NASA CONTRACTOR
REPORT

NASA CR-184245

MLITEMP - A COMPUTER PROGRAM TO PREDICT THE THERMAL EFFECTS ASSOCIATED WITH HYPERVELOCITY IMPACT DAMAGE TO THE SPACE STATION MLI

By W. K. Rule and V. Giridharan
Department of Engineering Mechanics
University of Alabama, Tuscaloosa

September 1991

Final Report

Prepared for
NASA, George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1991	3. REPORT TYPE AND DATES COVERED NASA Contractor Report

4. TITLE AND SUBTITLE

MLITEMP - A Computer Program to Predict the Thermal Effects Associated with Hypervelocity Impact Damage to Space Station MLI

5. FUNDING NUMBERS

NAS8-38555

6. AUTHOR(S)

W. K. Rule and V. Giridharan

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Department of Engineering Mechanics
University of Alabama
Box 870278
Tuscaloosa, AL 35487-0278

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, D.C. 20546

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR-184245

11. SUPPLEMENTARY NOTES

Contract Technical Monitor: Ben Hayashida, NASA, Geo. C. Marshall Space Flight Center, Marshall Space Flight Center, Alabama 35812

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified-Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes a family of user-friendly, DOS PC based, Microsoft BASIC programs written to provide spacecraft designers with empirical predictions of space debris damage to orbiting spacecraft. Spacecraft wall temperatures and condensate formation is also predicted. The spacecraft wall configuration is assumed to consist of multilayer insulation (MLI) placed between a Whipple style bumper and the pressure wall. Impact damage predictions are based on data sets of experimental results obtained from simulating debris impacts on spacecraft using light gas guns on earth. A module of the program facilitates the creation of the database of experimental results that is used by the damage prediction modules of the code. The user has the choice of three different prediction modules to predict damage to the bumper, the MLI, and the pressure wall. A finite difference technique is used to predict temperature distributions in the pressure wall, the MLI, and the bumper. Condensate layer thickness is predicted for the case where the pressure wall temperature drops below the dew point temperature of the spacecraft atmosphere.

14. SUBJECT TERMS

Multilayer Insulation, Hypervelocity Impact, Thermal Analysis, Condensation

15. NUMBER OF PAGES

79

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT

ACKNOWLEDGMENTS

The authors wish to thank Ben Hayashida, Scott Hill, Pedro Rodriguez, and Paul Thompson of Marshall Space Flight Center for providing experimental data and background information, and for enthusiastically supporting this effort.

. TABLE OF CONTENTS

General Notation

a = perpendicular distance between surfaces for view factor calculation

b = radius of circular radiating area for view factor calculation

c = radius of circular absorbing area for view factor calculation

C_i = coefficients in polynomial or nondimensional prediction functions

C_{ij} = thermal equilibrium influence coefficients

D_i = value of dependent variable (impact damage) at i-th data point

D_{MAX} = maximum diameter of the bumper hole

D_{MIN} = minimum diameter of the bumper hole

D_{MLI} = average diameter of the hole in the MLI

D_p = diameter of projectile

D_{pw} = average diameter of the pressure wall hole.

D_s = bumper stand-off distance

D = estimated value of dependent variable (impact damage) at an interpolation or prediction point

E = elastic modulus of the bumper plate material

F_{A1-A2} = view factor for radiating from circular area A_1 to circular area A_2

F_r = view factor for radiating from ring area to ring area

G = b/a

G_s = solar constant

h = the convective heat transfer coefficient

h_N = effective heat transfer coefficient of the dacron netting

H = c/a

k = in-plane thermal conductivity of a layer

L = the distance the module air travels along the pressure wall before meeting an obstruction

M = number of data points in database or number of material properties in each record of the materials data file

N = number of independent variables (impact parameters); or number of nodes per layer

q_c = the heat flux to the i-th node of the pressure wall

q_i = net heat flux into the i-th node of a layer

q_{in} = heat flux to a layer at radial position r from adjacent layers

q_N = heat flux through a layer of dacron netting

q_{out} = heat flux from a layer at radial position r to adjacent layers

q_r = radiation heat flux

r = radial position

R^2 = coefficient of determination

R_i = distance from i-th data point to interpolation or prediction point

S = length of influence of a data point

t = thickness of a layer

T = temperature in a layer

T_b = bumper thickness

T_i = the temperature of the i-th node

$T_{i,j}$ = temperature of the i-th node of the j-th layer

T_{pw} = pressure wall thickness

T_∞ = the free stream air temperature of the spacecraft module

u_∞ = the velocity of the module air next to the pressure wall

V = projectile velocity

V_s = speed of sound in the bumper material = $\sqrt{E/\rho}$

x = measured value in linear regression equation

$x_{j,i}$ = j-th coordinate (independent variable) of i-th data point

$x_{j,INT}$ = j-th coordinate (independent variable) of interpolation or prediction point

$\Delta x_{j,i} = x_{j,i} - x_{j,INT}$

Δ = radial distance between nodes in a layer

ε = emissivity of a radiating surface

ϕ = impact angle

ρ = mass density of the bumper plate material

σ = Stefan-Boltzmann constant = 5.6697E-8 W/(m^{2}K^4)

θ = weighting factor of a data point

Notation for Condensate Modeling

C_p = specific heat at constant pressure [J/kg K]

h = enthalpy [J/kg]

k = thermal conductivity [W/m K]

p = pressure [Pa]

q = heat flux [W/m^2]

r = radial coordinate [m]

rc = radial distance to center of T-cell [m]

ru = radial distance to center of u-cell [m]

Δr = width of T-cell in radial direction [m]

S_ϕ = source term for dependent variable ϕ

t = time [s]

T = temperature [K]

u = velocity in radial direction [m/s]

Δu = width of u-cell in radial direction [m]

v = velocity in axial direction [m/s]

Δv = width of v-cell in axial direction [m]

z = axial coordinate [m]

Δz = width of T-cell in axial direction

α = thermal diffusivity [m^2/s]

μ = coefficient of dynamic viscosity [N s/m^2]

ν = coefficient of kinematic viscosity [m^2/s]

ϕ = dependent variable

ρ = density [kg/m^3]

Γ = exchange or diffusion coefficient

Subscripts

i .... radial	L .... left face of cell
j .... axial	R .... right face of cell
u .... u-cell	F .... front face of cell
v .... v-cell	A .... aft face of cell

Superscripts

n .... present time level	~ .... upstream value of variable

1. INTRODUCTION

This report describes a computer program called MLITemp that is intended to be a design tool for aerospace engineers. The program first uses empirical equations to predict hypervelocity impact damage to spacecraft due to space debris in earth orbit. A Whipple [1] style of spacecraft wall configuration is assumed as is shown in Fig. 1.1. Then, the program predicts the thermal effects associated with impact damage, including the amount of condensate that would form.

MLITemp is written in Microsoft BASIC and is designed to run on an MS-DOS based personal computer. All of the various capabilities of the MLITemp are linked together in the seamless environment of a pull down menu system. A help file is provided to assist the user with the menu choices. A software user guide is provided in Section 2 of this report.

Three different techniques for empirically predicting the hypervelocity impact damage are provided. An explanation of how each of these techniques functions is provided in Sections 3 through 5. More details on these empirical prediction techniques are published in a recent NASA Technical Memorandum [2].

The theory behind the thermal analysis program is given in Section 6. The thermal analysis methodology that is used in MLITemp was validated by experimental testing [3]. Also, some thermal system parameters derived during the course of this experimental testing are used by MLITemp.

If the pressure wall of the spacecraft drops below the dew point temperature of the spacecraft module air then condensation will tend to occur. Such condensate could be hazardous to electrical equipment and could also promote the formation of mold. MLITemp estimates the volume of condensate that would form. The methodology used to do this is discussed in Section 7.

2. SOFTWARE USER GUIDE

2.1 COMPUTER SYSTEM REQUIREMENTS

The software developed for this project was written using the Microsoft BASIC Professional Development System (BPDS). However, the programs of MLITemp that do not use the menu, window, and mouse toolbox of BPDS can be modified and recompiled using Microsoft QuickBASIC or some other language. An EGA or VGA graphics card and monitor, and an Intel 80286, 80386 or 80486 CPU is required to run the software. A math coprocessor must be available. The software is provided on 360K computer disks.

2.2 PROGRAM AND DATA FILES OF MLITEMP

An annotated listing of the program and data files of MLITemp follows:

MLITEMP.BAS - source code for the main program that runs the other programs (ASCII).

MLITEMP.EXE - compiled version of the main program.

DATABASE.BAS - source code for the database creation program (ASCII).

DATABASE.EXE - compiled version of the database creation program.

DBASEDEL.BAS - source code for the database record deletion program (ASCII).

DBASEDEL.EXE - compiled version of the database record deletion program.

DBASEOUT.BAS - source code for the database viewing program (ASCII).

DBASEOUT.EXE - compiled version of the database viewing program.

EDITIMPA.BAS - source code for the impact parameters editing program which operates on the impact parameters file impact.par (ASCII).

EDITIMPA.EXE - compiled version of the impact parameters editing program.

EDITTHER.BAS - source code for the thermal parameters editing program which operates on the thermal parameters file thermal.par (ASCII).

EDITTHER.EXE - compiled version of the thermal parameters editing program.

INVRMETH.BAS - source code for the inverse R method damage prediction program (ASCII).

INVRMETH.EXE - compiled version of the inverse R method damage prediction program.

POLYMETH.BAS - source code for the polynomial function damage prediction program (ASCII).

POLYMETH.EXE - compiled version of the polynomial function damage prediction program.

NONDIMEN.BAS - source code for the nondimensional function damage prediction program (ASCII).

NONDIMEN.EXE - compiled version of the nondimensional function damage prediction program (ASCII).

PREVNOND.BAS - source code of the program that prompts the user if previously calculated nondimensional function coefficients should be used in the calculations (ASCII).

PREVNOND.EXE - compiled version of the prevnond.bas program.

SHOWIMPA.BAS - source code of the program that displays impact predictions on the screen (ASCII).

SHOWIMPA.EXE - compiled version of the showimpa.bas program.

UPDATE.BAS - source code of the program that updates the thermal parameters file with the latest impact results data (ASCII).

UPDATE.EXE - compiled version of the update.bas program.

THERMAL.BAS - source code of the program that performs the thermal analysis (ASCII).

THERMAL.EXE - compiled version of the thermal.bas program.

CONDEN.BAS - source code of the program that performs the condensation calculations (ASCII).

CONDEN.EXE - compiled version of the program that performs the condensation calculations.

SHOWTEMP.BAS - source code of the program that displays the results of the thermal analysis on the computer screen.

SHOWTEMP.EXE - compiled version of showtemp.bas.

HELP.BAS - source code of the program that displays the help file help.doc.

HELP.EXE - compiled version of the help.bas program.

MATERIAL.DAT - a typical database file of material properties which is used

by the INVRMETH program (ASCII).
MLI.DAT - a typical database file of experimental results (ASCII).
IMPACT.PAR - a typical impact parameters file (ASCII).
THERMAL.PAR - a typical thermal parameters file (ASCII).
NONDIMEN.OLD - a file storing the previously calculated nondimensional function coefficients (ASCII).
UPDATE.PAR - a file storing the coefficients of the diameter ratio function used by the update.exe program (ASCII).
CONDEN.PAR - a data file used for transferring information from the thermal.exe program to the conden.exe program (ASCII).
HELP.DOC - this is the help document displayed by the help.exe program.

2.3 SOFTWARE INSTALLATION AND EXECUTION

The software is installed by first creating a subdirectory on the hard disk and then copying all of the files from the computer disks into that subdirectory. If disk space is a problem then the source code files (*filename*.BAS) need not be copied. The program is started by typing MLITEMP. The options of MLITEMP can be selected from the keyboard or by using the mouse as will now be described.

WARNING - *Be sure you are using the correct units! The correct units for the various data files are given in Section 2.4.*

The standard procedure for running MLITemp is as follows. First, the impact parameters file (IMPACT.PAR) is edited to reflect the desired hypervelocity impact conditions. Then, the hypervelocity impact testing results file is edited if necessary. A typical impact testing results file called MLI.DAT is provided on disk. Next, one of the three impact damage prediction programs (inverse R, polynomial fit, nondimensional function) are run. The thermal parameters file (THERMAL.PAR) is then updated with the impact damage prediction results and possibly edited with respect to other thermal properties. Finally, the thermal analysis program is executed and the results viewed. More details on these procedures are provided below. The procedures described above can be performed by selecting tasks from the menu. The menu can be activated by clicking with the mouse, or by pressing the <ALT> key. Menu commands can be selected by using the mouse, by using the arrow keys and pressing <ENTER>, or by typing the red letter of each command. The menu commands of MLITemp are described below.

MAIN MENU - FILE

ADD TO IMPACT DATA FILE:

This allows the user to add data records to the hypervelocity impact testing results file. This file contains the data used for making empirical predictions of impact damage to the bumper, the multilayer insulation (MLI), and to the pressure wall. This menu pick runs the program DATABASE.EXE. The first record in the impact parameters file, IMPACT.PAR, contains the name of the impact testing results file that will be operated on. This filename can be changed by selecting the **EDIT IMPACT PARAMETERS FILE** menu pick, which is described below. The user can move from edit box to edit box in the editor window by pressing the <TAB> key, by pressing the <ENTER> key, or by pressing the up or down arrow keys. The user can move around within an edit box of an edit window using the <HOME>, <END>, and arrow keys. A button at the bottom of the edit window (add to database, cancel this data entry, exit program) can be activated by pressing <ENTER> after a button has been selected using the <TAB> or arrow keys. The mouse can also be used to move between edit boxes and buttons. A data input window is shown in Fig. 2.3.1. Note that the program automatically inserts defaults for data values that seldom vary.

REMOVE FROM IMPACT DATA FILE:

This allows the user to remove records from the hypervelocity impact testing results file. This file contains the data used for making empirical predictions of impact damage to the bumper, the MLI, and to the pressure wall. This menu pick runs the program DBASEDEL.EXE. The first record in the impact parameters file, IMPACT.PAR, contains the name of the impact testing results file that will be operated on. This filename can be changed by selecting the **EDIT IMPACT PARAMETERS FILE** menu pick, which is described below. A button at the bottom of the edit window (OK to remove, quit) can be activated by pressing <ENTER> after a button has been selected using the <TAB> or arrow keys. The mouse can also be used to move between buttons. The data window of DBASEDEL.EXE is shown in Fig. 2.3.2.

VIEW IMPACT DATA FILE:

This allows the user to view records in the hypervelocity impact testing results file. This file contains the data used for making empirical predictions of impact damage to the bumper, the MLI, and to the pressure wall. This menu pick runs the program DBASEOUT.EXE. The first record in the impact parameters file, IMPACT.PAR, contains the name of the impact testing results file that will be operated on. This filename can be changed by selecting the **EDIT IMPACT PARAMETERS FILE** menu pick, which is described below. A button at the bottom of the edit window (next data record, exit program) can be activated by pressing <ENTER> after a button has been selected using the <TAB> or arrow keys. The mouse can also be used to move between buttons. A typical data record view window is shown in Fig. 2.3.3.

EDIT IMPACT PARAMETERS FILE:

This allows the user to view and edit the impact parameters file, IMPACT.PAR. This menu pick runs the program EDITIMPA.EXE. The user can move from edit box to edit box in the editor window by pressing the <TAB> key, by pressing the <ENTER> key, or by pressing the up or down arrow keys. The user can move around within an edit box of an edit window using the <HOME>, <END>, and arrow keys. A button at the bottom of the edit window (save changes and exit, exit program) can be activated by pressing <ENTER> after a button has been selected using the <TAB> or arrow keys. The mouse can also be used to move between edit boxes and buttons. The impact parameters edit window is shown in Fig. 2.3.4.

EDIT THERMAL PARAMETERS FILE:

This allows the user to view and edit the thermal parameters file, THERMAL.PAR. This menu pick runs the program EDITTHER.EXE. The user can move from edit box to edit box in the editor window by pressing the <TAB> key, by pressing the <ENTER> key, or by pressing the up or down arrow keys. The user can move around within an edit box of an edit window using the <HOME>, <END>, and arrow keys. A button at the bottom of the edit window (next window, save changes and exit, exit program) can be activated by pressing <ENTER> after a button has been selected using the <TAB> or arrow keys. Four windows are required to view all of the thermal parameters. The user can proceed from window to window using the <NEXT WINDOW> button. The mouse can also be used to move between edit boxes and buttons. Fig. 2.3.5 illustrates the four windows of the thermal parameters editing program.

CURRENT DIRECTORY FILENAMES:

This menu pick causes the names of the files in the current directory to be listed on the screen. This may be useful if the user forgets the name of a data file.

DOS SHELL:

This menu causes a DOS shell to be created. This will allow the user to copy files and perform other tasks without leaving the MLITemp program permanently. Entering "exit" causes the DOS shell to close.

EXIT:

This menu pick will end the MLITemp program.

Main Menu - IMPACT

INVERSE R METHOD:

This menu pick will cause a hypervelocity impact damage prediction to be made using the "inverse - R" prediction algorithm (program INVRMETH.EXE is executed). The details of this prediction algorithm are described in Section 3. The empirical prediction is based on experimental data contained in the impact testing results file named in the impact parameters file (IMPACT.PAR). The impact parameters associated with the prediction are contained in file IMPACT.PAR. File IMPACT.PAR can be edited from the **EDIT IMPACT PARAMETERS FILE** menu pick under the **FILE** main menu. The bumper hole major and minor diameters, the MLI hole diameter, and the pressure wall hole diameter are predicted.

POLYNOMIAL FIT METHOD:

This menu pick will cause a hypervelocity impact damage prediction to be made using the "polynomial fit" prediction algorithm (program POLYMETH.EXE is executed). This prediction algorithm is described in Section 4. The empirical prediction is based on experimental data contained in the impact testing results file named in the impact parameters file (IMPACT.PAR). The impact parameters associated with the prediction are contained in file IMPACT.PAR. File IMPACT.PAR can be edited from the **EDIT IMPACT PARAMETERS FILE** menu pick under the **FILE** main menu. The bumper hole major and minor diameters, the MLI hole diameter, and the pressure wall hole diameter are predicted.

NONDIMENSIONAL FUNCTION METHOD:

This menu pick will cause a hypervelocity impact damage prediction to be made using the "nondimensional function" prediction algorithm (program NONDIMEN.EXE is executed). Details of this prediction algorithm are given in Section 5. The empirical prediction is based on experimental data contained in the impact testing results file named in the impact parameters file (IMPACT.PAR). The impact parameters associated with the prediction are contained in file IMPACT.PAR. File IMPACT.PAR can be edited from the **EDIT IMPACT PARAMETERS FILE** menu pick under the **FILE** main menu. The bumper hole major and minor diameters, the MLI hole diameter, and the pressure wall hole diameter are predicted. This program takes a relatively long time to run since 23 nonlinear function coefficients are being fit to the experimental data. At the end of the execution these coefficients are stored in a file named NONDIMEN.OLD. The following scheme was developed to speed up the calculations for the case where there has been no change in the impact testing results file (and thus the function coefficients would not change). Before running the NONDIMEN.EXE program a program called PREVNOND.EXE is run. PREVNOND.EXE prompts the user for whether the old nondimensional function coefficients should be used. If the user picks yes then the contents of file

NONDIMEN.OLD are copied to a file called NONDIMEN.NEW. If the program NONDIMEN.EXE senses the existence of file NONDIMEN.NEW, then no new function coefficients are calculated, and the old function coefficients (that were originally contained in file NONDIMEN.OLD) are used to make the damage predictions.

SHOW CURRENT IMPACT RESULTS:

This menu pick displays the current set of impact predictions (program SHOWIMPA.EXE is run). The impact predictions are stored in a file named in the impact parameters file IMPACT.PAR. Also, SHOWIMPA.EXE is automatically run after each of the damage prediction programs have completed their calculations. Typical output from this program is shown in Fig. 2.3.6. Press <ENTER> to exit from this program.

UPDATE THERMAL PARAMETERS FILE:

This menu pick (which executes program UPDATE.EXE) updates the thermal parameters file, THERMAL.PAR, with the bumper and MLI hole diameters obtained from the most recent run of an impact damage prediction program. UPDATE.EXE performs three operations. First, it determines an average bumper hole diameter by averaging the major and minor bumper hole diameters calculated by an impact prediction program. Then, UPDATE.EXE converts the average diameter from units of inches to units of meters, as required by the thermal analysis program. Finally, at the option of the user, the MLI hole diameter is adjusted with the "diameter ratio" parameter. The diameter ratio parameter is an empirical function of the impact parameters. The diameter ratio parameter is intended to account for the fact that the apparent MLI hole diameter from a thermal analysis context is in general different from the observed MLI hole diameter. The six empirical function coefficients used to calculate the diameter ratio parameter are stored in ASCII file UPDATE.PAR which may be modified by the user. Details on the diameter ratio function are given in Section 6.

MAIN MENU - TEMPERATURE

PERFORM THERMAL CALCULATIONS:

This menu pick runs the thermal analysis program THERMAL.EXE. The theory behind the thermal calculations is described in Section 6. The analysis is based on parameters contained in the thermal parameters file, THERMAL.PAR. The initial values used for the thermal analysis and the results of the thermal analysis are sent to files named in THERMAL.PAR. File THERMAL.PAR can be edited from the **EDIT THERMAL PARAMETERS FILE** menu pick under the **FILE** main menu. If pressure wall temperatures drop below the dew point temperature of the spacecraft module air, then the condensate thickness profile is calculated by program CONDEN.EXE. Details on the condensate calculation program are given in Section 7. The thermal analysis program, THERMAL.EXE, and the condensate analysis program, CONDEN.EXE, communicate to each other by means of the data file CONDEN.PAR. After the thermal and condensate analysis has been completed, the results are graphically illustrated on the screen by program SHOWTEMP.EXE. The results are also written to a file named in THERMAL.PAR. A typical thermal results file is shown in Table 2.3.1.

SHOW CURRENT THERMAL RESULTS:

This menu pick runs program SHOWTEMP.EXE which illustrates the results of thermal and condensate analyses on the computer screen. Color contour plots of the bumper and pressure wall temperature distributions are shown. Also, a cross section through the geometric configuration of the modeled section of the spacecraft wall and the condensate layer (if present) are drawn to scale on the screen. A typical display of results is illustrated in Fig. 2.3.7 (color contours can not be seen in the figure).

MAIN MENU - HELP

VIEW HELP DOCUMENT:

This menu pick will cause program HELP.EXE to run which displays an ASCII file containing instructions on how to use MLITemp. Additional information may be added to this file by using a text editing program if the user so desires.

2.4 DESCRIPTION OF THE DATA FILES OF MLITEMP

File - MATERIAL.DAT

The MATERIAL.DAT file that is provided on disk as an example of a typical materials data file. Any valid DOS name can be used for this file. Thus, the user may have several of this type of data file in a directory for different purposes. A file of this nature is required while running the inverse R program. The materials data file is an ASCII file that can be created and modified using any standard text editor. The format of the file is as follows:

■ *material property 1 name string*
■ *material property 2 name string*
•
•
•
■ *material property M name string*

▶ LISTING OF NAMES OF MATERIAL PROPERTIES TO BE MODELED (MAXIMUM OF 10) (25 CHARACTERS MAX)

(
■ *material 1 name string*
■ *material property 1 for material 1*

```
■ material property 2 for material 1
          .                        ▶ TYPICAL DATA RECORD
          .
          .
■ material property M for material 1
}
          .
          .      ▶ ANY NUMBER OF DATA RECORDS MAY BE USED
          .
```

A material data file provided with the MLITemp software is called MATERIAL.DAT and is reproduced below:

```
Density (lb/in^3)
Elastic Mod. (lb/in^2)
Ultimate Strgth (lb/in^2)
Sp. Heat (BTU/(lb-deg R))
Melting Temp (deg R)
{
1100
 9.780E-2
 1.000E7
 1.600E4
 2.140E-1
 1.680E3
}
{
2219-T87
 1.030E-1
 1.050E7
 6.300E4
 2.050E-1
 1.680E3
}
{
6061-T6
 9.800E-2
 9.900E6
 4.200E4
 2.100E-1
 1.680E3
}
```

The MATERIAL.DAT file listed above is set up to model the material properties: density, elastic modulus, ultimate strength, specific heat, and melting temperature. Other physical properties can be used to a maximum of 10. The units do not have to be included in the material property name string. MATERIAL.DAT contains three records of material data for materials: 1100, 2219-T87, and 6061-T6. The material names are treated as string variables and thus can be any combination of numbers and letters. Any number of records of material data may be included. The order of the material properties must be the same in every record and must be ordered as the material property name strings are listed. For instance, referring to file MATERIAL.DAT, the specific heat of material 2219-T87 is 2.050E-1.

The purpose of the material properties database file is to provide an efficient, yet very flexible scheme for inputting material property data into the inverse R method computer program. The user can easily change the material properties to be modeled without disturbing the database file of experimental results. If the materials used for the projectile, bumper and pressure wall do not vary in the database, then the contents of the material properties database file will have no effect on the damage predicted by the inverse R method program. The polynomial function method program assumes that the material properties do not vary in the database. The nondimensional function method program assumes that the material properties of the projectile and pressure wall do not vary in the database and assumes the bumper is constructed of an aluminum alloy.

HYPERVELOCITY IMPACT TESTING RESULTS FILE

The other database file required for running the impact damage prediction programs is associated with the experimental data. This file can be created (and enlarged) by running the database maintenance programs described in the previous section or it can be created using any standard text editor since it is an ASCII file. This file can be given any valid DOS file name. Currently, up to 100 data records can be placed in this file. The format for this file is as follows:

{

- *test ID number*
- *source of the data*
- *test date*
- *bumper material name - which must be of the same format as that listed in the material data file*
- *bumper thickness (inches)*
- *bumper standoff (inches)*
- *pressure wall material name - which must be of the same format as that listed in the material data file*
- *pressure wall thickness (inches)*
- *projectile material name - which must be of the same format as that listed in the material data file*
- *projectile diameter (inches)*
- *impact angle (degrees) - this is the angle between the normal to the bumper and the line of travel of the projectile*
- *projectile velocity (km/sec)*
- *major axis of bumper hole (inches)*
- *minor axis of bumper hole (inches)*
- *average MLI hole diameter (inches)*
- *average pressure wall hole diameter (inches)*

}

MLI.DAT is provided as an example of an experimental database file. This file is stored on the computer disks. It contains information on the specimens recently used for thermal testing in Sunspot Thermal Vacuum Chamber of MSFC. To help understand the format information given above, the first record of MLI.DAT is presented below for comparison:

```
{
1012
MSFC
05/08/90
6061-T6
 .08
 4
2219-T87
 .125
1100
 .313
 0
 6.72
 .729
 .729
 2.2
 .375
}
```

File - IMPACT.PAR

IMPACT.PAR specifies the following information related to predicting the damage associated with an impact event (required units shown in brackets):

- *filename of the hypervelocity impact testing results file*
- *filename of the material data file*
- *filename of the file to be used to store the output from the impact damage prediction programs - the impact damage prediction programs can be accessed under the IMPACT main menu*
- *bumper material name - which must be of the same format as that listed in the material data file*
- *bumper thickness (inches)*
- *bumper standoff (inches)*
- *pressure wall material name - which must be of the same format as that listed in the material data file*
- *pressure wall thickness (inches)*
- *projectile material name - which must be of the same format as that listed in the material data file*
- *projectile diameter (inches)*
- *impact angle (degrees) - this is the angle between the normal to the bumper and the line of travel of the projectile*
- *projectile velocity (km/sec)*

File - THERMAL.PAR

THERMAL.PAR specifies the following information related to predicting the temperature in the spacecraft walls and also condensate layer thickness (required units shown in brackets):

- *filename of the file to be used to store the output from the thermal analysis program - the thermal analysis program is under the TEMPERATURE main menu*
- *filename of the file uses to store initial temperatures and heat fluxes for the thermal analysis model - if the initial values file does not exist then estimated pressure wall and bumper temperatures will be used*
- *MLI hole diameter (m)*
- *MLI standoff (m) - this is the distance from the outer surface of the pressure wall to the centerline of the MLI blanket*
- *estimated pressure wall temperature (K) - this is only used if the initial values file does not exist*
- *estimated bumper temperature (K) - this is only used if the initial values file does not exist*
- *temperature conversion factor 1 - used for converting from degrees K to degrees F for display purposes*
- *temperature conversion factor 2 - used for converting from degrees K to degrees F for display purposes*
- *number of aluminized MLI layers*
- *radius of the area thermally modeled (m) - uniform temperatures are assumed to exist outside of this area*
- *pressure wall thickness (m)*
- *thickness of an aluminized MLI layer (m)*
- *beta cloth thickness (m)*
- *bumper thickness (m)*
- *bumper standoff (m)*
- *space thermal radiation - influx from far-field radiator (W/m^2) - if the area of interest is facing in the direction of deep space then this parameter should have a magnitude of zero - if the area of interest is facing the sun then this parameter should have a magnitude of 1353 W/m^2 (known as "solar constant", Gs)*
- *pressure wall thermal conductivity (W/mK)*
- *MLI aluminized layer thermal conductivity (W/mK)* dacron netting heat transfer coefficient (W/m^2K) - recent experiments have shown that this parameter should have a value of 1.0687 W/m^2K for baselined Space Station MLI*
- *beta cloth thermal conductivity (W/mK)*
- *bumper thermal conductivity (W/mK)*
- *pressure wall emissivity*
- *aluminized MLI layer emissivity*
- *beta cloth emissivity*
- *bumper emissivity outward - this allows for a special coating on the outside of the bumper*
- *bumper emissivity inward*
- *Stefan-Boltzmann constant = 5.6697E-8 W/m^2K^4*
- *maximum number of global iterations for a given mesh size*
- *convergence factor*
- *initial number of nodes in each layer of the model*
- *final number of nodes in each layer of the model*
- *bumper hole diameter (m)*
- *mean air temperature of the spacecraft module interior (K)*
- *spacecraft module air dew point temperature (K) - typically 290 K*
- *spacecraft wall convective heat transfer coefficient (W/m^2K)*
- *condensate density (kg/m^3)*
- *condensate kinematic viscosity (m^2/s)*

- *condensate thermal conductivity (W/mK)*
- *condensate constant pressure specific heat (J/kgK)*
- *spacecraft module air density (kg/m^3)*
- *spacecraft module air kinematic viscosity (m^2/s)*
- *spacecraft air thermal conductivity (W/mK)*
- *spacecraft air constant pressure specific heat (J/kgK)*

In the next three sections, the three techniques used for predicting impact damage are discussed.

3. The Inverse R Prediction Technique

The usual procedure for making predictions from experimental data is to assume some form for the equation relating the independent variables to the dependent variable. A function of this nature is described in Section 5 of this report. The equation typically contains empirical coefficients, the values of which are determined from a fit to the experimental data [4-9]. The method of least squares (maximizing the coefficient of determination, R^2) is an example of a popular technique for obtaining the coefficients from the experimental data. The final result is a closed form equation for making predictions.

This approach has been found to work very well for many engineering applications, however there are some disadvantages. A suitable form for the prediction equation must be developed. This is often difficult. Incorporating additional independent variables in an existing equation can pose problems. Usually, a well defined procedure for taking into account new experimental data is not put in place. Generally, a single set of empirical coefficients are used to make predictions over a fairly wide range of values of the independent variables. Thus, the best data in a database for making a prediction with a particular set of independent variables may not be used to best advantage. Also, it is usually difficult to assess the accuracy of a particular prediction.

In this section, a new method (called inverse R method) for making empirical predictions based on experimental data is discussed. The method uses a very general form of prediction equation that can be applied in the same manner to all problems. Thus, the user is not required to develop a suitable form for the prediction equation and additional independent variables can be easily incorporated. The new method is designed to work off a database that can be continuously updated as new experimental data becomes available. The method automatically takes advantage of the most appropriate data in the database for a given set of independent variables.

The new technique consists of four main steps which will now be described.

Step 1. Normalize the Independent Variables.

In general, the independent variables will vary greatly in magnitude. In hypervelocity impact work, dimensions can be of order 10 and velocities of order 10^6. The new technique requires that all variables be of the same order of magnitude. This was accomplished by scaling the independent variables such that their mean value was equal to unity. Other scaling methods could perhaps be used to improve the accuracy of this technique. For instance, the variables could be scaled such that predicted values of points in the database more closely match the measured values. This scaling technique was not tested. The dependent variables need not be scaled.

This technique works off a database that can and should be kept updated with the latest experimental data. Thus, the scaling factors will change as time progresses and the size of the database increases.

Step 2. Select a Series of Points in the Data Domain For Interpolation.

Two general requirements for prediction schemes are: the method should be capable of smoothing the data to (hopefully) cancel out the random scatter typically present in experimental measurements, and the technique should allow for making reliable predictions outside of the domain of the measured data. Here, these requirements are satisfied by using the data to make ten interpolations from within the domain of the data, which are then used for predicting the dependent variable at some point of interest. The ten "interpolation" points should provide for sufficient smoothing of the data and also capture the trend characteristics of the data for extrapolation purposes, if an extrapolation is required. The number of interpolation points to use was selected on the basis of trial and error. Note, in some cases extrapolation can produce misleading results regardless of the extrapolation technique used.

Fig. 3.1 provides an illustration of how the interpolation points are selected for a hypothetical case with two independent variables. An identical approach is used for the case of an arbitrary number of independent variables. In Fig. 3.1, the independent variables are in the plane of the page and the dependent variable takes the form of a surface out of the plane of the page.

First, a prediction "vector" is drawn from the origin through the point in the domain where a prediction of the dependent variable is required, which is called the "target" point. Then the "min" and "max" points (Fig. 3.1) are located on the prediction vector by considering the intersection points of perpendiculars from the data points to the prediction vector. The closest intersection point to the origin defines the min point, and that of the farthest, the max point. Ten equally spaced points (interpolation points) on the prediction vector between the min and max point are then used for the next step in the prediction process. If the target point lies between the min and max points then an interpolation is required, otherwise an extrapolation is required.

Step 3. Estimate Values of the Dependent Variable at Interpolation Points.

Next, values for the dependent variable must be estimated at the ten interpolation points. This is done as indicated in the following equation:

$$D = \frac{\sum_{i=1}^{M} \frac{D_i}{R_i^{N-1}}}{\sum_{i=1}^{M} \frac{1}{R_i^{N-1}}} \qquad (3.1)$$

The distances, R_i, are determined by the usual formula for determining the "distance" between two points in an N dimensional space:

$$R_i^2 = \sum_{j=1}^{N} (x_{j,i} - x_{j,INT})^2 \qquad (3.2)$$

where $x_{j,i}$ and $x_{j,INT}$ are the j-th coordinates (bumper thickness and so on) of the data point and the point to be predicted, respectively. The need for scaling the independent variables is evident from considering the form of Eq. (3.2).

The form of Eq. (3.1) will now be considered. It is assumed that if all measured data points are the same "distance" R from an interpolation point then all the measured data should be given equal weight. This situation is illustrated for the case of two independent variables (N = 2) in Fig. 3.2. This can be interpreted as saying that each data point has some "characteristic length of influence", S, that subtends an angle $\theta = S/R = S/R^{N-1}$ as indicated in Fig. 3.2. The θ can be taken as the weighting factor. For the constant R case shown in Fig. 3.2, all data points would be given the same weight. Fig. 3.3 illustrates the case for which the data points are considered to be equally valid (same S), but are located different "distances" from the interpolation point. Here, the weighting factors will be of the form $\theta_i = S/R_i^{N-1}$, and thus data points closer to the interpolation point will be given a higher weight. The value of the dependent variable at the interpolation point can be estimated from $D = \Sigma\theta_i D_i/\Sigma\theta_i$ which leads to Eq. (3.1) and hence this technique is given the name inverse R method. Note that a value for S is not required as it cancels out of the equation.

The three dimensional (three independent variables) application of this procedure leads to equations identical in form to those used for determining view factors in the field of radiation heat transfer [10]. Thus, the rationale for the inverse R method can be interpreted as follows. The measured data points are "radiating" information to the interpolation point. The farther the data point is away, the weaker the "radiation" (lower weight given to the information). In principle, the method can easily be extended to any number of independent variables, N.

Step 4. Fit a Polynomial Through the Interpolation Points and Make Prediction.

The final step in the process involves fitting a polynomial through the ten interpolation points and then using the polynomial to make a prediction of the dependent variable at the target point. The polynomial describes how the dependent variable behaves as a function of distance along the prediction vector. By trial and error it was found that a fourth-order polynomial worked well for this application. The polynomial could be used for interpolation or extrapolation depending on the location of the target point. There would of course be considerably more uncertainty in the prediction for the case of extrapolation. Errors in the ten interpolation points tend to get smoothed by the polynomial.

Because of it's uniqueness, the inverse R method was tested to ensure it would provide reliable predictions. These tests are reported in [2, 11].

4. The Polynomial Function Prediction Technique

In this section the polynomial function prediction technique is described. This method is based on the concepts associated with the finite element method (FEM). In FEM, relatively low order polynomials are used to interpolate the functions of interest (such as displacements, temperatures, and velocities) over a small portion of domain where the function is active called an element. The coefficients of the polynomial are derived from known values of the function of interest at points called nodes on the boundary of the element. For this application, the nodal values of the functions of interest (bumper hole size and so forth) were measured experimentally and are thus known quantities. This technique involves selecting a sufficient number of experimental data (node) points and then determining the coefficients of the polynomial from this data.

Ideally, the nodes "closest" to the prediction point in impact parameter space should be used to evaluate the polynomial coefficients and thus make a prediction. However, the set of closest nodes may not form linearly independent set of data, making it impossible to solve for the polynomial coefficients. Thus, remoter nodes must be considered in an attempt to find a linearly independent set of data. The technique used for selecting remoter nodes is discussed below. In general, the impact parameters will vary greatly in magnitude. In hypervelocity impact work, dimensions can be of order 10 and velocities of order 10^6. This polynomial function approach requires a reasonable scheme for determining "distances" between data points in impact parameter space. This is accomplished in the program by scaling the impact parameters (bumper thicknesses and so on) such that their mean value is equal to unity. Of course, the dependent variables, such as bumper hole size, need not be scaled. Having scaled the independent variables, the usual formula for determining the "distance", R_i, between two points (experimental data point and the prediction or interpolation point) in a multidimensional space can be used:

$$R_i^2 = \sum_{j=1}^{N} (x_{j,i} - x_{j,INT})^2 \qquad (4.1)$$

where $x_{j,i}$ and $x_{j,INT}$ are the j-th coordinates (bumper thickness and so on) of the data point and the point to be predicted, respectively. The need for scaling the independent variables is evident from considering the form of Eq. (4.1).

The form of the polynomial will now be considered. FEM theory dictates that a "complete" polynomial should produce the best results [12]. Here we have six independent variables (bumper thickness and so forth), $x_{j,i}$ ($j = 1$ to 6), associated with the i-th experimental data point to consider. It was decided to use $\Delta x_{j,i}$ ($= x_{j,i} - x_{j,INT}$) values in the polynomial equation to simplify the calculations. The lowest order complete polynomial for this case is:

$$D_i = C_1 + C_2 \bullet \Delta x_{1,i} + C_3 \bullet \Delta x_{2,i} + C_4 \bullet \Delta x_{3,i} + C_5 \bullet \Delta x_{4,i} + C_6 \bullet \Delta x_{5,i} + C_7 \bullet \Delta x_{6,i} \qquad (4.2)$$

Seven linearly independent data points, D_i, are required to determine the seven polynomial coefficients, C_i. Eq. (4.2) allows for a linear variation in damage along each coordinate axis in the design space. Obviously, allowing for a quadratic variation in the damage would provide a much better fit to the data. Unfortunately, a "complete" quadratic function with six variables would require too many linearly independent experimental data points to be of practical use considering the relatively small quantity of experimental data available.

Coefficient C_1 is the prediction of the damage at the point in the design space where the prediction is required, since this is the value of the polynomial (Eq. 4.2) when all $\Delta x_{j,i}$ are set equal to zero. If one or more of the prediction parameters, such as bumper thickness, does not vary in the experimental database file then program POLYMETH will sense this and automatically take that variable or variables out of Eq. (4.2). If one impact parameter does not vary, only six polynomial coefficients need be determined and thus only six linearly independent data points are required.

The method used to select the linearly independent set of data points from the database for determination of the function coefficients, C_i, of Eq. (4.2) will now be discussed. For illustration purposes, assume that three independent variables are active and thus four linearly independent data points are required to fit coefficients C_1 through C_4. First, the four closest data points are selected and tested for linear independence. If they are linearly independent, then the coefficients can be determined and the prediction made. If the four closest data points are not linearly independent, then groups of four data points (the closest data point plus three others) are selected from the closest five data points and tested for linear independence. The first linearly independent set of data points found is used for coefficient determination. If a set of suitable data points is not found, then sets of four data points are selected from the closest six data points and so on.

The number of ways to choose r items from n items, C(n,r), is given by the following equation:

$$C(n,\ r) = \frac{n!}{(n-r)!r!} \qquad (4.3)$$

From Eq. 4.3, there are 20 ways to choose 3 items from 6 items. Thus, as shown in Table 4.1, twenty sets of data would have to be tested for linear independence when selecting four point data sets (the closest plus three other data points) from the closest 7 data points. Note in Table 4.1 that the closest data sets are tested first and data point 1 is always used.

The effectiveness of this prediction technique is tested in [2].

5. The Nondimensional Parameter Prediction Technique

In many applications it has been found that empirical functions are best represented in terms of nondimensional parameters. Reynolds number is an example of a nondimensional parameter that has found widespread use in empirical equations of fluid mechanics. Program NONDIMEN uses a series of empirical functions based on nondimensional parameters of the form given in [13]:

BUMPER HOLE MINIMUM DIAMETER:

$$\frac{D_{MIN}}{D_p} = C_1 \left(\frac{V}{V_s} \right)^{C_2} \left(\frac{T_b}{D_p} \right)^{C_3} \left(\cos \phi \right)^{C_4} + C_5 \qquad (5.1)$$

BUMPER HOLE MAXIMUM DIAMETER:

$$\frac{D_{MAX}}{D_p} = C_6 \left(\frac{V}{V_s} \right)^{C_7} \left(\frac{T_b}{D_p} \right)^{C_8} \left(\cos \phi \right)^{C_9} + C_{10} \qquad (5.2)$$

MLI HOLE DIAMETER:

$$\frac{D_{MLI}}{D_p} = C_{11} \left(\frac{V}{V_s} \right)^{C_{12}} \left(\frac{T_b}{D_p} \right)^{C_{13}} \left(\frac{D_s}{D_p} \right)^{C_{14}} \left(\cos \phi \right)^{C_{15}} + C_{16} \qquad (5.3)$$

PRESSURE WALL AVERAGE HOLE DIAMETER:

$$\frac{D_{pw}}{D_p} = C_{17} \left(\frac{V}{V_s} \right)^{C_{18}} \left(\frac{T_b}{D_p} \right)^{C_{19}} \left(\frac{D_s}{D_p} \right)^{C_{20}} \left(\frac{T_{pw}}{D_p} \right)^{C_{21}} \left(\cos \phi \right)^{C_{22}} + C_{23} \qquad (5.4)$$

The function coefficients were determined using an optimization routine to adjust the values of the coefficients so as to maximize the coefficient of determination (R^2) of each of the functions. Thus, the nondimensional functions were adjusted to match the experimental results as closely as possible in a least squares sense. This approach to coefficient evaluation is suitable for any form of prediction function - linear or nonlinear. The nature of the optimization routine will now be described.

The magnitudes of the function coefficients can vary by several orders of magnitude. To avoid numerical problems it is advisable to work with percentage changes in the function coefficients. This approach also provides a simple way of controlling the amount of change in the function coefficients from one optimization iteration to the next. If the maximum allowable percentage change is too large, the optimizer could thrash back and forth around the optimum design point without ever converging to it. Alternatively, if the maximum allowable percentage change is too small, then it could take an impractical number of iterations to get to the optimum design point, or the optimizer could get "stuck" in a local maximum of the coefficient of determination function before getting to the global maximum.

The maximum allowable percentage change in the nondimensional function coefficient magnitudes used is 1.0 (equivalent to a 100% change). The optimizer is designed to reduce the magnitude of the search domain parameter as the optimization process proceeds. The final value will be 1/100 of the initial value. The idea here is to allow large changes in the design variables initially, to quickly get into the vicinity of the global maximum in the design space, and then use finer steps to precisely locate the global maximum. The user is free to change this parameter to attempt to improve optimization efficiency.

The initial values of the function coefficients are set equal to zero. Optimal values of the function coefficients could be positive, negative or zero.

The method chosen here for search vector selection is based on Powell's method [14]. This is a first order method that does not require the calculation of the gradient vector. Here, Powell's method was modified as follows. Initially, a number of search vectors equal to the number of function coefficients are created. The components of these vectors are random numbers between -1 and +1. The components of each random search vector are then scaled, such that the largest component has a magnitude of unity. These vectors are stored as columns of a "search matrix". Next, the coefficient of determination is evaluated at the current point in the design space and at design points given by +/- the search domain parameter times the first column of the search matrix. If either of the + or - design points has a coefficient of determination greater than that of the current design point, then the design point corresponding to the highest coefficient of determination will become the new design point. Otherwise, the design point does not change. The search vector multiplier (+/- search magnitude parameter or zero) used with the search vector is stored for later use. This procedure is then repeated with the remaining columns of the search matrix.

A new search vector is created after using all of the search vectors in the search matrix. This new vector is created by vectorially adding together all of the search vectors times their search vector multipliers. The new search vector is a vector sum of previous successful search vectors since unsuccessful search vectors have search multipliers of zero. Thus, the new search vector represents (stores) the trend of the optimization process. The new search vector is scaled such that the magnitude of it's largest component is unity and then is used to replace the first column of the search matrix. The procedure is repeated, a new search vector is determined, and then used to replace the second column of the search matrix, and so forth until only the last column of the search matrix remains untouched. Then an entirely new search matrix is created using the random number generator, and the process continues.

If at any time in the iterative process, a new search vector has a magnitude of zero (implying all current search directions are not beneficial), then a new random search matrix is createa immediately. The random number generator uses a seed based on the number of seconds from midnight on the computer's clock. Each successive run of the optimizer will use a different set of search vectors. Currently, the program runs the optimizer two times (each time using different sets of random search vectors) to help ensure that the global maximum of the coefficient of determination has been located in the design space. The number of random search matrices generated in each run is equal to twenty times the number of design variables.

The effectiveness of this prediction technique is described in [2].

6. The Thermal Analysis Program

A numerical model to predict the thermal behavior of impact damaged MLI was developed during this investigation. In this section the theory and assumptions associated with the thermal model are discussed.

The main goal of this project was to develop a microcomputer-based design tool to approximately predict the effects of damage to the MLI of Space Station Freedom. To be suitable as a design tool requires that the program be easy to use and that solution times be minimized to rapidly provide feedback for design studies. These requirements dictated that the numerical model be made as simple as possible while still retaining the capability to provide physically reasonable results.

The numerical model was based on the assumption of axial symmetry about the center of the MLI damage. A finite difference analysis approach was used to discretize the system, where an axially symmetric ring of material can be approximately modeled as a single node as shown in Fig. 6.1. Higher levels of accuracy can be obtained by using more nodes and spacing them closer together. Thus, only a single, radial line of calculation points (nodes) was required for each layer in the thermal system. The numerical model uses the same number of nodes in each layer. The time required to complete a set of calculations increases greatly as more nodes are used.

The computer program is designed to automatically refine the mesh of nodes until further refinement produces no change in the results or until the user specified maximum number of nodes per layer is reached. Each refinement halves the radial spacing between the nodes. The advantage of this refinement process is that a coarse mesh (large node spacing) is used to relatively quickly calculate an accurate set of nodal temperatures and heat fluxes which are then used as initial values for the refined mesh. Accurate initial values for the nodal temperatures and heat fluxes greatly enhance the rate convergence to a solution. An accurate solution can usually be obtained faster using a series of progressively finer meshes than if a single fine mesh is used. Also, the multi-mesh results provide the user with information on the sensitivity of the calculated results to the node spacing.

The pressure wall, the MLI blanket, and the bumper were assumed to radially extend out to infinity. The presence of ring frames and stringers is not modeled. These would be difficult and computationally expensive to model since they would be arbitrarily placed which would destroy the radial symmetry. Accurate studies of the effects of the ring frames and stringers would require a very detailed special purpose thermal model. Such studies are beyond the scope of the design tool under development in this study. However, the presence of the ring frame and stringers was accounted for indirectly during the thermal model parameter calibration process as discussed in [3].

As was discussed, it was assumed that all the MLI consisted of the same number of layers, and that no lap joints were present in the MLI. It was assumed that the damage in the MLI consisted of a circular hole of the same diameter through all of the MLI layers. Deviations from this assumed ideal hole geometry are provided by an experimentally determined parameter called the "diameter ratio" which will be described. Each layer of the MLI is explicitly modeled with an array of nodes. All aluminized layers are modeled in the same fashion. However, the beta cloth layer and the outer kapton layer are modeled as a single layer since they are not separated by a dacron netting spacer. Thus, for the baseline insulation system of the Space Station shown in Fig. 6.2, 22 layers have to be modeled: the pressure wall, 19 aluminized MLI layers, the combined kapton - beta cloth layer, and the bumper layer.

The numerical model was designed to model steady state conditions. Steady state means that the heat flux into each node in the system must equal the heat flux out of that node. Thus, an equation can be written for each node to calculate the nodal temperature such that heat influx will equal heat outflow. The MLI is in a vacuum so the modes of heat transfer are by conduction and radiation. Since radiation heat transfer is a function of temperature to the fourth power, the nodal heat flux equilibrium equations are nonlinear and must be solved by an iterative process. The thermal equilibrium equations are coupled as well - the temperature of a given node depends on the temperature of adjacent nodes in the same layer as well the nodal temperatures of adjacent layers. This complex pattern of heat flow is schematically illustrated in Fig. 6.3.

As adapted from [10] the following equation can be written to describe thermal equilibrium at a node:

$$-\frac{kt}{r}\frac{dT}{dr} - kt\frac{d^2T}{dr^2} = q_{in} - q_{out} = q_i \qquad (6.1)$$

where: k is the in-plane thermal conductivity of the layer, t is the thickness of the layer, r is the radial position of the node, T is temperature in the layer, q_{in} is the heat flux into the layer at position r from adjacent layers, q_{out} is the heat flux out of the layer at position r to the adjacent layers, and q_i is the net flux into the i-th node. Eq. 6.1 basically states that the heat conducted away from a node in the plane of the layer must equal the net influx of heat to that node from adjacent layers.

A standard finite difference approach was used to calculate the temperature derivatives at the i-th node:

$$\frac{dT}{dr} \approx \frac{\dfrac{T_{i+1} - T_i}{\Delta} + \dfrac{T_i - T_{i-1}}{\Delta}}{2} = \frac{T_{i+1} - T_{i-1}}{2\Delta} \qquad (6.2)$$

$$\frac{d^2T}{dr^2} \approx \frac{\dfrac{T_{i+1} - T_i}{\Delta} - \dfrac{T_i - T_{i-1}}{\Delta}}{\Delta} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta^2} \qquad (6.3)$$

where: Δ is the radial distance between nodes, T_i is the temperature of the i-th node, and T_{i-1} and T_{i+1} are the temperatures of the nodes on each side of the i-th node in the layer. The (i-1)-th node is closer to the origin of the coordinate system than the (i+1)-th node.

Note that the (1/r) factor prevents Eq. 6.1 from being used to calculate nodal temperatures at the origin of the coordinate system (r = 0 at node 1) for the pressure wall layer. The same problem occurs for the special case where there is no hole in the MLI, thus the first node of each MLI layer and the bumper is at r = 0. This singularity problem was solved in conjunction with treating the boundary conditions. For the case of a layer with no hole (node one at r = 0) axial symmetry dictates that the in-plane radial heat flux through the origin must be zero. This can be ensured by setting (dT/dr) and (d^2T/dr^2) equal to zero at node 1. Considering the form of Eqs. 6.2 and 6.3, this required setting $T_1 = T_2 = T_3$ and so the temperatures at nodes 1 and 2 were simply set equal to that of node 3. The same approach was also used for the MLI layers for the case where there was a hole in the MLI, since here there was also no radial flux at node 1 because of the presence of the free edge.

The technique used to treat the boundary conditions at the outer edge of the modeled area will now be discussed. The user of the program specifies the radius of the area to be modeled and the number of nodes, N, per layer. The N-th node would be located at on the outer edge of the modeled area. To preserve a type of symmetry in the matrix of governing equations, the computer program automatically adds an (N+1)-th node to each layer. It was assumed that at the N-th node the perturbing effects of the MLI hole have died out. Thus, the radial heat flux would be negligible and the radial temperature profile uniform at the N-th node of each layer. This boundary can be modeled by setting T_N and T_{N+1} equal to T_{N-1}. This same boundary condition would apply if the boundary of the area modeled was aligned with the outer edge of the pressure wall plate, the bumper plate, and the MLI blanket. Thus, the inner and outer boundary conditions for every layer were treated in an identical fashion.

Substituting Eqs. 6.2 and 6.3 into Eq. 6.1 produces the following equation which describes thermal equilibrium at the i-th node of a layer:

$$T_{i-1}\left(\frac{kt}{2r_i\Delta} - \frac{kt}{\Delta^2}\right) + T_i\left(\frac{2kt}{\Delta^2}\right) + T_{i+1}\left(-\frac{kt}{2r_i\Delta} - \frac{kt}{\Delta^2}\right) = q_i \qquad (6.4)$$

Eq. 6.4 can be written in a more compact form as:

$$C_{i1}T_{i-1} + C_{i2}T_i + C_{i3}T_{i+1} = q_i \qquad (6.5)$$

where the C_{ij} ($j = 1$ to 3) are thermal equilibrium influence coefficients for the i-th node of a layer which can be evaluated from Eq. 6.4. Eq. 6.5 can be expanded in matrix fashion to represent an entire layer of nodal temperatures:

$$\begin{bmatrix} C_{22} & & C_{23} & 0 & 0\cdots & & 0 & 0 \\ C_{31} & & C_{32} & C_{33} & 0\cdots & & & 0 \\ 0 & & C_{41} & C_{42} & C_{43} & & \vdots & \vdots \\ 0 & 0 & & & & & 0 & 0 \\ \vdots & \vdots & & & C_{(N-2)1} & C_{(N-2)2} & C_{(N-2)3} & 0 \\ 0 & & & & \cdots 0 & C_{(N-1)1} & C_{(N-1)2} & C_{(N-1)3} \\ 0 & 0 & & & \cdots 0 & 0 & C_{N1} & C_{N2} \end{bmatrix} \begin{Bmatrix} T_2 \\ T_3 \\ T_4 \\ \vdots \\ T_{N-2} \\ T_{N-1} \\ T_N \end{Bmatrix} = \begin{Bmatrix} q_1 - C_{12}T_1 \\ q_2 \\ q_3 \\ \vdots \\ q_{N-2} \\ q_{N-1} \\ q_N - C_{N3}T_{N+1} \end{Bmatrix} \qquad (6.6)$$

Eq. 6.6 consists of a tridiagonal system of equations which is very efficient to solve numerically. Note that T_1 and T_{N+1} are not explicitly solved for, rather they are set equal to T_3 and T_{N-1}, respectively, of the previous iteration as was mentioned in the boundary condition discussion. An iterative procedure is required to solve Eq. 6.6 because the q_i values are complicated nonlinear functions of the nodal temperatures.

The solution procedure consisted of solving Eq. 6.6 for the nodal temperatures of the first layer (pressure wall) and then proceeding to the next layer, solving for the nodal temperatures and so forth, until finally solving for the nodal temperatures of the final layer (bumper). This procedure is repeated until the nodal temperatures converge with respect to a user defined tolerance.

The calculation of the q_i values will now be discussed. The formulas used to calculate the q_i values varied from layer to layer. Accordingly, the method of q_i calculation will be discussed on this basis.

PRESSURE WALL NODAL NET HEAT INFLUX CALCULATIONS

The pressure wall will interact thermally with the atmosphere of the inside of the spacecraft in the form of forced convection heat transfer. The equation describing this process is [10]:

$$q_c = h(T_\infty - T_i) \tag{6.7}$$

where q_c is the convected heat flux to the i-th node of the pressure wall, h is the convective heat transfer coefficient, T_∞ is the free stream air temperature of the spacecraft module, and T_i is the temperature of the i-th pressure wall node. Thus, if the pressure wall is cooler than the module air temperature, then heat from the spacecraft module air will flow into the pressure wall and vice versa.

The convective heat transfer coefficient can be estimated from the following equation which was adapted from information presented in [10]:

$$h \approx 4\left(\frac{u_\infty}{L}\right)^{1/2} \tag{6.8}$$

where u_∞ is the velocity of the module air next to the pressure wall (m/s) and L is the distance (m) that the air travels along the pressure wall before meeting an obstruction such as a ring frame.

The pressure wall radiates heat towards the MLI blanket and into the air of the spacecraft module. This heat flux, q_r, is described by the following equation [10]:

$$q_r = \varepsilon\sigma T_i^4 \tag{6.9}$$

where ε is the emissivity of the radiating surface, σ is the Stefan-Boltzmann constant [5.6697E-8 W/(m^{2}K^4)], and T_i is the temperature of the radiating node. Emissivity values can vary between 0 and 1 depending on the material the surface is made from and the condition of the surface (polished or tarnished and so forth). Emissivity values can vary as a function of time and temperature. In this investigation all emissivities were assumed to be constant.

The pressure wall was exposed to heat flux radiated down from the adjacent MLI layer, and from the bumper and space environment if a MLI hole is present. Not all of the radiation impinging on the pressure wall was absorbed. The fraction absorbed is called the absorptivity. To simplify calculations, the absorptivity is commonly assumed to equal the emissivity [10]. Radiated energy that is not absorbed is reflected. To preserve conservation of energy, the computer model keeps track of the magnitudes of emitted, absorbed, and reflected radiation. Actually, a portion of the radiation striking the pressure wall from the adjacent MLI layer and from the bumper and space environment through the MLI hole is reflected radiation from these layers.

The simplest case of no MLI hole will be considered first. Here, all the nodal temperatures of the MLI layer next to the pressure wall will be identical after equilibrium is attained. Thus, the thermal radiation emitted and reflected will be the same for each node in the MLI layer. Also, no thermal energy from the bumper or space environment will strike the pressure wall. Accordingly, for this simple case, the computer program uses the thermal radiation (both emitted and reflected) from i-th node of the MLI next to the pressure wall when calculating the heat influx to the i-th node of the pressure wall.

The more general case with a hole in the MLI is considerably more complicated. Here, the thermal radiation coming from each node of the MLI layer next to the pressure wall will vary. Also, the thermal radiation from the bumper and space environment will pass through the hole in the MLI and strike the pressure wall plate. The concept of view factors [10] was used to treat this problem.

View factors give the fraction of the thermal radiation given off from a surface that will strike another surface of known geometry and position. Consider Fig. 6.4 where thermal energy is radiating from circular area A_1 to circular area A_2. In Fig. 6.4, the plane of area A_1 is parallel to the plane of area A_2. The view factor associated with this geometry, F_{A1-A2}, is given by [10]:

$$F_{A1-A2} = \frac{1 + G^2 + H^2 - \sqrt{(1 + G^2 + H^2)^2 - 4G^2H^2}}{2G^2} \qquad (6.10)$$

where G = b/a and H = c/a (see Fig. 6.4). Note, for example, that F_{A1-A2} approaches unity as c (and thus G) approaches infinity as one would expect because for this case area A_1 would be radiating into an infinite plane and thus all radiation would be captured.

As is illustrated in Fig. 6.1, the numerical model developed during the course of this investigation is based on the assumption of axial symmetry. Thus, a view factor, F_r, for radiating from ring area to ring area is required here. F_r can be obtained by repeatedly applying Eq. 6.10 (see Fig. 6.5):

$$F_r = \frac{A_{11}(F_{A11-A21} - F_{A11-A22}) - A_{12}(F_{A12-A21} - F_{A12-A22})}{A_{11} - A_{12}} \qquad (6.11)$$

Thus, F_r specifies the fraction of the energy that is radiated by ΔA_1 (= A_{11} - A_{12}) that will strike ΔA_2 (= A_{21} - A_{22}).

The total heat influx to the ring corresponding to the i-th node of the pressure wall from the adjacent MLI layer was calculated by a summation formed from repeatedly using Eq. 6.11 for each node (and thus corresponding ring) of the MLI layer. This is shown schematically in Fig. 6.6. The outer boundary of the ring corresponding to the N-th node of the MLI layer was extended out a large distance beyond the user specified radius of modeled area. This was done to be compatible with the assumption that the layers extend out to infinity in all directions.

The heat flux from the bumper to the pressure wall through the hole in the MLI was treated in two steps. First the ring approach of Eq. 6.11 was used to calculate to total heat flux to the MLI hole from each node on the bumper. For this calculation, A_{22} (Fig. 6.5) was set to zero and A_{21} corresponded to the area of the MLI hole. Also included in this calculation is the thermal radiation from the space environment (space thermal radiation) which would pass through the bumper hole. Then, the thermal energy impinging on the MLI hole was allocated to the pressure wall node under consideration by using Eq. 6.11 again. For this calculation, A_{12} was set to zero and A_{11} was set equal to the area of the MLI hole. This process is illustrated in Fig

6.7.

The sun continuously emits thermal radiation, a small fraction of which strikes the earth. This incident solar radiation flux from the sun has an average magnitude (called solar constant G_s) of approximately 1353 W/m^2 [10]. The thermal analysis program uses a parameter (in file THERMAL.PAR) called space thermal radiation to account for thermal radiation striking the bumper from the space environment. Thus, if the region of the spacecraft wall being modeled directly faces the sun, then the space thermal radiation parameter should be set to 1353 W/m^2. If the modeled region faces deep space then this parameter should be set to 0 W/m^2. A cylindrical module with one side facing the sun and the other facing deep space would experience an average heat flux of 431 W/m^2.

NODAL NET HEAT INFLUX CALCULATIONS FOR MLI LAYER NEXT TO PRESSURE WALL

The MLI layer next to the pressure wall (first MLI layer) can radiate energy to both the pressure wall and the next MLI layer. Thus, the q_r for this layer will be twice that given by Eq. 6.9.

The pressure wall can subject the nearest MLI layer to both emitted and reflected thermal radiation. This was treated in exactly the same way that MLI heat flux impinging on the pressure wall was treated (reverse of Fig. 6.6), which has been discussed. Note that the MLI layer next to the pressure wall is blocked from receiving radiation directly from the bumper or the space environment.

The MLI layer nearest the pressure wall will also be subjected to emitted and reflected radiation from the next MLI layer (second MLI layer). Since the MLI layers are so close to each other a view factor approach of Eq. 6.11 was not used here. The thermal radiation flux from the second MLI layer striking the i-th node of the first MLI layer was assumed to equal the thermal radiation flux from the i-th node of the second MLI layer.

Direct conduction between the first and second MLI layers was inhibited by the presence of a layer of dacron netting. The heat flux to the first MLI layer from the second MLI layer through the dacron netting, q_N, was assumed to be of the following form:

$$q_N = h_N(T_{i,2} - T_{i,1}) \qquad (6.12)$$

where h_N is the effective netting heat transfer coefficient, and $T_{i,1}$ and $T_{i,2}$ are the temperatures of the i-th node of the first and second MLI layers, respectively. A value for h_N equal to 1.0687 W/m^{2}K was determined for the Space Station MLI by fitting the computer model to experimental data as discussed in [3]. It was assumed that the netting heat transfer coefficient was the same for all netting layers.

NODAL NET HEAT INFLUX CALCULATIONS FOR A TYPICAL ALUMINIZED MLI LAYER

Here the net heat influx, q_1, to the nodes of the MLI layers between the the first (next to pressure wall) and last (next to bumper) MLI layers are considered. The q_1 values for the nodes of these layers are calculated in a similar fashion to what was done for the first MLI layer, except here there are two MLI layers radiating into the MLI layer under consideration. No view factor calculations are required here, since the layers are assumed to be close together. Also, there are two layers of dacron netting next to each MLI layer, and thus Eq. 6.12 will have to be applied twice - once for the layer above and once for the layer below.

NODAL NET HEAT INFLUX CALCULATIONS FOR LAYER NEXT TO BUMPER

As was noted previously, the last aluminized MLI layer (closest to bumper) and the beta cloth layer are not separated by a layer of dacron netting (see Fig. 6.2). Accordingly, these layers were analyzed as a single layer with the inside surface having the emissivity of an aluminized layer and the outside surface having the emissivity of the beta cloth layer. The thermal conductivity of the layer was assumed to equal the weighted average (on the basis of thickness) of the two layers. For q_1 calculation purposes, this combined layer was treated in exactly the same manner as the first MLI layer except that here the takes the place of the pressure wall.

NODAL NET HEAT INFLUX CALCULATIONS FOR THE BUMPER LAYER

The net heat influx to the bumper layer was calculated in a very similar manner to that of the pressure wall. The bumper will be subjected to heat influx from the pressure wall through the MLI hole just as the pressure wall was from the bumper. However, unlike the pressure wall, there is no convective heat transfer to the bumper from the spacecraft module air. Instead, the bumper interacts with the thermal radiation from space.

This concludes the discussion of q_1 calculation for the various layers of the thermal system.

Nodal temperatures were calculated layer by layer starting with the pressure wall, and finishing with the bumper. A set of calculations covering all layers once is considered one global iteration. The user sets the number of global iterations in the thermal parameters file THERMAL.PAR.

It typically takes many global iterations until the nodal temperatures converge. After convergence is reached, the program refines the mesh and then starts calculations for the new mesh. If the finest mesh is being used when global convergence occurs, then the program stops. If the maximum allowable number of global iterations (a user input parameter) is used before convergence is obtained, then the program refines the mesh and begins calculations again. If the finest mesh is being used and convergence is not obtained before the maximum allowable number of global iterations has been exhausted, then the program issues a warning and stops.

Ten global iterations are conducted between each check for convergence. Convergence is assessed by calculating magnitude of the change that occurred in the temperatures of the inside and outside edge nodes of the pressure wall and bumper layers during the ten global iterations. The change in temperature is divided by the magnitudes of the temperatures to produce a nondimensional relative temperature change. The relative temperature change is compared with a user input convergence factor stored in the thermal parameters file. If the relative temperature change is less than the convergence factor, then the calculations were considered to have converged.

As has been discussed, the computer program has been designed to automatically refine the mesh by halving the distance between the nodes. The idea is to have coarse meshes provide accurate initial values for successively finer meshes. This serves two purposes: the rate of convergence is enhanced and information on the sensitivity of the calculated results to the mesh density is provided. Ideally, the mesh should be refined until there is an acceptably small change in the calculated results.

A factor called "diameter ratio" was developed in [3] to account for the fact that the apparent MLI hole diameter with respect to thermal behavior tends to be different from the measured MLI hcle diameter. Thus, the diameter ratio accounts for damage effects such as the charring and crinkling of the MLI beyond the edge of the MLI hole. It also indirectly accounts for the fact that the MLI blanket has some thickness. The diameter ratio is defined as the thermally apparent diameter ratio divided by the visually measured diameter ratio. Thus, for thermal calculation purposes, the visually measured MLI hole diameter should be multiplied by the diameter ratio parameter.

The empirical function for diameter ratio that was derived during the course of the investigation reported in [3] is:

$$\text{diameter ratio} = D_p\left[0.1978\ V^{3.466}\ T_b^{5.356}\ D_p^{-7.135}(\cos\ \phi)^{1.694} + 2.575\right] \qquad (6.13)$$

where D_p is the diameter of the projectile, V is the velocity of the projectile, T_b is the thickness of the bumper, and ϕ is the impact angle (see Fig. 1.1). Of course, Eq. 6.13 should only be used to predict diameter ratios for impact conditions similar to those of the investigation reported in [3]. Otherwise, a diameter ratio of unity can be used as an approximation. The form of Eq. 6.13 was derived from the nondimensional prediction equations discussed in Section 5.

7. The Condensate Prediction Program

7.1 INTRODUCTION

The condensation process of water vapor from moist air over a circular surface is studied here. The main objective of the study is to determine the condensate height for a given temperature distribution on the surface. Typically, condensation problems have been dealt with using boundary layer techniques. Two sets of conservation equations are solved: one for the condensate layer and one for the vapor layer, with appropriate interface conditions. But the boundary layer theory breaks down near the center of the circular region [15]. Hence the full Navier-Stokes (momentum conservation) equations are to be considered.

In the following case, the three basic conservation equations of mass, momentum, and energy are solved with changes in thermophysical properties being accounted for with changes in temperature. The height of the condensate is determined from the final temperature distribution over the region. The input parameters required are the radial positions of the nodes along the pressure wall and the corresponding temperatures, the radius of the surface on the pressure wall, the ambient and dew point temperatures of moist air, and the velocity of air over the surface.

7.2 LITERATURE REVIEW

Analyses of laminar film condensation problems have generally been done for flow of a vapor or vapor-gas mixture along horizontal or vertical surfaces [18-25]. All of the studies dealing with vertical surfaces involve gravity as a body force. The case of film condensation on a horizontal flat plate in the absence of gravity has been studied [25]. But in this study, a boundary layer formulation has been used and as explained earlier, this solution breaks down in the vicinity of $r = 0$. It should be noted that in most of the above cases an uniform wall temperature was assumed.

7.3 PHYSICAL MODEL AND COORDINATES

The model for the problem along with the coordinate system is shown in Figure 7.1. The flow of moist air is directed radially away from the center with a velocity v_{inlet}. For an axisymmetric flow situation, the centerline $r = 0$, which forms the left boundary of the domain, is an axis of symmetry. The wall along which the temperature is prescribed as a function of the radial position, $T = T(r)$, forms the front boundary. Since the edge of the circular region is insulated, the temperature equals that of the ambient condition for $r \geq rmax$. The right boundary ($r = rmax$) is an open outflow boundary. For the present case, it is assumed that ambient conditions are achieved at a distance of one radius from the surface in the axial direction, i.e. zmax = rmax. The condensate layer will extend in the radial direction till the point at which the wall temperature exceeds the dew point temperature, while the height in the axial direction will vary with r depending on the value of z at which the vapor reaches its dew point temperature.

7.4 GOVERNING EQUATIONS

The general form of the conservation or transport equation in cylindrical coordinates (for an axisymmetric problem) for any variable ϕ is given by [16]

$$\frac{\partial(\rho\phi)}{\partial t} + \frac{1}{r}\frac{\partial(\rho r u\phi)}{\partial r} + \frac{\partial(\rho v\phi)}{\partial z} = \frac{1}{r}\frac{\partial(r\Gamma_\phi \frac{\partial\phi}{\partial r})}{\partial r} + \frac{\partial(\Gamma_\phi \frac{\partial\phi}{\partial z})}{\partial z} + S_\phi \qquad (7.1)$$

The first term on the left is the unsteady term which accounts for changes in ϕ with respect to time. The second and third terms on the left represent changes in ϕ due to convection. ϕ is called the convected quantity while u and v are the convecting velocities. The first two terms on the right represent changes in ϕ due to diffusion. Γ_ϕ is called the diffusion or exchange coefficient. The last term on the right is called the source term. Effects not represented in the other terms, for example the pressure gradient term, are included in S_ϕ.

For $\phi = 1$, eqn.(7.1) becomes the mass conservation or continuity equation. If ϕ is taken as u or v, we get the momentum conservation or Navier-Stokes equations. If ϕ is the enthalpy h, we get the energy conservation equation. Since $h = C_pT$, where C_p is the specific heat at constant pressure and T is the temperature, the energy conservation equation can be written in terms of T. The conservation equations mainly differ in the form of the diffusion or exchange coefficient and source term S_ϕ as shown in Table 7.1. Here μ is the absolute viscosity, k is the thermal conductivity, and $\frac{D}{Dt}$ is the substantial derivative and is written as

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial r} + v\frac{\partial}{\partial z}$$

For an incompressible flow situation, ρ can be considered to be a constant. Hence the continuity equation is

$$\frac{1}{r}\frac{\partial(ru)}{\partial r} + \frac{\partial v}{\partial z} = 0 \qquad (7.2)$$

The momentum conservation equations are written as

$$r: \frac{\partial u}{\partial t} + \frac{1}{r}\frac{\partial (ruu)}{\partial r} + \frac{\partial (vu)}{\partial z} = \frac{1}{r}\frac{\partial (r\nu\frac{\partial u}{\partial r})}{\partial r} + \frac{\partial (\nu\frac{\partial u}{\partial z})}{\partial z} + \left[-\frac{1}{\rho}\frac{\partial p}{\partial r} + \frac{1}{r}\frac{\partial (r\nu\frac{\partial u}{\partial r})}{\partial r} + \frac{\partial (\nu\frac{\partial v}{\partial r})}{\partial z} - 2\nu\frac{u}{r^2}\right] \qquad (7.3)$$

$$z: \frac{\partial v}{\partial t} + \frac{1}{r}\frac{\partial (ruv)}{\partial r} + \frac{\partial (vv)}{\partial z} = \frac{1}{r}\frac{\partial (r\nu\frac{\partial v}{\partial r})}{\partial r} + \frac{\partial (\nu\frac{\partial v}{\partial z})}{\partial z} + \left[-\frac{1}{\rho}\frac{\partial p}{\partial z} + \frac{1}{r}\frac{\partial (r\nu\frac{\partial u}{\partial z})}{\partial r} + \frac{\partial (\nu\frac{\partial v}{\partial z})}{\partial z}\right] \qquad (7.4)$$

where $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity.

For the energy conservation equation, we write the enthalpy $h= C_pT$. Considering C_p to be constant we have

$$\frac{\partial T}{\partial t} + \frac{1}{r}\frac{\partial (ruT)}{\partial r} + \frac{\partial (vT)}{\partial z} = \frac{1}{r}\frac{\partial (r\alpha\frac{\partial T}{\partial r})}{\partial r} + \frac{\partial (\alpha\frac{\partial T}{\partial z})}{\partial z} + \frac{1}{\rho C_p}\frac{\mathrm{D}p}{\mathrm{D}t} \qquad (7.5)$$

where α is the thermal diffusivity which equals $\frac{k}{\rho C_p}$.

7.5 COMPUTATIONAL PROCEDURE

7.5.1 Description Of The Grid

The computer code for solving the governing equations is based on the marker and cell (MAC) method. This method uses a nonuniform mesh system. The scalar quantities like pressure, temperature, and density are placed at the center of the cells while the velocity components are placed normal to the cell walls as shown in Figure 7.2. This kind of placement of variables is called a staggered mesh system. Hence, there are three different control volumes or cells for u, v, and T which are used in solving the momentum and energy conservation equations, Figure 7.3. The boundaries for the cells are denoted as the left, right, front, and aft faces.

A fictitious layer of cells is added to all four sides of the computational domain. Hence, at any of the boundaries the normal velocity components lie directly on the boundary while the tangential velocity components and any scalar quantities are displaced by half a cell width within the flow domain. The additional layer of cells makes it easy to apply various types of boundary conditions as shown in the next section.

The widths of the T-cell are Δr_i and Δz_j in the radial and axial directions respectively. Since the velocities are displaced half a cell width, the u and v cells used in the computation are displaced by the same distance in the radial and axial directions respectively. The widths of the u-cell and v-cell in the r and z directions respectively are defined as

$$\Delta u_i = \Delta r_i/2 + \Delta r_{i+1}/2$$

$$\Delta v_j = \Delta z_j/2 + \Delta z_{j+1}/2$$

The different dimensions of the cells are shown in Figure 7.3. The distances to the center of the cells are rc_i and zc_j while the distances to the u and v locations are ru_i and zv_j in the radial and axial directions respectively.

7.5.2 Boundary Conditions

It is assumed that ambient conditions are reached at a distance equal one radius from the origin in the axial direction. Accordingly, the dimension of the computational domain in the z-direction is taken to be equal to the radius.

The moist air enters the domain through the aft boundary, i.e. flow is radially away from the center. The front boundary is a rigid no-slip surface and the velocity components are zero on this boundary. The temperature is set according to the given distribution. The left boundary of the domain is the centerline (r=0) and hence a line of symmetry. Thus, the normal velocity component, u, is zero on this boundary while the tangential component, v, and the temperature have zero normal gradient across the boundary. The right boundary of the domain is the outflow boundary and continuative conditions are applied here. The gradients for both the velocity components are set equal to zero. The temperature along the right boundary was set equal to the ambient temperature. The following boundary condition was imposed at the aft (inlet) boundary. The axial velocity v was set equal to the inlet velocity v_{inlet} and the radial velocity u was set equal to zero.

The boundary conditions can be summarized as follows:

<u>Aft Boundary (inlet)</u>

$$v_{i,jmax} = v_{i,jmax-1} = v_{\text{inlet}}$$

$$u_{i,jmax} = -u_{i,jmax-1} \quad (u = 0)$$

$$(T_{i,jmax} + T_{i,jmax-1})/2 = T_{\text{ambient}}$$

or $$T_{i,jmax} = 2(T_{\text{ambient}}) - T_{i,jmax-1}$$

Front Boundary (rigid no-slip)

$$u_{i,1} = -u_{i,2} \qquad (u = v = 0)$$
$$v_{i,1} = -v_{i,2}$$

$$T_{i,1} = 2{*}T(r) - T_{i,2}$$

where $T(r)$ is the given radial temperature distribution.

Left Boundary (symmetry)

$$u_{1,j} = -u_{2,j} \qquad (u = 0)$$
$$v_{1,j} = v_{2,j}$$
$$T_{1,j} = T_{2,j}$$

Right Boundary (outflow)

$$u_{imax,j} = u_{imax-1,j}$$
$$v_{imax,j} = v_{imax-1,j}$$
$$T_{imax,j} = 2(T_{\text{ambient}}) - T_{imax-1,j}$$

The outflow boundary conditions are imposed only after each time step and not after each step of the pressure iteration procedure which is explained later on in the text. This is because the normal velocity at the outflow boundary may vary with changes in pressure.

7.5.3 Solution Method

The steady state problem of condensation in this case is solved using an unsteady time-dependent technique [16]. The solution proceeds by marching forward in time, with the time-marching procedure being continued till there is negligible difference in the values of the variables between two consecutive time steps, i.e. a steady state has been reached.

Since it is an incompressible flow problem, the density is not considered to be a variable but is updated after each time step since it is temperature dependent. The other transport properties like viscosity and thermal diffusivity are also updated after each time step.

The governing equations are solved by means of an explicit finite difference method [16]. The subscripts i and j denote the radial and axial directions respectively while the superscript n stands for time level t. No superscript is used for time level t+Δt. The detailed derivation of the finite difference approximations to the governing equations is given in the next section. The convection terms are discretized using the first upwind differencing scheme while the diffusion terms are evaluated using central differencing. Forward differencing is used for the time derivatives.

At each time step the velocities u and v are calculated explicitly from the momentum conservation equations and the temperature T from the energy conservation equation. The calculated velocities are used as initial values for the next time step. These velocities do not in general satisfy the continuity equation. The reason for this is that the pressure field is not known a priori (an initial guess for the pressure field has to be made at the beginning of the solution) and since velocity is affected by pressure changes, the cell pressures have to be adjusted such that the velocities u and v satisfy the mass conservation equation. This procedure is explained later on in the text.

7.5.4 Finite Difference Approximations To Governing Equations

Following are the finite difference schemes used to evaluate the different terms of the governing equations:

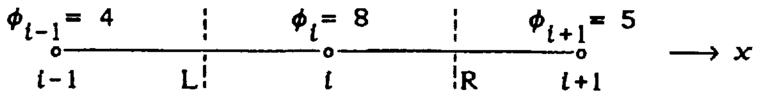
(i) Time Derivatives - Forward Differencing

(ii) Convection Terms - First Upwind Differencing Scheme

(iii) Diffusion Terms - Central Differencing

(iv) Source Terms - Central Differencing

Though central differencing could be used for the convection terms too, it has been found that [26] representing the convected quantity ϕ by central differencing leads to instabilities in the solution. Let us take the simplest example of an uniform one-dimensional mesh with x as the coordinate and a uniform velocity u through the mesh.



Mesh for one dimensional problem

Consider the term $\frac{\partial(u\phi)}{\partial x}$. Using central differencing, we have

$$\phi_R = (\phi_i + \phi_{i+1})/2 \quad ; \quad \phi_L = (\phi_i + \phi_{i-1})/2$$

$$\frac{\partial(u\phi)}{\partial x} \approx \frac{u_R * \phi_R - u_L * \phi_L}{\Delta x_l} = u\frac{(\phi_l + \phi_{l+1})/2 - (\phi_l + \phi_{i-1})/2}{\Delta x_l}$$

$$= u\frac{\phi_{l+1} - \phi_{l-1}}{2\Delta x_l} \qquad \text{(since } u_R = u_L = u\text{)}$$

This implies that changes inside the l^{th} cell are not affected by the value of ϕ_l, but only by those of ϕ_{l+1} and ϕ_{l-1}. It is clear that whatever the values of ϕ_{l+1} and ϕ_{l-1} may be, the value of ϕ_l could still lie outside this range. For example, let $\phi_{l-1}= 4$, $\phi_l= 8$, and $\phi_{l+1}= 5$ as shown in the mesh diagram above. Assuming that no sources or sinks are present, the ϕ field has to be monotonically increasing or decreasing and no sudden jumps are allowed. Hence the above field is not physically plausible.

The easiest alternative to this is the upwind differencing scheme [26] in which the value of ϕ at any cell face is chosen to be the upstream value according to the sign of u at that cell face, i.e.

$$\begin{aligned} \phi_R &= \phi_l \text{ if } u_R > 0 \\ &= \phi_{l+1} \text{ if } u_R < 0 \end{aligned} \qquad ; \qquad \begin{aligned} \phi_L &= \phi_{l-1} \text{ if } u_L > 0 \\ &= \phi_l \text{ if } u_L < 0 \end{aligned}$$

All other quantities whose values are not directly available at the cell faces are obtained by linear interpolation from the adjacent cell values. The thermal conductivity k is represented in a different manner than the other properties as explained below. Again, let us consider the uniform one dimensional mesh. The heat flux at the right face of the cell is given by

$$q_R = k_R\frac{T_l - T_{l+1}}{\Delta u_l}$$

Now consider the cell between the l^{th} and l+1^{th} grid points as a composite slab. Then the heat flux, from basic principles, is

$$q_R = \frac{T_l - T_{l+1}}{\frac{\Delta r_l/2}{k_l} + \frac{\Delta r_{l+1}/2}{k_{l+1}}}$$

Comparing the two expressions for q_R, we get

$$\frac{\Delta u_i}{k_R} = \frac{\Delta r_{i+1}/2}{k_{i+1}} + \frac{\Delta r_i/2}{k_i}$$

or $$k_R = \frac{k_i * k_{i+1} * \Delta u_i}{k_i * \Delta r_{i+1}/2 + k_{i+1} * \Delta r_i/2}$$

We use this form since it gives the exact representation of the heat flux across the cell face.

The r-momentum equation is discretized with respect to the u-cell.

(i) $$\frac{\partial u}{\partial t} \approx \frac{u_{i,j} - u^n_{i,j}}{\Delta t}$$

(ii) (a) $$\frac{1}{r}\frac{\partial(ruu)}{\partial r} \approx \frac{1}{ru_i}\,\frac{rc_{i+1} * u^n_{R,u} * \tilde{u}_R - rc_i * u^n_{L,u} * \tilde{u}_L}{\Delta u_i}$$

$$u^n_{R,u} = (u^n_{i+1,j} + u^n_{i,j})/2 \quad ; \quad u^n_{L,u} = (u^n_{i,j} + u^n_{i-1,j})/2$$

$$\begin{aligned} \tilde{u}_R &= u^n_{i,j} \text{ if } u^n_{R,u} > 0 & \qquad \tilde{u}_L &= u^n_{i-1,j} \text{ if } u^n_{L,u} > 0 \\ &= u^n_{i+1,j} \text{ if } u^n_{R,u} < 0 & ; \qquad &= u^n_{i,j} \text{ if } u^n_{L,u} < 0 \end{aligned}$$

(b) $$\frac{\partial(vu)}{\partial z} \approx \frac{v^n_{A,u} * \tilde{u}_A - v^n_{F,u} * \tilde{u}_F}{\Delta z_j}$$

$$v^n_{A,u} = \frac{v^n_{i,j} * \Delta r_{i+1}/2 + v^n_{i+1,j} * \Delta r_i/2}{\Delta u_i}$$

$$v^n_{F,u} = \frac{v^n_{i,j-1} * \Delta r_{i+1}/2 + v^n_{i+1,j-1} * \Delta r_i/2}{\Delta u_i}$$

$$\begin{aligned} \tilde{u}_A &= u^n_{i,j} \text{ if } v^n_{A,u} > 0 & \qquad \tilde{u}_F &= u^n_{i,j-1} \text{ if } v^n_{F,u} > 0 \\ &= u^n_{i,j+1} \text{ if } v^n_{A,u} < 0 & ; \qquad &= u^n_{i,j} \text{ if } v^n_{F,u} < 0 \end{aligned}$$

(iii) (a) $$\frac{1}{r}\frac{\partial}{\partial r}(r\nu\frac{\partial u}{\partial r}) \approx \frac{1}{ru_l}\,\frac{r\nu\frac{\partial u}{\partial r}\big|_{R} - r\nu\frac{\partial u}{\partial r}\big|_{L}}{\Delta u_l}$$

$$= \frac{rc_{l+1}{*}\ \nu^{n}_{R,u}\left(\frac{u^{n}_{l+1,j} - u^{n}_{l,j}}{\Delta r_{l+1}}\right) - rc_{l}{*}\ \nu^{n}_{L,u}\left(\frac{u^{n}_{l,j} - u^{n}_{l-1,j}}{\Delta r_{l}}\right)}{ru_{l}{*}\ \Delta u_{l}}$$

$$\nu^{n}_{R,u} = \nu^{n}_{l+1,j} \quad ; \quad \nu^{n}_{L,u} = \nu^{n}_{l,j}$$

(b) $$\frac{\partial}{\partial z}(\nu\frac{\partial u}{\partial z}) \approx \frac{\nu\frac{\partial u}{\partial z}\big|_{A} - \nu\frac{\partial u}{\partial z}\big|_{F}}{\Delta z_j}$$

$$= \frac{\nu^{n}_{A,u}\left(\frac{u^{n}_{l,j+1} - u^{n}_{l,j}}{\Delta v_{j}}\right) - \nu^{n}_{F,u}\left(\frac{u^{n}_{l,j} - u^{n}_{l,j-1}}{\Delta v_{j-1}}\right)}{\Delta z_j}$$

Values of $\nu^{n}_{A,u}$ and $\nu^{n}_{F,u}$ are obtained by linearly interpolating from the four neighboring ν values.

$$\nu^{n}_{A,u} = \frac{1}{4\Delta u_l \Delta v_j}\Big[\nu^{n}_{l+1,j}\Delta r_l \Delta z_{j+1} + \nu^{n}_{l+1,j+1}\Delta r_l \Delta z_j + \nu^{n}_{l,j}\Delta r_{l+1}\Delta z_{j+1} + \nu^{n}_{l,j+1}\Delta r_{l+1}\Delta z_j\Big]$$

$$\nu^{n}_{F,u} = \frac{1}{4\Delta u_l \Delta v_{j-1}}\Big[\nu^{n}_{l+1,j}\Delta r_l \Delta z_{j-1} + \nu^{n}_{l+1,j-1}\Delta r_l \Delta z_j + \nu^{n}_{l,j}\Delta r_{l+1}\Delta z_{j-1} + \nu^{n}_{l,j-1}\Delta r_{l+1}\Delta z_j\Big]$$

(iv) (a) $$-\frac{1}{\rho}\frac{\partial p}{\partial r} \approx -\frac{1}{\rho_u}\frac{p^n_{l+1,j} - p^n_{l,j}}{\Delta u_l}$$

$$\rho_u = \frac{(\rho_l * \Delta r_{l+1}/2 + \rho_{l+1} * \Delta r_l/2)}{\Delta u_l}$$

(b) $$\frac{\partial(\nu\frac{\partial v}{\partial r})}{\partial z} \approx \frac{\nu\frac{\partial v}{\partial r}|_{\mathrm{A}} - \nu\frac{\partial v}{\partial r}|_{\mathrm{F}}}{\Delta z_j}$$

$$= \frac{\nu^n_{\mathrm{A},u}(\frac{v^n_{l+1,j} - v^n_{l,j}}{\Delta u_l}) - \nu^n_{\mathrm{F},u}(\frac{v^n_{l+1,j-1} - v^n_{l,j-1}}{\Delta u_l})}{\Delta z_j}$$

(c) $$-2\nu\frac{u}{r^2} = -2\nu^n_u\frac{u^n_{l,j}}{ru^2_l}$$

$$\nu^n_u = (\nu^n_{l+1,j} + \nu^n_{l,j})/2$$

The z-momentum equation is discretized with respect to the v-cell.

(i) $$\frac{\partial v}{\partial t} \approx \frac{v_{l,j} - v^n_{l,j}}{\Delta t}$$

(ii) (a) $$\frac{1}{r}\frac{\partial(ruv)}{\partial r} \approx \frac{1}{rc_l}\frac{ru_l * u^n_{\mathrm{R},v} * \tilde{v}_{\mathrm{R}} - ru_{l-1} * u^n_{\mathrm{L},v} * \tilde{v}_{\mathrm{L}}}{\Delta r_l}$$

$$u^n_{\mathrm{R},v} = \frac{u^n_{l,j} * \Delta z_{j+1}/2 + u^n_{l,j+1} * \Delta z_j/2}{\Delta v_j}$$

$$u^n_{\mathrm{L},v} = \frac{u^n_{l-1,j} * \Delta z_{j+1}/2 + u^n_{l-1,j+1} * \Delta z_j/2}{\Delta v_j}$$

$$\begin{aligned} \tilde{v}_{\mathrm{R}} &= v^{n}_{l,j} \text{ if } u^{n}_{\mathrm{R},v} > 0 \\ &= v^{n}_{l+1,j} \text{ if } u^{n}_{\mathrm{R},v} < 0 \end{aligned} \quad ; \quad \begin{aligned} \tilde{v}_{\mathrm{L}} &= v^{n}_{l-1,j} \text{ if } u^{n}_{\mathrm{L},v} > 0 \\ &= v^{n}_{l,j} \text{ if } u^{n}_{\mathrm{L},v} < 0 \end{aligned}$$

(b) $$\frac{\partial(vv)}{\partial z} \approx \frac{v^{n}_{\mathrm{A},v} * \tilde{v}_{\mathrm{A}} - v^{n}_{\mathrm{F},v} * \tilde{v}_{\mathrm{F}}}{\Delta v_{j}}$$

$$v^{n}_{\mathrm{A},v} = (v^{n}_{l,j+1} + v^{n}_{l,j})/2 \quad ; \quad v^{n}_{\mathrm{F},v} = (v^{n}_{l,j} + v^{n}_{l,j-1})/2$$

$$\begin{aligned} \tilde{v}_{\mathrm{A}} &= v^{n}_{l,j} \text{ if } v^{n}_{\mathrm{A},v} > 0 \\ &= v^{n}_{l,j+1} \text{ if } v^{n}_{\mathrm{A},v} < 0 \end{aligned} \quad ; \quad \begin{aligned} \tilde{v}_{\mathrm{F}} &= v^{n}_{l,j-1} \text{ if } v^{n}_{\mathrm{F},v} > 0 \\ &= v^{n}_{l,j} \text{ if } v^{n}_{\mathrm{F},v} < 0 \end{aligned}$$

(iii) (a) $$\frac{1}{r}\frac{\partial(rv\frac{\partial v}{\partial r})}{\partial r} \approx \frac{1}{rc_{l}} \frac{rv\frac{\partial v}{\partial r}\big|_{\mathrm{R}} - rv\frac{\partial v}{\partial r}\big|_{\mathrm{L}}}{\Delta r_{l}}$$

$$= \frac{ru_{l} * v^{n}_{\mathrm{R},v}\left(\frac{v^{n}_{l+1,j} - v^{n}_{l,j}}{\Delta u_{l}}\right) - ru_{l-1} * v^{n}_{\mathrm{L},v}\left(\frac{v^{n}_{l,j} - v^{n}_{l-1,j}}{\Delta u_{l-1}}\right)}{rc_{l} * \Delta r_{l}}$$

The values of $v^{n}_{\mathrm{R},v}$ and $v^{n}_{\mathrm{L},v}$ are obtained by linearly interpolating from the four neighboring v values.

$$v^{n}_{\mathrm{R},v} = \frac{1}{4\Delta u_{l}\Delta v_{j}}\Big[v^{n}_{l,j+1}\Delta r_{l+1}\Delta z_{j} + v^{n}_{l+1,j+1}\Delta r_{l}\Delta z_{j} + v^{n}_{l,j}\Delta r_{l+1}\Delta z_{j+1} + v^{n}_{l+1,j}\Delta r_{l}\Delta z_{j+1}\Big]$$

$$v^{n}_{\mathrm{L},v} = \frac{1}{4\Delta u_{l-1}\Delta v_{j}}\Big[v^{n}_{l,j+1}\Delta r_{l-1}\Delta z_{j} + v^{n}_{l-1,j+1}\Delta r_{l}\Delta z_{j} + v^{n}_{l,j}\Delta r_{l-1}\Delta z_{j+1} + v^{n}_{l-1,j}\Delta r_{l}\Delta z_{j+1}\Big]$$

(b) $$\frac{\partial(\nu\frac{\partial v}{\partial z})}{\partial z} \approx \frac{\nu\frac{\partial v}{\partial z}|_{\mathrm{A}} - \nu\frac{\partial v}{\partial z}|_{\mathrm{F}}}{\Delta v_j}$$

$$= \frac{\nu^n_{\mathrm{A},v}(\frac{v^n_{i,j+1} - v^n_{i,j}}{\Delta z_{j+1}}) - \nu^n_{\mathrm{F},v}(\frac{v^n_{i,j} - v^n_{i,j-1}}{\Delta z_j})}{\Delta v_j}$$

$$\nu^n_{\mathrm{A},v} = \nu^n_{i,j+1} \quad ; \quad \nu^n_{\mathrm{F},v} = \nu^n_{i,j}$$

(iv) (a) $$-\frac{1}{\rho}\frac{\partial p}{\partial z} \approx -\frac{1}{\rho_v}\frac{p^n_{i,j+1} - p^n_{i,j}}{\Delta v_j}$$

$$\rho_v = \frac{\rho^n_{i,j}* \Delta z_{j+1}/2 + \rho^n_{i,j+1}* \Delta z_j/2}{\Delta v_j}$$

(b) $$\frac{1}{r}\frac{\partial(r\nu\frac{\partial u}{\partial z})}{\partial r} \approx \frac{1}{rc_i}\frac{r\nu\frac{\partial u}{\partial z}|_{\mathrm{R}} - r\nu\frac{\partial u}{\partial z}|_{\mathrm{L}}}{\Delta r_i}$$

$$= \frac{ru_i* \nu^n_{\mathrm{R},v}(\frac{u^n_{i,j+1} - u^n_{i,j}}{\Delta v_j}) - ru_{i-1}* \nu^n_{\mathrm{L},v}(\frac{u^n_{i-1,j+1} - u^n_{i-1,j}}{\Delta v_j})}{rc_i* \Delta r_i}$$

The energy conservation equation is discretized with respect to the T-cell.

(i) $$\frac{\partial T}{\partial t} \approx \frac{T_{i,j} - T^n_{i,j}}{\Delta t}$$

(ii) (a) $$\frac{1}{r}\frac{\partial(ruT)}{\partial r} \approx \frac{1}{rc_i}\frac{ru_i* u^n_{\mathrm{R},T}* \tilde{T}_{\mathrm{R}} - ru_{i-1}* u^n_{\mathrm{L},T}* \tilde{T}_{\mathrm{L}}}{\Delta r_i}$$

$$u^n_{\mathrm{R},T} = u^n_{i,j} \quad ; \quad u^n_{\mathrm{L},T} = u^n_{i-1,j}$$

$$\begin{array}{lll} \tilde{T}_{\mathrm{R}} = T^{n}_{l,j} \text{ if } u^{n}_{\mathrm{R},T} > 0 & & \tilde{T}_{\mathrm{L}} = T^{n}_{l-1,j} \text{ if } u^{n}_{\mathrm{L},T} > 0 \\ & ; & \\ \quad = T^{n}_{l+1,j} \text{ if } u^{n}_{\mathrm{R},T} < 0 & & \quad = T^{n}_{l,j} \text{ if } u^{n}_{\mathrm{L},T} < 0 \end{array}$$

(b) $$\frac{\partial(vT)}{\partial z} \approx \frac{v^{n}_{\mathrm{A},T} * \tilde{T}_{\mathrm{A}} - v^{n}_{\mathrm{F},T} * \tilde{T}_{\mathrm{F}}}{\Delta z_{j}}$$

$$v^{n}_{\mathrm{A},T} = v^{n}_{l,j} \quad ; \quad v^{n}_{\mathrm{F},T} = v^{n}_{l,j-1}$$

$$\begin{array}{lll} \tilde{T}_{\mathrm{A}} = T^{n}_{l,j} \text{ if } v^{n}_{\mathrm{A},T} > 0 & & \tilde{T}_{\mathrm{F}} = T^{n}_{l,j-1} \text{ if } v^{n}_{\mathrm{F},T} > 0 \\ & ; & \\ \quad = T^{n}_{l,j+1} \text{ if } v^{n}_{\mathrm{A},T} < 0 & & \quad = T^{n}_{l,j} \text{ if } v^{n}_{\mathrm{F},T} < 0 \end{array}$$

(iii) (a) $$\frac{1}{r}\frac{\partial(r\alpha\frac{\partial T}{\partial r})}{\partial r} \approx \frac{1}{rc_{l}} \frac{r\alpha\frac{\partial T}{\partial r}\big|_{\mathrm{R}} - r\alpha\frac{\partial T}{\partial r}\big|_{\mathrm{L}}}{\Delta r_{l}}$$

$$= \frac{ru_{l} * \alpha^{n}_{\mathrm{R},T}\left(\frac{T^{n}_{l+1,j} - T^{n}_{l,j}}{\Delta u_{l}}\right) - ru_{l-1} * \alpha^{n}_{\mathrm{L},T}\left(\frac{T^{n}_{l,j} - T^{n}_{l-1,j}}{\Delta u_{l-1}}\right)}{rc_{l} * \Delta r_{l}}$$

$$\alpha^{n}_{\mathrm{R},T} = \frac{\alpha^{n}_{l,j} * \alpha^{n}_{l+1,j} * \Delta u_{l}}{\alpha^{n}_{l,j} * \Delta r_{l+1}/2 + \alpha^{n}_{l+1,j} * \Delta r_{l}/2}$$

$$\alpha^{n}_{\mathrm{L},T} = \frac{\alpha^{n}_{l,j} * \alpha^{n}_{l-1,j} * \Delta u_{l-1}}{\alpha^{n}_{l,j} * \Delta r_{l-1}/2 + \alpha^{n}_{l-1,j} * \Delta r_{l}/2}$$

(b) $$\frac{\partial(\alpha\frac{\partial T}{\partial z})}{\partial z} \approx \frac{\alpha\frac{\partial T}{\partial z}|_{\mathrm{A}} - \alpha\frac{\partial T}{\partial z}|_{\mathrm{F}}}{\Delta z_{j}}$$

$$= \frac{\alpha^{n}_{\mathrm{A},T}(\frac{T^{n}_{i,j+1} - T^{n}_{i,j}}{\Delta v_{j}}) - \alpha^{n}_{\mathrm{F},T}(\frac{T^{n}_{i,j} - T^{n}_{i,j-1}}{\Delta v_{j-1}})}{\Delta z_{j}}$$

$$\alpha^{n}_{\mathrm{A},T} = \frac{\alpha^{n}_{i,j}*\ \alpha^{n}_{i,j+1}*\ \Delta v_{j}}{\alpha^{n}_{i,j}*\ \Delta z_{j+1}/2 + \alpha^{n}_{i,j+1}*\ \Delta z_{j}/2}$$

$$\alpha^{n}_{\mathrm{F},T} = \frac{\alpha^{n}_{i,j}*\ \alpha^{n}_{i,j-1}*\ \Delta v_{j-1}}{\alpha^{n}_{i,j}*\ \Delta z_{j-1}/2 + \alpha^{n}_{i,j-1}*\ \Delta z_{j}/2}$$

(iv) (a) $$\frac{\partial p}{\partial t} \approx \frac{p_{i,j} - p^{n}_{i,j}}{\Delta t}$$

(b) $$u\frac{\partial p}{\partial r} \approx u_{T}\frac{p_{\mathrm{R},T} - p_{\mathrm{L},T}}{\Delta r_{i}}$$

$$u_{T} = 0.5*(u_{i,j} + u_{i-1,j})$$

$$p_{\mathrm{R},T} = \frac{p_{i,j}*\ \Delta r_{i+1}/2 + p_{i+1,j}*\ \Delta r_{i}/2}{\Delta u_{i}}$$

$$p_{\mathrm{L},T} = \frac{p_{i,j}*\ \Delta r_{i-1}/2 + p_{i-1,j}*\ \Delta r_{i}/2}{\Delta u_{i-1}}$$

(c) $$v\frac{\partial p}{\partial z} \approx v_{T}\frac{p_{\mathrm{A},T} - p_{\mathrm{F},T}}{\Delta z_{j}}$$

$$v_{T} = 0.5*(v_{i,j} + v_{i,j-1})$$

$$p_{A,T} = \frac{p_{i,j} * \Delta z_{j+1}/2 + p_{i,j+1} * \Delta z_j/2}{\Delta v_j}$$

$$p_{F,T} = \frac{p_{i,j} * \Delta z_{j-1}/2 + p_{i,j-1} * \Delta z_j/2}{\Delta v_{j-1}}$$

The mass conservation equation is discretized with respect to the *T*-cell.

(a) $\frac{1}{r}\frac{\partial(ru)}{\partial r} \approx \frac{1}{rc_i}\frac{ru|_R - ru|_L}{\Delta r_i}$

$$ru|_R = ru_i * u_{i,j} \quad ; \quad ru|_L = ru_{i-1} * u_{i-1,j}$$

(b) $\frac{\partial v}{\partial z} \approx \frac{v_{i,j} - v_{i,j-1}}{\Delta z_j}$

7.5.5 Pressure Iteration Procedure [16]

The discretized form of the mass conservation equation is given by

$$\frac{1}{rc_i}\frac{(ru_i u_{i,j} - ru_{i-1}u_{i-1,j})}{\Delta r_i} + \frac{v_{i,j} - v_{i,j-1}}{\Delta z_j} = 0$$

Since the velocities computed at each time step do not in general satisfy the above equation, let the left hand side be denoted as D_{ij}(not equal to zero). At every time level, the velocities are adjusted by adjusting cell pressures until D_{ij} is sufficiently small for all i and j.

The pressure adjustment is obtained using the momentum equations. The discretized *r*-momentum equation is

$$u_{i,j} = u_{i,j}^n - \frac{\Delta t}{\rho}\frac{p_{i+1,j} - p_{i,j}}{\Delta u_i} + \text{other terms}$$

If $p_{i,j}$ is increased by $\Delta p_{i,j}$ without changing anything else, the adjusted velocity is given by

$$u'_{i,j} = u^{n}_{i,j} - \frac{\Delta t}{\rho}\,\frac{p_{i+1,j} - p_{i,j}}{\Delta u_i} + \frac{\Delta t\ \Delta p_{i,j}}{\rho\ \Delta u_i} + \text{other terms}$$

or $$u'_{i,j} = u_{i,j} + \frac{\Delta t\ \Delta p_{i,j}}{\rho\ \Delta u_i}$$

Similarly $$u'_{i-1,j} = u_{i-1,j} - \frac{\Delta t\ \Delta p_{i,j}}{\rho\ \Delta u_{i-1}}$$

$$v'_{i,j} = v_{i,j} + \frac{\Delta t\ \Delta p_{i,j}}{\rho\ \Delta v_j}$$

$$v'_{i,j-1} = v_{i,j-1} - \frac{\Delta t\ \Delta p_{i,j}}{\rho\ \Delta v_{j-1}}$$

Now let the adjusted velocities satisfy the continuity equation. Then by substituting the expressions for the adjusted velocities in the continuity equation, we get

$$\frac{1}{rc_i}\,\frac{ru_i(u_{i,j} + \frac{\Delta t\ \Delta p_{i,j}}{\rho\Delta u_i}) - ru_{i-1}(u_{i-1,j} - \frac{\Delta t\ \Delta p_{i,j}}{\rho\Delta u_{i-1}})}{\Delta r_i} + \frac{(v_{i,j} + \frac{\Delta t\ \Delta p_{i,j}}{\rho\Delta v_j}) - (v_{i,j-1} - \frac{\Delta t\ \Delta p_{i,j}}{\rho\Delta v_{j-1}})}{\Delta z_j} = 0$$

or $$\frac{1}{rc_i}\,\frac{(ru_i u_{i,j} - ru_{i-1}u_{i-1,j})}{\Delta r_i} + \frac{v_{i,j} - v_{i,j-1}}{\Delta z_j} + \frac{\Delta t\ \Delta p_{i,j}}{\rho}\left\{\frac{1}{rc_i\Delta r_i}\left[\frac{ru_i}{\Delta u_i} + \frac{ru_{i-1}}{\Delta u_{i-1}}\right] + \frac{1}{\Delta z_j}\left[\frac{1}{\Delta v_j} + \frac{1}{\Delta v_{j-1}}\right]\right\} = 0$$

or $$D_{ij} + \frac{\Delta t\ \Delta p_{i,j}}{\rho}\left\{\frac{1}{rc_i\Delta r_i}\left[\frac{ru_i}{\Delta u_i} + \frac{ru_{i-1}}{\Delta u_{i-1}}\right] + \frac{1}{\Delta z_j}\left[\frac{1}{\Delta v_j} + \frac{1}{\Delta v_{j-1}}\right]\right\} = 0$$

Denoting the term in braces as β_{ij} we have

$$\Delta p_{i,j} = - \frac{\rho D_{ij}}{\Delta t \ \beta_{ij}} \tag{7.6}$$

It can be seen that the velocity components in each cell are affected by the pressure adjustments in all the neighboring cells. So the D_{ij} for all the cells need not necessarily be close to zero after just a single step of the pressure iteration procedure. Hence the cell pressures and velocities are adjusted over and over again in an iterative process, the most recent values of velocities being used to calculate D_{ij}. The process is repeated until the D_{ij} for all the cells is less than a specified small number ε. The right hand side of eqn.(7.6) is multiplied by a relaxation factor ω to enhance the convergence of the pressure iteration procedure. The optimum value of the relaxation factor is approximately $\omega_{opt} \cong 1.8$.

The temperature is computed from the energy conservation equation using the velocities satisfying the continuity equation which are obtained after the pressure iteration procedure.

7.6 STABILITY CRITERION

The time step Δt used in the computation cannot be greater than a particular value or the calculations will become unstable. The first restriction is that the fluid cannot flow through more than one cell in one single time step. This is because the finite difference expressions assume mass or momentum fluxes only between adjacent cells. Hence the time step Δt should be less than the minimum time taken for the fluid to pass through one cell, taken over all the cells in the grid.

Therefore, $$\Delta t < \min \left[\frac{\Delta r_i}{|u_{i,j}|}, \frac{\Delta z_j}{|v_{i,j}|} \right]$$

The Δt used for computation is usually taken to be 0.25-0.33 times that obtained from above [16].

The second restriction arises from the fact that the fluid should not diffuse through more than one cell in a single time step. The expression obtained after performing a linear stability analysis [17] is

$$\frac{\nu \Delta t}{\Delta r_i^{\,2}} + \frac{\nu \Delta t}{\Delta z_j^{\,2}} < \frac{1}{2}$$

$$\text{or} \quad \Delta t < \frac{1}{2\nu} \frac{1}{\dfrac{1}{\Delta r_i^{\,2}} + \dfrac{1}{\Delta z_J^{\,2}}}$$

7.7 List of Variables Used In The Code

Symbol — Description

CONDHEIGHT — Subroutine used to calculate condensate heights

thickr# — Condensate height at radial position radpos#

zcond# — Condensate height at location of cell center rc#

CONVCT — Subroutine which evaluates the convection term of the temperature equation: $\frac{1}{r}\frac{\partial(ruT)}{\partial r} + \frac{\partial(vT)}{\partial z}$

uLT#, uRT#, vFT#, vAT# — Velocities $u^n_{\mathrm{L,T}}$, $u^n_{\mathrm{R,T}}$, $v^n_{\mathrm{F,T}}$ and $v^n_{\mathrm{A,T}}$ at the left, right, front, and aft faces of the u-cell respectively

tLwig#, tRwig#, tFwig#, tAwig# — Upstream values of temperature $\tilde{T}_{\mathrm{L}}$, $\tilde{T}_{\mathrm{R}}$, $\tilde{T}_{\mathrm{F}}$ and $\tilde{T}_{\mathrm{A}}$ at the left, right, front, and aft faces of the T-cell respectively

CONVCU — Subroutine which evaluates the convection term of the r-momentum equation: $\frac{1}{r}\frac{\partial(ruu)}{\partial r} + \frac{\partial(vu)}{\partial z}$

uLun#, uRun#, vFun#, vAun# — Convecting velocities $u^n_{\mathrm{L},u}$, $u^n_{\mathrm{R},u}$, $v^n_{\mathrm{F},u}$ and $v^n_{\mathrm{A},u}$ at the left, right, front, and aft faces of the u-cell respectively

uLwig#, uRwig#, uFwig#, uAwig# — Convected velocities $\tilde{u}_{\mathrm{L}}$, $\tilde{u}_{\mathrm{R}}$, $\tilde{u}_{\mathrm{F}}$, and $\tilde{u}_{\mathrm{A}}$ at the left, right, front, and aft faces of the u-cell respectively

CONVCV — Subroutine which evaluates the convection term of the z-momentum equation: $\frac{1}{r}\frac{\partial(ruv)}{\partial r} + \frac{\partial(vv)}{\partial z}$

uLvn#, uRvn#, vFvn#, vAvn# — Convecting velocities $u^n_{\mathrm{L},v}$, $u^n_{\mathrm{R},v}$, $v^n_{\mathrm{F},v}$, and $v^n_{\mathrm{A},v}$ at the left, right, front, and aft faces of the v-cell respectively vLwig#, vRwig#, vFwig#, vAwig# — Convected velocities $\tilde{v}_{\mathrm{L}}$, $\tilde{v}_{\mathrm{R}}$, $\tilde{v}_{\mathrm{F}}$, and $\tilde{v}_{\mathrm{A}}$ at the left, right, front, and aft faces of the v-cell respectively

delt# — Time step, Δt

DIFFNT — Subroutine which computes the diffusion term of the temperature equation: $\frac{1}{r}\frac{\partial(\alpha r\frac{\partial T}{\partial r})}{\partial r} + \frac{\partial(\alpha\frac{\partial T}{\partial z})}{\partial z}$

alphaL#, alphaR#, alphaF#, alphaA# — Thermal diffusivities $\alpha^{n}_{\mathrm{L},T}$, $\alpha^{n}_{\mathrm{R},T}$, $\alpha^{n}_{\mathrm{F},T}$, and $\alpha^{n}_{\mathrm{A},T}$ at the left, right, front, and aft faces of the T-cell respectively

dtrL#, dtrR#, dtzF#, dtzA# — Derivatives $\frac{\partial T}{\partial r}|_{\mathrm{L}}$, $\frac{\partial T}{\partial r}|_{\mathrm{R}}$, $\frac{\partial T}{\partial z}|_{\mathrm{F}}$ and $\frac{\partial T}{\partial z}|_{\mathrm{A}}$ at the left, right, front and aft faces of the T-cell respectively

DIFFNU — Subroutine which computes the diffusion term of the r-momentum equation: $\frac{1}{r}\frac{\partial(\nu r\frac{\partial u}{\partial r})}{\partial r} + \frac{\partial(\nu\frac{\partial u}{\partial z})}{\partial z}$

nuLun#, nuRun#, nuFun#, nuAun# — Kinematic viscosities $\nu^{n}_{\mathrm{L},u}$, $\nu^{n}_{\mathrm{R},u}$, $\nu^{n}_{\mathrm{F},u}$ and $\nu^{n}_{\mathrm{A},u}$ at the left, right, front and aft faces of the u-cell respectively

durL#, durR#, duzF#, duzA# — Derivatives $\frac{\partial u}{\partial r}|_{\mathrm{L}}$, $\frac{\partial u}{\partial r}|_{\mathrm{R}}$, $\frac{\partial u}{\partial z}|_{\mathrm{F}}$ and $\frac{\partial u}{\partial z}|_{\mathrm{A}}$ at the left, right, front and aft faces of the u-cell respectively

DIFFNV — Subroutine which computes the diffusion term of the z-momentum equation: $\frac{1}{r}\frac{\partial(\nu r\frac{\partial v}{\partial r})}{\partial r} + \frac{\partial(\nu\frac{\partial v}{\partial z})}{\partial z}$

nuLvn#, nuRvn#, nuFvn#, nuAvn# — Kinematic viscosities $\nu^{n}_{\mathrm{L},v}$, $\nu^{n}_{\mathrm{R},v}$, $\nu^{n}_{\mathrm{F},v}$ and $\nu^{n}_{\mathrm{A},v}$ at the left, right, front and aft faces of the v-cell respectively

dvrL#, dvrR#, dvzF#, dvzA# — Derivatives $\frac{\partial v}{\partial r}|_{\mathrm{L}}$, $\frac{\partial v}{\partial r}|_{\mathrm{R}}$, $\frac{\partial v}{\partial z}|_{\mathrm{F}}$ and $\frac{\partial v}{\partial z}|_{\mathrm{A}}$ at the left, right, front and aft faces of the v-cell respectively

epst#,epsv#	Convergence criteria for the pressure iteration procedure and temperature field respectively
errt#	Error, (temp-tempn), after each time step
HEATFLUX	Subroutine calculating heat fluxes
hfluxr#	Heat flux at radial position radpos#
qout#	Heat flux at cell center location rc#
iflag%	Parameter used to determine whether outflow boundary conditions for velocity are to be applied or not. iflag% = 0 for pressure iteration when outflow conditions are not applied. iflag% = 1 otherwise
INITIAL	Subroutine used to set the initial conditions for velocities and temperature
INPUTS	Subroutine which reads model parameters from files "thermal.par" and "conden.par"
cph2o#, kh2o#, nuh2o#, rhoh2o#, cpair#, kair#, nuair#, rhoair#	Specific heat at constant pressure, C_p, thermal conductivity, k, kinematic viscosity, ν, and density, ρ of water and moist air respectively
INTERP1	Subroutine used to interpolate temperatures from radial positions radpos# to locations at cell centers rc#
tempr#	Temperature at position radpos#
tw#	Temperature at position rc#
INTERP2	Subroutine used to interpolate condensate heights and heat fluxes from cell center locations rc# to specified radial positions radpos#
OUTPUTS	Subroutine which writes out results to file "conden.par"

PROPERTY	Subroutine defining physical properties at grid points
cp#, k#, rho#, visc#	Specific heat at constant pressure, thermal conductivity, density and kinematic viscosity
p#, pn#	Pressure p at present and previous time levels respectively
PRSITR	Subroutine which adjusts cell pressures and velocities until continuity is satisfied
beta#	Geometric factor β_{ij}
delp#	pressure adjustment Δp
dij#	Left hand side of the mass conservation equation
omega#	relaxation factor ω
REGION	Subroutine which calculates geometric parameters associated with the mesh
delr#, delrb2#, delu#, rc#, ru#, delz#, delzb2#, delv#, zc#, zv#	The distances Δr, $\Delta r/2$, Δu, rc, ru, Δz, $\Delta z/2$, Δv, zc and zv respectively
slope#	Term used to add the effects of convection, diffusion and source terms e.g.

$$T_{i,j} = T^{n}_{i,j} + \Delta\text{t} * \text{slope}_{i,j} \text{ where slope}_{i,j} = \left(\frac{1}{r}\frac{\partial(ruT)}{\partial r} + \frac{\partial(vT)}{\partial z}\right)\Big|^{n} +$$

$$\frac{1}{r}\frac{\partial(\alpha r\frac{\partial T}{\partial r})}{\partial r} + \frac{\partial(\alpha\frac{\partial T}{\partial z})}{\partial z}\Big|^{n} + \frac{1}{\rho C_p}\frac{\text{D}p}{\text{D}\text{t}}\Big|^{n}$$

SOURCT	Subroutine which evaluates the source term of the temperature equation: $\frac{1}{\rho C_p}\frac{\text{D}p}{\text{D}\text{t}}$
dpt#, udpr#, vdpz	$\frac{\partial p}{\partial t}$, $u\frac{\partial p}{\partial r}$ and $v\frac{\partial p}{\partial z}$ respectively
pL#, pR#, pF#, pA#	Pressure values at the left, right, front and aft faces of the T-cell respectively
uT#, vT#	Velocities at the center of the T-cell
SOURCU	Subroutine which evaluates the source term of

the r-momentum equation: $\frac{1}{r}\frac{\partial(\nu r\frac{\partial u}{\partial r})}{\partial r} + \frac{\partial(\nu\frac{\partial v}{\partial r})}{\partial z} - 2\nu\frac{u}{r^2} - \frac{1}{\rho}\frac{\partial p}{\partial r}$

dpr#, dvrF#, dvrA#, $-\frac{1}{\rho}\frac{\partial p}{\partial r}$ and derivatives $\frac{\partial v}{\partial r}|_F$ and $\frac{\partial v}{\partial r}|_A$ at the front and aft faces of the u-cell respectively

rhou# Density at the center of the u-cell

srcu1#, srcu2#, srcu3# $\frac{1}{r}\frac{\partial(\nu r\frac{\partial u}{\partial r})}{\partial r}$, $\frac{\partial(\nu\frac{\partial v}{\partial r})}{\partial z}$ and $-2\nu\frac{u}{r^2}$ respectively

SOURCV Subroutine which evaluates the source term of the z-momentum equation: $\frac{1}{r}\frac{\partial(\nu r\frac{\partial u}{\partial z})}{\partial r} + \frac{\partial(\nu\frac{\partial v}{\partial z})}{\partial z} - \frac{1}{\rho}\frac{\partial p}{\partial z}$

dpz#, duzL#, duzR#, $-\frac{1}{\rho}\frac{\partial p}{\partial z}$ and derivatives $\frac{\partial u}{\partial z}|_L$ and $\frac{\partial u}{\partial z}|_R$ at the left and right faces of the v-cell respectively

rhov# Density at the center of the v-cell

srcv1#, srcv2# $\frac{1}{r}\frac{\partial(\nu r\frac{\partial u}{\partial z})}{\partial r}$ and $\frac{\partial(\nu\frac{\partial v}{\partial z})}{\partial z}$ respectively

tamb#, tdewpt# Ambient and dew point temperatures of the moist air respectively

temp#, tempn# Temperature T at the present and previous time levels respectively

TBOUNDS Subroutine used to set the boundary conditions for temperature

u#, un#, v#, vn# Velocities u and v at the present and previous time levels respectively

vin# Inlet velocity of moist air

VBOUNDS Subroutine which sets the velocity boundary conditions

7.8 Test Cases

To check the condensate model three test cases were run, each with a different pressure wall temperature profile. For all the test cases, the following values were used for the model parameters:

Ambient temperature = 294 K = 21 °C
Dew point temperature = 288 K = 15 °C
Coefficient of heat transfer for moist air = 5 W/(m^2 K)
Radius of pressure wall surface = 1 m
Inlet velocity = 0.005 m/s ≅ 0.015 ft/s

Physical properties of water were prescribed at the dew point temperature while those of moist air were prescribed at the ambient temperature. The temperature and condensate height distribution along the radial position for test cases I, II and III are shown in Figures 7.4, 7.5 and 7.6, and listed in Tables 7.2, 7.3, and 7.4, respectively. The results obtained appear to be physically reasonable. The authors could locate no appropriate experimental data in the literature for comparison with the calculated results.

REFERENCES

1. Whipple, F. L.,"Meteorites and Space Travel," Astronomical Journal, No. 1161, 1947, p. 131.

2. Rule, W. K. and Hayashida, K. B., "Empirical Predictions of Hypervelocity Impact Damage to the Space Station," to appear as a NASA Technical Memorandum, 1991.

3. Rule, W. K. and Hayashida, K. B., "SUNSPOT - A Program to Model the Behavior of Hypervelocity Impact Damaged Multilayer Insulation in the Sunspot Thermal Vacuum Chamber of Marshall Space Flight Center," to appear as a NASA Technical Memorandum, 1991.

4. Schonberg, W. P. and Taylor, R. A., "Penetration and Ricochet Phenomena in Oblique Hypervelocity Impact," AIAA Journal, Vol. 27, May 1989, pp. 639-646.

5. Coronado, A. R., Gibbins, M. N., Wright, M. A. and Stern, P. H., "Space Station Integrated Wall Design and Penetration Damage Control," Boeing Aerospace Company, Seattle, WA, D180-30550-1, July 1987.

6. Fraas, A. P., "Protection of Spacecraft from Meteoroids and Orbital Debris," Oak Ridge National Laboratory, Oak Ridge, TN, ORNL/TM-9904, March 1986.

7. Gehring, J. W.,"Theory of Impact on Thin Targets and Shields and Correlation with Experiment," High-Velocity Impact Phenomena, Kinslow, R. (ed)., 1st ed., Academic Press, New York, 1970, pp. 117-147.

8. Maiden, C. J., Gehring, J. W., and McMillan, A. R., "Investigation of Fundamental Mechanism of Damage to Thin Targets by Hypervelocity Projectiles," GM Defense Research Laboratories, Santa Barbara, CA, GM-DRL-TR-63-225, Sept. 1963.

9. Bouma, D. D., Burkitt, W. C., "Multivariable Analysis of the Mechanics of Penetration of High Speed Particles," Martin Marietta Corporation, NASA CR-664, Dec. 1966.

10. Özişik, M. N., HEAT TRANSFER A Basic Approach, 1st ed., McGraw-Hill Book Company, New York, 1985.

11. Rule, W. K., "An Interpolation/Extrapolation Technique with Application to Hypervelocity Impact of Space Debris," to appear in AIAA J. of Spacecraft and Rockets.

12. Cook, R. D., Malkus, D. S., and Plesha, M. E., Concepts and Applications of Finite Element Analysis, 3rd ed., John Wiley and Sons, New York, 1989, pp. 82-83.

13. Schonberg, W. P., Bean, A. J., Darzi, K. "Hypervelocity Impact Physics," University of Alabama in Huntsville, Final Report for Contract No. NAS8-36955/D.O.16, July 1990.

14. Vanderplaats, G. N., Numerical Optimization Techniques for Engineering Design: With Applications, McGraw-Hill Book Company, New York, 1984, pp. 84-87.

15. Gunaji, M.V, Pederson R. J., and Leslie, I. H., "Numerical Study of Natural Convection Over a Finite Heated Disc," Numerical Heat Transfer, 1990.

16. Sharif, M.A.R., "Accurate Modeling of Transient Three Dimensional Transport Problems," Ph.D. dissertation, Clarkson University, 1988, pp. 6-25.

17. Hoffmann, K. A., Computational Fluid Dynamics for Engineers, Engineering Education System, Austin, Texas, 1989.

18. Tsay, Y.L., Lin, T. F., and Yan, W. M., "Cooling of a Falling Liquid Film Through Interfacial Heat and Mass Transfer," Int. J. Multiphase Flow, Vol. 16, No. 5, pp. 853-865, 1990.

19. Mori, Y. and Hijikata, K., "Free Convective Condensation Heat Transfer With Noncondensable Gas On a Vertical Surface", Int. J. Heat Mass Transfer, Vol. 16, pp. 2229-2240, 1973.

20. Fujii, T. and Uehara, H., "Laminar Filmwise Condensation On a Vertical Surface," Int. J. Heat Mass Transfer, Vol. 15, pp. 217-233, 1972.

21. Denny, V. E., Mills, A. F., and Jusionis, V. J., "Laminar Film Condensation From a Steam-Air Mixture Undergoing Forced Flow Down a Vertical Surface," J. Heat Transfer, pp. 297-304, August 1971.

22. Minkowycz, W. J. and Sparrow, E. M., "Condensation Heat Transfer In the Presence of Noncondensables, Interfacial Resistance, Superheating, Variable Properties and Diffusion," Int. J. Heat Mass Transfer, Vol. 9, pp.1125-1144, 1966.

23. Sparrow, E. M. and Lin, S. H., "Condensation Heat Transfer in the Presence of a Noncondensable Gas," Trans. ASME, pp. 430-436, 1964.

24. Chen, M. M. "An Analytical Study of Laminar Film Condensation: Part 1- Flat Plates," Trans. ASME, pp. 48-54, February 1961.

25. Cess, R. D., "Laminar Film Condensation on a Flat Plate in the Absence of a Body Force," Z. angew Math Phy.(ZAMP), Vol. 11, pp. 426-433, 1960.

26. Patankar, S. V., Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Co., 1980.

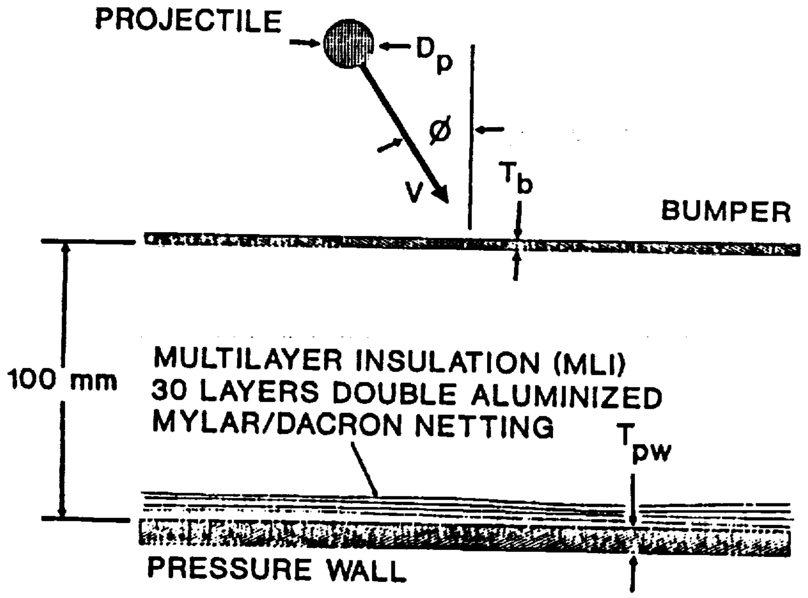
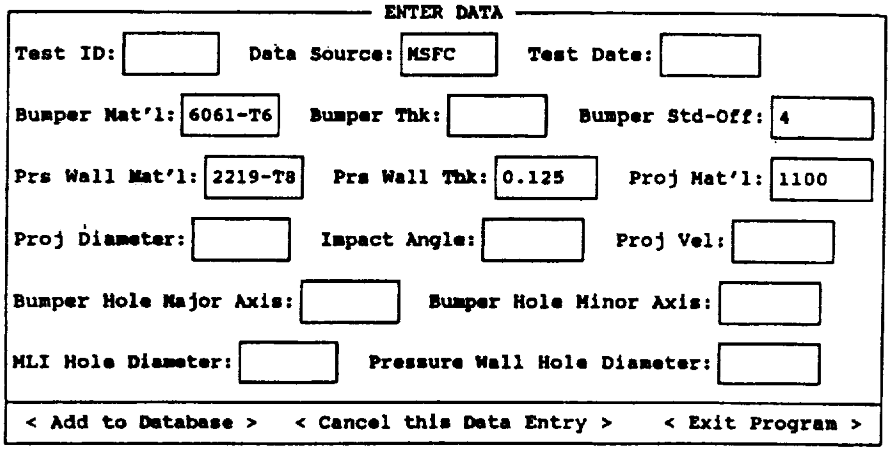


Fig. 1.1 Schematic drawing of impact specimen.



Test ID: Data Source: MSFC Test Date:

Bumper Mat'l: 6061-T6 Bumper Thk: Bumper Std-Off: 4

Prs Wall Mat'l: 2219-T8 Prs Wall Thk: 0.125 Proj Mat'l: 1100

Proj Diameter: Impact Angle: Proj Vel:

Bumper Hole Major Axis: Bumper Hole Minor Axis:

MLI Hole Diameter: Pressure Wall Hole Diameter:

Fig. 2.3.1 Data entry window for adding records to the impact data file.

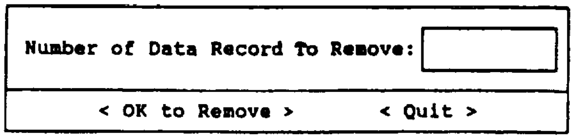
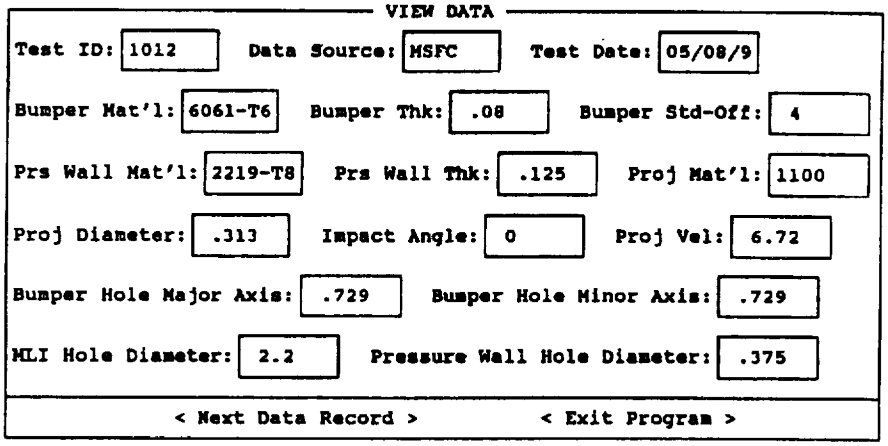

Number of Data Record To Remove:

Fig. 2.3.2 Data entry window for deleting records from the impact data file.


VIEW DATA

Test ID: 1012 Data Source: MSFC Test Date: 05/08/9

Bumper Mat'l: 6061-T6 Bumper Thk: .08 Bumper Std-Off: 4

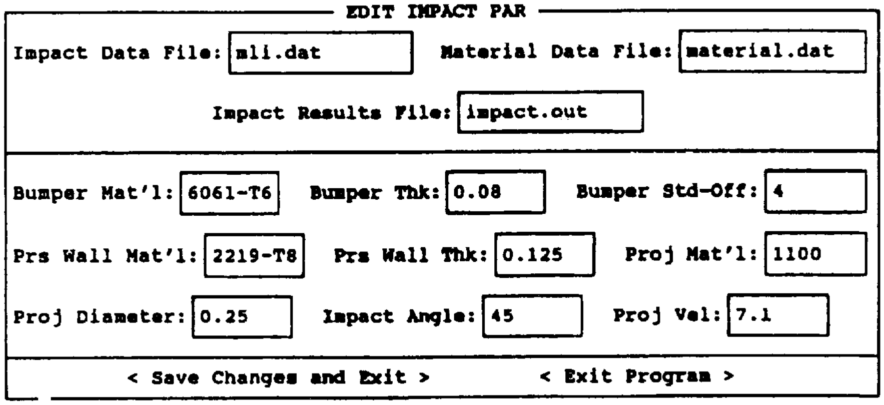
Prs Wall Mat'l: 2219-T8 Prs Wall Thk: .125 Proj Mat'l: 1100

Proj Diameter: .313 Impact Angle: 0 Proj Vel: 6.72

Bumper Hole Major Axis: .729 Bumper Hole Minor Axis: .729

MLI Hole Diameter: 2.2 Pressure Wall Hole Diameter: .375

Fig. 2.3.3 Data entry window for viewing records in the impact data file.


EDIT IMPACT PAR

Impact Data File: mli.dat Material Data File: material.dat

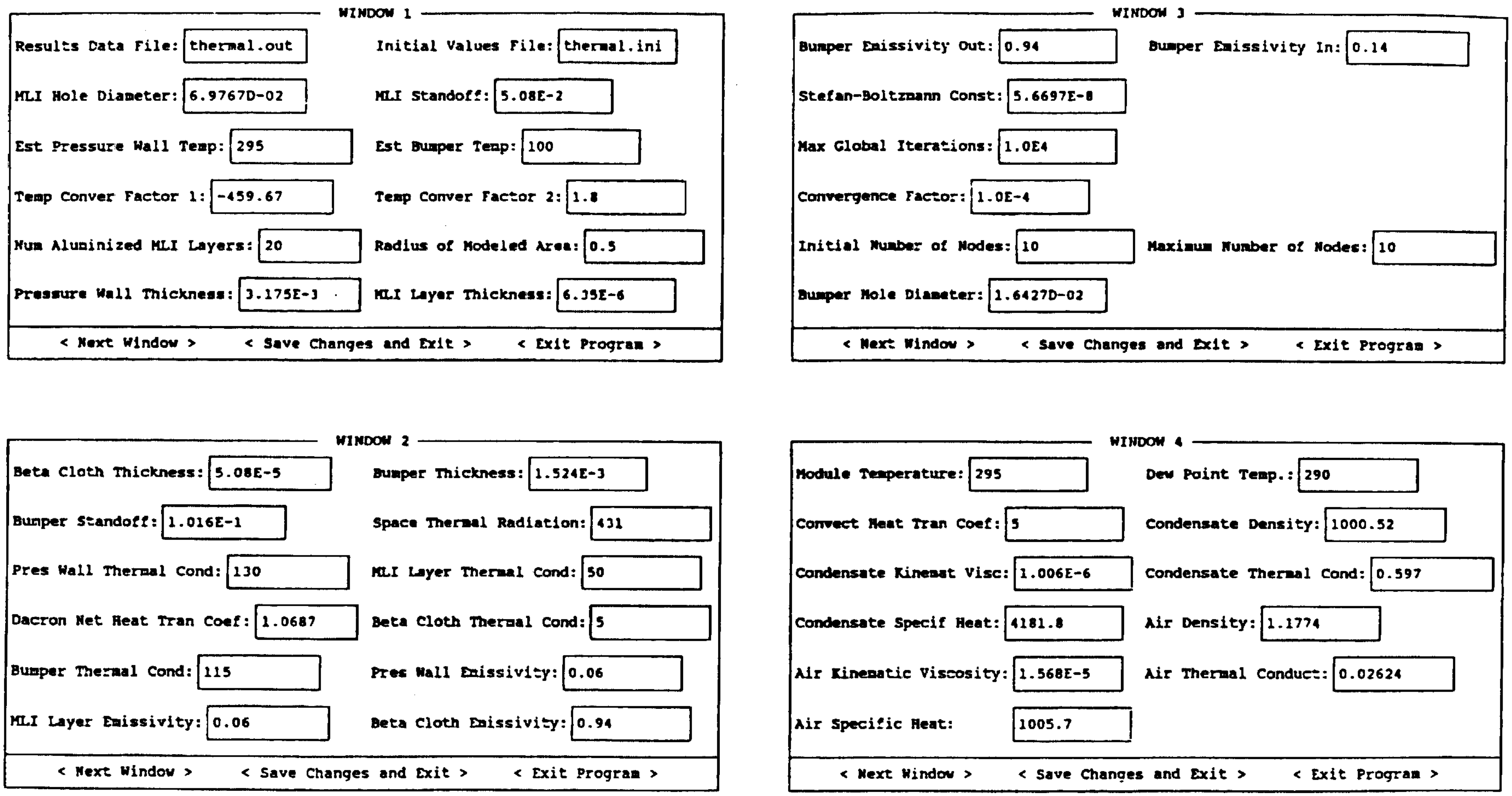
Impact Results File: impact.out

Bumper Mat'l: 6061-T6 Bumper Thk: 0.08 Bumper Std-Off: 4

Prs Wall Mat'l: 2219-T8 Prs Wall Thk: 0.125 Proj Mat'l: 1100

Proj Diameter: 0.25 Impact Angle: 45 Proj Vel: 7.1

Fig. 2.3.4 Data entry window for editing the impact parameters file.



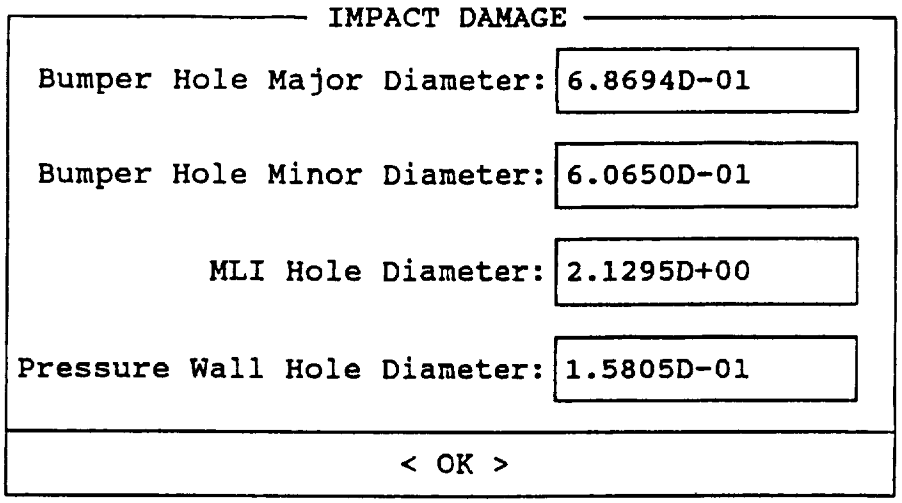
```
WINDOW 1
Results Data File: thermal.out          Initial Values File: thermal.ini
MLI Hole Diameter: 6.9767D-02           MLI Standoff: 5.08E-2
Est Pressure Wall Temp: 295             Est Bumper Temp: 100
Temp Conver Factor 1: -459.67           Temp Conver Factor 2: 1.8
Num Aluminized MLI Layers: 20           Radius of Modeled Area: 0.5
Pressure Wall Thickness: 3.175E-3       MLI Layer Thickness: 6.35E-6
< Next Window >    < Save Changes and Exit >    < Exit Program >
```

```
WINDOW 3
Bumper Emissivity Out: 0.94             Bumper Emissivity In: 0.14
Stefan-Boltzmann Const: 5.6697E-8
Max Global Iterations: 1.0E4
Convergence Factor: 1.0E-4
Initial Number of Nodes: 10             Maximum Number of Nodes: 10
Bumper Hole Diameter: 1.6427D-02
< Next Window >    < Save Changes and Exit >    < Exit Program >
```

```
WINDOW 2
Beta Cloth Thickness: 5.08E-5           Bumper Thickness: 1.524E-3
Bumper Standoff: 1.016E-1               Space Thermal Radiation: 431
Pres Wall Thermal Cond: 130             MLI Layer Thermal Cond: 50
Dacron Net Heat Tran Coef: 1.0687       Beta Cloth Thermal Cond: 5
Bumper Thermal Cond: 115                Pres Wall Emissivity: 0.06
MLI Layer Emissivity: 0.06              Beta Cloth Emissivity: 0.94
< Next Window >    < Save Changes and Exit >    < Exit Program >
```

```
WINDOW 4
Module Temperature: 295                 Dew Point Temp.: 290
Convect Heat Tran Coef: 5               Condensate Density: 1000.52
Condensate Kinemat Visc: 1.006E-6       Condensate Thermal Cond: 0.597
Condensate Specif Heat: 4181.8          Air Density: 1.1774
Air Kinematic Viscosity: 1.568E-5       Air Thermal Conduct: 0.02624
Air Specific Heat:       1005.7
< Next Window >    < Save Changes and Exit >    < Exit Program >
```

Fig. 2.3.5 Data entry windows for editing the thermal parameters file.



IMPACT DAMAGE

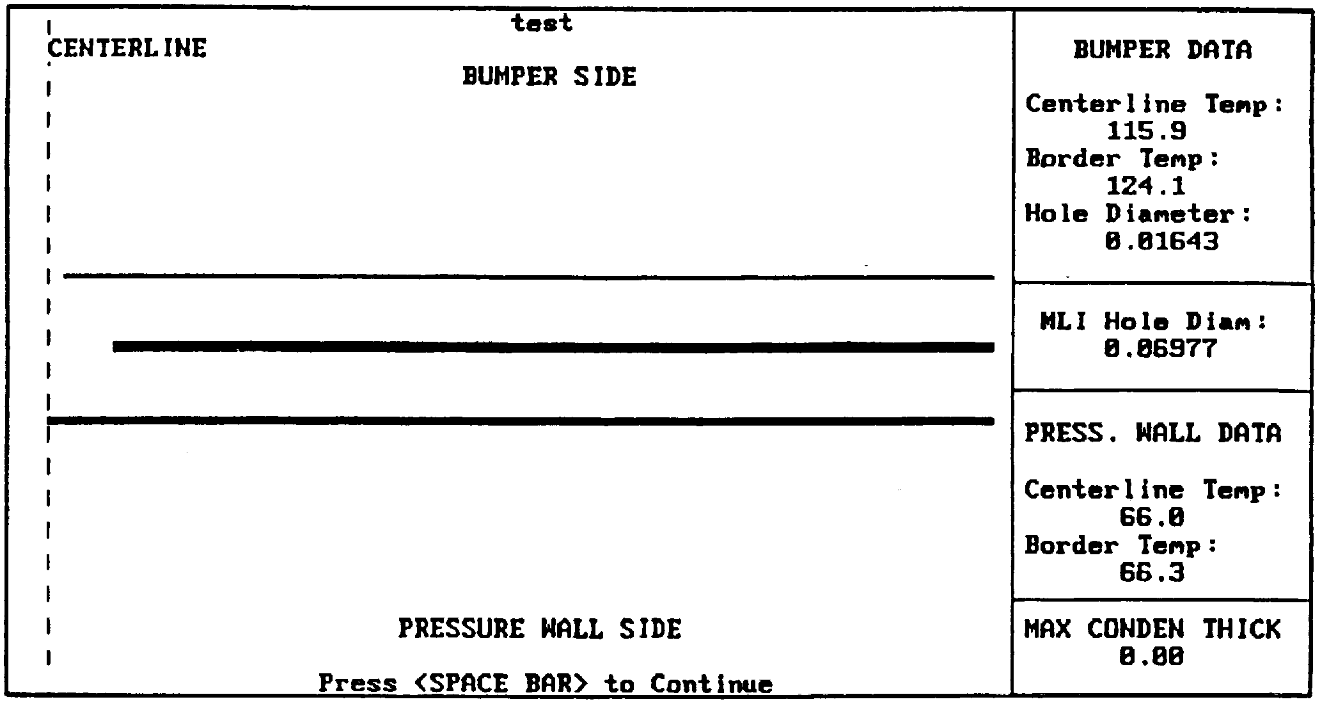
Bumper Hole Major Diameter: 6.8694D-01

Bumper Hole Minor Diameter: 6.0650D-01

MLI Hole Diameter: 2.1295D+00

Pressure Wall Hole Diameter: 1.5805D-01

Fig. 2.3.6 Data window for displaying impact results.



test

BUMPER DATA

Centerline Temp:
115.9
Border Temp:
124.1
Hole Diameter:
0.01643

MLI Hole Diam:
0.06977

PRESS. WALL DATA

Centerline Temp:
66.0
Border Temp:
66.3

MAX CONDEN THICK
0.00

Press <SPACE BAR> to Continue

Fig. 2.3.7 Graphics screen showing thermal and condensation calculation results.

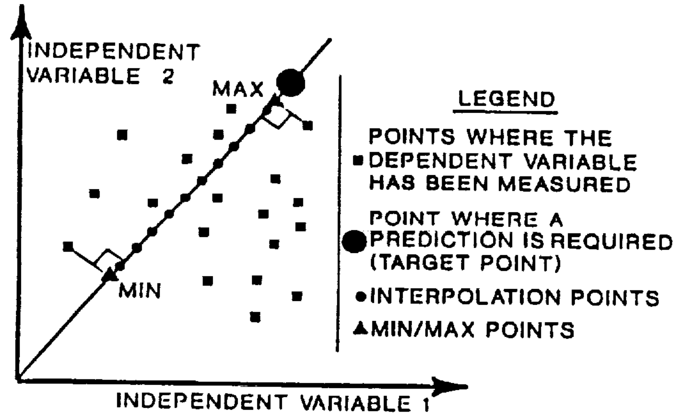

INDEPENDENT VARIABLE 2
MAX
MIN
INDEPENDENT VARIABLE 1
LEGEND
■ POINTS WHERE THE DEPENDENT VARIABLE HAS BEEN MEASURED
● POINT WHERE A PREDICTION IS REQUIRED (TARGET POINT)
• INTERPOLATION POINTS
▲ MIN/MAX POINTS

Fig. 3.1 Technique for selecting interpolation point locations for the case of two independent variables.

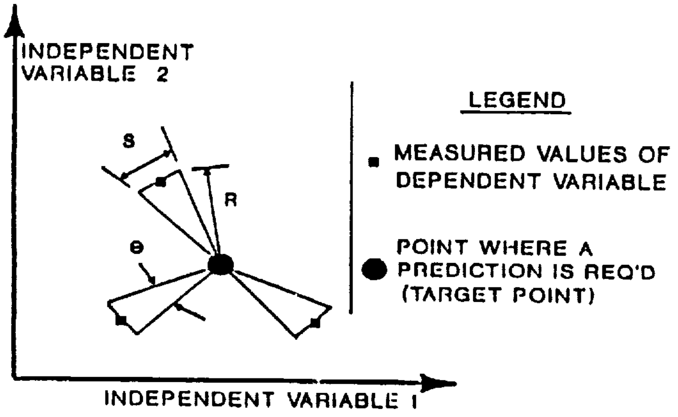

INDEPENDENT VARIABLE 2
S
R
θ
INDEPENDENT VARIABLE 1
LEGEND
■ MEASURED VALUES OF DEPENDENT VARIABLE
● POINT WHERE A PREDICTION IS REQ'D (TARGET POINT)

Fig. 3.2 Interpolation scheme for equally spaced data points.

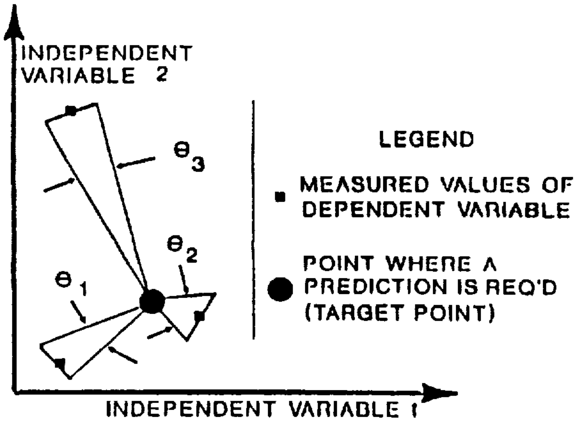

INDEPENDENT VARIABLE 2
θ3
θ2
θ1
INDEPENDENT VARIABLE 1
LEGEND
■ MEASURED VALUES OF DEPENDENT VARIABLE
● POINT WHERE A PREDICTION IS REQ'D (TARGET POINT)

Fig. 3.3 Interpolation scheme for unequally spaced data points.

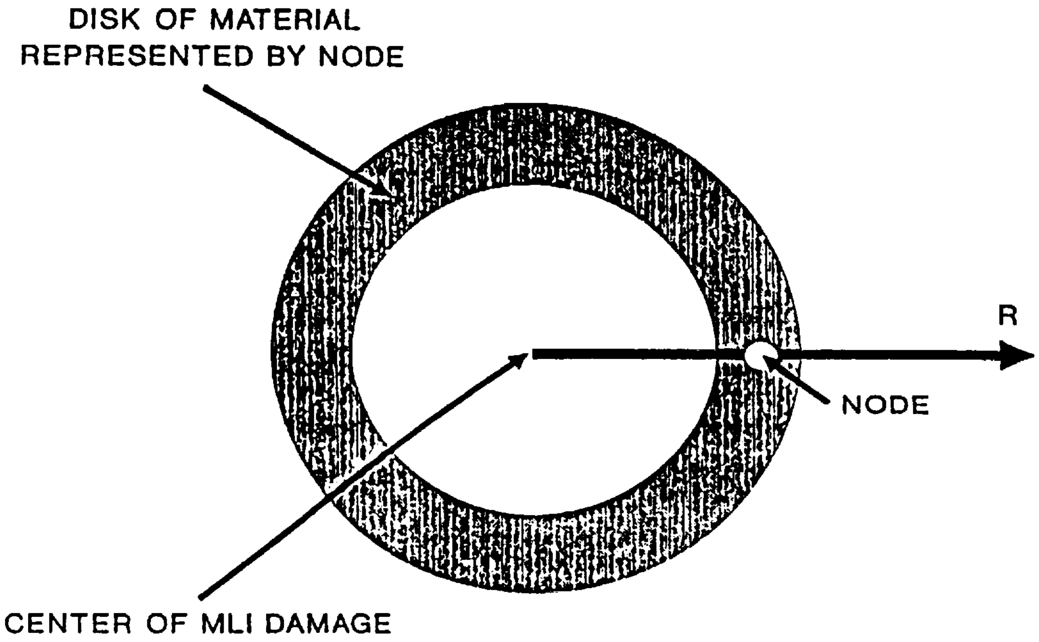


Fig. 6.1 Finite difference discretization scheme where an axially symmetric disk of material is represented by a single node.

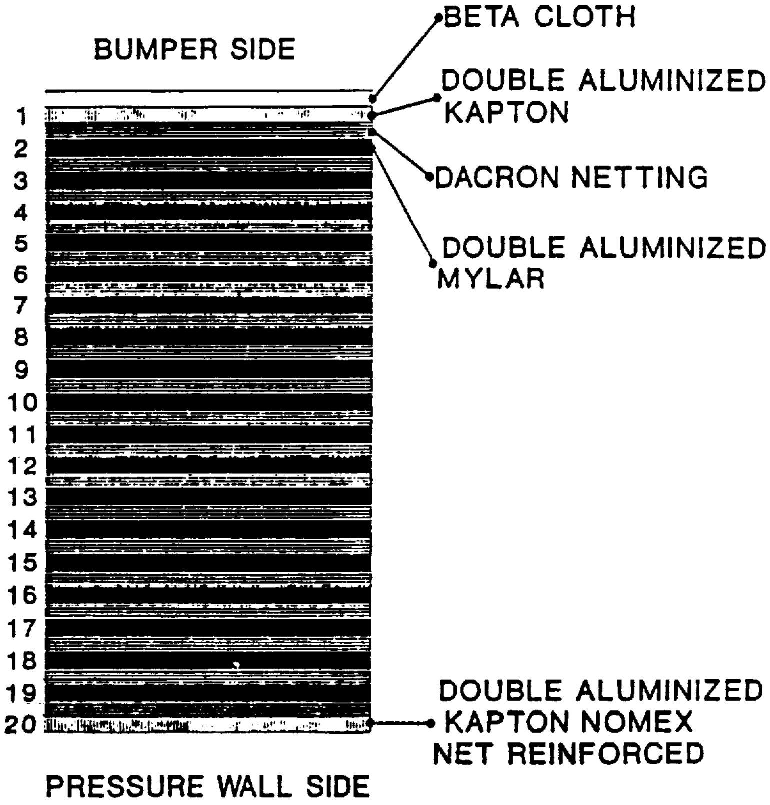


Fig. 6.2 Space Station MLI layup.

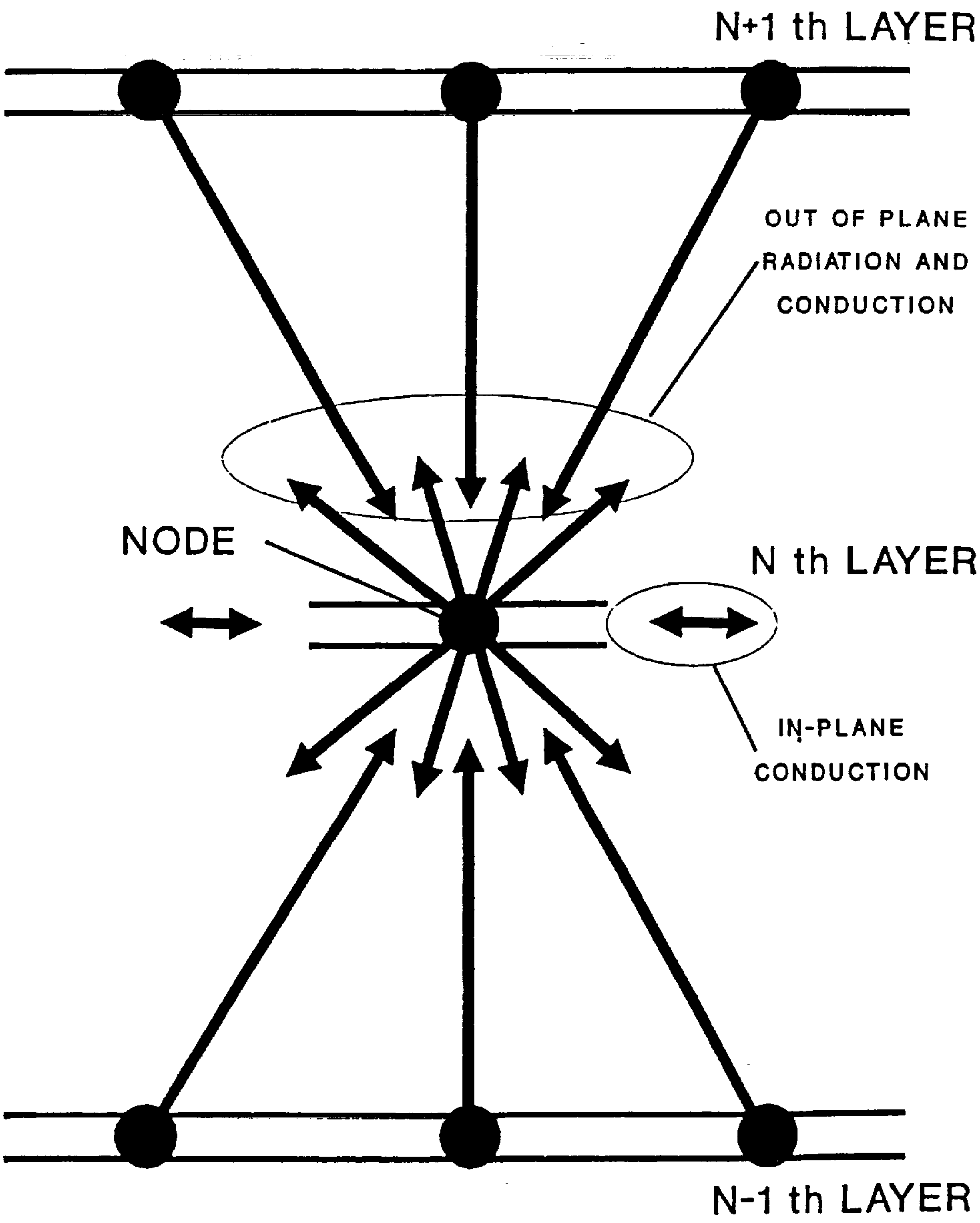


Fig. 6.3 Schematic drawing of heat flow into and out of a typical node.

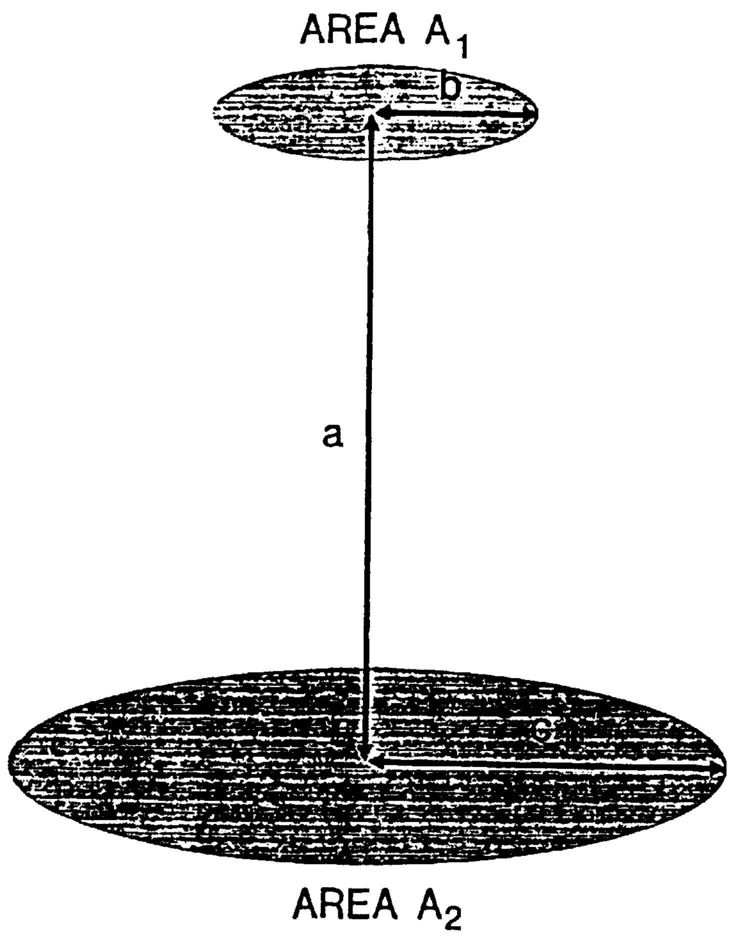


Fig. 6.4 View factor geometry.

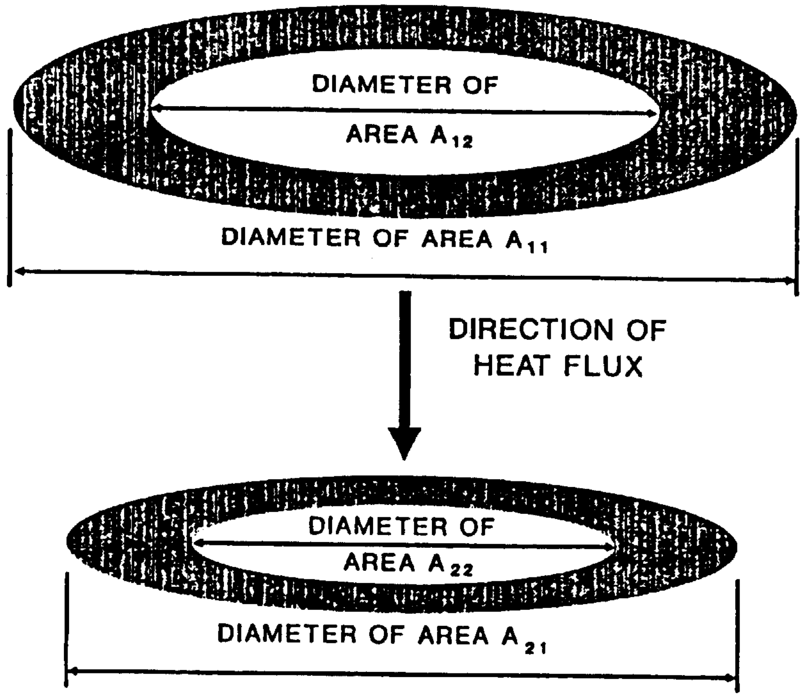


Fig. 6.5 Ring to ring view factor geometry.

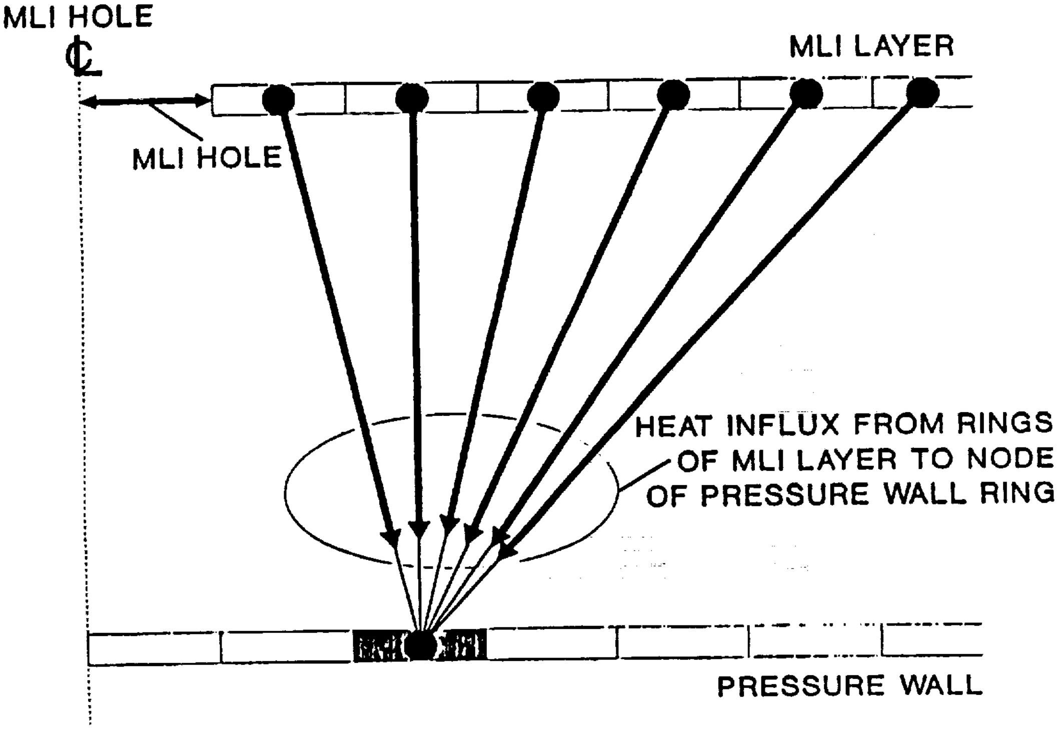


Fig. 6.6 Heat flux from rings of the first MLI layer to a node in a pressure wall ring.

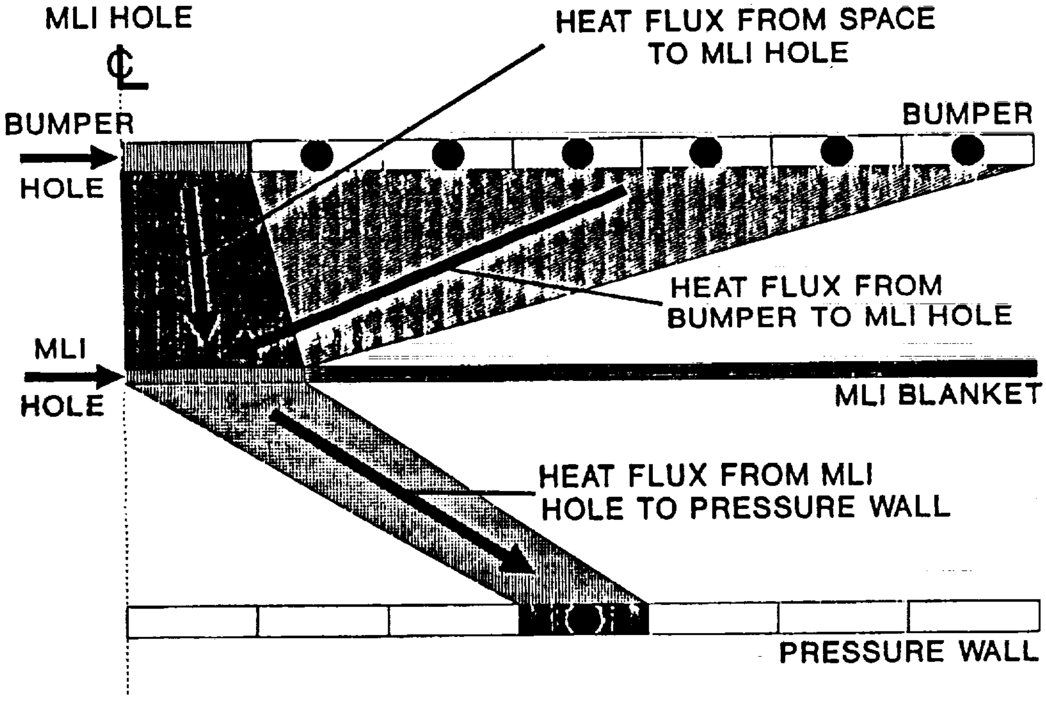


Fig. 6.7 Schematic drawing illustrating the method of calculating heat flux to the pressure wall from the bumper and space environment through the MLI hole.

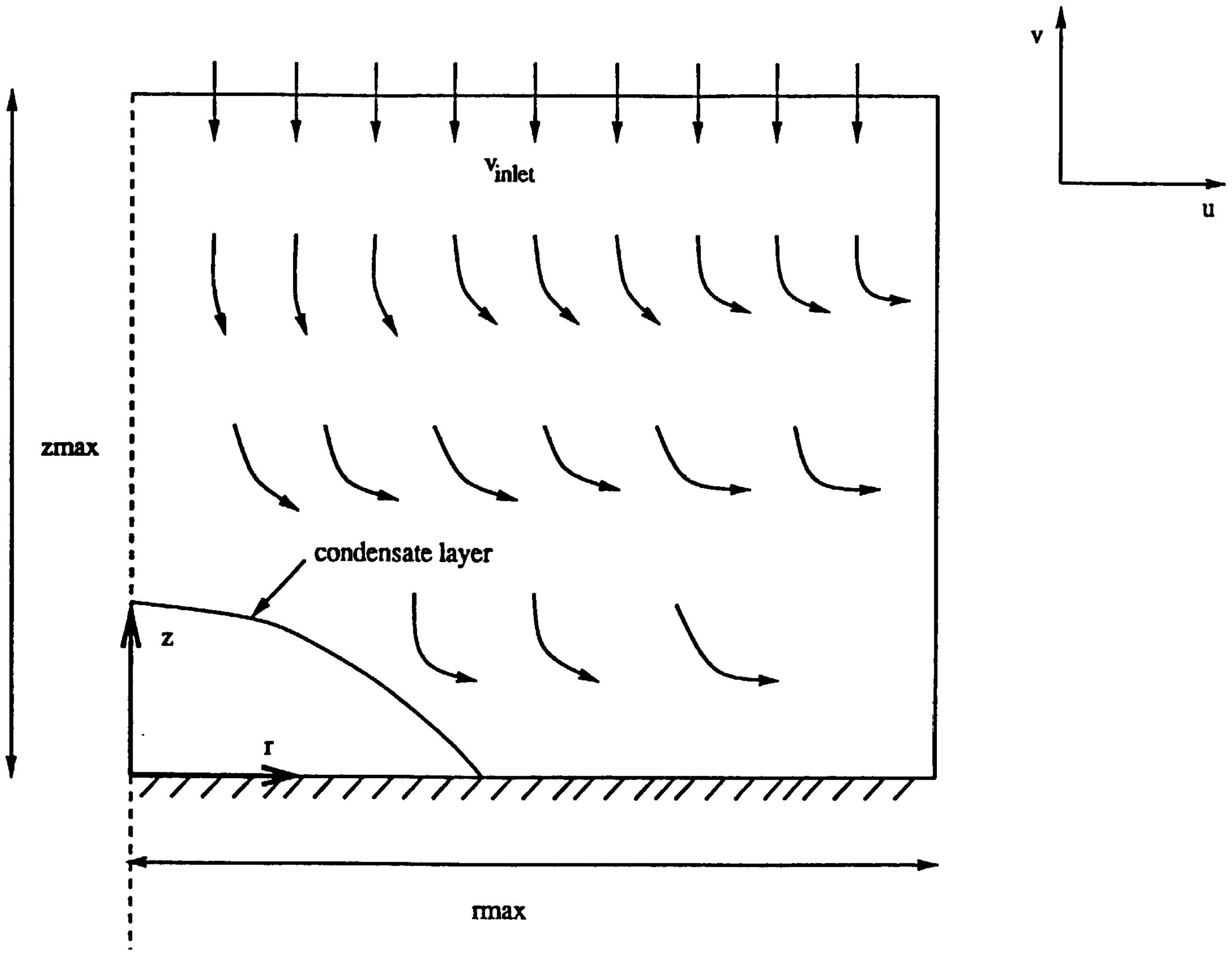


Fig. 7.1 Physical model and coordinate system.

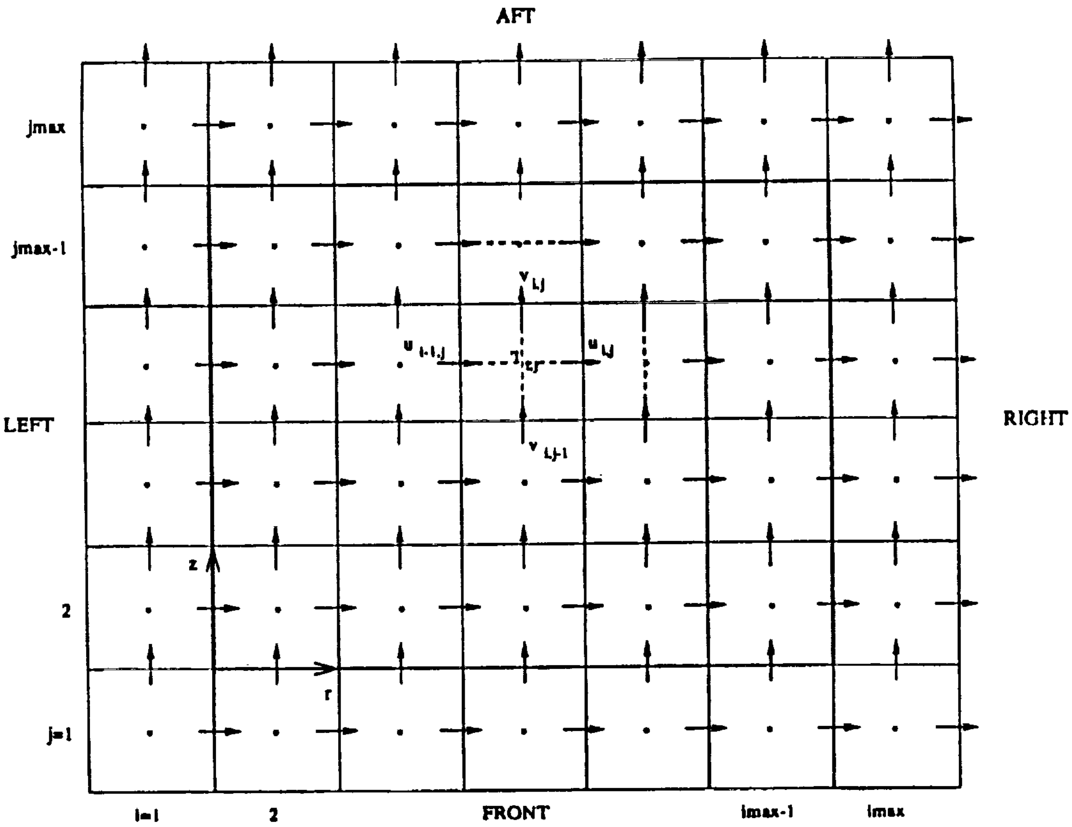


Fig. 7.2 Staggered mesh system showing locations of velocities and temperatures.

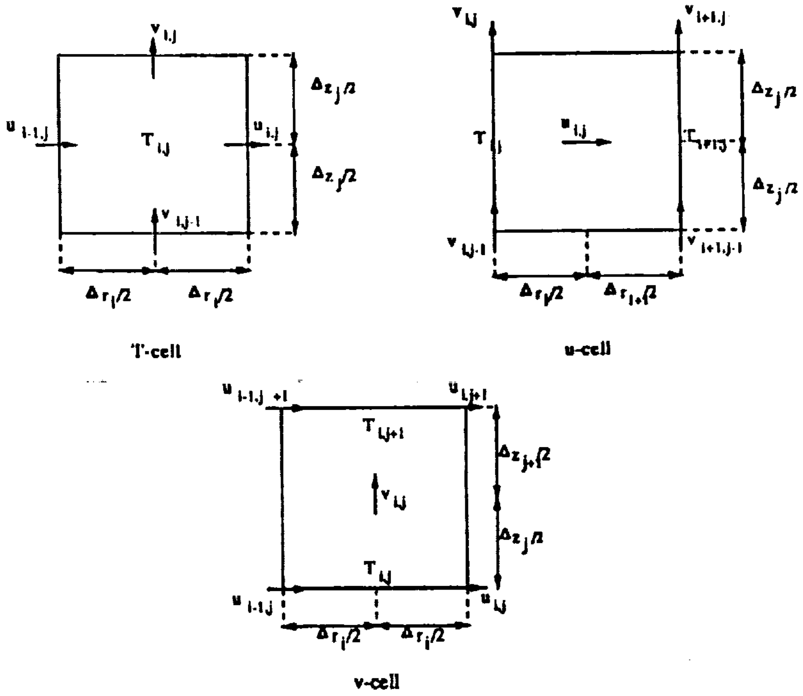


Fig. 7.3 Dimensions of different cells used in 'the computation of u,v and t.

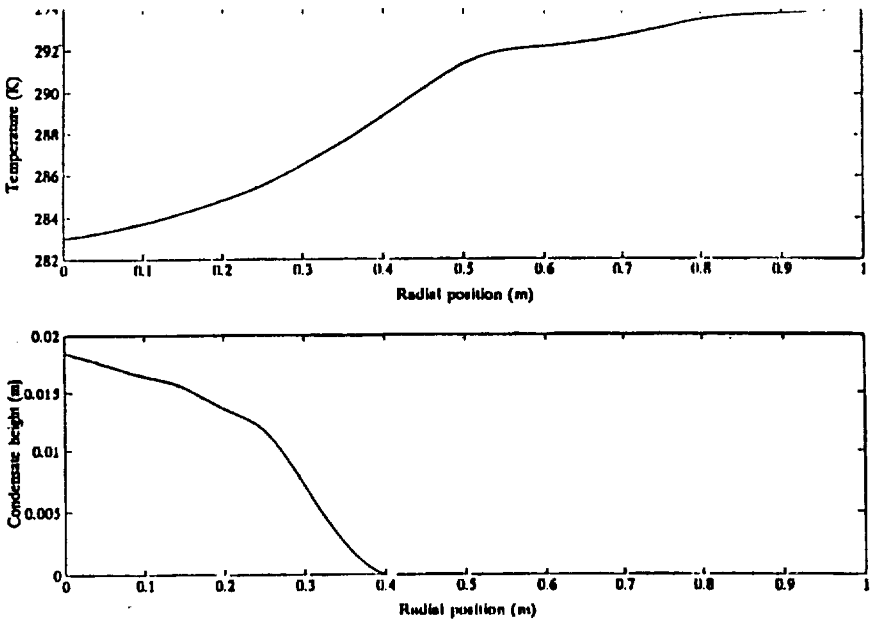


Fig. 7.4 Results of test case I.

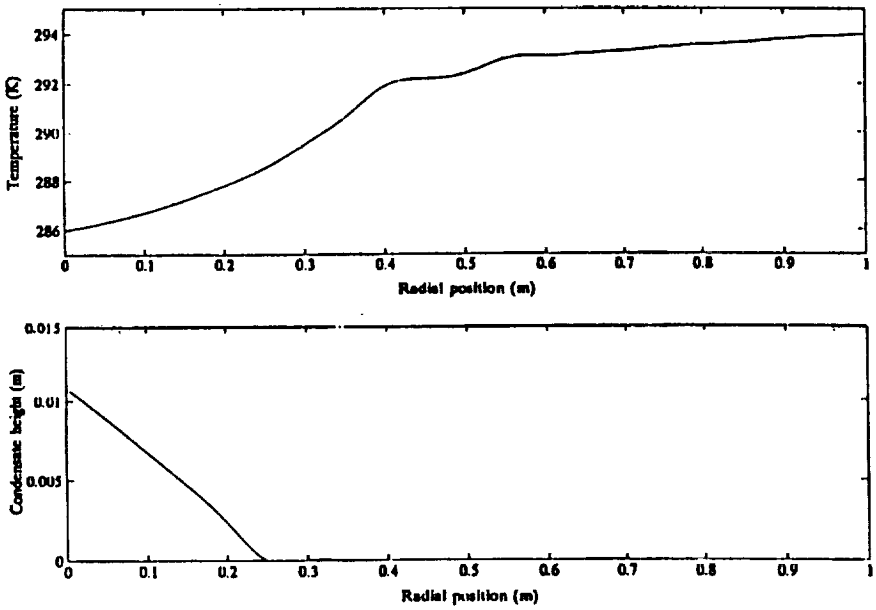


Fig. 7.5 Results of test case II.

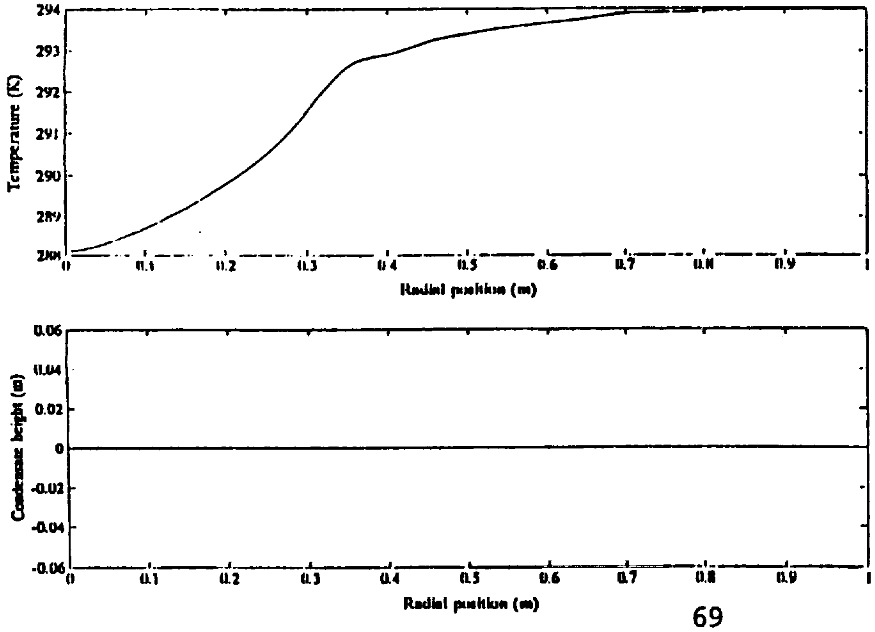
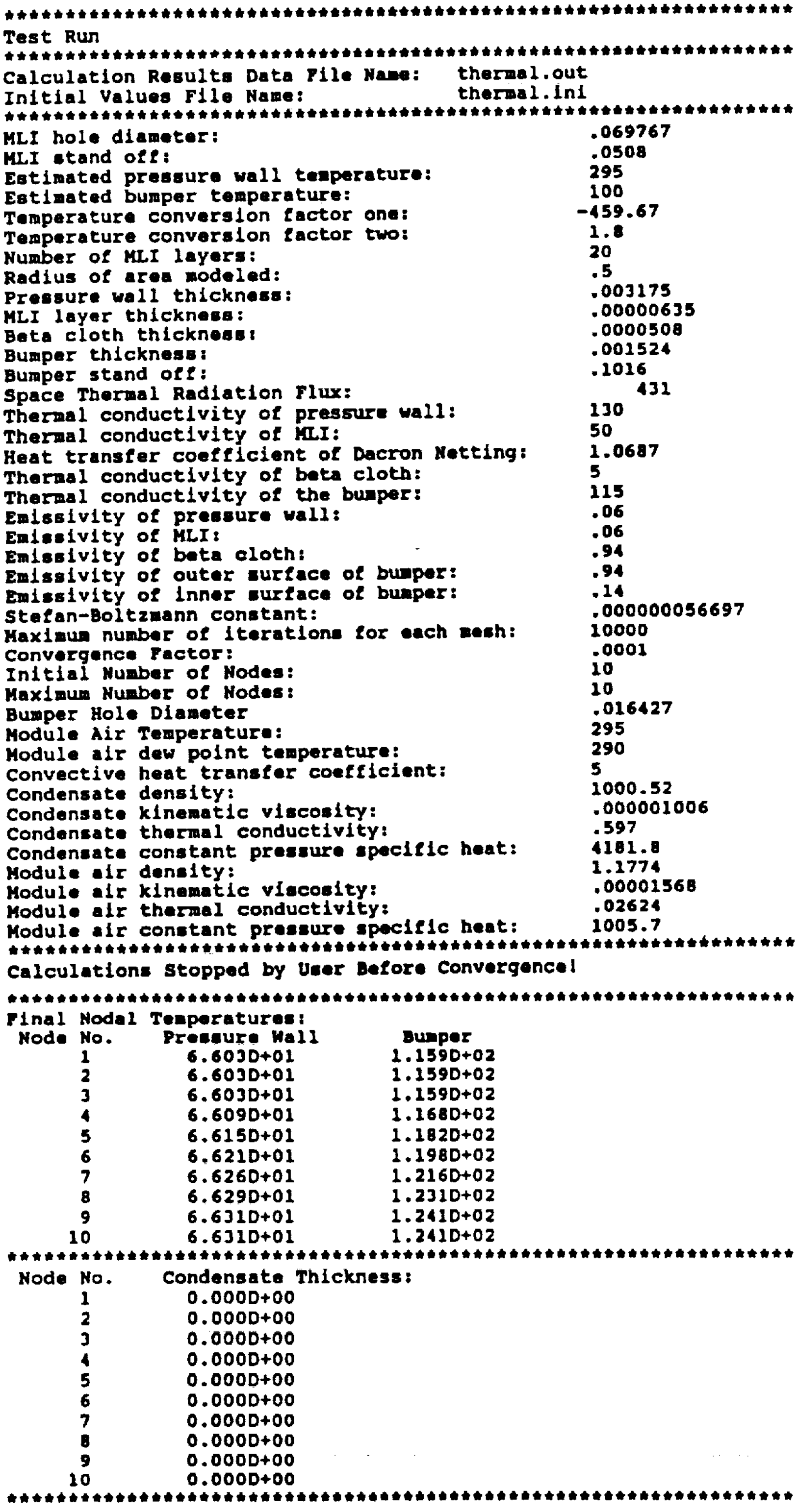


Fig. 7.6 Results of test case III.



```
*********************************************************************
Test Run
*********************************************************************
Calculation Results Data File Name:   thermal.out
Initial Values File Name:             thermal.ini
*********************************************************************
MLI hole diameter:                              .069767
MLI stand off:                                  .0508
Estimated pressure wall temperature:            295
Estimated bumper temperature:                   100
Temperature conversion factor one:             -459.67
Temperature conversion factor two:              1.8
Number of MLI layers:                           20
Radius of area modeled:                         .5
Pressure wall thickness:                        .003175
MLI layer thickness:                            .00000635
Beta cloth thickness:                           .0000508
Bumper thickness:                               .001524
Bumper stand off:                               .1016
Space Thermal Radiation Flux:                       431
Thermal conductivity of pressure wall:          130
Thermal conductivity of MLI:                    50
Heat transfer coefficient of Dacron Netting:    1.0687
Thermal conductivity of beta cloth:             5
Thermal conductivity of the bumper:             115
Emissivity of pressure wall:                    .06
Emissivity of MLI:                              .06
Emissivity of beta cloth:                       .94
Emissivity of outer surface of bumper:          .94
Emissivity of inner surface of bumper:          .14
Stefan-Boltzmann constant:                      .000000056697
Maximum number of iterations for each mesh:     10000
Convergence Factor:                             .0001
Initial Number of Nodes:                        10
Maximum Number of Nodes:                        10
Bumper Hole Diameter                            .016427
Module Air Temperature:                         295
Module air dew point temperature:               290
Convective heat transfer coefficient:           5
Condensate density:                             1000.52
Condensate kinematic viscosity:                 .000001006
Condensate thermal conductivity:                .597
Condensate constant pressure specific heat:     4181.8
Module air density:                             1.1774
Module air kinematic viscosity:                 .00001568
Module air thermal conductivity:                .02624
Module air constant pressure specific heat:     1005.7
*********************************************************************
Calculations Stopped by User Before Convergence!

*********************************************************************
Final Nodal Temperatures:
 Node No.     Pressure Wall       Bumper
      1         6.603D+01       1.159D+02
      2         6.603D+01       1.159D+02
      3         6.603D+01       1.159D+02
      4         6.609D+01       1.168D+02
      5         6.615D+01       1.182D+02
      6         6.621D+01       1.198D+02
      7         6.626D+01       1.216D+02
      8         6.629D+01       1.231D+02
      9         6.631D+01       1.241D+02
     10         6.631D+01       1.241D+02
*********************************************************************
 Node No.     Condensate Thickness:
      1         0.000D+00
      2         0.000D+00
      3         0.000D+00
      4         0.000D+00
      5         0.000D+00
      6         0.000D+00
      7         0.000D+00
      8         0.000D+00
      9         0.000D+00
     10         0.000D+00
*********************************************************************
```

Table 2.3.1   Typical thermal and condensation calculations results file.

Order In Which Data Sets Are Tested For Linear Independence	Data Points Selected To Form Set Of Four (Ordered From Closest To Prediction Point To Farthest)						
	1	2	3	4	5	6	7
1	1	2	3	4			
2	1	2	3		4		
3	1	2	3			4	
4	1	2	3				4
5	1	2		3	4		
6	1	2		3		4	
7	1	2		3			4
8	1	2			3	4	
9	1	2			3		4
10	1	2				3	4
11	1		2	3	4		
12	1		2	3		4	
13	1		2	3			4
14	1		2		3	4	
15	1		2		3		4
16	1		2			3	4
17	1			2	3	4	
18	1			2	3		4
19	1			2		3	4
20	1				2	3	4

Table 4.1 Scheme for Selecting Four Data Point Sets from the Closest Seven Nodes for Damage Function Coefficient Determination.

Equation	ϕ	Γ_ϕ	S_ϕ
Continuity	1	0	0
r-Momentum	u	μ	$-\frac{\partial p}{\partial r} + \frac{1}{r}\frac{\partial(r\mu\frac{\partial u}{\partial r})}{\partial r} + \frac{\partial(\mu\frac{\partial v}{\partial r})}{\partial z} - 2\mu\frac{u}{r^2}$
z-Momentum	v	μ	$-\frac{\partial p}{\partial z} + \frac{1}{r}\frac{\partial(r\mu\frac{\partial u}{\partial z})}{\partial r} + \frac{\partial(\mu\frac{\partial v}{\partial z})}{\partial z}$
Energy	h	k/C_p	$\frac{Dp}{Dt}$

Table 7.1 Summary of Equations.

Table 7.2 Results of condensate test case I

Radial Position (m)	Temperature (K)	Heat Flux (W/m^2)	Condensate Height (m)
0.00000	283.00000	167.22690	0.01851
0.05000	283.30000	160.47165	0.01749
0.10000	283.70000	153.71641	0.01646
0.15000	284.20000	146.96116	0.01544
0.20000	284.80000	137.19782	0.01357
0.25000	285.50000	127.43448	0.01171
0.30000	286.50000	111.61815	0.00710
0.35000	287.60000	95.80183	0.00249
0.40000	288.90000	57.40092	0.00000
0.45000	290.20000	19.00000	0.00000
0.50000	291.40000	14.50000	0.00000
0.55000	292.00000	10.00000	0.00000
0.60000	292.20000	9.00000	0.00000
0.65000	292.40000	8.00000	0.00000
0.70000	292.70000	6.25000	0.00000
0.75000	293.10000	4.50000	0.00000
0.80000	293.50000	3.00000	0.00000
0.85000	293.70000	1.50000	0.00000
0.90000	293.80000	1.00000	0.00000
0.95000	293.90000	0.50000	0.00000
1.00000	294.00000	0.25000	0.00000

Table 7.3 Results of condensate test case II

Radial Position (m)	Tempearture (K)	Heat Flux (W/m^2)	Condensate Height (m)
0.00000	286.00000	122.26894	0.01084
0.05000	286.30000	115.49804	0.00879
0.10000	286.70000	108.72715	0.00674
0.15000	287.20000	101.95625	0.00468
0.20000	287.80000	64.72812	0.00234
0.25000	288.50000	27.50000	0.00000
0.30000	289.50000	22.25000	0.00000
0.35000	290.60000	17.00000	0.00000
0.40000	291.90000	13.00000	0.00000
0.45000	292.20000	9.00000	0.00000
0.50000	292.40000	7.00000	0.00000
0.55000	293.00000	5.00000	0.00000
0.60000	293.10000	4.50000	0.00000
0.65000	293.20000	4.00000	0.00000
0.70000	293.30000	3.37500	0.00000
0.75000	293.45000	2.75000	0.00000
0.80000	293.55000	2.25000	0.00000
0.85000	293.65000	1.75000	0.00000
0.90000	293.80000	1.12500	0.00000
0.95000	293.90000	0.50000	0.00000
1.00000	294.00000	0.25000	0.00000

Table 7.4 Results of condensate test case III

Radial Position (m)	Temperature (K)	Heat Flux (W/m^2)	Condensate Height (m)
0.00000	288.10000	29.50000	0.00000
0.05000	288.30000	28.50000	0.00000
0.10000	288.70000	26.50000	0.00000
0.15000	289.20000	24.00000	0.00000
0.20000	289.80000	21.00000	0.00000
0.25000	290.50000	17.50000	0.00000
0.30000	291.50000	12.50000	0.00000
0.35000	292.60000	7.00000	0.00000
0.40000	292.90000	5.50000	0.00000
0.45000	293.20000	4.00000	0.00000
0.50000	293.40000	3.00000	0.00000
0.55000	293.55000	2.25000	0.00000
0.60000	293.65500	1.72500	0.00000
0.65000	293.75900	1.20500	0.00000
0.70000	293.88000	0.60000	0.00000
0.75000	293.91000	0.45000	0.00000
0.80000	293.93000	0.35000	0.00000
0.85000	293.95500	0.22500	0.00000
0.90000	293.97500	0.12500	0.00000
0.95000	293.99000	0.05000	0.00000
1.00000	294.00000	0.00000	0.00000

☆U.S. GOVERNMENT PRINTING OFFICE 1991—631-060/40164