EP2,C1C2,ET2,C4C2,
& IXMON3,IYMON3,IZMON3,KE3,EP3,C1C3,ET3,C4C3,
& IXMON4,IYMON4,IZMON4,KE4,EP4,C1C4,ET4,C4C4,
& IXMON5,IYMON5,IZMON5,KE5,EP5,C1C5,ET5,C4C5,
& IXMON6,IYMON6,IZMON6,KE6,EP6,C1C6,ET6,C4C6,
& IXMON7,IYMON7,IZMON7,KE7,EP7,C1C7,ET7,C4C7,
& IXMON8,IYMON8,IZMON8,KE8,EP8,C1C8,ET8,C4C8,
& IXMON9,IYMON9,IZMON9,KE9,EP9,C1C9,ET9,C4C9,
& IXMON10,IYMON10,IZMON10,KE10,EP10,C1C10,ET10,C4C10
1978 FORMAT(1X,'MONITORING VALUES : '2X,'KE',11X,'EP',11X,'C1',10X,
& 'ENUT',9X,'TEMP'/,10(1X,'LO(',I2,',',I2,',',I2,')':1P,5E13.5:/))
IF(MOD(ISWEEP,TSTSWP).EQ.0.AND.IG(40).EQ.1) WRITE(6,1979)
& IXMON1,IYMON1,IZMON1,H1H1,C2C1,C3C1,CPC1,CXC1,
& IXMON2,IYMON2,IZMON2,H1H2,C2C2,C3C2,CPC2,CXC2,
& IXMON3,IYMON3,IZMON3,H1H3,C2C3,C3C3,CPC3,CXC3,
& IXMON4,IYMON4,IZMON4,H1H4,C2C4,C3C4,CPC4,CXC4,
& IXMON5,IYMON5,IZMON5,H1H5,C2C5,C3C5,CPC5,CXC5,
& IXMON6,IYMON6,IZMON6,H1H6,C2C6,C3C6,CPC6,CXC6,
& IXMON7,IYMON7,IZMON7,H1H7,C2C7,C3C7,CPC7,CXC7,
& IXMON8,IYMON8,IZMON8,H1H8,C2C8,C3C8,CPC8,CXC8,
& IXMON9,IYMON9,IZMON9,H1H9,C2C9,C3C9,CPC9,CXC9,
& IXMON10,IYMON10,IZMON10,H1H10,C2C10,C3C10,CPC10,CXC10
1979 FORMAT(1X,'MONITORING VALUES : '2X,'H1',11X,'C2',11X,'C3',11X,
& 'CP',10X,'SPAR'/,10(1X,'PT(',I2,',',I2,',',I2,')':1P,5E13.5:/))

```

C
C-pd---Printout heat info-----
C

```

IF (IG(41).EQ.1) THEN
  CALL GETSOR('HEATTR1E',H1,QDOT1)
  CALL GETSOR('HEATTR1W',H1,QDOT2)
  CALL GETSOR('HEATTR1N',H1,QDOT3)
  CALL GETSOR('HEATTR1S',H1,QDOT4)
  CALL WRITBL
  CALL WRIT4R(' Qdot 1 ',QDOT1,', Qdot 2 ',QDOT2,
& ', Qdot 3 ',QDOT3,', Qdot 4 ',QDOT4)
ENDIF

```

C
C-pd---Printout sorc and calc pumping ratio-----
C

```

IF(MOD(ISWEEP,NPRMON).EQ.0) THEN
  CALL GETSOR('XDYNOUT',R1,XMDOT1)
  CALL GETSOR('XOPEN2', R1,XMDOT2)
  CALL GETSOR('XENGOUT',R1,XMDOT3)
  CALL GETSOR('XOPEN3', R1,XMDOT4)
  CALL GETSOR('XDYNIN',R1,XMDOT5)
  CALL GETSOR('XENGIN', R1,XMDOT6)
  CALL GETSOR('XOPEN1', R1,XMDOT7)
  CALL GETSOR('XDYNOUT',V1,XVVEL1)
  CALL GETSOR('XENGOUT',W1,XWVEL1)
  XPR1=(-XMDOT2-XMDOT1)/XMDOT1
  XPR2=(-XMDOT4-XMDOT3)/XMDOT3
  XERR1=RESD(P1)*RESREF(P1)*RG(701)*100.0/XMDOT7
  XERR2=RESD(W1)*RESREF(W1)*100.0/(XWVEL1+XVVEL1)
  CALL WRIT4R(' Mdot 1 ',XMDOT1,', Mdot 2 ',XMDOT2,
& ', Mdot 3 ',XMDOT3,', Mdot 4 ',XMDOT4)

```

```

      CALL WRIT2R(' PR Dyn ',XPR1,',PR Eng ',XPR2)
      CALL WRIT4R(' DYN IN',XMDOT5/RG(35),',DYN OUT',XMDOT1/RG(35),
&      ',ENG IN',XMDOT6/RG(35),',ENG OUT',XMDOT3/RG(35))
      XFUL=(XMDOT3+XMDOT6)/RG(35)
      CALL WRIT1R(' FUEL IN',XFUL)
      CALL WRIT2R(' ErrMdot',XERR1,',ErrVel ',XERR2)
    ENDIF

```

```

C
C-pd---Printout max and min-----
C

```

```

      IF(MOD(ISWEEP,NPRMON).EQ.0) THEN
        WRITE(6,*)' P1MAX LOC ',XP1MAX,IXPMAX,IYPMAX,IZPMAX
        WRITE(6,*)' P1MIN LOC ',XP1MIN,IXPMIN,IYPMIN,IZPMIN
        WRITE(6,*)' U1MAX LOC ',XU1MAX,IXUMAX,IYUMAX,IZUMAX
        WRITE(6,*)' U1MIN LOC ',XU1MIN,IXUMIN,IYUMIN,IZUMIN
        WRITE(6,*)' V1MAX LOC ',XV1MAX,IXVMAX,IYVMAX,IZVMAX
        WRITE(6,*)' V1MIN LOC ',XV1MIN,IXVMIN,IYVMIN,IZVMIN
        WRITE(6,*)' W1MAX LOC ',XW1MAX,IXWMAX,IYWMAX,IZWMAX
        WRITE(6,*)' W1MIN LOC ',XW1MIN,IXWMIN,IYWMIN,IZWMIN
        WRITE(6,*)' H1MAX LOC ',XH1MAX,IXHMAX,IYHMAX,IZHMAX
        WRITE(6,*)' H1MIN LOC ',XH1MIN,IXHMIN,IYHMIN,IZHMIN
        WRITE(6,*)' T1MAX LOC ',XT1MAX,IXTMAX,IYTMAX,IZTMAX
        WRITE(6,*)' T1MIN LOC ',XT1MIN,IXTMIN,IYTMIN,IZTMIN
        WRITE(6,*)' KEMAX LOC ',XKEMAX,IXKMAX,IYKMAX,IZKMAX
        WRITE(6,*)' KEMIN LOC ',XKEMIN,IXKMIN,IYKMIN,IZKMIN
        WRITE(6,*)' EPMAX LOC ',XEPMAX,IXEMAX,IYEMAX,IZEMAX
        WRITE(6,*)' EPMIN LOC ',XEPMIN,IXEMIN,IYEMIN,IZEMIN
        WRITE(6,*)' ETMAX LOC ',XETMAX,IXXMAX,IYXMAX,IZXMAX
        WRITE(6,*)' ETMIN LOC ',XETMIN,IXXMIN,IYXMIN,IZXMIN
      ENDIF

```

```

      IF(ISWEEP.EQ.FSWEEP+2) NPRMON=INPR
      IF(MOD(ISWEEP,TSTSWP).NE.0) WRITE(6,*)' ISWEEP = ',ISWEEP

```

```

C
C-pd---Printout heat total-----
C

```

```

      IF(ISWEEP.EQ.LSWEEP.OR.MOD(ISWEEP,IG(901)).EQ.0) THEN
        CALL WRITBL
        CALL WRIT1R('Qdot,Tot',QDTTOT)
        CALL WRIT4R(' QTOT 1 ',QDOT01,',QTOT 2 ',QDOT02,
&        ',QTOT 3 ',QDOT03,',QTOT 4 ',QDOT04)
        QDTTOT=0.0
        QDOT01=0.0
        QDOT02=0.0
        QDOT03=0.0
        QDOT04=0.0
        CALL WRITBL
      ENDIF

```

```

C
C-pd---Check to stop run-----
C

```

```

      INQUIRE(FILE='ABORT',EXIST=LSG1)
      IF(LSG1) THEN
        OPEN(91,FILE='ABORT')
        CLOSE(91,STATUS='DELETE')
        LSWEEP=ISWEEP+2
        WRITE(6,*)' ==> ABORT CALLED: STOP IN 2 SWEEPS '
        LSG1=.FALSE.
      ENDIF

```

```

C
C-pd---Modify relaxation without killing run-----

```

C

```

INQUIRE(FILE='RELAXP',EXIST=LSG2)
IF(LSG2) THEN
  OPEN(92,FILE='RELAXP')
  WRITE(6,*)' ==> MODIFYING RELAX P1          OLD VALVE=',
&      DTFALS(P1)
  READ(92,1971)XRELP1
  DTFALS(P1)=XRELP1
  CLOSE(92,STATUS='DELETE')
  WRITE(6,*)' ==>          ISWEEP      &      NEW VALVE=',
&      DTFALS(P1),ISWEEP
  ITST=TSTSWP
  TSTSWP=1
  IOPEN=1
  LSG2=.FALSE.
ENDIF

```

C

```

INQUIRE(FILE='RELAXT',EXIST=LSG3)
IF(LSG3) THEN
  OPEN(93,FILE='RELAXT')
  WRITE(6,*)' ==> MODIFYING RELAX KE & EP      OLD VALVES=',
&      DTFALS(KE),DTFALS(EP)
  READ(93,1972)XRELKE,XRELEP
  DTFALS(KE)=XRELKE
  DTFALS(EP)=XRELEP
  CLOSE(93,STATUS='DELETE')
  WRITE(6,*)' ==>          ISWEEP      &      NEW VALVES=',
&      DTFALS(KE),DTFALS(EP),ISWEEP
  IF(IOPEN.EQ.0) THEN
    ITST=TSTSWP
    TSTSWP=1
    IOPEN=1
  ENDIF
  LSG3=.FALSE.
ENDIF

```

C

```

INQUIRE(FILE='RELAXS',EXIST=LSG4)
IF(LSG4) THEN
  OPEN(94,FILE='RELAXS')
  IF(SOLVE(C2)) THEN
    WRITE(6,*)' ==> MODIFYING RELAX H1 C1 & C2 OLD VALVES=',
&      DTFALS(H1),DTFALS(C1),DTFALS(C2)
    READ(94,1973)XRELH1,XRELC1,XRELC2
    DTFALS(H1)=XRELH1
    DTFALS(C1)=XRELC1
    DTFALS(C2)=XRELC2
    WRITE(6,*)' ==>          ISWEEP      &      NEW VALVES=',
&      DTFALS(H1),DTFALS(C1),DTFALS(C2),ISWEEP
  ELSE
    WRITE(6,*)' ==> MODIFYING RELAX H1 & C1      OLD VALVES=',
&      DTFALS(H1),DTFALS(C1)
    READ(94,1972)XRELH1,XRELC1
    DTFALS(H1)=XRELH1
    DTFALS(C1)=XRELC1
    WRITE(6,*)' ==>          ISWEEP      &      NEW VALVES=',
&      DTFALS(H1),DTFALS(C1),ISWEEP
  ENDIF
  CLOSE(94,STATUS='DELETE')
  IF(IOPEN.EQ.0) THEN
    ITST=TSTSWP

```

```

        TSTSWP=1
        IOPEN=1
    ENDIF
    LSG4=.FALSE.
ENDIF

```

C

```

INQUIRE(FILE='RELAXV',EXIST=LSG5)
IF(LSG5) THEN
    OPEN(95,FILE='RELAXV')
    WRITE(6,*)' ==> MODIFYING RELAX U1 V1 & W1 OLD VALVES=',
    & DTFALS(U1),DTFALS(V1),DTFALS(W1)
    READ(95,1973)XRELU1,XRELV1,XRELW1
    DTFALS(U1)=XRELU1
    DTFALS(V1)=XRELV1
    DTFALS(W1)=XRELW1
    WRITE(6,*)' ==> ISWEEP & NEW VALVES=',
    & DTFALS(U1),DTFALS(V1),DTFALS(W1),ISWEEP
    CLOSE(95,STATUS='DELETE')
    IF(IOPEN.EQ.0) THEN
        ITST=TSTSWP
        TSTSWP=1
        IOPEN=1
    ENDIF
    LSG5=.FALSE.
ENDIF

```

C

```

INQUIRE(FILE='DUMPIT',EXIST=LSG6)
IF(LSG6) THEN
    OPEN(96,FILE='DUMPIT')
    CLOSE(96,STATUS='DELETE')
    CALL AUTCHA(ISWEEP)
    LSG6=.FALSE.
ELSEIF (MOD(ISWEEP,IG(902)).EQ.0) THEN
    CALL AUTCHA(ISWEEP)
ENDIF

```

C

```

INQUIRE(FILE='TSTMOD',EXIST=LSG7)
IF(LSG7) THEN
    OPEN(97,FILE='TSTMOD')
    WRITE(6,*)' ==> MODIFYING TSTSWP OLD VALVE=',
    & TSTSWP
    READ(97,1974)TSTSWP
    IF(IOPEN.EQ.0) THEN
        ITST=TSTSWP
        IOPEN=1
    ENDIF
    WRITE(6,*)' ==> ISWEEP & NEW VALVE=',
    & TSTSWP,ISWEEP
    CLOSE(97,STATUS='DELETE')
    LSG7=.FALSE.
ENDIF

```

C

```

INQUIRE(FILE='NPRMOD',EXIST=LSG8)
IF(LSG8) THEN
    OPEN(98,FILE='NPRMOD')
    WRITE(6,*)' ==> MODIFYING NPRMON OLD VALVE=',
    & NPRMON
    READ(98,1974)NPRMON
    WRITE(6,*)' ==> ISWEEP & NEW VALVE=',
    & NPRMON,ISWEEP

```

```
        CLOSE(98,STATUS='DELETE')
        LSG8=.FALSE.
    ENDIF
```

```
C
    INQUIRE(FILE='IGGMOD',EXIST=LSG9)
    IF(LSG9) THEN
        OPEN(99,FILE='IGGMOD')
        WRITE(6,*)' ==> MODIFYING IG(38-41)           OLD VALVES=',
&          IG(38),IG(39),IG(40),IG(41)
        READ(99,1975)IG(38),IG(39),IG(40),IG(41)
        WRITE(6,*)' ==>          ISWEEP           &      NEW VALVES=',
&          IG(38),IG(39),IG(40),IG(41),ISWEEP
        CLOSE(99,STATUS='DELETE')
        LSG9=.FALSE.
    ENDIF
```

```
C
    INQUIRE(FILE='ML2MOD',EXIST=LSG9)
    IF(LSG9) THEN
        OPEN(100,FILE='ML2MOD')
        WRITE(6,*)' ==> MODIFYING IXYZMON2           OLD VALVES=',
&          IXMON2,IYMON2,IZMON2
        READ(100,1976)IXMON2,IYMON2,IZMON2
        WRITE(6,*)' ==>          ISWEEP           &      NEW VALVES=',
&          IXMON2,IYMON2,IZMON2,ISWEEP
        CLOSE(100,STATUS='DELETE')
        LSG9=.FALSE.
    ENDIF
```

```
C
    INQUIRE(FILE='ML3MOD',EXIST=LSG9)
    IF(LSG9) THEN
        OPEN(101,FILE='ML3MOD')
        WRITE(6,*)' ==> MODIFYING IXYZMON3           OLD VALVES=',
&          IXMON3,IYMON3,IZMON3
        READ(101,1976)IXMON3,IYMON3,IZMON3
        WRITE(6,*)' ==>          ISWEEP           &      NEW VALVES=',
&          IXMON3,IYMON3,IZMON3,ISWEEP
        CLOSE(101,STATUS='DELETE')
        LSG9=.FALSE.
    ENDIF
```

```
C
    INQUIRE(FILE='RAXVMD',EXIST=LSG9)
    IF(LSG9) THEN
        OPEN(102,FILE='RAXVMD')
        WRITE(6,*)' ==> READING MODIFICATION FOR RAX VEL '
        READ(102,1971)RAXFTV
        WRITE(6,*)' ==>          ISWEEP           &      FACTOR=',
&          RAXFTV,ISWEEP
        IRAXV=1
        CLOSE(102,STATUS='DELETE')
        LSG9=.FALSE.
    ENDIF
```

```
C
    INQUIRE(FILE='RAXTMD',EXIST=LSG9)
    IF(LSG9) THEN
        OPEN(102,FILE='RAXTMD')
        WRITE(6,*)' ==> READING MODIFICATION FOR RAX TURB '
        READ(102,1971)RAXFTT
        WRITE(6,*)' ==>          ISWEEP           &      FACTOR=',
&          RAXFTT,ISWEEP
        IRAXT=1
```

```

        CLOSE(102,STATUS='DELETE')
        LSG9=.FALSE.
    ENDIF
C
    INQUIRE(FILE='RAXSMD',EXIST=LSG9)
    IF(LSG9) THEN
        OPEN(102,FILE='RAXSMD')
        WRITE(6,*)' ==> READING MODIFICATION FOR RAX SCAL '
        READ(102,1971)RAXFTS
        WRITE(6,*)' ==>          ISWEEP      &          FACTOR=',
    &          RAXFTS,ISWEEP
        IRAXS=1
        CLOSE(102,STATUS='DELETE')
        LSG9=.FALSE.
    ENDIF
C
1971 FORMAT(F12.8)
1972 FORMAT(2F12.8)
1973 FORMAT(3F12.8)
1974 FORMAT(I5)
1975 FORMAT(4I2)
1976 FORMAT(3I3)
C
    RETURN
198 CONTINUE
C * ----- SECTION 8 ---- FINISH OF TIME STEP.
    RETURN
C*****
C
C--- GROUP 20. Preliminary print-out
C
    20 CONTINUE
    RETURN
C*****
C * Make changes for this group only in group 19.
C--- GROUP 21. Print-out of variables
C--- GROUP 22. Spot-value print-out
C*****
C
C--- GROUP 23. Field print-out and plot control
    23 CONTINUE
    RETURN
C*****
C
C--- GROUP 24. Dumps for restarts
C
    24 CONTINUE
    RETURN
    END
C*****
    SUBROUTINE TEMPER(HSTAT,T0,T,CPDR,RGAS,SC,NSC,NFO)
C*****
C TEMPER uses an iterative procedure to calculate temperature
C given H1 and a guess for temperature
C-----
C
    DIMENSION SC(NSC)
    DATA NITER,DT0,TMIN/12,50.,12.345/
C
    CALL ENTHAL(T0,HHH,CPDR,SC,NSC,NFO)

```

```

C
CP=CPDR*RGAS
ENTH=CP*T0
DT=(HSTAT-ENTH)/(CP+1.E-15)
TEMPL=T0
IF(NFO.GE.4) WRITE(6,900) T0,ENTH,HSTAT,RGAS,SC(1),SC(2),SC(3)
TEMP=T0+DT
ITER=0
100 ENTHL=ENTH
ITER=ITER+1
CALL ENTHAL(TEMP,HHH,CPDR,SC,NSC,NFO)
ENTH=CPDR*RGAS*TEMP
RENTH=(HSTAT-ENTHL)/((ENTH-ENTHL)+1.E-9)
IF(NFO.GE.4) WRITE(6,910) ITER,TEMP,ENTH,ENTHL,HSTAT,RENTH
IF(ABS(ENTH-ENTHL).LT..001*ABS(ENTH)) RENTH=1.
TEMP1=TEMPL+(TEMP-TEMPL)*RENTH
TEMP1=AMAX1(TEMP1,.5*TEMP,TMIN)
TEMP1=AMIN1(TEMP1,1.5*TEMP,5000.)
TEMPL=TEMP
TEMP=TEMP1
AR=ABS(RENTH)
IF( (AR.GT.1.005 .OR. AR.LT..995) .AND. ITER.LT.NITER) GO TO 100
T=TEMP
RETURN

```

```

C
900 FORMAT(' T0 E HS RG SC',1P,7E12.4)
910 FORMAT(' IT T E EL HS RE',I3,1P,5E12.4)

```

```

C
END

```

```

C
C*****
C      SUBROUTINE ENTHAL(TEMP,HSUM,CPSUM,SC,N NFO)
C*****
C      ENTHAL calculates H/RT from JANNAF data. The order of
C      species is N O C H.
C-----

```

```

C
      DIMENSION SC(*),ZS(7,2,4)
      DATA ZS/ 0.28532899E+01, 0.16022128E-02, -0.62936893E-06,
&              0.11441022E-09, -0.78057465E-14, -0.89008093E+03,
&              0.63964897E+01, 0.37044177E+01, -0.14218753E-02,
&              0.28670392E-05, -0.12028885E-08, -0.13954677E-13,
&              -0.10640795E+04, 0.22336285E+01,
&              0.36122139E+01, 0.74853166E-03, -0.19820647E-06,
&              0.33749008E-10, -0.23907374E-14, -0.11978151E+04,
&              0.36703307E+01, 0.37837135E+01, -0.30233634E-02,
&              0.99492751E-05, -0.98189101E-08, 0.33031825E-11,
&              -0.10638107E+04, 0.36416345E+01,
&              0.44608041E+01, 0.30981719E-02, -0.12392571E-05,
&              0.22741325E-09, -0.15525954E-13, -0.48961442E+05,
&              -0.98635982E+00, 0.24007797E+01, 0.87350957E-02,
&              -0.66070878E-05, 0.20021861E-08, 0.63274039E-15,
&              -0.48377527E+05, 0.96951457E+01,
&              0.27167633E+01, 0.29451374E-02, -0.80224374E-06,
&              0.10226682E-09, -0.48472145E-14, -0.29905826E 05,
&              0.66305671E+01, 0.40701275E+01, -0.11084499E-02,
&              0.41521180E-05, -0.29637404E-08, 0.80702103E-12,
&              -0.30279722E+05, -0.32270046E+00 /

```

```

C
      K=1

```

```

      IF(TEMP.LT.1000.) K=2
      TEMP2=TEMP*TEMP
      HSUM=0.
      CPSUM=0.
      DO 100 IS=1,NS
      CP1=ZS(1,K,IS)
      CP2=ZS(2,K,IS)*TEMP
      CP3=ZS(3,K,IS)*TEMP2
      CP4=ZS(4,K,IS)*TEMP2*TEMP
      CP5=ZS(5,K,IS)*TEMP2*TEMP2
      CPSUM=CPSUM+SC(IS)*(CP1+CP2+CP3+CP4+CP5)
100 HSUM =HSUM+
      1 SC(IS)*(CP1+.5*CP2+.33333*CP3+.25*CP4+.2*CP5+ZS(6,K,IS)/TEMP)
C
      RETURN
      END
C
C*****
C      SUBROUTINE XGETCV(N,M,C,V)
C*****
C      XGETCV used to set up procedure to get a patch co and val.
C-----
C
C      COMMON/IDATA/IDFIL1(70),NUMREG,IDFIL2(49)
C      COMMON/NPAT/NAMPAT(100)
C      CHARACTER N*(*),NAMPAT*8
C
C      IR=IRPAT(N)
C      CALL XCV(IR,M,C,V)
C
C      RETURN
C      END
C
C*****
C      SUBROUTINE XCV(IR,MPHID,C,V)
C*****
C      XCV used to get a patch co and val.
C-----
C
C      COMMON F(1)
C      COMMON/ICOVL/M04,IOPHI
C      LOGICAL QLT
C      INCLUDE 'SATEAR'
C
C      MPHI=MPHID
C      IO=0
C      IF(EARTH) IO=IORTCV
C      IF(QLT(F(IO+10*IR-8),23.0).AND.MPHI.LE.2) MPHI=MPHI+8
C      IOPHI=IORCV(MPHI)
C      IF(IOPHI.EQ.IO+NRTC) GO TO 5
C      IOPHI=IO-4
C      DO 2 I=1,NUMREG
C      IOPHI=IO+4
C      IOL=IORCVL(MPHI)
C      IF(EARTH) IOL=IO+IORCVF(MPHI)-4
C      IF(IOPHI.EQ.IOL+4) GO TO 5
C      IF(IABS(IFIX(F(IOPHI+1))).NE.IR) GO TO 2
C      C=F(IOPHI+2)
C      V=F(IOPHI+3)
C      GO TO 7

```

```

C   2   CONTINUE
C   5   C=-999.0
C       V=0.0
C   7   CONTINUE
C
C       RETURN
C       END
C
C*****
C       SUBROUTINE XSETCV(N,M,C,V,CF,VF)
C*****
C       XGETCV used to set up procedure to modify a patch co and val.
C-----
C
C       COMMON/IDATA/IDFIL1(70),NUMREG,IDFIL2(49)
C       COMMON/NPAT/NAMPAT(100)
C       CHARACTER N*(*),NAMPAT*8
C
C       IR=IRPAT(N)
C       CALL XSCV(IR,M,C,V,CF,VF)
C
C
C       RETURN
C       END
C
C*****
C       SUBROUTINE XSCV(IR,MPHID,C,V,CF,VF)
C*****
C       XCV used to get a patch co and val.
C-----
C
C       COMMON F(1)
C       COMMON/ICOVL/MO4,IOPHI
C       LOGICAL QLT
C       INCLUDE 'SATEAR'
C
C       MPHI=MPHID
C       IO=0
C           IF(EARTH) IO=IORTCV
C           IF(QLT(F(IO+10*IR-8),23.0).AND.MPHI.LE.2) MPHI=MPHI+8
C       IOPHI=IORCV(MPHI)
C           IF(IOPHI.EQ.IO+NRTC) GO TO 5
C       IOPHI=IO-4
C           DO 2 I=1,NUMREG
C       IOPHI=IO+4
C       IOL=IORCVL(MPHI)
C           IF(EARTH) IOL=IO+IORCVF(MPHI)-4
C           IF(IOPHI.EQ.IOL+4) GO TO 5
C           IF(IABS(IFIX(F(IOPHI+1))).NE.IR) GO TO 2
C       C=F(IOPHI+2)
C       V=F(IOPHI+3)
C       WRITE(6,*)' IN SETCV   VAR & OLD VALUES= ',MPHI,C,V
C       F(IOPHI+2)=F(IOPHI+2)*CF
C       F(IOPHI+3)=F(IOPHI+3)*VF
C       C=F(IOPHI+2)
C       V=F(IOPHI+3)
C       WRITE(6,*)' IN SETCV   VAR & NEW VALUES= ',MPHI,C,V
C       GO TO 7
C   2   CONTINUE
C   5   C=-999.0

```

```

C      V=0.0
C      7 CONTINUE
C
C      RETURN
C      END
C
C*****
      SUBROUTINE AUTCHA(ISW)
C*****
C  AUTUCH writes phida file.
C-----
      DIMENSION JDATE(6)
C
      CALL DUMP
C-pd---WARNING:  The following two calls may be machine dependent-----
      CALL IDATE(JDATE(1))
      CALL ITIME(JDATE(4))
      WRITE(6,*)' **** DUMP CALLED ****      ISWEEP=',ISW
      WRITE(6,*)'  DAY  MONTH  YEAR      +++      HOUR  MINUTE  SECOND'
      WRITE(6,1974)JDATE
1974  FORMAT(I4,I6,I8,8X,I6,I7,I8)
C-----
C
      RETURN
      END

```

DISTRIBUTION QUESTIONNAIRE

The Naval Civil Engineering Laboratory is revising its primary distribution lists.

SUBJECT CATEGORIES

1 SHORE FACILITIES

- 1A Construction methods and materials (including corrosion control, coatings)
- 1B Waterfront structures (maintenance/deterioration control)
- 1C Utilities (including power conditioning)
- 1D Explosives safety
- 1E Aviation Engineering Test Facilities
- 1F Fire prevention and control
- 1G Antenna technology
- 1H Structural analysis and design (including numerical and computer techniques)
- 1J Protective construction (including hardened shelters, shock and vibration studies)
- 1K Soil/rock mechanics
- 1L Airfields and pavements
- 1M Physical security

2 ADVANCED BASE AND AMPHIBIOUS FACILITIES

- 2A Base facilities (including shelters, power generation, water supplies)
- 2B Expedient roads/airfields/bridges
- 2C Over-the-beach operations (including breakwaters, wave forces)
- 2D POL storage, transfer, and distribution
- 2E Polar engineering

3 ENERGY/POWER GENERATION

- 3A Thermal conservation (thermal engineering of buildings, HVAC systems, energy loss measurement, power generation)
- 3B Controls and electrical conservation (electrical systems, energy monitoring and control systems)
- 3C Fuel flexibility (liquid fuels, coal utilization, energy from solid waste)

- 3D Alternate energy source (geothermal power, photovoltaic power systems, solar systems, wind systems, energy storage systems)

- 3E Site data and systems integration (energy resource data, integrating energy systems)

- 3F EMCS design

4 ENVIRONMENTAL PROTECTION

- 4A Solid waste management
- 4B Hazardous/toxic materials management
- 4C Wastewater management and sanitary engineering
- 4D Oil pollution removal and recovery
- 4E Air pollution
- 4F Noise abatement

5 OCEAN ENGINEERING

- 5A Seafloor soils and foundations
- 5B Seafloor construction systems and operations (including diver and manipulator tools)
- 5C Undersea structures and materials
- 5D Anchors and moorings
- 5E Undersea power systems, electromechanical cables, and connectors
- 5F Pressure vessel facilities
- 5G Physical environment (including site surveying)
- 5H Ocean-based concrete structures
- 5J Hyperbaric chambers
- 5K Undersea cable dynamics

ARMY FEAP

- BDG Shore Facilities
- NRG Energy
- ENV Environmental/Natural Responses
- MGT Management
- PRR Pavements/Railroads

TYPES OF DOCUMENTS

D - Techdata Sheets; R - Technical Reports and Technical Notes; G - NCEL Guides and Abstracts; I - Index to TDS; U - User Guides; ☐ None - remove my name

Old Address:

Telephone No.:

New Address:

Telephone No.:

INSTRUCTIONS

The Naval Civil Engineering Laboratory has revised its primary distribution lists. To help us verify our records and update our data base, please do the following:

- Add - circle number on list
- Remove my name from all your lists - check box on list.
- Change my address - line out incorrect line and write in correction (DO NOT REMOVE LABEL).
- Number of copies should be entered after the title of the subject categories you select.
- Are we sending you the correct type of document? If not, circle the type(s) of document(s) you want to receive listed on the back of this card.

Fold on line, staple, and drop in mail.

DEPARTMENT OF THE NAVY
Naval Civil Engineering Laboratory
560 Laboratory Drive
Port Hueneme CA 93043-4328

Official Business
Penalty for Private Use, \$300

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 12503 WASH D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

COMMANDING OFFICER
CODE L34
560 LABORATORY DRIVE
NAVAL CIVIL ENGINEERING LABORATORY
PORT HUENEME CA 93043-4328

NCEL DOCUMENT EVALUATION

You are number one with us; how do we rate with you?

We at NCEL want to provide you our customer the best possible reports but we need your help. Therefore, I ask you to please take the time from your busy schedule to fill out this questionnaire. Your response will assist us in providing the best reports possible for our users. I wish to thank you in advance for your assistance. I assure you that the information you provide will help us to be more responsive to your future needs.

R. N. Storer

R. N. STORER, Ph.D., P.E.
Technical Director

DOCUMENT NO. _____ TITLE OF DOCUMENT: _____

Date: _____ Respondent Organization: _____

Name: _____ Activity Code: _____
Phone: _____ Grade/Rank: _____

Category (please check):

Sponsor _____ User _____ Proponent _____ Other (Specify) _____

Please answer on your behalf only; not on your organization's. Please check (use an X) only the block that most closely describes your attitude or feeling toward that statement:

SA Strongly Agree A Agree O Neutral D Disagree SD Strongly Disagree

	SA	A	O	D	SD		SA	A	O	D	SD
1. The technical quality of the report is comparable to most of my other sources of technical information.	()	()	()	()	()	6. The conclusions and recommendations are clear and directly supported by the contents of the report.	()	()	()	()	()
2. The report will make significant improvements in the cost and or performance of my operation.	()	()	()	()	()	7. The graphics, tables, and photographs are well done.	()	()	()	()	()
3. The report acknowledges related work accomplished by others.	()	()	()	()	()						
4. The report is well formatted.	()	()	()	()	()						
5. The report is clearly written.	()	()	()	()	()						

Do you wish to continue getting
NCEL reports?

☐☐

YES

NO

Please add any comments (e.g., in what ways can we improve the quality of our reports?) on the back of this form.

Comments:

Fold on line, staple, and drop in mail.

DEPARTMENT OF THE NAVY
Naval Civil Engineering Laboratory
560 Laboratory Drive
Port Hueneme CA 93043-4328

Official Business
Penalty for Private Use, \$300

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 12503 WASH D.C.

POSTAGE WILL BE PAID BY ADDRESSEE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

COMMANDING OFFICER
CODE L03
560 LABORATORY DRIVE
NAVAL CIVIL ENGINEERING LABORATORY
PORT HUENEME CA 93043-4328