D180-19024-1

INPUTB

(A THERMAL/STRUCTURAL DATA INTERFACE PROGRAM FOR 2-DIMENSIONAL AND 3-DIMENSIONAL INTERPOLATION)

Contract NAS8-30615

September 1, 1975

Prepared by

R. G. Vos

J. W. Straayer, Program Manager

Prepared for

National Aeronautics and Space Administration
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

(NASA-CR-144119) INPUTB: A THERMAL/STRUCTURAL DATA INTERFACE PROGRAM FOR 2-DIMENSIONAL AND 3-DIMENSIONAL INTERPOLATION (Boeing Aerospace Co., Seattle, Wash.) 58 p HC $4.50 CSCL 09B G3/60

N76-16810

Unclas 07426

Boeing Aerospace Company
Research and Engineering Division
Seattle, Washington 98124

TABLE OF CONTENTS

PART III. PROGRAMMER MANUAL

1.0 INTRODUCTION

INPUTB is a program for interpolation in both space (2-dimensional or 3-dimensional) and time. It is based on a linear interpolation scheme using simplex spatial regions (triangles and tetrahedra). INPUTB was developed mainly to provide data interfacing between the output from thermal analyzers and the input to the BOPACE 3-D (Boeing plastic analysis capability for 3-dimensional solids using isoparametric finite elements) program. The INPUTB interpolator is of a quite general nature, however, and could be used effectively for other tasks (with appropriate changes in input/output formats, if necessary).

The INPUTB program utilizes temperature values which are given at some sequence of times for a list of strategically located "thermal nodes." It operates on these values by performing a double interpolation (in time and space) to provide temperature values at another desired sequence of times for a list of "structural nodes." The thermal/structural interface procedure is shown in Figure 1-1.

INPUTB is written in FORTRAN IV and is available on both the IBM 360/370 and the UNIVAC 1108 machines. The program has a core storage requirement of 30K words, and it presently has a capability for handling 1000 structural nodes and 500 thermal nodes.

The INPUTB document consists of three major parts:

PART I	Theoretical Manual
PART II	User Manual
PART III	Programmer Manual

Recognition is due to Curt Whiting for his work in preparation of example problems, and in debugging of the program.

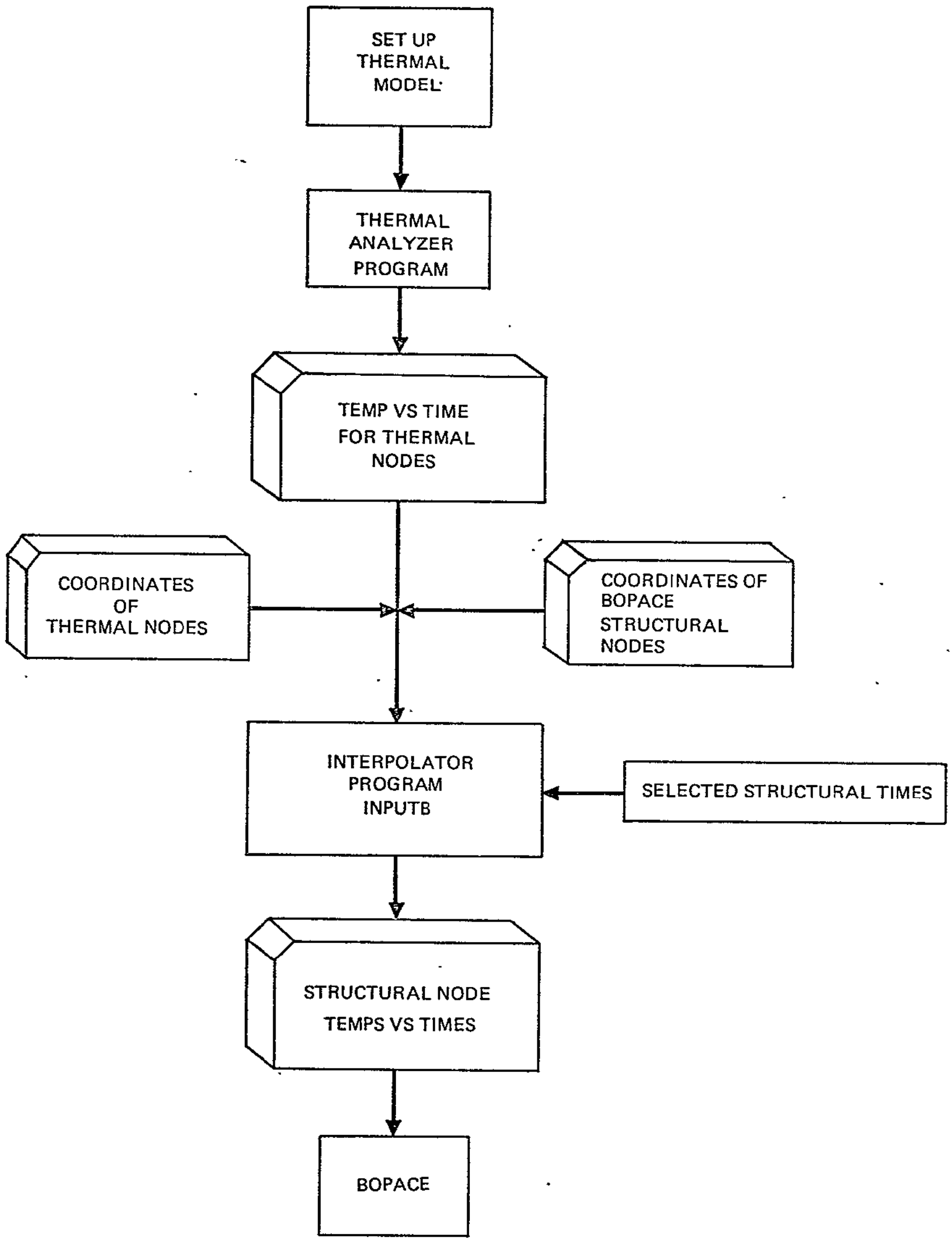

Figure 1-1. Thermal/Structural Interface for INPUTB/BOPACE Run

INPUTB

PART I. THEORETICAL MANUAL

2.0 INPUTB INTERPOLATION THEORY

INPUTB performs a double interpolation in time and space, to convert data associated with one set of times and spatial locations (i.e., thermal node data) to data associated with another set (i.e., structural node data). In order to progressively develop the necessary concepts, the timewise interpolation is discussed first, followed by discussions of the 2-dimensional and 3-dimensional spatial interpolation.

2.1 TIMEWISE INTERPOLATION

The interpolation in time is a simple linear, 1-dimensional scheme. Given the temperature of a point at a sequence of "thermal" times, two of these times are used to compute the temperature at each desired "structural" time. The two thermal times, of course, must be those nearest to the structural time, such that the first is less than or equal to, and the second is greater than or equal to, the structural time.

2.2 SPATIAL 2-DIMENSIONAL INTERPOLATION

The 2-dimensional interpolation scheme is illustrated by Figure 2-1. There the open circles represent thermal nodes with given temperature values, and the closed circle (point N) represents one of the structural nodes at which the temperature must be computed.

In general, three thermal node points must be selected (forming an enclosing triangle around the structural node) in order to accomplish an exact linear interpolation. The sequence for selecting these three points is as follows:

1. Find point 1 as the thermal node closest to N.
2. Locate point 2 as the next closest thermal node such that the smaller angle between lines N-1 and N-2 is greater than 90°. (The restriction on the angle is necessary to avoid possible selection later of the third point such that a long narrow triangle would be formed. Note it also ensures that a

perpendicular from N to the line 1-2 will pass between points 1 and 2).

3. Locate point 3 as the next closest thermal node such that N is enclosed within the triangle 1-2-3. (A necessary and sufficient condition for satisfying this requirement is that the three vector cross products N-1 X N-2, N-2 X N-3, and N-3 X N-1, all have the same direction).

The structural node N divides the triangle into three triangular sub-areas A_1, A_2, and A_3, with total area of the triangle defined by $A = A_1 + A_2 + A_3$. Interpolation weighting factors are then defined by $W_1 = A_1/A$, $W_2 = A_2/A$, and $W_3 = A_3/A$, so that $W_1 + W_2 + W_3 = 1$. Finally, the temperature T_N at point N is computed by

$$T_N = W_1 T_1 + W_2 T_2 + W_3 T_3 \qquad (2\text{-}1)$$

where T_i is the temperature of the i th thermal node forming the triangle.

In case a third thermal node is not found to form an enclosing triangle, a perpendicular is defined from N to the line 1-2, intersecting 1-2 at say point N^1. The weights W_1 and W_2 are then computed as the distances from N^1 to points 2 and 1, respectively, divided by the length of line 1-2. T_N can then again be computed from Equation 2-1 (INPUTB logic in this case sets $W_3 = 0$, and assumes node 3 = node 1). In case a second node is also not found according to the above selection procedure, INPUTB logic sets $W_2 = W_3 = 0$, and assumes node 2 = node 3 = node 1, and again makes use of Equation 2-1.

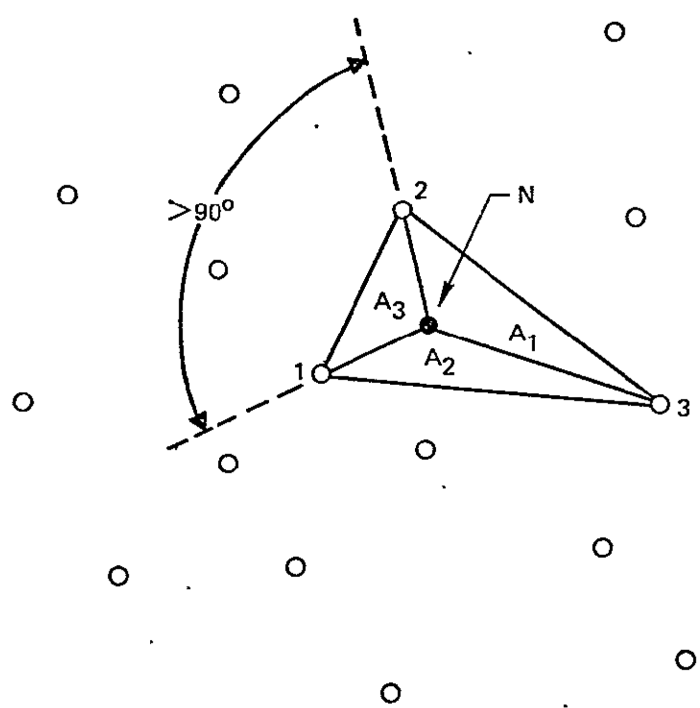

Figure 2-1. 2-Dimensional Spatial Interpolation

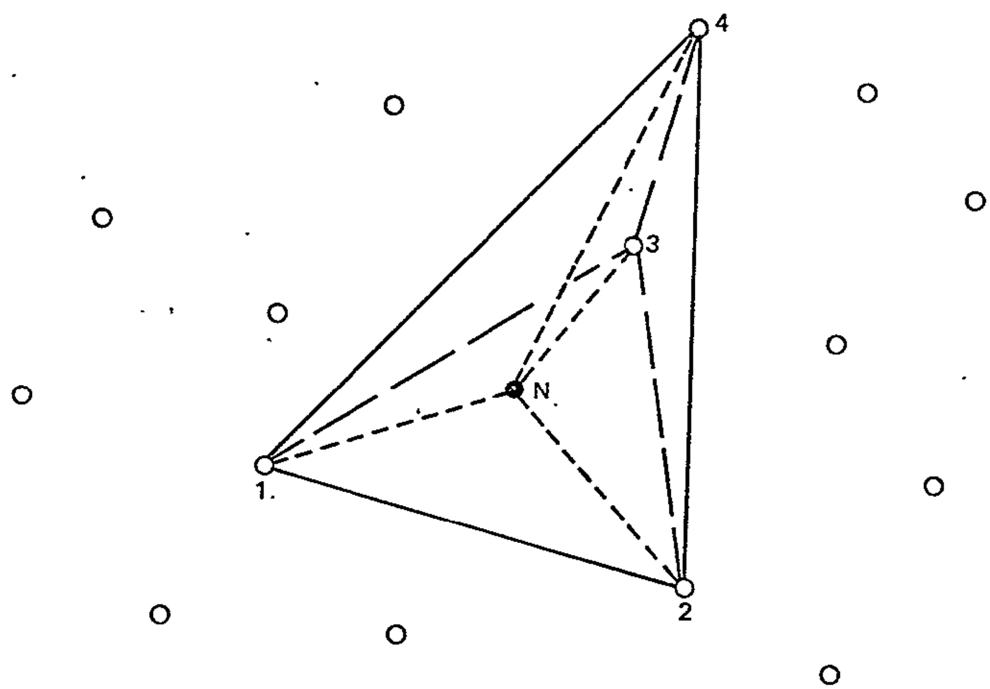

Figure 2-2. 3-Dimensional Spatial Interpolation

2.3 SPATIAL 3-DIMENSIONAL INTERPOLATION

The 3-dimensional interpolation scheme is similar to the 2-dimensional scheme, except that tetrahedra rather than triangles are involved. As shown by Figure 2-2, the open circles again represent thermal nodes with given temperature values, and the closed circle (point N) represents a structural node at which the temperature must be computed.

Four thermal node points must now be selected to form an enclosing tetrahedron around the structural node in order to accomplish an exact linear interpolation. The sequence for selecting these four points is as follows:

1. Find point 1 as the thermal node closest to N.

2. Locate point 2 as the next closest thermal node such that the smaller angle between lines N-1 and N-2 is greater than 90°.

3. Locate point 3 as the next closest thermal node such that a perpendicular from N to the plane 1-2-3 intersects this plane within the triangle 1-2-3. (A vector S is defined normal to the plane 1-2-3. The requirement is satisfied if the three vector dot-cross products S · (N-1 x N-2), S · (N-2 x N-3), and S · (N-3 x N-1), all have the same sign.)

4. Locate point 4 as the next closest thermal node such that N is enclosed within the tetrahedron 1-2-3. (A necessary and sufficient condition for satisfying this requirement is that the four vector dot-cross products -N-1·(N-3 x N-4), + N-2·(N-4 x N-1), -N-3·(N-1 x N-2), and + N-4·(N-2 x N-3), all have the same sign.)

The structural node N divides the tetrahedron into four tetrahedral sub-volumes, V_1, V_2, V_3, and V_4, with total volume of the tetrahedron defined by $V = V_1 + V_2 + V_3 + V_4$ (V_i is the volume of the tetrahedron opposite node point i). Interpolation weighting factors are then defined by $W_i = V_1/V$, $W_2 = V_2/V$, $W_3 = V_3/V$, and $W_4 = V_4/V$, so that $W_1 + W_2 + W_3 + W_4 = 1$. Finally the temperature T_N at point N is computed by

$$T_N = W_1T_1 + W_2T_2 + W_3T_3 + W_4T_4 \qquad (2\text{-}2)$$

where T_i is the temperature of the i th thermal node forming the tetrahedron.

In case a fourth thermal node is not found to form an enclosing tetrahedron, point N is projected perpendicularly onto the 1-2-3 plane, and the 2-dimensional interpolation scheme using triangle 1-2-3 is employed. In case a third or second node is also not found, INPUTB logic again proceeds as discussed for 2-dimensional interpolation.

3.0 PROGRAM FLOW

The major steps accomplished during an INPUTB run are shown in the main program flow summary of Figure 3-1. Step 1 reads input/output file unit number, and allows for printed and/or punched output. READM and READMT read the structural and thermal mesh (node locations) respectively, with structural nodes given in BOPACE 3-D format. Step 4 reads a series of thermal node temperature vectors, corresponding to a given sequence of times. In step 5, the major interpolation logic is employed, as discussed in Sections 2.2 and 2.3, to select four thermal nodes and compute associated weighting factors, for each structural node. The four nodes and weighting factors are then stored, so that they can later be used to give interpolated structural node temperatures at each desired time. In step 6, the interpolation is actually performed, using these stored values and the thermal node time-temperature vectors read in Step 4. Multiple problems may be executed during a single program run.

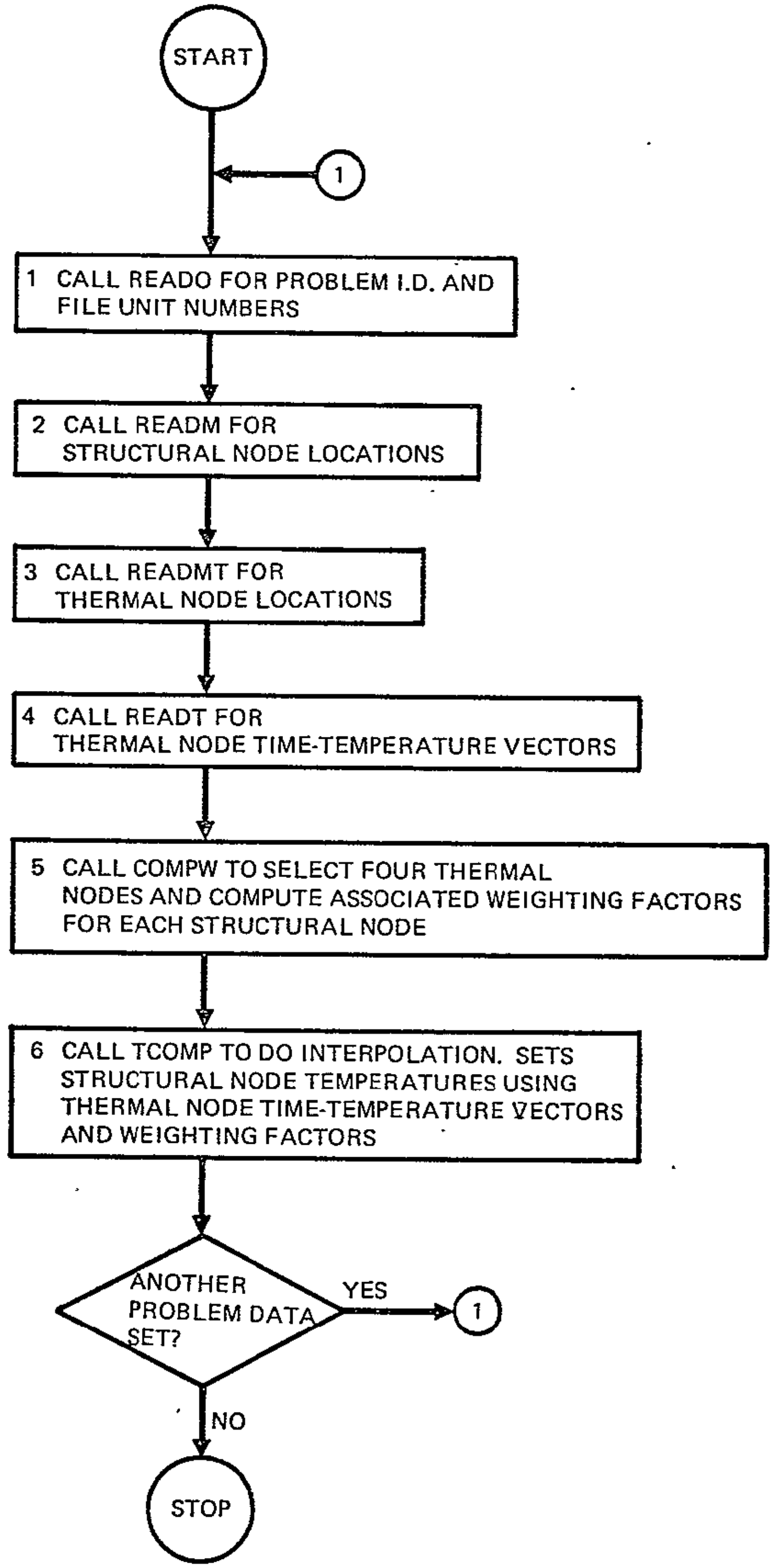

Figure 3-1. Main Problem Flow

INPUTB

PART II. USER MANUAL

4.0 SUMMARY OF INPUTB INPUT DATA

A pictorial of the INPUTB input deck is shown in Figure 4-1. The input data consists of the following two general types.

Type C: Data on the usual card file.

Type I: Data on File I. These include almost all of the input data. File I is defined by the user so that his data can be input via cards, tape, etc.

The data included on these files are described below. Formats are consistent with FORTRAN IV conventions.

C-1. Start code and file unit numbers:

a. Insert the code "START".

b. Unit number for file I. (Must be given).

c. Unit number for first output file (e.g., printer). (If not given, this file will not be output.)

d. Unit number for second output file (e.g. punch). (If not given, this file will not be output.)

Format (A4, 6X, 3I5)

I-1. Problem I.D. title (any 80 characters).

Format (20A4).

I-2 Structural node location data (BOPACE 3-D format). For each structural node give node I.D. number, identification number of coordinate system to define location (= 0, 1 or 2), coordinates X-Y-Z (or R-Θ-Z or R-Θ-Φ), and

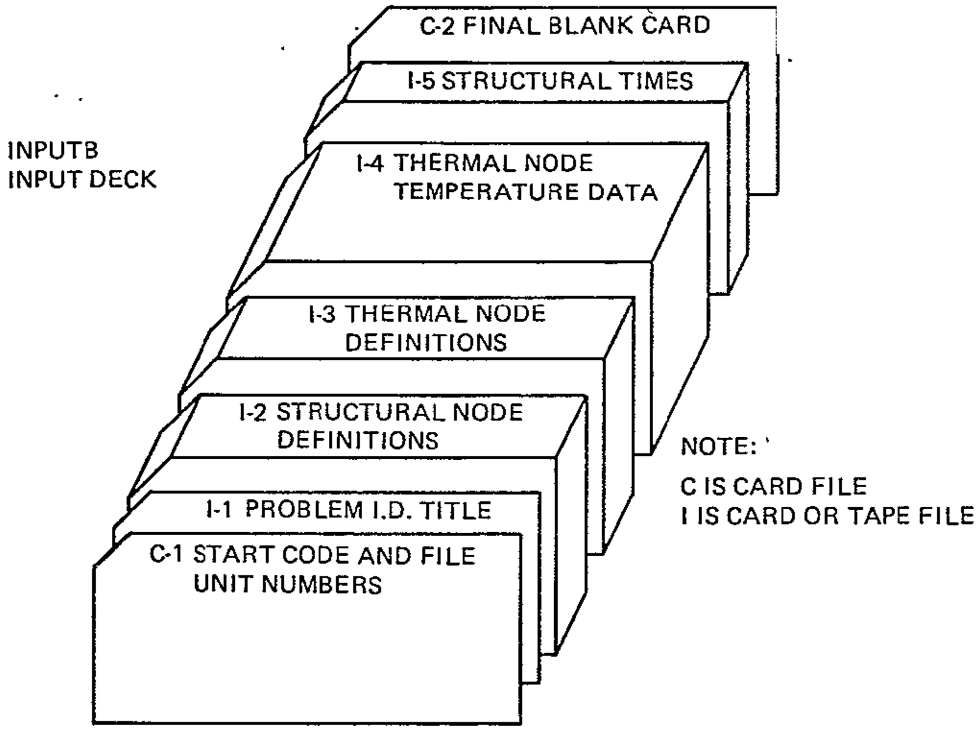

Figure 4-1. INPUTB Input Deck Setup

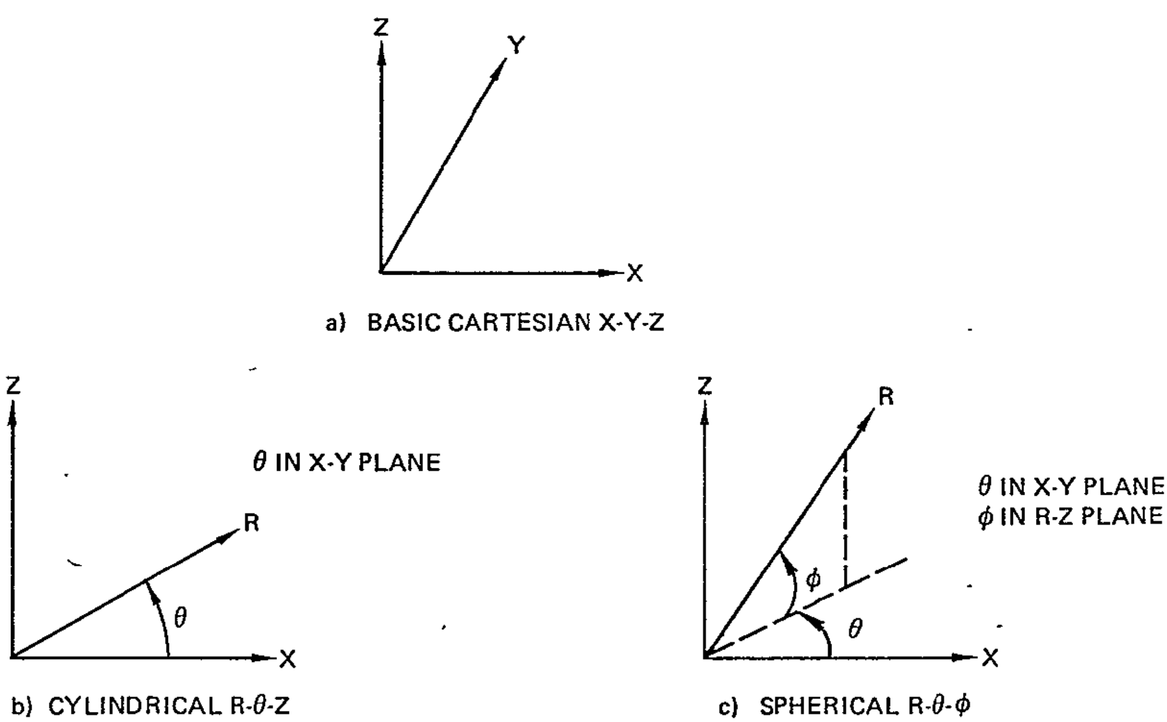

Figure 4-2. Coordinate Systems

identification number of coordinate system to define displacements (= 0, 1, 2 or other). (Coordinate identification number 0 specifies the basic Cartesian system, 1 specifies the basic spherical system, and 2 specifies a special Cartesian system. Note: The displacement coordinate system I.D. is part of the BOPACE format, but is simply read and printed by INPUTB; also, for a 2-dimensional problem, the third coordinate is input as blank or zero. The basic coordinate systems are illustrated in Figure 4-2.)

Format (2I5, 3F10.0, I5)

Blank card after last structural node.

I-3. Thermal node location data. For each thermal node give node I.D. number, identification number of coordinate system to define location (= 0, 1 or 2), coordinates X-Y-Z (or R-Θ-Z or R-Θ-Φ). (For a 2-dimensional problem, the third coordinate is input as blank or zero.) User has option of one or two thermal nodes per card.

Format (2 (2I5, 3F10.0))

Blank card after last thermal node.

I-4. Thermal node time-temperature data.

a. Number of thermal times (≥2), initial default temperature.

Format (I10, F10.0)

b. Time-temperature vectors. For each vector:

Time

Format (F10.0)

Thermal node temperatures. For each node give node I.D. and temperature. User has option of from one to four thermal nodes per card.

Format (4(I10, F10.0))

Blank card after last value of each vector.

Note: If temperature is not given for any node, it is assigned a value by the program. For the first thermal time, this value is the default temperature in input item I-4. For later times, it is the value from the preceeding time.

I-5. Structural times. Give time value for each time at which interpolated structural temperature output is desired.

Format (F10.0)

After last structural time, insert a card with an integer 9 in column 20.

C-2 Blank card after last problem. (Multiple problem data sets are run consecutively, by stacking each data behind the previous one).

5.0 SUMMARY OF OUTPUT

The description of INPUTB output is here divided into two parts. The first covers output which is primarily an echo check of the input data and also includes some intermediate calculated results. This output is contained only on the first output file (UOUT1). The second part of the output consists of the final interpolated structural results to be used with BOPACE. This output is contained on both the first and second output files (UOUT1 and UOUT2).

5.1 ECHO CHECK OF INPUT DATA, AND INTERMEDIATE RESULTS

Title - The first page of INPUTB output for a problem contains the 80-character problem I.D. title input as item I-1.

Structural Node Definitions - The information given in input item I-2 is printed. Values are the structural node number, node I.D., location coordinate system number (0 = basic Cartesian, 1 = basic cylindrical, 2 = basic spherical), the location coordinates (X-Y-Z, R-Θ-Z, or R-Θ-Φ), and the direction coordinate system number (0 = basic Cartesian, 1 = basic cylindrical, 2 = basic spherical, >2 = I.D. of special system).

Thermal Node Definitions - The information given in input item I-3 is printed. Values are the thermal node number, node I.D., location coordinate system number (0 = basic Cartesian, 1 = basic cylindrical, 2 = basic spherical) and the location coordinates (X-Y-Z, R-Θ-Z, or R-Θ-Φ).

Thermal Node Time - Temperature Vectors - The information given in input item I-4 is printed. First the number of thermal times (temperature vectors) is printed, along with the initial default temperature. Then for each given thermal time is printed the time value, followed by the list of node I.D.'s and associated temperature values for each specified node.

Intermediate Interpolation Results - This output is provided for the user so that he may review, if desired, the thermal nodes and corresponding interpolation weighting factors which are associated with each structural node. Values printed are the structural node I.D., and associated four thermal node point I.D.'s and corresponding weighting factors. If only three thermal nodes were used to interpolate temperatures for a particular structural node, then the fourth weight is printed as zero and the fourth thermal node point is printed equal to the first. Similarly if only two thermal nodes were used, the third weight is also printed as zero and the third thermal point is also printed equal to the first. If only one thermal point could be used, all weights but the first are printed as zero and all thermal points are printed equal to the first.

5.2 FINAL RESULTS

These are the final interpolated structural temperatures at each specified structural time, and are written onto both the first and second output files, in BOPACE format. For each structural time the time value is printed on the first line or card (with the remainder of the card filled with asterisks to help the user identify the first card). Following this, is the vector of structural node I.D.'s and corresponding temperature values, for all structural nodes. At the end of each vector is printed a blank card as required for direct input to BOPACE.

6.0 SIZE LIMITATIONS

The following variables are used to specify maximum size limitations in INPUTB. The values set for these variables are given in Table 6-1.

NMAX2	=	maximum number of structural nodes.
NMAX4	=	maximum structural node I.D. number.
NMAX52	=	maximum number of thermal nodes.
NMAX54	=	maximum thermal node I.D. number.
NMAX81	=	maximum number of thermal node times.

TABLE 6-1: INPUTB MAXIMUM SIZE LIMITATIONS

VARIABLE	VALUE
NMAX2	1,000
NMAX4	5,000
NMAX52	500
NMAX54	2,500
NMAX81	100

7.0 INPUTB ERROR MESSAGES

INPUTB uses the FORTRAN STOP codes described in this section to indicate error conditions which may occur during execution of the program. The following are explanations of the error STOP codes, listed by subroutine in which they occur.

<u>READO</u>

9999 Normal program exit (not an error) caused by reading final blank card after all problems are run.

<u>READM</u>

701 Structural node I.D. exceeds maximum.

702 I.D. of a structural node location coordinate system not equal to 0, 1 or 2.

704 Number of structural nodes exceeds maximum.

705 No structural nodes input.

<u>READMT</u>

801 Thermal node I.D. exceeds maximum.

802 I.D. of a thernal node location coordinate system not equal to 0, 1 or 2.

804 Number of thermal nodes exceeds maximum.

805 No thermal nodes input.

<u>READT</u>

901 Undefined thermal node I.D. used to specify temperature at a thermal time.

902 Number of thermal times is less than 2 or exceeds maximum.

<u>TCOMP</u>

1001 Structural time is outside range of thermal times, or times are not in increasing order.

8.0 EXAMPLE PROBLEMS

The example problems provided are intended to familiarize the user with the INPUTB program. Two problems are provided, a 2-dimensional problem which introduces INPUTB in a more easily visualized 2-dimensional format, and a 3-dimensional problem which performs essentially the same functions in three dimensions.

8.1 2-DIMENSIONAL PROBLEM

The mesh used in this problem is shown in Figure 8.1-1. The closed circles correspond to the structural nodes and the open circles to the thermal nodes. Temperature values, at TIME = 1.0, were assigned to the thermal nodes using the equation:

$$T = 50x - 100y$$

A listing of temperature values is shown in Table 8.1-1.

Certain precautions were taken in arriving at input data for this example problem, in order to most effectively illustrate the interpolation procedure, and to guarantee that identical results would be obtained on different computer systems in the presence of small roundoff errors. For these reasons it was important that certain thermal points should not coincide exactly with any structural points, and that two thermal nodes and a structural node should not lie on the same line. It was also important that each structural node be enclosed within a triangle formed by three thermal nodes, which, although not always possible, increases the accuracy of the interpolation. Of course, the user running an actual problem needs to be less concerned with precautions such as those described here.

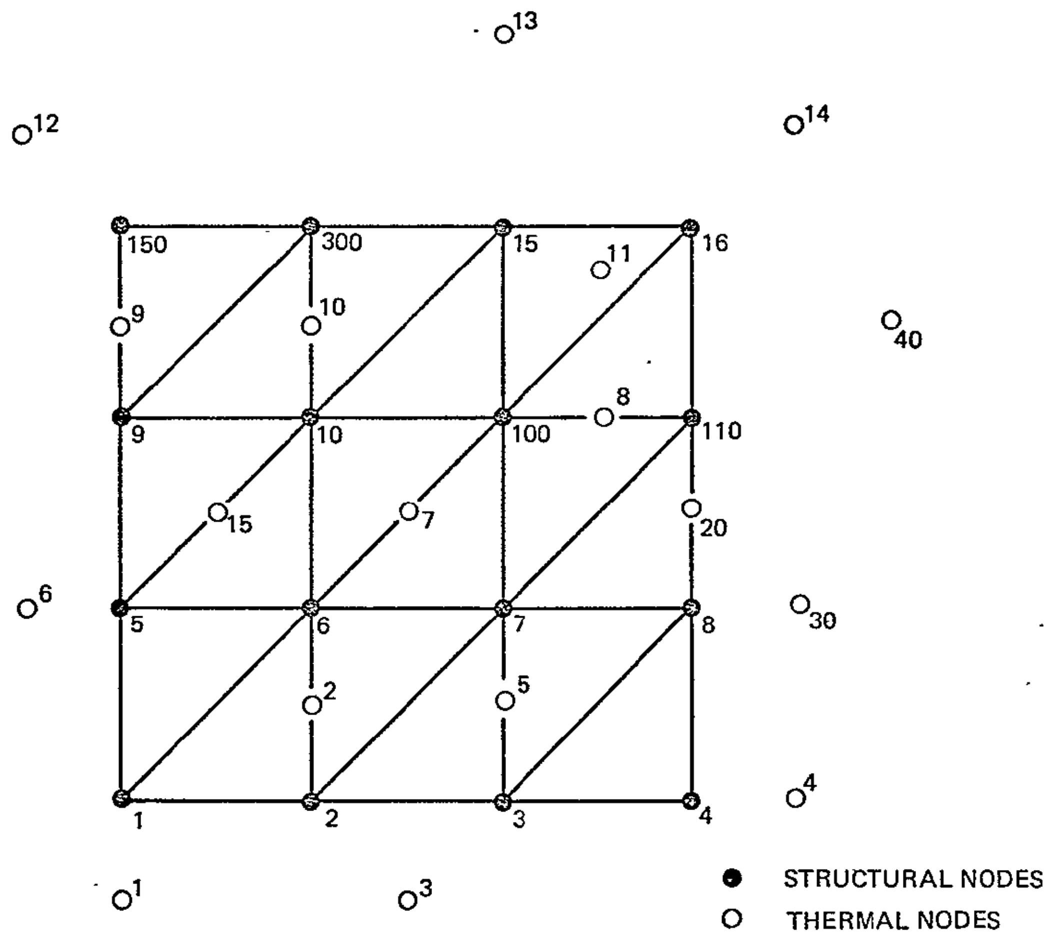

Figure 8.1-1. 2-D Problem Mesh

*Table 8.1-1. Thermal Node Temperatures for 2-D Problem at Time = 1.0**

Node	Temperature	Node	Temperature
1	50	20	0
2	0	9	-250
3	125	10	-200
4	175	8	- 75
5	50	11	-150
6	-125	40	- 50
7	- 75	12	-325
15	-125	13	-300
30	75	14	-175

*Note: Default Temperature at time 0.0 = 25.0.

```
START          5    6    7
INPUTB CHECKOUT PROBLEM (2-D MESH)                                              08/15/75
   1    0 0.000     0.000     0.000        0
   2    0 1.000     0.000     0.000        0
   3    0 2.000     0.000     0.000        0
   4    0 3.000     0.000     0.000        0
   5    0 0.000     1.000     0.000        0
   6    0 1.000     1.000     0.000        0
   7    2 2.236    26.525     0.000        0
   8    0 3.000     1.000     0.000        0
   9    0 0.000     2.000     0.000        0
  10    0 1.000     2.000     0.000        0
 100    0 2.000     2.000     0.000        0
 110    0 3.000     2.000     0.000        0
 150    0 0.000     3.000     0.000        0
 300    0 1.000     3.000     0.000        0
  15    0 2.000     3.000     0.000        0
  16    1 4.242    45.000     0.000        0

   1    0 0.000    -0.500     0.000        2    0 1.000      0.500     0.000
   3    0 1.500    -0.500     0.000        4    0 3.500      0.000     0.000
   5    0 2.000     0.500     0.000        6    0 -0.500     1.000     0.000
   7    0 1.500     1.500     0.000       30    0 3.500      1.000     0.000
  15    2 1.581    71.565     0.000       20    0 3.000      1.500     0.000
  40    0 4.000     2.500     0.000       12    0 -0.500     3.500     0.000
   8    0 2.500     2.000     0.000        9    0 0.000      2.500     0.000
  10    0 1.000     2.500     0.000       11    0 2.500      2.750     0.000
  13    0 2.000     4.000     0.000       14    1 4.950     45.000     0.000

        2 0.0
0.0

1.0
        1 50.00            3 125.00           4 175.00           5 50.00
        6 -125.00          7 -75.00          15 -125.00         30 75.00
        9 -250.00         10 -200.00          8 -75.00          11 -150.00
       40 -50.00          12 -325.00         13 -300.00         14 -175.00

.25               0
.50               0
1.0               0
                  9
```

** NODE **

NO.	I.D.	LOCATE	X1	X2	X3	DISPLACE
1	1	0	0.0	0.0	0.0	0
2	2	0	0.10000E 01	0.0	0.0	0
3	3	0	0.20000E 01	0.0	0.0	0
4	4	0	0.30000E 01	0.0	0.0	0
5	5	0	0.0	0.10000E 01	0.0	0
6	6	0	0.10000E 01	0.10000E 01	0.0	0
7	7	2	0.22360E 01	0.26525E 02	0.0	0
8	8	0	0.30000E 01	0.10000E 01	0.0	0
9	9	0	0.0	0.20000E 01	0.0	0
10	10	0	0.10000E 01	0.20000E 01	0.0	0
11	100	0	0.20000E 01	0.20000E 01	0.0	0
12	110	0	0.30000E 01	0.20000E 01	0.0	0
13	150	0	0.0	0.30000E 01	0.0	0
14	200	0	0.10000E 01	0.30000E 01	0.0	0
15	15	0	0.20000E 01	0.30000E 01	0.0	0
16	16	1	0.42420E 01	0.45000E 02	0.0	0

* T NODE *

NO.	I.D.	LOCATE	X1	X2	X3
1	1	0	0.0	-0.50000E 00	0.0
2	2	0	0.10000E 01	0.50000E 00	0.0
3	3	0	0.15000E 01	-0.50000E 00	0.0
4	4	0	0.35000E 01	0.0	0.0
5	5	0	0.20000E 01	0.50000E 00	0.0
6	6	0	-0.50000E 00	0.10000E 01	0.0
7	7	0	0.15000E 01	0.15000E 01	0.0
8	30	0	0.35000E 01	0.10000E 01	0.0
9	15	2	0.15810E 01	0.71[illegible]65E 02	0.0
10	20	0	0.30000E 01	0.15000E 01	0.0
11	40	0	0.40000E 01	0.25000E 01	0.0
12	12	0	-0.50000E 00	0.35000E 01	0.0
13	8	0	0.25000E 01	0.20000E 01	0.0
14	9	0	0.0	0.25000E 01	0.0
15	10	0	0.10000E 01	0.25000E 01	0.0
16	11	0	0.25000E 01	0.27500E 01	0.0
17	13	0	0.20000E 01	0.40000E 01	0.0
18	14	1	0.49500E 01	0.45000E 02	0.0

NUMBER OF TIMES (TEMPERATURE VECTORS) = 2   DEFAULT TEMPERATURE = 0.0

TEMPERATURE VECTOR FOR TIME = 0.0

THERMAL PT.   TEMPERATURE

TEMPERATURE VECTOR FOR TIME = 0.1000E 01

THERMAL PT.	TEMPERATURE
1	50.00
3	125.00
4	175.00
5	50.00
6	125.00
7	-75.00
15	-125.00
30	75.00
9	-250.00
10	-200.00
8	-75.00
11	-150.00
40	-50.00
12	325.00
13	-300.00
14	-175.00

NODE	PT 1	PT 2	PT 3	PT 4	WT 1	WT 2	WT 3	WT 4
1	1	2	6	1	0.6250	0.1250	0.2500	0.0
2	2	3	1	2	0.5000	0.3333	0.1667	0.0
3	5	3	4	5	0.4286	0.4286	0.1429	0.0
4	4	5	5	4	0.7143	0.1429	0.1429	0.0
5	6	15	2	6	0.6000	0.2000	0.2000	0.0
6	2	15	7	2	0.5000	0.2500	0.2500	0.0
7	5	7	20	5	0.5014	0.3320	0.1666	0.0
8	30	20	5	30	0.5000	0.2500	0.2500	0.0
9	9	15	6	4	0.6000	0.2000	0.2000	0.0
10	10	7	15	10	0.5000	0.2500	0.2500	0.0
100	8	7	11	8	0.1667	0.5000	0.3333	0.0
110	20	8	40	20	0.2500	0.5000	0.2500	0.0
150	9	12	10	9	0.2500	0.5000	0.2500	0.0
300	10	13	9	10	0.3333	0.3333	0.3333	0.0
15	11	13	10	11	0.5000	0.2500	0.2500	0.0
16	11	14	8	11	0.3337	0.4995	0.1668	0.0

8.1-6

0.2500**

1	0.0	2	12.50	3	25.00	4	37.50
5	-25.00	6	-12.50	7	0.04	8	12.50
9	-50.00	10	-37.50	100	-25.00	110	-12.50
150	-66.75	300	-62.50	15	-50.00	16	-37.49

0.5000**

1	0.0	2	25.00	3	50.00	4	75.00
5	-50.00	6	-25.00	7	0.09	8	25.00
9	-100.00	10	-75.00	100	-50.00	110	-25.00
150	-137.50	300	-125.00	15	-100.00	16	-74.99

1.0000**

1	0.0	2	50.00	3	100.00	4	150.00
5	-100.00	6	-50.00	7	0.17	8	50.00
9	-200.00	10	-150.00	100	-100.00	110	-50.00
150	-275.00	300	-250.00	15	-200.00	16	-149.97

8.2 3-DIMENSIONAL PROBLEM

The structural network for this problem consists of three parallel planes, with the structural grids stacked one on top of the other, as shown in Figure 8.2-1. The arrangement of the nodes on each respective plane is shown in Figure 8.2-2. The closed circles correspond to the structural nodes and the open circles to the thermal nodes. The temperature values at TIME = 1.0 were assigned to the thermal nodes using the equation:

$$T = 50x - 100y + 50z$$

A listing of the temperature values is shown in Table 8.2-1.

The same precautions were taken when formulating input data for this example problem as were taken for the 2-dimensional example, with additions being made for a 3-dimensional format.

Thermal nodes were placed on levels .999, 1.999, and 3.001 so that thermal nodes and structural nodes would always be in different horizontal planes. These steps were utilized in order to avoid any inconsistencies in output data due to roundoff on different computer systems.

It was also necessary to try to place the thermal nodes in such a way that each structural node was enclosed within a tetrahedron formed by four thermal nodes. When this condition could not be met the program would substitute the node found for point one in place of the missing points. This case is evident for nodes 12, 13, 30, etc., in the output data.

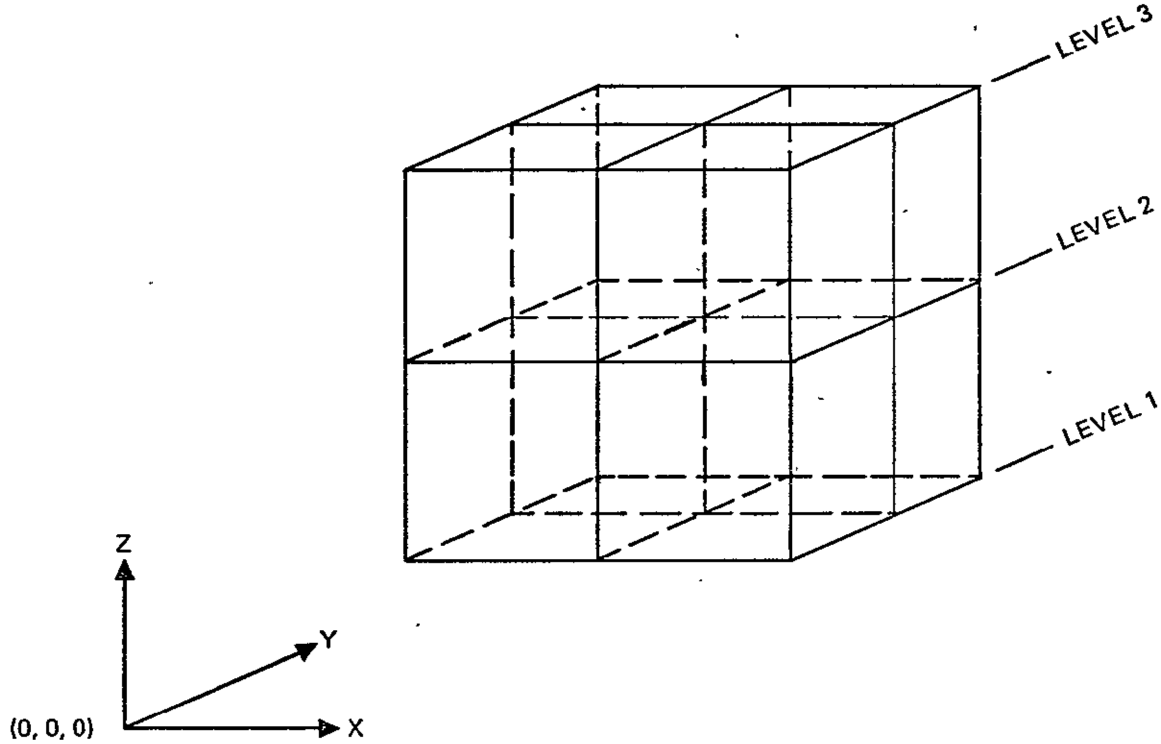

Figure 8.2-1. 3-D Problem Geometry Definition

*Table 8.2-1. Thermal Node Temperatures for 3-D Problem at Time = 1.0**

Node	Temperature	Node	Temperature
1	24.95	15	124.95
2	25.05	16	-100.10
3	- 75.05	17	75.00
4	0.00	18	- 75.10
5	62.45	19	- 25.15
6	-124.95	20	-150.00
7	- 75.05	21	175.15
8	-175.15	22	75.05
9	- 75.15	23	25.05
10	25.50	24	- 50.00
11	174.95	25	50.10
12	175.10	26	-174.95
13	- 0.05	27	-124.95
14	25.0		

*Note: Default temperature at time 0.0 = 0.0.

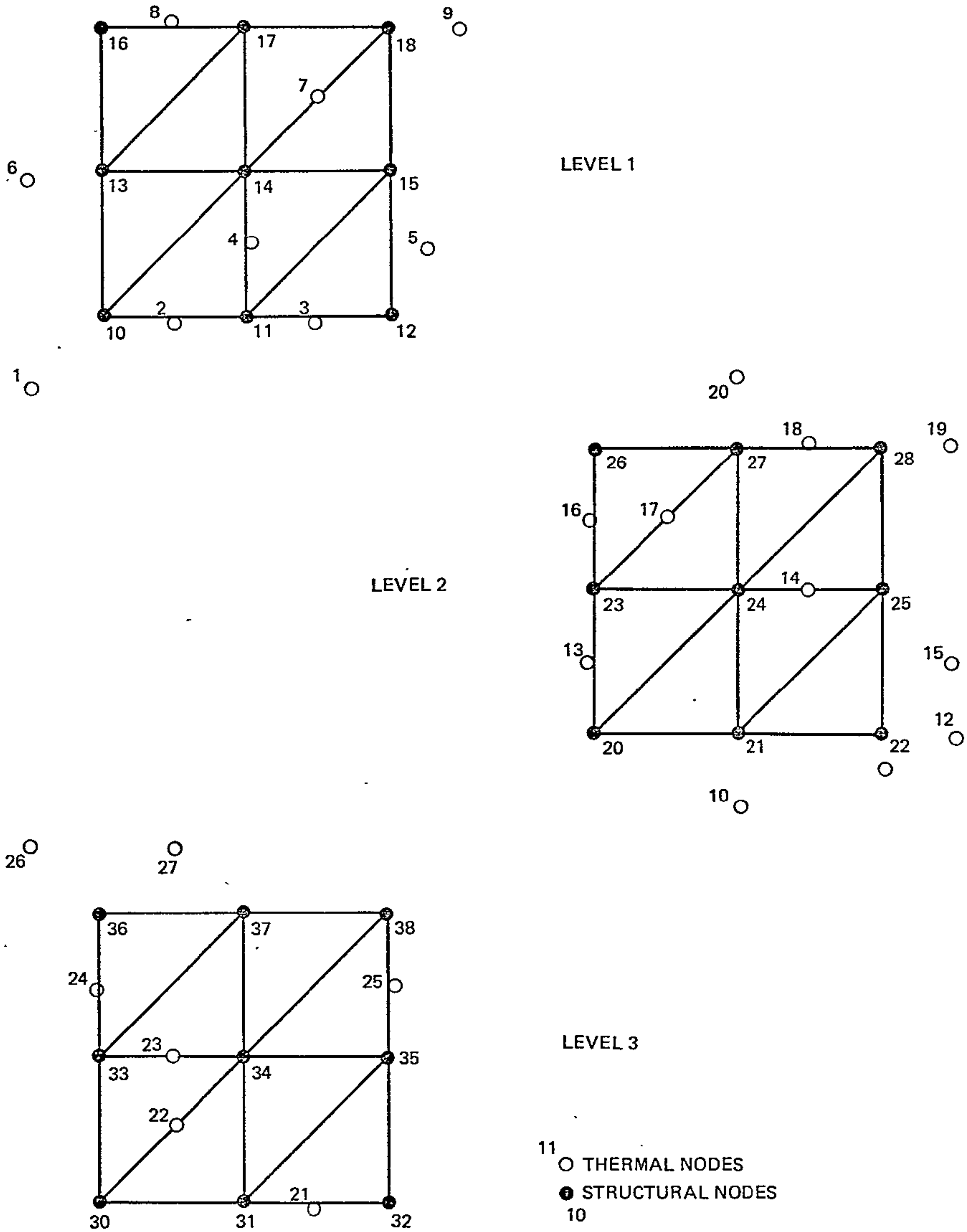

Figure 8.2-2. 3-D Problem Mesh Points

START 5 6 7

INPUTB CHECKOUT PROBLEM (3-D MESH) 08/15/75

10	0	1.000	1.000	1.000	0
11	0	2.000	1.000	1.000	0
12	0	3.000	1.000	1.000	0
13	0	1.000	2.000	1.000	0
14	0	2.000	2.000	1.000	0
15	1	3.606	33.690	1.000	0
16	0	1.000	3.000	1.000	0
17	2	3.742	56.310	15.501	0
18	0	3.000	3.000	1.000	0
20	0	1.000	1.000	2.000	0
21	0	2.000	1.000	2.000	0
22	0	3.000	1.000	2.000	0
23	0	1.000	2.000	2.000	0
24	0	2.000	2.000	2.000	0
25	0	3.000	2.000	2.000	0
26	1	3.162	71.565	2.000	0
27	0	2.000	3.000	2.000	0
28	2	4.690	45.000	25.239	0
30	0	1.000	1.000	3.000	0
31	2	3.742	26.565	53.301	0
32	0	3.000	1.000	3.000	0
33	0	1.000	2.000	3.000	0
34	1	2.828	45.000	3.000	0
35	0	3.000	2.000	3.000	0
36	0	1.000	3.000	3.000	0
37	0	2.000	3.000	3.000	0
38	0	3.000	3.000	3.000	0

1	0	0.500	0.500	0.999	2	0	1.500	0.999	0.999
3	0	2.500	0.999	0.999	4	0	2.001	1.500	0.999
5	0	3.250	1.500	0.999	6	0	0.500	1.999	0.999
7	0	2.500	2.500	0.999	8	1	3.355	63.443	0.999
9	2	4.717	40.611	12.226	10	0	2.001	0.500	1.999
11	0	3.000	0.750	1.999	12	0	3.500	0.999	1.999
13	2	2.693	56.336	47.977	14	0	2.500	2.000	1.999
15	0	3.500	1.500	1.999	16	0	0.999	2.500	1.999
17	1	2.915	59.036	1.999	18	0	2.500	3.001	1.999
19	0	3.500	3.001	1.999	20	0	2.000	3.500	1.999

21	1	2.693	21.782	3.001	22	0	1.500	1.500	3.001
23	0	1.500	2.000	3.001	24	0	0.999	2.500	3.001
25	2	4.925	39.796	37.536	26	0	0.500	3.500	3.001
27	0	1.500	3.500	3.001					

2 25.00

0.0

1.0

1	24.95	2	25.05	3	75.05	4	0.00
5	62.45	6	-124.95	7	-75.05	8	-175.15
9	-75.15	10	25.50	11	174.95	12	175.10
13	-0.05	15	124.95	16	-100.1	17	-75.00
18	-75.10	19	-25.15	20	-150.00	21	175.15
22	75.05	23	25.05	24	-50.00	25	50.10
26	-174.95	27	-124.95				

.25	0
.50	0
1.0	0
	9

** NODE **

NO.	I.D.	LOCATE	X1	X2	X3	DISPLACE
1	10	0	0.10000E 01	0.10000E 01	0.10000E 01	0
2	11	0	0.20000E 01	0.10000E 01	0.10000E 01	0
3	12	0	0.30000E 01	0.10000E 01	0.10000E 01	0
4	13	0	0.10000E 01	0.20000E 01	0.10000E 01	0
5	14	0	0.20000E 01	0.20000E 01	0.10000E 01	0
6	15	1	0.36060E 01	0.33690E 02	0.10000E 01	0
7	16	0	0.10000E 01	0.30000E 01	0.10000E 01	0
8	17	2	0.37420E 01	0.56310E 02	0.15501E 02	0
9	18	0	0.30000E 01	0.30000E 01	0.10000E 01	0
10	20	0	0.10000E 01	0.10000E 01	0.20000E 01	0
11	21	0	0.20000E 01	0.10000E 01	0.20000E 01	0
12	22	0	0.30000E 01	0.10000E 01	0.20000E 01	0
13	23	0	0.10000E 01	0.20000E 01	0.20000E 01	0
14	24	0	0.20000E 01	0.20000E 01	0.20000E 01	0
15	25	0	0.30000E 01	0.20000E 01	0.20000E 01	0
16	26	1	0.31620E 01	0.71565E 02	0.20000E 01	0
17	27	0	0.20000E 01	0.30000E 01	0.20000E 01	0
18	28	2	0.46400E 01	0.45000E 02	0.25234E 02	0
19	30	0	0.10000E 01	0.10000E 01	0.30000E 01	0
20	31	2	0.37420E 01	0.26565E 02	0.53301E 02	0
21	32	0	0.30000E 01	0.10000E 01	0.30000E 01	0
22	33	0	0.10000E 01	0.20000E 01	0.30000E 01	0
23	34	1	0.21280E 01	0.45000E 02	0.30000E 01	0
24	35	0	0.30000E 01	0.20000E 01	0.30000E 01	0
25	36	0	0.10000E 01	0.30000E 01	0.30000E 01	0
26	37	0	0.20000E 01	0.30000E 01	0.30000E 01	0
27	38	0	0.30000E 01	0.30000E 01	0.30000E 01	0

ORIGINAL PAGE IS OF POOR QUALITY

* T NODE *

NO.	I.D.	LOCATE	X1	X2	X3
1	1	0	0.50000E 00	0.50000E 00	0.99900E 00
2	2	0	0.15000E 01	0.99900E 00	0.99900E 00
3	3	0	0.25000E 01	0.99900E 00	0.99900E 00
4	4	0	0.20010E 01	0.15000E 01	0.99900E 00
5	5	0	0.32500E 01	0.15000E 01	0.99900E 00
6	6	0	0.50000E 00	0.19990E 01	0.99900E 00
7	7	0	0.25000E 01	0.25000E 01	0.99900E 00
8	8	1	0.23[illegible]50E 01	0.63443E 02	0.99900E 00
9	9	2	0.47170E 01	0.40611E 02	0.12226E 02
10	10	0	0.20010E 01	0.50000E 00	0.19990E 01
11	11	0	0.30000E 01	0.75000E 00	0.19990E 01
12	12	0	0.35000E 01	0.99900E 00	0.19990E 01
13	13	2	0.26930E 01	0.56336E 02	0.47977E 02
14	14	0	0.25000E 01	0.20000E 01	0.19990E 01
15	15	0	0.35000E 01	0.15000E 01	0.19990E 01
16	16	0	0.99900E 00	0.25000E 01	0.19990E 01
17	17	1	0.29150E 01	0.55026E 02	0.19990E 01
18	18	0	0.25000E 01	0.30010E 01	0.19990E 01
19	19	0	0.35000E 01	0.30010E 01	0.19990E 01
20	20	0	0.20000E 01	0.35000E 01	0.19990E 01
21	21	1	0.26950E 01	0.21712E 02	0.30010E 01
22	22	0	0.15000E 01	0.15000E 01	0.30010E 01
23	23	0	0.15000E 01	0.20000E 01	0.30010E 01
24	24	0	0.99900E 00	0.25000E 01	0.30010E 01
25	25	2	0.49250E 01	0.39796E 02	0.37536E 02
26	26	0	0.50000E 00	0.35000E 01	0.30010E 01
27	27	0	0.15000E 01	0.35000E 01	0.30010E 01

NUMBER OF TIMES (TEMPERATURE VECTORS) = 2 DEFAULT TEMPERATURE = 25.0000

TEMPERATURE VECTOR FOR TIME = 0.0

THERMAL PT. TEMPERATURE

TEMPERATURE VECTOR FOR TIME = 0.1000E 01

THERMAL PT.	TEMPERATURE
1	24.95
2	25.05
3	75.05
4	0.0
5	62.45
6	-124.95
7	-75.05
8	175.15
9	-75.15
10	25.50
11	174.95
12	175.10
13	-0.05
15	124.95
16	-100.10
17	-75.00
18	-75.10
19	-25.15
20	-150.00
21	175.15
22	75.05
23	25.05
24	50.00
25	50.10
26	-174.95
27	124.95

NODE	PT 1	PT 2	PT 3	PT 4	WT 1	WT 2	WT 3	WT 4
10	2	1	6	13	0.4995	0.3329	0.1666	0.0010
11	2	3	4	10	0.4980	0.4980	0.0030	0.0010
12	3	5	11	2	0.4626	0.4279	0.1095	0.0
13	6	16	15	6	0.7997	0.1011	0.0992	0.0
14	4	7	8	14	0.6000	0.1984	0.2006	0.0010
15	5	7	9	16	0.5451	0.3623	0.0916	0.0010
16	8	16	6	8	0.7494	0.0012	0.2494	0.0
17	8	7	20	9	0.7478	0.0024	0.0011	0.2487
18	9	7	16	8	0.7488	0.0016	0.0011	0.2485
20	13	2	10	15	0.7490	0.0005	0.2499	0.0
21	10	14	13	21	0.6243	0.2493	0.1256	0.0008
22	11	15	10	21	0.4001	0.3995	0.1994	0.0010
23	13	16	17	23	0.5001	0.4982	0.0015	0.0002
24	14	17	13	22	0.6000	0.1996	0.1997	0.0007
25	14	15	25	18	0.2505	0.4995	0.0010	0.2490
26	16	20	24	16	0.7489	0.2501	0.0010	0.0
27	20	17	18	27	0.3317	0.3331	0.3342	0.0010
28	18	19	14	25	0.4990	0.4993	0.0009	0.0008
30	22	1	22	22	0.8332	0.1668	0.0	0.0
31	21	22	10	21	0.5166	0.3797	0.1035	0.0
32	21	11	25	21	0.6893	0.2005	0.1102	0.0
33	23	24	13	23	0.3333	0.4996	0.1671	0.0
34	23	25	22	14	0.3339	0.3324	0.3328	0.0009
35	25	21	15	25	0.6617	0.2615	0.0768	0.0
36	24	26	27	16	0.4990	0.2495	0.2505	0.0010
37	27	23	25	20	0.5549	0.1113	0.3329	0.0009
38	25	13	19	25	0.7996	0.1007	0.0997	0.0

0.2500**

10	16.75	11	31.22	12	33.90	13	-8.76
14	6.25	15	18.75	16	-21.88	17	-18.75
18	-6.25	20	20.34	21	-24.32	22	-50.04
23	6.25	24	18.76	25	31.26	26	-9.38
27	-6.24	28	6.26	30	35.42	31	49.16
32	-59.08	33	14.59	34	31.25	35	-40.69
36	-6.25	37	6.25	38	26.25		

0.5000**

10	12.50	11	37.44	12	52.80	13	-42.52
14	-12.50	15	12.50	16	-66.77	17	-62.51
18	-37.50	20	15.67	21	23.64	22	-75.08
23	-12.50	24	12.52	25	37.51	26	-43.76
27	-37.48	28	-12.49	30	45.85	31	73.53
32	-95.17	33	4.18	34	37.51	35	-50.77
36	-37.50	37	-12.50	38	27.49		

1.0000**

10	-0.00	11	49.16	12	60.60	13	-110.04
14	-50.00	15	-0.01	16	-162.54	17	-150.01
18	-100.00	20	6.35	21	22.29	22	125.17
23	-50.00	24	0.05	25	50.02	26	-112.53
27	[illegible]	[illegible]	-49.97	30	60.09	31	171.66
32	161.33	33	-10.04	34	50.02	35	88.55
36	-100.00	37	-50.00	38	29.90		

INPUTB

PART III. PROGRAMMER MANUAL

9.0 SUBROUTINES

All the subroutines of the INPUTB program are described in this section, with their calling sequence and argument variable definitions. The main program is described in Section 3.

9.1 READO

Subroutine READO reads the problem identification card, and the input/output file unit numbers.

Call READO (UIN, UOUT1, UOUT2)

UIN	-	input file unit number (e.g., cards).
UOUT1	-	first output file unit number (e.g., printer).
UOUT2	-	second output file unit number (e.g., punch).

9.2 READM

Subroutine READM reads the structural node locations, using BOPACE 3-D data format.

Call READM (UIN, UOUT1, NOD, COORD, NODE, NMAX2, NMAX4)

UIN	-	input file unit number.
UOUT1	-	first output file unit number
NOD	-	number of structural nodes.
COORD	-	COORD (J, I) = Jth coordinate of node I..
NODE	-	NODE (I) = external node I.D. for internal node I.
NMAX2	-	maximum number of nodes.
NMAX 4	-	maximum node I.D. number.

9.3 READMT

Subroutine READMT reads the thermal node locations.

Call READMT (UIN, UOUT1, NODT, COORDT, NODET, NODIT, NMAX52, NMAX54).

UIN	-	input file unit number.
UOUT1	-	first output file unit number.
NODT	-	number of thermal nodes.
COORDT	-	COORDT (J, I) = J th coordinate of node I.
NODET	-	NODET (I) = external node I.D. for internal node I.
NODIT	-	NODIT (I) = node internal number for external I.D. I.
NMAX52	-	maximum number of thermal nodes.
NMAX54	-	maximum thermal node I.D. number.

9.4 READT

Subroutine READT reads the vector of thermal node temperatures, at each given thermal time.

Call READT (UIN, UOUT1, IOT, NTIME, TIME, NODT, NODIT, NMAX54, NMAX 81)

UIN	-	input file unit number
UOUT1	-	first output file unit number.
IOT	-	file unit number for storing thermal node temperature vectors.
NTIME	-	number of temperature vectors (time values) given for thermal nodes.
TIME	-	TIME (I) = I th time value for thermal nodes.
NODT	-	number of thermal nodes.
NODIT	-	NODIT (I) - node internal number for external I.D. I.
NMAX54	-	maximum thermal node I.D. number.
NMAX81	-	maximum number of thermal node temperature vectors (time values).

9.5 COMPW

Subroutine COMPW selects four thermal nodes and associated weighting factors for each structural node, and stores them for later interpolation.

Call COMPW (UOUT1, NOD, NODT, COORD, COORDT, NON, NOW, NODE, NODET)

UOUT1	-	first output file unit number
NOD	-	number of structural nodes.
NODT	-	number of thermal nodes.
COORD	-	COORD (J, I) - Jth coordinate of structural node I.
COORDT	-	COORDT (J, I) = Jth coordinate of thermal node I.
NON	-	NON (J, I) = selected Jth thermal node (J= 1-4) for weighting temperature of structural node I.
NOW	-	NOW (J, I) = computed J the thermal node weight (J = 1-4) for structural node I.
NODE	-	NODE (I) = external I.D. for internal structural node I.
NODET	-	NODET (I) = external I.D. for internal thermal node I.

9.6 TCOMP

Subroutine TCOMP performs the actual interpolation in space and time, to compute structural node temperatures, using the thermal node temperature vectors and the weighting factors.

Call TCOMP (UIN, UOUT1, UOUT2, IOT, NOD, NODT, NON, NOW, NTIME, TIME, NODE)

UIN	-	input file unit number
UOUT1	-	first output file unit number.
UOUT2	-	second output file unit number.

IOT	-	file unit number for storing thermal node temperature vectors.
NOD	-	number of structural nodes.
NODT	-	number of thermal nodes.
NON	-	NON (J, I) - Jth thermal node (J = 1-4) for weighting temperature of structural node I.
NOW	-	NOW (J, I) = J the thermal node weight (J = 1-4) for structural node I.
NTIME	-	number of time values (temperature vectors) for thermal nodes.
TIME	-	TIME (I) = I th time value for thermal nodes.
NODE	-	NODE (I) = external I.D. for internal structural node I.

10.0 COMMON BLOCKS

The only common used in the INPUTB program are the two labeled common blocks COMTO and COMT1, which provide storage space for thermal node temperature vectors.

Common/COMTO/TO (used in MAIN, TCOMP)

Common/COMT1/T1 (used in MAIN, READT, TCOMP)

The MAIN program contains both TO and T1, and is used to dimension these vectors. Subroutine READT uses the T1 vector as temporary storage to read in each thermal node temperature vector before writing it onto the file IOT. Subroutine TCOMP uses both the TO and T1 vectors, to read and store thermal node temperature vectors from file IOT, at successive thermal time values. These vectors are then used to interpolate all structural node temperatures within the time interval from TO to T1.

11.0 FILE USAGE

INPUTB uses FORTRAN I/O to access several files. Some of the files are defined by the user and the others are defined in the program. A list of files by logical unit number follows:

UNIT NUMBER	DESCRIPTION	DEFINED BY
5	Input card file	READO
UIN	Input data file	user
UOUT1	First output file. Contains all output results, including echo check of input data, and intermediate results.	user
UOUT2	Second output file. Contains only final output results, i.e., interpolated structural node temperature data.	user
IOT (=11)	Storage file for thermal node temperature vectors.	MAIN

APPENDIX – INPUT B PROGRAM LISTING

```fortran
C     ******************************************************************
C      I N P U T B (THERMAL/STRUCTURAL DATA INTERFACE PROGRAM)
C      INPUTB IBM 360/370 VERSION (1000 NODES) DATED 09/15/75
C      BOEING AEROSPACE COMPANY, P.O. BOX 3999, SEATTLE, WASH. 98124
C      ENGINEER/PROGRAMMER: R.G. VOS, PHONE 773-2946, BLDG. 18-05
C     ******************************************************************
C      SPACE-TIME INTERPOLATION FOR 1-, 2- OR 3-DIMENSIONAL SPACES.
C      PROBLEM SIZE CAPABILITY INCLUDES 1000 STRUCTURAL NODES,
C      500 THERMAL NODES, 100 THERMAL NODE TIMES, ARBITRARY NUMBER OF
C      STRUCTURAL NODE TIMES.
C      MAXIMUM STRUCTURAL,THERMAL NODE I.D. NUMBERS ARE 5000,2500.
C      NODES ARE LOCATED IN BASIC CARTESIAN,CYLINDRICAL OR SPHERICAL
C      COORDINATES.
       INTEGER UIN,UOUT1,UOUT2,IOT,NOD,NODT,NTIME,NMAX2,NMAX4,NMAX52,
      1NMAX54,NMAX81
       INTEGER NODE(1000),NODET(500),NODIT(2500),NON(4,1000)
       REAL TO(500),T1(500),TIME(100),COORD(3,1000),COORDT(3,500),
      1NOW(4,1000)
       COMMON/COMTO/TO/COMT1/T1
       DATA IOT/11/
       DATA NMAX2,NMAX4,NMAX52,NMAX54,NMAX81/1000,5000,500,2500,100/
C      READ PROBLEM I.D. AND FILE UNIT NUMBERS.
     1 CALL READO(UIN,UOUT1,UOUT2)
C      READ STRUCTURAL NODE DEFINITIONS.
       CALL READM(UIN,UOUT1,NOD,COORD,NODE,NMAX2,NMAX4)
C      READ THERMAL NODE DEFINITIONS.
       CALL READMT(UIN,UOUT1,NODT,COORDT,NODET,NODIT,NMAX52,NMAX54)
C      READ THERMAL NODE TIME-TEMPERATURE VECTORS.
       CALL READT(UIN,UOUT1,IOT,NTIME,TIME,NODT,NODIT,NMAX54,NMAX81)
C      SELECT THERMAL POINTS AND COMPUTE CORRESPONDING WEIGHTS TO GIVE
C      STRUCTURAL TEMPERATURES AS FUNCTIONS OF THERMAL NODE TEMPERATURES.
       CALL COMPW(UOUT1,NOD,NODT,COORD,COORDT,NON,NOW,NODE,NODET)
C      COMPUTE AND OUTPUT STRUCTURAL NODE TEMPERATURES AT GIVEN TIMES.
       CALL TCOMP(UIN,UOUT1,UOUT2,IOT,NOD,NODT,NON,NOW,NTIME,TIME,NODE)
       GO TO 1
       END

       SUBROUTINE READO(UIN,UOUT1,UOUT2)
C      READ FILE UNIT NUMBERS AND PROBLEM IDENTIFICATION
```

```
C     UIN = INPUT FILE UNIT NUMBER.
C     UOUT1,UOUT2 = OUTPUT FILE UNIT NUMBERS.
C     STOP 9999 = NORMAL PROGRAM EXIT AFTER LAST PROBLEM IS INPUT.
      INTEGER UIN,UOUT1,UOUT2
      INTEGER I,START,STAR,IDENT(20)
  101 FORMAT(A4,6X,3I5)
  102 FORMAT(20A4)
  201 FORMAT(1H1,20A4)
      DATA STAR/4HSTAR /
      READ(5,101)START,UIN,UOUT1,UOUT2
      IF(START.NE.STAR)STOP 9999
      READ(UIN,102)(IDENT(I),I=1,20)
      IF(UOUT1.GT.0)WRITE(UOUT1,201)(IDENT(I),I=1,20)
      IF(UOUT2.GT.0)WRITE(UOUT2,102)(IDENT(I),I=1,20)
      RETURN
      END


      SUBROUTINE READM(UI,UO,NOD,COORD,NODE,NMAX2,NMAX4)
C     READ STRUCTURAL NODE DATA.
C     UI,UO = INPUT,OUTPUT FILE UNIT NUMBERS.
C     NOD = NUMBER OF STRUCTURAL NODES.
C     COORD(J,I) = COORDINATES OF NODE I.
C     NODE(I) = NODE EXTERNAL I.D. FOR INTERNAL NUMBER I.
C     NMAX2 = MAXIMUM NUMBER OF NODES.
C     NMAX4 = MAXIMUM NODE I.D. NUMBER.
C     STOP 701 = STRUCTURAL NODE I.D. EXCEEDS MAXIMUM.
C     STOP 702 = I.D. OF A STRUCTURAL NODE LOCATION COORDINATE SYSTEM
C     NOT EQUAL TO 0,1 OR 2.
C     STOP 704 = NUMBER OF STRUCTURAL NODES EXCEEDS MAXIMUM.
C     STOP 705 = NO STRUCTURAL NODES INPUT.
      INTEGER UI,UO,NOD,NODE(1),NMAX2,NMAX4
      REAL COORD(3,1)
      INTEGER I,LCOORD,DCOORD
      REAL F,ANGLE,X,Y,Z
  101 FORMAT (2I5,3F10.0,I5)
  201 FORMAT(1H1,10H** NODE **/1H ,18H NO.  I.D.  LOCATE,6X,
     12HX1,11X,2HX2,11X,2HX3,8X,8HDISPLACE)
  202 FORMAT(2I5,2X,I5,3X,3(E12.5,1X),1X,I5)
      F = 3.1415927/180.
```

```
      WRITE(UO,201)
      NCD = 0
    6 READ(UI,101)I,LCOORD,X,Y,Z,DCOORD
      IF(I.LE.0)GO TO 150
      IF(I.GT.NMAX4)STOP 701
      IF(LCOORD.NE.0.AND.LCOORD.NE.1.AND.LCOORD.NE.2)STOP 702
      NOD = NOD+1
      IF(NOD.GT.NMAX2)STOP 704
      IF(UO.GT.0)WRITE(UO,202)NOD,I,LCOORD,X,Y,Z,DCOORD
      NODE(NOD) = I
      IF(LCOORD.EQ.0)GO TO 12
      IF(LCOORD.EQ.1)GO TO 11
C     SPHERICAL COORDINATES.
      ANGLE = Z*F
      Z = X*SIN(ANGLE)
      X = X*COS(ANGLE)
C     CYLINDRICAL COORDINATES.
   11 ANGLE = Y*F
      Y = X*SIN(ANGLE)
      X = X*COS(ANGLE)
C     BASIC CARTESIAN COORDINATES.
   12 COORD(1,NCD) = X
      COORD(2,NOD) = Y
      CCORD(3,NCD) = Z
      GO TO 6
  150 IF(NOD.EQ.0)STOP 705
      RETURN
      END

      SUBROUTINE READMT(UI,UO,NODT,COORDT,NODET,NODIT,NMAX52,NMAX54)
C     READ THERMAL NODE DATA.
C     UI,UO = INPUT,OUTPUT FILE UNIT NUMBERS.
C     NCDT = NUMBER OF THERMAL NODES.
C     COORDT(J,I) = COORDINATES OF NODE I.
C     NODET(I) = NODE EXTERNAL I.D. FOR INTERNAL NUMBER I.
C     NCDIT(I) = NODE INTERNAL NUMBER FOR EXTERNAL I.D. I.
C     NMAX52 = MAXIMUM NUMBER OF THERMAL NODES.
C     NMAX54 = MAXIMUM THERMAL NODE I.D. NUMBER.
C     STOP 801 = THERMAL NODE I.D. EXCEEDS MAXIMUM.
```

```
C     STOP 802 = I.D. OF A THERMAL NODE LOCATION COORDINATE SYSTEM
C     NOT EQUAL TO 0,1 OR 2.
C     STOP 804 = NUMBER OF THERMAL NODES EXCEEDS MAXIMUM.
C     STOP 805 = NO THERMAL NODES INPUT.
      INTEGER UI,UO,NODT,NODET(1),NODIT(1),NMAX52,NMAX54
      REAL COORDT(3,1)
      INTEGER I,LCOORD,K,ISTOR(2),LSTOR(2)
      REAL F,ANGLE,X,Y,Z,XSTOR(2),YSTOR(2),ZSTOR(2)
  101 FORMAT (2(2I5,3F10.0))
  201 FORMAT(1H1,10H* T NODE */1H ,18H NO.  I.D.  LOCATE,6X,
     12HX1,11X,2HX2,11X,2HX3)
  202 FORMAT(2I5,2X,I5,3X,3(E12.5,1X))
      F = 3.1415927/180.
      WRITE(UO,201)
      DO 2 I=1,NMAX54
    2 NODIT(I) = 0
      NODT = 0
    6 READ(UI,101)(ISTOR(K),LSTOR(K),XSTOR(K),YSTOR(K),ZSTOR(K),K=1,2)
      DO 8 K=1,2
      IF(ISTOR(K).NE.0)GO TO 9
    8 CONTINUE
      IF(NODT.EQ.0)STOP 805
      RETURN
    9 DO 20 K=1,2
      I = ISTOR(K)
      IF(I.LE.0)GO TO 20
      IF(I.GT.NMAX54)STOP 801
      LCOORD = LSTOR(K)
      IF(LCOORD.NE.0.AND.LCOORD.NE.1.AND.LCOORD.NE.2)STOP 802
      NODT = NODT+1
      IF(NODT.GT.NMAX52)STOP 804
      X = XSTOR(K)
      Y = YSTOR(K)
      Z = ZSTOR(K)
      IF(UO.GT.0)WRITE(UO,202)NODT,I,LCOORD,X,Y,Z
      NODET(NODT) = I
      NODIT(I) = NODT
      IF(LCOORD.EQ.0)GO TO 12
      IF(LCOORD.EQ.1)GO TO 11
C     SPHERICAL COORDINATES.
```

```
      ANGLE = Z*F
      Z = X*SIN(ANGLE)
      X = X*COS(ANGLE)
C     CYLINDRICAL COORDINATES.
   11 ANGLE = Y*F
      Y = X*SIN(ANGLE)
      X = X*COS(ANGLE)
C     BASIC CARTESIAN COORDINATES.
   12 COORDT(1,NODT) = X
      COORDT(2,NODT) = Y
      COORDT(3,NODT) = Z
   20 CONTINUE
      GO TO 6
      END


      SUBROUTINE READT(UI,UO,IOT,NTIME,TIME,NODT,NODIT,NMAX54,NMAX81)
C     READ THERMAL NODE TIME-TEMPERATURE VECTORS.
C     UI,UO = INPUT,OUTPUT FILE UNIT NUMBERS.
C     IOT = FILE UNIT NUMBER FOR THERMAL NODE TIME-TEMPERATURE VECTORS.
C     NTIME = NUMBER OF TIME VALUES FOR THERMAL NODES.
C     TIME(I) = ITH TIME VALUE FOR THERMAL NODES.
C     NODT = NUMBER OF THERMAL NODES.
C     NODIT(I) = THERMAL NODE INTERNAL NUMBER FOR EXTERNAL I.D. I.
C     NMAX54 = MAXIMUM THERMAL NODE I.D. NUMBER.
C     NMAX81 = MAXIMUM NUMBER OF THERMAL TIMES.
C     STOP 901 = UNDEFINED THERMAL NODE I.D. USED TO SPECIFY TEMPERATURE
C     STOP 902 = NUMBER OF THERMAL TIMES IS LESS THAN 2 OR EXCEEDS
C     MAXIMUM.
      COMMON/ COMT1/T1
C     T1(I) = TEMPERATURE OF ITH THERMAL NODE AT GIVEN TIME.
      INTEGER UI,UO,UOT,NTIME,NODT,NODIT(1),NMAX54,NMAX81
      REAL TIME(1),T1(1)
      INTEGER I,K,ITIME,ISTOR(4)
      REAL TEMPO,STOR(4)
  101 FORMAT(I10,F10.0)
  102 FORMAT(F10.0)
  103 FORMAT(4(I10,F10.0))
  201 FORMAT(1H1,40HNUMBER OF TIMES (TEMPERATURE VECTORS) = ,I3,
     110X,22HDEFAULT TEMPERATURE = ,F10.4)
```

```
  202 FORMAT(1H1,30HTEMPERATURE VECTOR FOR TIME = ,E12.4/
     11H0,25HTHERMAL PT.    TEMPERATURE)
  203 FORMAT(1H ,I5,7X,F12.2)
      READ(UI,101)NTIME,TEMPO
      IF(UO.GT.0)WRITE(UO,201)NTIME,TEMPO
      IF(NTIME.LT.2.OR.NTIME.GT.NMAX81)STOP 902
      REWIND IOT
      DO 100 ITIME=1,NTIME
      READ(UI,102)TIME(ITIME)
      IF(UO.GT.0)WRITE(UO,202)TIME(ITIME)
      IF(ITIME.GT.1)GO TO 13
      DO 5 I=1,NODT
    5 T1(I) = TEMPO
   13 READ(UI,103)(ISTOR(K),STOR(K),K=1,4)
      DO 15 K=1,4
      IF(ISTOR(K).NE.0)GO TO 16
   15 CONTINUE
      GO TO 99
   16 DO 20 K=1,4
      I = ISTOR(K)
      IF(I.LE.0)GO TO 20
      IF(UO.GT.0)WRITE(UO,203)I,STOR(K)
      IF(I.GT.NMAX54)STOP 901
      I = NODIT(I)
      IF(I.LE.0)STOP 901
      T1(I) = STOR(K)
   20 CONTINUE
      GO TO 13
   99 WRITE(IOT)(T1(I),I=1,NODT)
  100 CONTINUE
      RETURN
      END

      SUBROUTINE COMPW(UO,NOD,NODT,COORD,COORDT,NON,NOW,NODE,NODET)
C     SELECT THERMAL POINTS AND SET WEIGHTS FOR EACH STRUCTURAL NODE.
C     UO = OUTPUT FILE UNIT NUMBER.
C     NOD = NUMBER OF STRUCTURAL NODES.
C     NODT = NUMBER OF THERMAL NODES.
C     COORD(J,I) = JTH COORDINATE OF STRUCTURAL NODE I.
```

```
C     COORDT(J,I) = JTH COORDINATE OF THERMAL NODE I.
C     NCN(J,I) = JTH THERMAL NODE NUMBER (J=1-4) FOR STRUCTURAL NODE I.
C     NOW(J,I) = JTH THERMAL NODE WEIGHT (J=1-4) FOR STRUCTURAL NODE I.
C     NODE(I) = EXTERNAL I.D. FOR STRUCTURAL NODE I.
C     NODET(I) = EXTERNAL I.D. FOR THERMAL NODE I.
      INTEGER UO,NOD,NODT,NON(4,1),NODE(1),NODET(1)
      REAL COORD(3,1),COORDT(3,1),NOW(4,1)
      INTEGER INOD,I,J,I1,I2,I3,I4
      REAL C,R,W,W1,W2,W3,W4,P(3),PO(3),P1(3),P2(3),P3(3),P4(3),
     1S(3),S12(3),S23(3)
  201 FORMAT(1H1,5H NODE,1X,4(4H  PT,I2),4(6X,2HWT,I2))
  202 FORMAT(I5,2X,4(I5,1X),2X,4F10.4)
      IF(UO.GT.0)WRITE(UO,201)(J,J=1,4),(J,J=1,4)
      DO 1000 INOD=1,NOD
      DO 5 J=1,3
    5 PO(J) = COORD(J,INOD)
C     LOCATE 1ST THERMAL PT AS NEAREST PT
      R = 10.E30
      DO 500 I=1,NODT
      DO 456 J=1,3
  456 P(J) = COORDT(J,I) - PO(J)
      IF(P(1)**2+P(2)**2+P(3)**2.GE.R)GO TO 500
      I1 = I
      R = P(1)**2+P(2)**2+P(3)**2
  500 CONTINUE
      IF(R.EQ.0.)GO TO 951
      DO 512 J=1,3
  512 P1(J) = COORDT(J,I1) - PO(J)
C     LOCATE 2ND THERMAL PT AS NEAREST PT AT LEAST 90 DEGREES FROM 1ST
      R = 10.E30
      I2 = 0
      DO 600 I=1,NODT
      DO 556 J=1,3
  556 P(J) = COORDT(J,I) - PO(J)
      IF(P(1)**2+P(2)**2+P(3)**2.GE.R.OR.I.EQ.I1)GO TO 600
      IF(P(1)*P1(1)+P(2)*P1(2)+P(3)*P1(3).GT.0.)GO TO 600
      I2 = I
      R = P(1)**2+P(2)**2+P(3)**2
  600 CONTINUE
      IF(I2.EQ.0)GO TO 951
```

```
      DO 612 J=1,3
      P2(J) = COORDT(J,I2) - PO(J)
  612 S12(J) = P2(J) - P1(J)
C     LOCATE 3RD THERMAL PT AS NEAREST PT FORMING ENCLOSING TRIANGLE
C     FOR PERPENDICULAR TO PLANE
      R = 10.E30
      I3 = 0
      DO 700 I=1,NODT
      DO 656 J=1,3
  656 P(J) = COORDT(J,I) - PO(J)
      IF(P(1)**2+P(2)**2+P(3)**2.GE.R.OR.I.EQ.I1.OR.I.EQ.I2)GO TO 700
      DO 662 J=1,3
  662 S23(J) = P(J) - P2(J)
      S(1) = S12(2)*S23(3)-S12(3)*S23(2)
      S(2) = S12(3)*S23(1)-S12(1)*S23(3)
      S(3) = S12(1)*S23(2)-S12(2)*S23(1)
      IF(S(1).EQ.0..AND.S(2).EQ.0..AND.S(3).EQ.0.) GO TO 700
      W1 = S(1)*(P2(2)*P(3)-P2(3)*P(2))+S(2)*(P2(3)*P(1)-P2(1)*P(3))
     1+S(3)*(P2(1)*P(2)-P2(2)*P(1))
      W2 = S(1)*(P(2)*P1(3)-P(3)*P1(2))+S(2)*(P(3)*P1(1)-P(1)*P1(3))
     1+S(3)*(P(1)*P1(2)-P(2)*P1(1))
      W3 = S(1)*(P1(2)*P2(3)-P1(3)*P2(2))+S(2)*(P1(3)*P2(1)-P1(1)*P2(3))
     1+S(3)*(P1(1)*P2(2)-P1(2)*P2(1))
      IF((W1.LE.0..AND.W2.LE.0..AND.W3.LE.0.).OR.
     1(W1.GE.0..AND.W2.GE.0..AND.W3.GE.0.))GO TO 690
      GO TO 700
  690 I3 = I
      R = P(1)**2+P(2)**2+P(3)**2
  700 CONTINUE
      IF(I3.EQ.0)GO TO 961
      DO 712 J=1,3
  712 P3(J) = COORDT(J,I3) - PO(J)
C     LOCATE 4TH THERMAL PT AS NEAREST PT FORMING ENCLOSING TETRAHEDRON
      R = 10.E30
      I4 = 0
      DO 800 I=1,NODT
      DO 756 J=1,3
  756 P(J) = COORDT(J,I) - PO(J)
      IF(P(1)**2+P(2)**2+P(3)**2.GE.R.OR.I.EQ.I1.OR.I.EQ.I2.OR.I.EQ.I3)
     1GO TO 800
```

```
      W1 = P(1)*(P2(2)*P3(3)-P2(3)*P3(2))
     1+P(2)*(P2(3)*P3(1)-P2(1)*P3(3))+P(3)*(P2(1)*P3(2)-P2(2)*P3(1))
      W2 =-P1(1)*(P3(2)*P(3)-P3(3)*P(2))
     1-P1(2)*(P3(3)*P(1)-P3(1)*P(3))-P1(3)*(P3(1)*P(2)-P3(2)*P(1))
      W3 = P2(1)*(P(2)*P1(3)-P(3)*P1(2))
     1+P2(2)*(P(3)*P1(1)-P(1)*P1(3))+P2(3)*(P(1)*P1(2)-P(2)*P1(1))
      W4 =-P3(1)*(P1(2)*P2(3)-P1(3)*P2(2))
     1-P3(2)*(P1(3)*P2(1)-P1(1)*P2(3))-P3(3)*(P1(1)*P2(2)-P1(2)*P2(1))
      IF((W1.LE.0..AND.W2.LE.0..AND.W3.LE.0..AND.W4.LE.0.).OR.
     1(W1.GE.0..AND.W2.GE.0..AND.W3.GE.0..AND.W4.GE.0.))GO TO 790
      GO TO 800
  790 I4 = I
      R = P(1)**2+P(2)**2+P(3)**2
  800 CONTINUE
      IF(I4.EQ.0)GO TO 971
      DO 812 J=1,3
  812 P4(J) = CCORDT(J,I4) - P0(J)
C     4 THERMAL PTS CAN BE USED
      W1 = P4(1)*(P2(2)*P3(3)-P2(3)*P3(2))
     1+P4(2)*(P2(3)*P3(1)-P2(1)*P3(3))+P4(3)*(P2(1)*P3(2)-P2(2)*P3(1))
      W2 =-P1(1)*(P3(2)*P4(3)-P3(3)*P4(2))
     1-P1(2)*(P3(3)*P4(1)-P3(1)*P4(3))-P1(3)*(P3(1)*P4(2)-P3(2)*P4(1))
      W3 = P2(1)*(P4(2)*P1(3)-P4(3)*P1(2))
     1+P2(2)*(P4(3)*P1(1)-P4(1)*P1(3))+P2(3)*(P4(1)*P1(2)-P4(2)*P1(1))
      W4 =-P3(1)*(P1(2)*P2(3)-P1(3)*P2(2))
     1-P3(2)*(P1(3)*P2(1)-P1(1)*P2(3))-P3(3)*(P1(1)*P2(2)-P1(2)*P2(1))
      W = W1+W2+W3+W4
      IF(W.EQ.0.)GO TO 971
      W1 = W1/W
      W2 = W2/W
      W3 = W3/W
      W4 = W4/W
      GO TO 991
C     ONLY 1 THERMAL PT CAN BE USED
  951 I2 = I1
      I3 = I1
      I4 = I1
      W1 = 1.0
      W2 = 0.
      W3 = 0.
```

```
      W4 = 0.
      GC TO 991
C     ONLY 2 THERMAL PTS CAN BE USED
  961 I3 = I1
      I4 = I1
      W3 = 0.
      W4 = 0.
      W = 0.
      R = 0.
      DO 965 J=1,3
      W = W + P2(J)*(P2(J)-P1(J))
  965 R = R + (P2(J)-P1(J))**2
      IF(R.EQ.0.)GO TO 951
      W1 = W/R
      W2 = 1.0-W1
      GO TO 991
C     ONLY 3 THERMAL PTS CAN BE USED
  971 I4 = I1
      W4 = 0.
      DO 975 J=1,3
  975 S23(J) = P3(J) - P2(J)
      S(1) = S12(2)*S23(3)-S12(3)*S23(2)
      S(2) = S12(3)*S23(1)-S12(1)*S23(3)
      S(3) = S12(1)*S23(2)-S12(2)*S23(1)
      IF(S(1).EQ.0..AND.S(2).EQ.0..AND.S(3).EQ.0.) GO TO 961
      W1 = S(1)*(P2(2)*P3(3)-P2(3)*P3(2))+S(2)*(P2(3)*P3(1)-P2(1)*P3(3))
     1+S(3)*(P2(1)*P3(2)-P2(2)*P3(1))
      W2 = S(1)*(P3(2)*P1(3)-P3(3)*P1(2))+S(2)*(P3(3)*P1(1)-P3(1)*P1(3))
     1+S(3)*(P3(1)*P1(2)-P3(2)*P1(1))
      W3 = S(1)*(P1(2)*P2(3)-P1(3)*P2(2))+S(2)*(P1(3)*P2(1)-P1(1)*P2(3))
     1+S(3)*(P1(1)*P2(2)-P1(2)*P2(1))
      W = W1+W2+W3
      IF(W.EQ.0.)GO TO 961
      W1 = W1/W
      W2 = W2/W
      W3 = W3/W
  991 NCN(1,INOD) = I1
      NON(2,INOD) = I2
      NCN(3,INOC) = I3
      NCN(4,INOC) = I4
```

```
      NOW(1,INOD) = W1
      NOW(2,INOD) = W2
      NOW(3,INOD) = W3
      NOW(4,INOD) = W4
      IF(UO.GT.0)WRITE(UO,202)NODE(INOD),NODET(I1),NODET(I2),NODET(I3),
     1NODET(I4),W1,W2,W3,W4
 1000 CONTINUE
      RETURN
      END


      SUBROUTINE TCOMP(UI,UO1,UO2,IOT,NOD,NODT,NON,NOW,NTIME,TIME,NODE)
C     COMPUTE AND OUTPUT STRUCTURAL NODE TEMPERATURES AT GIVEN TIMES.
C     UI,UO1,UO2 = INPUT,OUTPUT FILE UNIT NUMBERS.
C     IOT = FILE UNIT NUMBER FOR THERMAL NODE TIME-TEMPERATURE VECTORS.
C     NOD = NUMBER OF STRUCTURAL NODES.
C     NODT = NUMBER OF THERMAL NODES.
C     NON(J,I) = JTH THERMAL NODE NUMBER (J=1-4) FOR STRUCTURAL NODE I.
C     NOW(J,I) = JTH THERMAL NODE WEIGHT (J=1-4) FOR STRUCTURAL NODE I.
C     NTIME = NUMBER OF TIME VALUES FOR THERMAL NODES.
C     TIME(I) = ITH TIME VALUE FOR THERMAL NODES.
C     NODE(I) = EXTERNAL I.D. FOR STRUCTURAL NODE I.
C     STOP 1001 = STRUCTURAL TIME IS OUTSIDE RANGE OF THERMAL TIMES, OR
C     TIMES ARE NOT IN INCREASING ORDER.
      COMMON/COMT0/T0/COMT1/T1
C     T0(I),T1(I) = TEMPERATURES OF ITH THERMAL NODE AT TIMES 0,1.
      INTEGER UI,UO1,UO2,IOT,NOD,NODT,NON(4,1),NTIME,NODE(1)
      REAL NOW(4,1),TIME(1),T0(1),T1(1)
      INTEGER I,J,K,L,II,ISTOP,IT,ISTOR(4)
      REAL STIME,C,W,TIM0,TIM1,F0,F1,STOR(4)
  101 FORMAT(F10.0,I10)
  201 FORMAT(F10.4,70(1H*))
  202 FORMAT(4(I10,F10.2))
  203 FORMAT(1H )
  205 FORMAT(1H1)
      IF(UO1.GT.0)WRITE(UO1,205)
      REWIND IOT
      IT = 2
      READ(IOT)(T1(I),I=1,NODT)
      DO 2 I=1,NODT
```

```
    2 TO (I) =T1(I)
      READ(IOT)(T1(I),I=1,NODT)
    1 READ(UI,101)STIME,ISTOP
      IF(ISTOP.NE.0)GO TO 501
      IF(UO1.GT.0)WRITE(UO1,201)STIME
      IF(UO2.GT.0)WRITE(UO2,201)STIME
      IF(STIME.LT.TIME(IT-1))STOP 1001
    3 IF(STIME.LE.TIME(IT))GO TO 9
      IT = IT+1
      IF(IT.GT.NTIME)STOP 1001
      DO 6 I=1,NODT
    6 TO(I) = T1(I)
      READ(IOT)(T1(I),I=1,NODT)
      GO TO 3
    9 TIMO = TIME(IT-1)
      TIM1 = TIME(IT)
      FO = TIM1-TIMO
      IF(FO.GT.0.)FO = (TIM1-STIME)/FO
      F1 = 1.0-FO
      I = 0
   11 L = 0
      DO 20 K=1,4
      I = I+1
      IF(I.GT.NOD)GO TO 20
      L = L+1
      ISTOR(L) = NODE(I)
      C = 0.
      DO 15 J=1,4
      II = NON(J,I)
      W = NOW(J,I)
   15 C = C + W*(FO*TO(II)+F1*T1(II))
      STOR(L) = C
   20 CONTINUE
      IF(L.NE.0)GO TO 51
      IF(UO1.GT.0)WRITE(UO1,203)
      IF(UO2.GT.0)WRITE(UO2,203)
      GO TO 1
   51 IF(UO1.GT.0)WRITE(UO1,202)(ISTOR(K),STOR(K),K=1,L)
      IF(UO2.GT.0)WRITE(UO2,202)(ISTOR(K),STOR(K),K=1,L)
      GO TO 11
  501 RETURN
      END
```