DEPARTMENT OF AEROSPACE ENGINEERING
COLLEGE OF ENGINEERING & TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529

THERMAL STRUCTURE ANALYSES FOR CSM TESTBED (COMET)

By

David Y. Xue

and

Chuh Mei, Principal Investigator

Final Report
For the period May 16, 1992 to August 15, 1994

Prepared for
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

Under
Research Grant NAG-1-1431
Dr. John T. Wang, Technical Monitor
SMD-Computational Mechanics Branch

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, VA 23508

October 1994

CONTENTS

Thermal Structure Analyses for CSM Testbed (COMET)

This document is the final report for the project entitled "Thermal Structure Analyses for CSM Testbed (COMET)," for the period of May 15, 1992 - August 15, 1994. The project was focused on the investigation and development of finite element analysis capability of the computational Structural Mechanics (CSM) Testbed (COMET) software system in the field of thermal structural responses. The stages of this project consisted of investigating present capabilities, developing new functions, analysis demonstrations and research topics. The Appendices of this report listed the detailed documents of major accomplishments and demonstration runstreams for future references.

1. Summary of Accomplishments

* Implemented thermal stress analysis capability for solid element ES3
* Confirmed the thermal stress analysis capability of shell elements ES1, ES5 and ES7 by comparing results with classical solution or EAL results. Table I shows the thermal stress analysis capability in COMET ESi elements:

Table I

ES1	ES2	ES3	ES4	ES5	ES6	ES7	ES8	ES10
Shell	Shell	Solid	Shell	Shell	Beam	Shell	Shell	Solid
Yes	No	Yes	No	Yes	Yes	Yes	No	No

* Confirmed thermal stress analysis of composites in COMET.
* Demonstrated temperature program input (Appendix I).
* Completed temperature input from PATRAN to COMET by adding a subroutine GCPTEMP in PT2T. All kinds of temperatures which can be generated in PATRAN can be transferred to COMET (Appendix II).
* Developed innovative PATRAN temperature input for temperature varying across the thickness (multi-values of temperature at one node.) All kinds of multi-values of temperatures of COMET formats can be generated in PATRAN then transferred to COMET runstream (Appendix II).
* Integrated thermal / structural analyses. Using different elements and meshes in thermal and structural analyses according to the different thermal and structural loading and boundary conditions (Appendix III).

* Developed temperature transformation between different meshes (Appendix III).

2. Some Existing Problems

1) The incomplete thermal stress analysis capability for elements ES2, ES4, ES8 and ES10.

2) The incompleteness of PATRAN input.

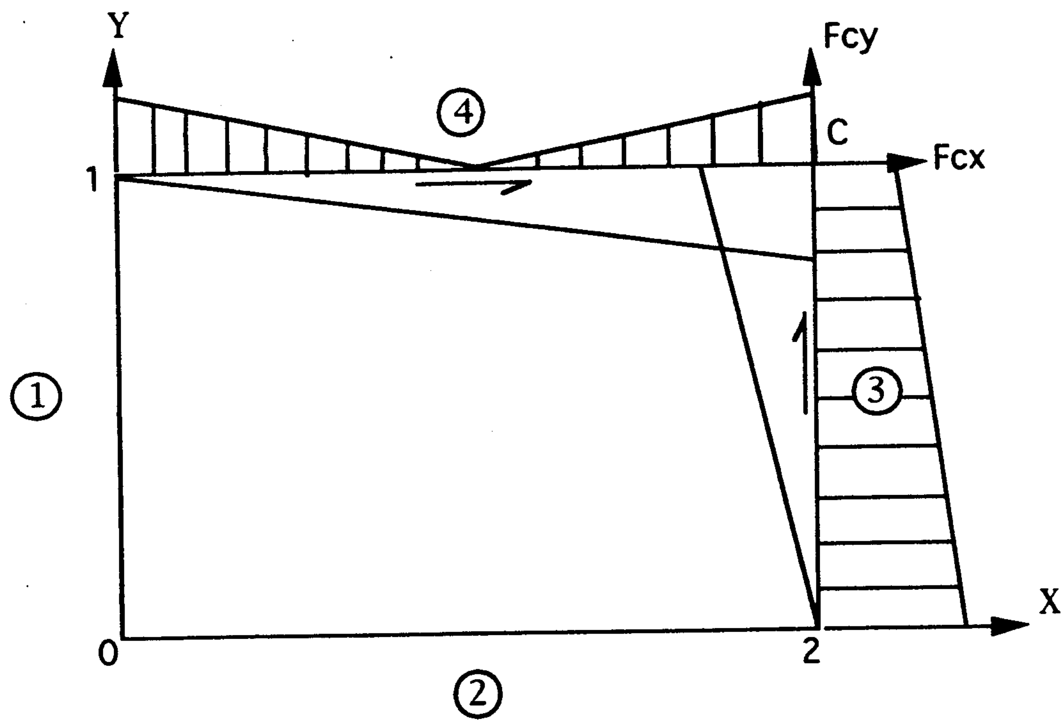
PATRAN is a powerful tool for the pre-processing of finite element analysis specially for complex structures and applied loads. It has been widely accepted by industry and research institutions. If we have a well completed PATRAN interface, it can be expected that more users will use COMET. By using PATRAN input user can obtain a COMET runstream, this feature greatly helps COMET users. However, the translator PT2T has not been well completed, users still need to do some modifications (specially in material properties) on the runstream produced from PT2T.

3) Need of a mini-manual.

COMET needs a mini-manual to lead new users to quickly start using COMET.

Appendix I Programmed Temperatures

This appendix shows an example of programmed temperature input using GCP processor. A rectangular isotropic plate is subjected to a combined mechanical and thermal loads as shown in Fig. 1. The results compared well with EAL results and classical solutions.



Boundary conditions: u=v=0 at edges 1 and 2

Temperature and Mechanical loads:

Case 1 T=x+y

Edge 4:	Fx= 0.76923+0.76923x	Fy=-0.76923+0.76923x
Edge 3:	Fx=-1.53850+0.76923y	Fy= 1.53850+0.76923y
Node C	Fcx=0.769231,	Fcy=1.538462

Case 2 $T=x^2+y^2$

Edge 4:	Fx= 0.0	$Fy=-0.0027778+3.3333x^2$
Edge 3:	$Fx=-0.0027778-3.3333y^2$	Fy= 0.0
Node C	Fcx=-1.7542,	Fcy=6.5014

Fig. 1 Rectangular plate subjected combined thermal and mechanical loading

*** COMET runstream**

(programmed temperature load)

```
cp /csm/prc/proclib.gal .
chmod +w proclib.gal
time testbed << \endinput
*open/new 1, case1.DBC
*open 28, 'proclib.gal'
*set plib = 28
*set echo,off
*def/i ES_GCP=<true>
*add rec1.clp
.
.   *set echo on, ma, md
*call X1
*eof
endinput
```

```
*procedure X1
. Initial Model Def. Procedure
*call ES (function = 'DEFINE ELEMENTS'; es_proc = ES1; es_name = EX47)
[xqt TAB
  START 231 3 4 5 6
  JLOC
  1 0. 0. 0. 0. 1. 0. 11 1 21
  11 2. 0. 0. 2. 1. 0.
.
[xqt GCP
    fabrication
      shell
        fabid = 1
        nlayers = 1
        matid = 1
        thick = 1.0
        intpts = 1
        angle = 0.0
      end
    endfab
    material
      isoel 1 1 1
      1.0E7 0.3 0.0 1.0E-6 0.0 0.0 0.0
    endmat
.
[xqt ELD
 <ES_EXPE_CMD>
   NSECT = 1
    1 12 13 2 1 20 10
.
[xqt TAB
    CON 1
    ZERO 1 2: 1,11: 12,221,11
.  *set echo on, ma, md
[xqt ES1
*def/a es_name = EX47
*do $k =1, 2
  *def/i  es_load_set = <$k>
DEFINE TEMPERATURE
          *find 1, DEF.* /seq=dsn
          *rem DEF.* : dsn=<dsn>
          *find 1, JLOC.BTAB.2.5 /seq=dsnjloc
          *rem JLOC.BTAB.2.5 : dsnjloc=<dsnjloc>
          *do $j=1, 200              .    200 Elements
            ELEMENT = <$j>
             *do $i = 12, 15         .   Node #1 - #4 in DEF.Ex47.
              *def/i ioff1 = <$i>
              *gal2mac  /Name=nnode /Ioff=<ioff1> --
                        /Maxn=1 1, <dsn> DATA.<$j>
            NODE = <<$i>-11>
              *def/i ioff2 = <<<<nnode>-1>*3>>
         *gal2mac /Name=coord[1:2] /Ioff=<ioff2> /Maxn=2 1, <dsnjloc> --
                       DATA.1
              *def/f   x=<coord[1]>
              *def/f   y=<coord[2]>
          *if <<$k> /eq 1 > /then
                  T = <<x>+<y>>            .    T=x+y
          *else
                  T = <<<x>*<x>>-<<y>*<y>>>
          *endif
             *enddo
          *enddo
    END_DEFINE_TEMPERATURE
          *enddo
.
[xqt AUS
    SYSVEC: APPL FORC 1
    CASE 1
         I=1, 2
         *do $i=1, 19
             J=<11+<11*<$i>>>: <.76923+<.076923*<$i>>>, <-.7692308+  --
               <.076923*<$i>>>
         *enddo
         *do $i=1, 9
             J=<221+<$i>>: <-1.538501+<.076923*<$i>>>, <1.5385+    --
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                <.076923*<$i>>>
        *enddo
            J=231:       .769231,    1.538462
    SYSVEC: APPL FORC 2
    CASE 1
        I=2
        *do $i=1, 19
        *def/f xx=<<<$i>*<$i>>*0.01>
            J=<11+11*<$i>>: <0.0027778+<3.3333*<xx>>>
        *enddo
        I=1
        *do $i=1, 9
        *def/f yy=<<<$i>*<$i>>*0.01>
            J=<221+<$i>>: <0.0027778-<3.3333*<yy>>>
        *enddo
        I=1 2: J=231: -1.7542, 6.5014
stop

 *call L_STATIC (location=centroids; nval_meth=1; stress=<true>;  --
                print=<true>; load_set=1)
 *call L_STATIC (location=centroids; nval_meth=1; stress=<true>;  --
                print=<true>; load_set=2)

 *fopen 8 result.all
 *set unit prt=8
*remark
*remark RESULTS
*remark
*do $k =1, 2
[xqt VPRT
lines 70
*remark
*remark
*remark          DISPLACEMENTS (U=1, V=2)
*if <<$k> /eq 1 > /then
    *remark (TEMPERATURE DISTRIBUTION T(X,Y)=X+Y AND LOAD CASE 1)
*else
    *remark (TEMPERATURE DISTRIBUTION T(X,Y)=X*X-Y*Y AND LOAD CASE 2)
*endif
*remark
        jointk 2,222,22: 6,226,22: 9,229,22: 11,231,22
        print STAT DISP <$k>
*remark
*remark
[xqt PESR
   select/line=100
*remark
*remark
*remark       STRESSES AT ELEMENT CENTERS
*if <<$k> /eq 1 > /then
    *remark (TEMPERATURE DISTRIBUTION T(X,Y)=X+Y AND LOAD CASE 1)
*else
    *remark (TEMPERATURE DISTRIBUTION T(X,Y)=X*X-Y*Y AND LOAD CASE 2)
*endif
*remark
   print /id=1:19:2,20,61:79:2,80,121:139:2,140,181:199:2,200  --
         /load_set=<$k>
    stop
*enddo
*fclose 8
*set unit prt=6
*end
```

ORIGINAL PAGE IS
OF POOR QUALITY

Appendix II PATRAN Temperature Input

This appendix states the update of PATRAN-COMET temperature transformation capability of processor PT2T. As reported before, the processor PT2T has been improved to be able to transfer temperature distribution from PATRAN to COMET. Any kinds of temperature distribution which can be generated in PATRAN (into neutral files) by using "NODE TEMPERATURES" can be transformed into COMET. But PATRAN does not generate temperature across the thickness for shell or beam elements, because such temperature distribution requires more then one temperature value at one element node and PATRAN can only generate one temperature value at one node.

In the thermal stress analysis of COMET, the temperature distributions across the thickness of shell or beam elements are inputted in various formats using DEFINE TEMPERATURE. These formats include: (see section II.1)

CONSTANT ------- one value at one node,
LINEAR -------- two or three values at one node,
QUADRATIC ------ three values at one node,
CUBIC -------------- four values at one node,
EXPONENTIAL --- three values at one node,
PWLIN ----------- NL+1 values at one node, and
TABLE ------------ NT*2 values at one node

To transfer such kinds of temperature distributions from PATRAN to COMET, an alternate transformation has been developed and the subroutine GCPTEMP in PT2T has been further updated to have the capability to transfer all above formats for the temperature distribution across the thickness of shell and beam elements form PATRAN to COMET. To apply these temperature transformation certain temperature input formats have to be followed in PATRAN temperature generation. The basic idea of this new temperature transformation is to define different sets of temperatures in PATRAN using designed formats. Then to combine them in the translator processor PT2T and obtain the COMET runstream for the temperature distribution across the thickness.

The detailed steps of how to use PT2T to transfer temperature from PATRAN to COMET are presented in section II.2. The updated processor PT2T does not affect the original temperature distribution transformation. If in a structure temperature does not varying across the thickness, user does not need to use these designed formats to input temperatures and just generates temperatures using PATRAN formats. The feature of this new PATRAN temperature transformation is stated in the part a) of section II.3. The running steps of thermal stress analysis on COMET using PATRAN input is summarized in the part b) of section II.3.

Two examples, a temperature distribution without variation across the thickness is given in section II.4 and a temperature distribution with different formats of variation across the thickness is given in section II.5.

The updated subroutine GCPTEMP is listed in section II.6.

II.1

COMET Temperature Input Formats

Table A.1 Temperature Input Formats

Format	Solid		Layered Shell		Beam	
CONST*	$\bar{T}$	✓	$\bar{T}$	✓	$\bar{T}$	✓
LINEAR	—		T_{top}, T_{bot}	✓	$\bar{T}, \frac{\partial T}{\partial y}, \frac{\partial T}{\partial z}$	✓
PWLIN	—		$T_1, T_2, \ldots, T_{NL+1}$	✓	—	
QUADRATIC	—		T_0, T_1, T_2	✓	—	
CUBIC	—		T_0, T_1, T_2, T_3		—	
EXPON	—		a, b, c $(T(z) = ae^{b(z-c)})$		—	
TABLE	—		$T_1, z_1,\ T_2, z_2,\ \ldots,\ T_{NT}, z_{NT}$	✓	—	

Key	Meaning
*	Default qualifier
✓	Currently implemented
$\bar{T}$	Average (ref. surface) temperature
T_{top}	Top surface temperature
T_{bot}	Bottom surface temperature
NL	Number of layers
NT	Number of tabulated points

For the **/LINEAR** format, the shell through-thickness temperature distribution is computed as follows:

$$T(z) = \frac{T_{top} + T_{bot}}{2} + z\,\frac{T_{top} - T_{bot}}{h}$$

while the beam through-thickness temperature distribution is computed as

$$T(y,z) = \overline{T} + y\frac{\partial T}{\partial y} + z\frac{\partial T}{\partial z}$$

For the **/PWLIN** format, the shell temperatures are interpolated linearly within each layer, using the layer interface temperatures, $T_i (i = 1, NL)$.

For the **/QUADRATIC** format, the shell through-thickness temperature distribution is computed as:

$$T(z) = T_0 + T_1 z + T_2 z^2$$

For the **/CUBIC** format, the shell through-thickness temperature distribution is computed as:

$$T(z) = T_0 + T_1 z + T_2 z^2 + T_3 z^3$$

For the **/EXPONENTIAL** format, the shell through-thickness temperature distribution is computed via:

$$T(z) = a e^{b(z-c)}$$

Finally, for the **/TABLE** format, the shell temperatures are tabulated at the through-thickness coordinates, $z_1, z_2, \ldots, z_{NT}$, and piecewise linearly interpolated between these points.

Note that the thickness coordinate z, appearing in the above definitions, is measured relative to the shell *mid*-surface, and h is the shell thickness.

The structural temperature distribution may be defined with the **DEFINE TEMPERATURE** command at any time after the element connectivity has been defined

II.2

Temperature transformation from PATRAN to COMET

1) Node temperature variations only

The word " node temperature variations only" refers: (a) solid elements have temperatures defined on nodes only, or (b) shell or beam elements have temperatures defined in inplane nodes only. That means each node has only one temperature value.

There is no special requirements for these temperature generations, user can just follow PATRAN manual to generate temperature distributions. Any kind of temperature distribution which can be generated (into neutral file) in PATRAN by using "NODE TEMPERATURES" can be transformed into COMET runstream. The new subroutine in PT2T also transfers different temperature load sets defined in PATRAN. The only restriction is:

The number of temperature load set < 10000.

It is recommended to choose load set number starting with 1, 2, 3, ... sequentially.

2) Inplane and across the thickness temperature variations for shell or beam elements.

Temperate, which varies not only in inplane but also across the thickness of shell or beam elements, requests to transfer more then one values at one node of the shell or beam elements. New subroutine in PT2T transfers those temperatures according to the formats provided in COMET (see Attachment I). This transformation requires that user follows designed load set numbers to generate temperature load in PATRAN.

Each designed load set number has five digits. The first digit (1xxxx - 9xxxx) indicates the format of the temperature; the second and third digits (x01xxx - x99xxx) indicate the sequential number of input values at one node; the fourth and fifth digits (xxx01 - xxx99) indicate the load set number of the temperature distribution. The first, second and third digits are designated and the last two digits are user's choice. For example a linear variation of temperature across a shell element needs input T_{top} and T_{bot} two values at each nodes. The designated load set numbers for PATRAN input are:

20103 for T_{top}

and **20203** for T_{bot}

where the first digit **2** denotes the linear variation through the thickness in shell elements; the following **01** and **02** are for T_{top} and T_{bot}; and **03** is load set number in COMET.

The designed PATRAN load set formats are listed in Table I.

Table I COMET Temperature Input Formats / Designated PATRAN Load Set Numbers

COMET Format	Element	Node values	Designated load set number for PATRAN input (00 < xx < 100) *
CONST $T = \bar{T}$	Solid Layered Shell Beam	$\bar{T}$	101xx
LINEAR $T(z) = \frac{T_{top}+T_{bot}}{2} + z\frac{T_{top}-T_{bot}}{h}$	Layered Shell	T_{top} T_{bot}	201xx 202xx
LINEAR $T(y,z) = \bar{T} + y\frac{\partial T}{\partial y} + z\frac{\partial T}{\partial z}$	Beam	$\bar{T}$ $\frac{\partial T}{\partial y}$ $\frac{\partial T}{\partial z}$	301xx 302xx 303xx
PWLIN $T(z) = \frac{T_{i+1}+T_i}{2} + z\frac{T_{i+1}-T_i}{h_i}$ h_i: Thickness of layer i	Layered Shell	T_1 T_2 T... T_{NL+1}	401xx 402xx 4...xx 4(NL+1)xx (NL<99)
QUADRATIC $T(z) = T_0+T_1z+T_2z^2$	Layered Shell	T_0 T_1 T_2	501xx 502xx 503xx
**CUBIC $T(z) = T_0+T_1z+T_2z^2+T_3z^3$	Layered Shell	T_0 T_1 T_2 T_3	601xx 602xx 603xx 604xx
**EXPON $T(z) = ae^{b(z-c)}$	Layered Shell	a b c	701xx 702xx 703xx
TABLE $T(z) = \frac{T_{i+1}+T_i}{2} + z\frac{T_{i+1}-T_i}{Z_{i+1}-Z_i}$	Layered Shell	T_1 Z_1 T_2 Z_2 ... T_{NT} Z_{NT}	801xx 802xx 803xx 804xx 8...xx 8(NT*2-1)xx 8(NT*2)xx (NT<100)
** Not currently implemented in COMET			* xx is COMET load set number

In PATRAN input the designated five digit numbers are only for the load set number of "NODE TEMPERATURE". They are not the real load set number in COMET. The real load set number in COMET are the fourth and fifth digits of the designated five digit number. That means the same temperature load set in COMET should have the same fourth and fifth digits in PATRAN input. For examples, a shell structure has two temperature load sets: Load 1 and Load 5. Load 1 contains CONST and QUADRATIC temperature distributions and Load 5 has CONST, TABLE and LINEAR temperatures. For Load 1 the temperature set 10101 for T, 50101,50201 and 50301 for T0, T1 and T2 should be generated in PATRAN. And for temperature Load 5 the temperature set 10105 for T; 80105, 80205, 80305, 80405, ... for T1, Z1, T2, Z2,... and 20105, 20205 for T_{top},T_{bot} should be generated in PATRAN.

Basically this transformation treats node temperature multi-values as different designed temperature sets in PATRAN, then combine them in PT2T to become COMET formats.

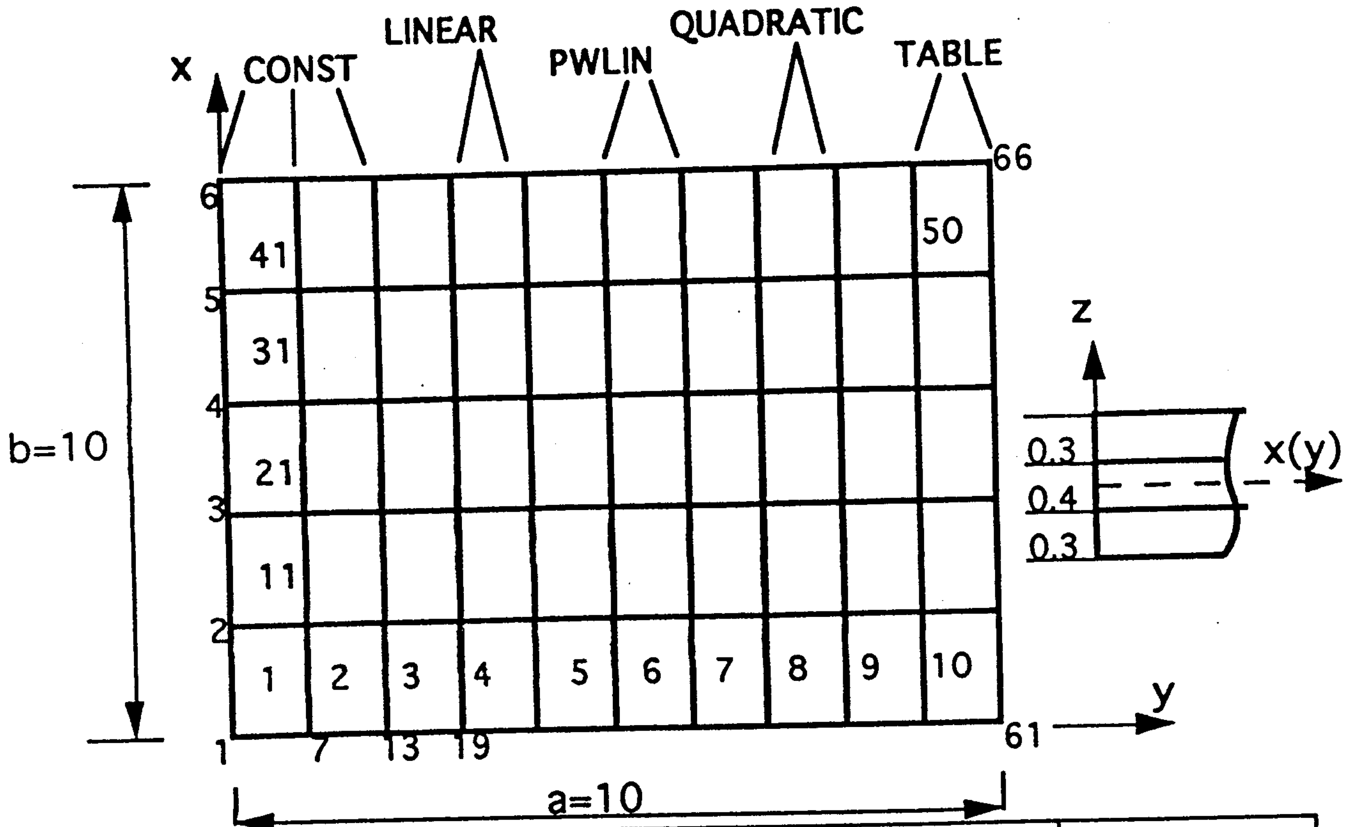
II.3

Running thermal stress analysis in COMET using PATRAN input

3) Type 'pt2ttemp' to execute translator PT2T. A runstream file (e.g. PT2T.PRC) will be created.
4) Modify the COMET runsteam in PT2T.PRC or prepare user's own runstream file which calls the transformed subroutines in PT2T.PRC.
5) Create an executable file xxx.com, and execute an analysis.

II.4

An example with general temperature distributions



		Node	SET 1	SET 2
CONST	$T(Z) = \overline{T}$	1-18	$\overline{T} = x+y$	
LINEAR	$T(z) = \frac{T_{top}+T_{bot}}{2} + z\frac{T_{top}-T_{bot}}{h}$	19-30	$T_{top}=200$ $T_{bot}=-200$	
PWLIN	$T(z) = \frac{T_{i+1}+T_i}{2} + z\frac{T_{i+1}-T_i}{h_i}$ h_i: Thickness of layer i $h_1=h_3=0.03$, $h_2=0.04$	31-42	$T_1=410$ $T_2=420$ $T_3=430$ $T_4=440$	
QUADRATIC	$T(z) = T_0+T_1z+T_2z^2$	43-54	$T_0=510$ $T_1=520$ $T_2=530$	$T_0=512$ $T_1=522$ $T_2=532$
TABLE	$T(z) = \frac{T_{i+1}+T_i}{2}+z\frac{T_{i+1}-T_i}{Z_{i+1}-Z_i}$	55-66	$T_1=810$, $Z_1=-.05$ $T_2=830$, $Z_2=-.02$ $T_3=850$, $Z_3=.01$ $T_4=870$, $Z_4=.05$	$T_1=812$, $Z_1=-.05$ $T_2=832$, $Z_2=-.02$ $T_3=852$, $Z_3=.01$ $T_4=872$, $Z_4=.05$

1) TEMP.ses .... PATRAN input

```
GO
1
1
2
PA,1,,10/10
GFEG,P1,,11/6
CFEG,P1,QUAD
DFEG,P1,DISP,0/0/0,,ED1/4
PF,P1,QU,1/0.1
PMAT,1,ISO,3.0E7,,0.3,,6.0E-6
FIELD,2,RECT/S,1/100/1/100
DFEG,P1,TEMP/N,S2,10101,N1T18
DFEG,P1,TEMP/N,200,20101,N19T30
DFEG,P1,TEMP/N,-200,20201,N19T30
DFEG,P1,TEMP/N,410,40101,N31T42
DFEG,P1,TEMP/N,420,40201,N31T42
DFEG,P1,TEMP/N,430,40301,N31T42
DFEG,P1,TEMP/N,440,40401,N31T42
DFEG,P1,TEMP/N,510,50101,N43T54
DFEG,P1,TEMP/N,520,50201,N43T54
DFEG,P1,TEMP/N,530,50301,N43T54
DFEG,P1,TEMP/N,810,80101,N55T66
DFEG,P1,TEMP/N,-0.05,80201,N55T66
DFEG,P1,TEMP/N,830,80301,N55T66
DFEG,P1,TEMP/N,-0.02,80401,N55T66
DFEG,P1,TEMP/N,850,80501,N55T66
DFEG,P1,TEMP/N,0.01,80601,N55T66
DFEG,P1,TEMP/N,870,80701,N55T66
DFEG,P1,TEMP/N,0.05,80801,N55T66
DFEG,P1,TEMP/N,512,50102,N43T54
DFEG,P1,TEMP/N,522,50202,N43T54
DFEG,P1,TEMP/N,532,50302,N43T54
DFEG,P1,TEMP/N,812,80102,N55T66
DFEG,P1,TEMP/N,-0.05,80202,N55T66
DFEG,P1,TEMP/N,832,80302,N55T66
DFEG,P1,TEMP/N,-0.02,80402,N55T66
DFEG,P1,TEMP/N,852,80502,N55T66
DFEG,P1,TEMP/N,0.01,80602,N55T66
DFEG,P1,TEMP/N,872,80702,N55T66
DFEG,P1,TEMP/N,0.05,80802,N55T66
END
```

2) TEMP.PRC .... PT2T output

```
*PROCEDURE PT2T_START
 START 66
*END
*PROCEDURE PT2T_TITLE
 TITLE'  TEMP1
*END
*PROCEDURE PT2T_JLOC
 JOINT LOCATIONS
       1          0.0          0.0          0.0
       2          0.0     2.000000          0.0
       3          0.0     4.000000          0.0
       4          0.0     6.000000          0.0
       5          0.0     8.000000          0.0
       6          0.0     10.00000          0.0
       7     1.000000          0.0          0.0
       8     1.000000     2.000000          0.0
       9     1.000000     4.000000          0.0
      10     1.000000     6.000000          0.0
      11     1.000000     8.000000          0.0
      12     1.000000     10.00000          0.0
      13     2.000000          0.0          0.0
      14     2.000000     2.000000          0.0
      15     2.000000     4.000000          0.0
      16     2.000000     6.000000          0.0
      17     2.000000     8.000000          0.0
      18     2.000000     10.00000          0.0
      19     3.000000          0.0          0.0
      20     3.000000     2.000000          0.0
      21     3.000000     4.000000          0.0
      22     3.000000     6.000000          0.0
      23     3.000000     8.000000          0.0
      24     3.000000     10.00000          0.0
      25     4.000000          0.0          0.0
      26     4.000000     2.000000          0.0
      27     4.000000     4.000000          0.0
      28     4.000000     6.000000          0.0
      29     4.000000     8.000000          0.0
      30     4.000000     10.00000          0.0
      31     5.000000          0.0          0.0
      32     5.000000     2.000000          0.0
      33     5.000000     4.000000          0.0
      34     5.000000     6.000000          0.0
      35     5.000000     8.000000          0.0
      36     5.000000     10.00000          0.0
      37     6.000000          0.0          0.0
      38     6.000000     2.000000          0.0
      39     6.000000     4.000000          0.0
      40     6.000000     6.000000          0.0
      41     6.000000     8.000000          0.0
      42     6.000000     10.00000          0.0
      43     7.000000          0.0          0.0
      44     7.000000     2.000000          0.0
      45     7.000000     4.000000          0.0
      46     7.000000     6.000000          0.0
      47     7.000000     8.000000          0.0
      48     7.000000     10.00000          0.0
      49     8.000000          0.0          0.0
      50     8.000000     2.000000          0.0
      51     8.000000     4.000000          0.0
      52     8.000000     6.000000          0.0
      53     8.000000     8.000000          0.0
      54     8.000000     10.00000          0.0
      55     9.000000          0.0          0.0
      56     9.000000     2.000000          0.0
      57     9.000000     4.000000          0.0
      58     9.000000     6.000000          0.0
```

```
      59       9.000000    8.000000         0.0
      60       9.000000    10.00000         0.0
      61       10.00000         0.0         0.0
      62       10.00000    2.000000         0.0
      63       10.00000    4.000000         0.0
      64       10.00000    6.000000         0.0
      65       10.00000    8.000000         0.0
      66       10.00000    10.00000         0.0
*END
*PROCEDURE PT2T_MAT
 MATERIAL CONSTANTS
       1                   3.0000+7   0.300000                 6.0000-6
*END
*PROCEDURE PT2T_TEMP
[XQT <ES_PROC>
   *DEF/A ES_NAME = <ES_NAME>
   *DEF/I ES_LOAD_SET =        1
    DEFINE TEMPERATURE
     ELEMENT =            1
     NODE =     1
   T =    0.00000000E+00
     NODE =     2
   T =    0.10000000E+03
     NODE =     3
   T =    0.30000000E+03
     NODE =     4
   T =    0.20000000E+03
     ELEMENT =            2
     NODE =     1
   T =    0.10000000E+03
     NODE =     2
   T =    0.20000000E+03
     NODE =     3
   T =    0.40000000E+03
     NODE =     4
   T =    0.30000000E+03
     ELEMENT =            3
     NODE =     1
   T =    0.20000000E+03
     NODE =     2
   T/LINEAR =    0.20000000E+03   -0.20000000E+03
     NODE =     3
   T/LINEAR =    0.20000000E+03   -0.20000000E+03
     NODE =     4
   T =    0.40000000E+03
     ELEMENT =            4
     NODE =     1
   T/LINEAR =    0.20000000E+03   -0.20000000E+03
     NODE =     2
   T/LINEAR =    0.20000000E+03   -0.20000000E+03
     NODE =     3
   T/LINEAR =    0.20000000E+03   -0.20000000E+03
     NODE =     4
   T/LINEAR =    0.20000000E+03   -0.20000000E+03
     ELEMENT =            5
     NODE =     1
   T/LINEAR =    0.20000000E+03   -0.20000000E+03
     NODE =     2
   T/PWLIN =    0.41000000E+03  ++
                0.42000000E+03  ++
                0.43000000E+03  ++
                0.44000000E+03  ++
   :
     NODE =     3
   T/PWLIN =    0.41000000E+03  ++
                0.42000000E+03  ++
```

```
T/TABLE =     0.81000000E+03 -0.50000001E-01  ++
              0.83000000E+03 -0.20000000E-01  ++
              0.85000000E+03  0.99999998E-02  ++
              0.87000000E+03  0.50000001E-01  ++

:
 NODE =    4
T/QUADRATIC =    0.51000000E+03    0.52000000E+03    0.53000000E+03
 ELEMENT =      10
 NODE =    1
T/TABLE =     0.81000000E+03 -0.50000001E-01  ++
              0.83000000E+03 -0.20000000E-01  ++
              0.85000000E+03  0.99999998E-02  ++
              0.87000000E+03  0.50000001E-01  ++

:
 NODE =    2
T/TABLE =     0.81000000E+03 -0.50000001E-01  ++
              0.83000000E+03 -0.20000000E-01  ++
              0.85000000E+03  0.99999998E-02  ++
              0.87000000E+03  0.50000001E-01  ++

:
 NODE =    3
T/TABLE =     0.81000000E+03 -0.50000001E-01  ++
              0.83000000E+03 -0.20000000E-01  ++
              0.85000000E+03  0.99999998E-02  ++
              0.87000000E+03  0.50000001E-01  ++

:
 NODE =    4
T/TABLE =     0.81000000E+03 -0.50000001E-01  ++
              0.83000000E+03 -0.20000000E-01  ++
              0.85000000E+03  0.99999998E-02  ++
              0.87000000E+03  0.50000001E-01  ++

:
 ELEMENT =      11
 NODE =    1
T =    0.20000000E+03
 NODE =    2
T =    0.30000000E+03
 NODE =    3
T =    0.50000000E+03
 NODE =    4
T =    0.40000000E+03
 ELEMENT =      12
 NODE =    1
T =    0.30000000E+03
 NODE =    2
T =    0.40000000E+03
 NODE =    3
T =    0.60000000E+03
 NODE =    4
T =    0.50000000E+03
 ELEMENT =      13
 NODE =    1
T =    0.40000000E+03
 NODE =    2
T/LINEAR =    0.20000000E+03    -0.20000000E+03
 NODE =    3
T/LINEAR =    0.20000000E+03    -0.20000000E+03
 NODE =    4
T =    0.60000000E+03
 ELEMENT =      14
 NODE =    1
T/LINEAR =    0.20000000E+03    -0.20000000E+03
 NODE =    2
T/LINEAR =    0.20000000E+03    -0.20000000E+03
 NODE =    3
T/LINEAR =    0.20000000E+03    -0.20000000E+03
```

```
  NODE =   1
T/QUADRATIC =    0.51200000E+03     0.52200000E+03     0.53200000E+03
  NODE =   2
T/TABLE =      0.81200000E+03 -0.50000001E-01   ++
               0.83200000E+03 -0.20000000E-01   ++
               0.85200000E+03  0.99999998E-02   ++
               0.87200000E+03  0.50000001E-01   ++

:
  NODE =   3
T/TABLE =      0.81200000E+03 -0.50000001E-01   ++
               0.83200000E+03 -0.20000000E-01   ++
               0.85200000E+03  0.99999998E-02   ++
               0.87200000E+03  0.50000001E-01   ++

:
  NODE =   4
T/QUADRATIC =    0.51200000E+03     0.52200000E+03     0.53200000E+03
  ELEMENT =      50
  NODE =   1
T/TABLE =      0.81200000E+03 -0.50000001E-01   ++
               0.83200000E+03 -0.20000000E-01   ++
               0.85200000E+03  0.99999998E-02   ++
               0.87200000E+03  0.50000001E-01   ++

:
  NODE =   2
T/TABLE =      0.81200000E+03 -0.50000001E-01   ++
               0.83200000E+03 -0.20000000E-01   ++
               0.85200000E+03  0.99999998E-02   ++
               0.87200000E+03  0.50000001E-01   ++

:
  NODE =   3
T/TABLE =      0.81200000E+03 -0.50000001E-01   ++
               0.83200000E+03 -0.20000000E-01   ++
               0.85200000E+03  0.99999998E-02   ++
               0.87200000E+03  0.50000001E-01   ++

:
  NODE =   4
T/TABLE =      0.81200000E+03 -0.50000001E-01   ++
               0.83200000E+03 -0.20000000E-01   ++
               0.85200000E+03  0.99999998E-02   ++
               0.87200000E+03  0.50000001E-01   ++

:
 END_DEFINE_TEMPERATURE
*END
*PROCEDURE PT2T_CONN
<ES_EXPE_CMD>              . QUAD4     ELEMENT TYPE
    NMAT=      1
    NSECT=     1
     1      7      8      2                         .  PATRAN ELEMENT # 1
     7     13     14      8                         .  PATRAN ELEMENT # 2
    13     19     20     14                         .  PATRAN ELEMENT # 3
    19     25     26     20                         .  PATRAN ELEMENT # 4
    25     31     32     26                         .  PATRAN ELEMENT # 5
    31     37     38     32                         .  PATRAN ELEMENT # 6
    37     43     44     38                         .  PATRAN ELEMENT # 7
    43     49     50     44                         .  PATRAN ELEMENT # 8
    49     55     56     50                         .  PATRAN ELEMENT # 9
    55     61     62     56                         .  PATRAN ELEMENT # 10
     2      8      9      3                         .  PATRAN ELEMENT # 11
     8     14     15      9                         .  PATRAN ELEMENT # 12
    14     20     21     15                         .  PATRAN ELEMENT # 13
    20     26     27     21                         .  PATRAN ELEMENT # 14
    26     32     33     27                         .  PATRAN ELEMENT # 15
    32     38     39     33                         .  PATRAN ELEMENT # 16
    38     44     45     39                         .  PATRAN ELEMENT # 17
    44     50     51     45                         .  PATRAN ELEMENT # 18
    50     56     57     51                         .  PATRAN ELEMENT # 19
```

```
   56     62     63     57                                   . PATRAN ELEMENT # 20
    3      9     10      4                                   . PATRAN ELEMENT # 21
    9     15     16     10                                   . PATRAN ELEMENT # 22
   15     21     22     16                                   . PATRAN ELEMENT # 23
   21     27     28     22                                   . PATRAN ELEMENT # 24
   27     33     34     28                                   . PATRAN ELEMENT # 25
   33     39     40     34                                   . PATRAN ELEMENT # 26
   39     45     46     40                                   . PATRAN ELEMENT # 27
   45     51     52     46                                   . PATRAN ELEMENT # 28
   51     57     58     52                                   . PATRAN ELEMENT # 29
   57     63     64     58                                   . PATRAN ELEMENT # 30
    4     10     11      5                                   . PATRAN ELEMENT # 31
   10     16     17     11                                   . PATRAN ELEMENT # 32
   16     22     23     17                                   . PATRAN ELEMENT # 33
   22     28     29     23                                   . PATRAN ELEMENT # 34
   28     34     35     29                                   . PATRAN ELEMENT # 35
   34     40     41     35                                   . PATRAN ELEMENT # 36
   40     46     47     41                                   . PATRAN ELEMENT # 37
   46     52     53     47                                   . PATRAN ELEMENT # 38
   52     58     59     53                                   . PATRAN ELEMENT # 39
   58     64     65     59                                   . PATRAN ELEMENT # 40
    5     11     12      6                                   . PATRAN ELEMENT # 41
   11     17     18     12                                   . PATRAN ELEMENT # 42
   17     23     24     18                                   . PATRAN ELEMENT # 43
   23     29     30     24                                   . PATRAN ELEMENT # 44
   29     35     36     30                                   . PATRAN ELEMENT # 45
   35     41     42     36                                   . PATRAN ELEMENT # 46
   41     47     48     42                                   . PATRAN ELEMENT # 47
   47     53     54     48                                   . PATRAN ELEMENT # 48
   53     59     60     54                                   . PATRAN ELEMENT # 49
   59     65     66     60                                   . PATRAN ELEMENT # 50
*END
*PROCEDURE PT2T_BC
 CONSTRAINT DEFINITION  1
     ZERO      1 2 3
      1
      2
      3
      4
      5
      6
      7
     13
     19
     25
     31
     37
     43
     49
     55
     61
*END
*PROCEDURE PT2T_MODEL
*CALL ES ( FUNCTION = 'DEFINE ELEMENTS' ; --
           ES_NAME = <ES_NAME>          ; --
           ES_PROC = <ES_PROC>          ; --
           ES_PARS = <ES_PARS>          )
[XQT TAB
*CALL PT2T_START
*CALL PT2T_TITLE
*CALL PT2T_JLOC
*CALL PT2T_BC
STOP
*CALL MATDAT  . USER DEFINED SECTION PROPERTIES.
[XQT LAU
[XQT ELD
```

```
*CALL PT2T_CONN
*CALL ES ( FUNCTION = 'DEFINE FREEDOMS' )
[XQT AUS
*CALL PT2T_TEMP
*END
```

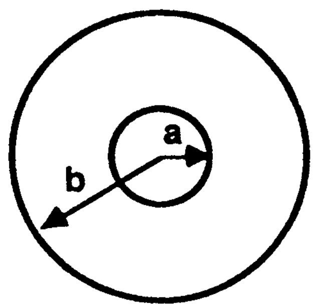
```
cp /csm/prc/proclib.gal  .
chmod +w proclib.gal
time testbed << \endinput
*open/new 1, temp.DBC
*open 28, 'proclib.gal'
*set plib = 28
*set echo,off
*def/i ES_GCP=<true>
*add PT2T.PRC
*add temp.clp
.
.  *set echo on, ma, md
*call TEMP
*eof
endinput
```

```
*procedure TEMP
  *def/a es_proc = ES1
  *def/a es_name = EX47
*call ES (function = 'DEFINE ELEMENTS'; es_proc = <es_proc>; --
          es_name = <es_name>)
[xqt TAB
   START 66
.  START 121
. *call PT2T_START
*call PT2T_TITLE
*call PT2T_JLOC
*call PT2T_BC
stop
. *call ES ( function = 'DEFINE FREEDOMS')
. [xqt ES1
. *def/a es_name = EX47
. *call MATDAT
. [xqt LAU
[xqt GCP
          fabrication
          shell
          fabid=1
          nlayers=3
          matid=3@1
          thick=0.03,0.04,0.03
          intpts=3@2
          angle=3@0.0
          end
          endfab
          material
          isoel 1 1 1
          3.0E7 0.3 0.0 6.0E-6 0.0 0.0 0.0
          endmat
[xqt ELD
<ES_EXPE_CMD>
*call PT2T_CONN
   *set echo on, ma, md
*call PT2T_TEMP
. [xqt AUS
. *call PT2T_AD
. *call PT2T_AF
  *call L_STATIC (stress=<true>; print=<false>)
  *call STRESS
  [xqt NVAL
  reset DIM = 2 METHOD = 3 RECNAME = NODES_S1   --
  DSNAME = STRS.EX47.*.*
  [xqt PNSR
.   print /id=5:115:11,56:66:2  /METHOD = 3
    print /id=3:63:6,31:36  /METHOD = 3
  stop
*end
```

II.5

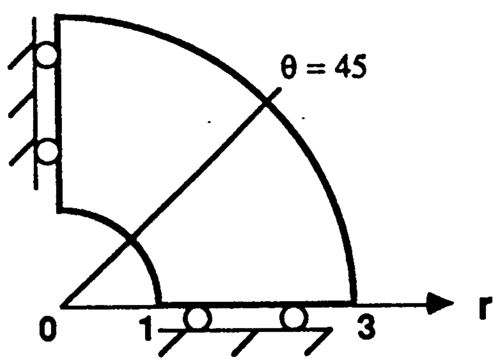
An example with inplane temperature only

Inplane Tempertature Variation Only



a = 1.0
b = 3.0
T = 100 + 200r

1/4 Model



```
 ........
  414     1.332717   2.576307      0.0
  415     1.128811   2.672103      0.0
  416     0.916624   2.752156      0.0
  417     0.696903   2.815718      0.0
  418     0.470395   2.862044      0.0
  419     0.237845   2.890387      0.0
  420          0.0   2.900000      0.0
  421     3.000000        0.0      0.0
  422     2.990055   0.246047      0.0
  423     2.960735   0.486615      0.0
  424     2.912812   0.720935      0.0
  425     2.847058   0.948232      0.0
  426     2.764245   1.167735      0.0
  427     2.665145   1.378673      0.0
  428     2.550531   1.580272      0.0
  429     2.421174   1.771761      0.0
  430     2.277846   1.952368      0.0
  431     2.121320   2.121320      0.0
  432     1.952368   2.277846      0.0
  433     1.771761   2.421174      0.0
  434     1.580272   2.550531      0.0
  435     1.378673   2.665145      0.0
  436     1.167735   2.764245      0.0
  437     0.948232   2.847058      0.0
  438     0.720935   2.912812      0.0
  439     0.486616   2.960735      0.0
  440     0.246047   2.990055      0.0
  441          0.0   3.000000     .0.0
*END
*PROCEDURE PT2T_MAT
 MATERIAL CONSTANTS
    1                3.0000+7   0.300000              6.0000-6
*END
*PROCEDURE PT2T_TEMP
[XQT <ES_PROC>
  *DEF/A ES_NAME = <ES_NAME>
  *DEF/I ES_LOAD_SET =      1
   DEFINE TEMPERATURE
    ELEMENT =         1
    NODE =    1
    T = 0.300000000E+03
    NODE =    2
    T = 0.300010742E+03
    NODE =    3
    T = 0.320011841E+03
    NODE =    4
    T = 0.320000031E+03
    ELEMENT =         2
    NODE =    1
    T = 0.300010742E+03
    NODE =    2
    T = 0.300030518E+03
    NODE =    3
    T = 0.320033600E+03
    NODE =    4
    T = 0.320011841E+03
    ELEMENT =         3
    NODE =    1
    T = 0.300030518E+03
    NODE =    2
    T = 0.300046875E+03
    NODE =    3
    T = 0.320051605E+03
    NODE =    4
    T = 0.320033600E+03
    ELEMENT =         4
    NODE =    1
    T = 0.300046875E+03
    NODE =    2
    T = 0.300054260E+03
    NODE =    3
    T = 0.320059692E+03
    NODE =    4
```

```
T = 0.320051605E+03
ELEMENT =          5
NODE =    1
T = 0.300054260E+03
NODE =    2
T = 0.300051758E+03
NODE =    3
T = 0.320056946E+03
NODE =    4
T = 0.320059692E+03
ELEMENT =          6
NODE =    1
T = 0.300051758E+03
NODE =    2
T = 0.300041534E+03
NODE =    3
T = 0.320045715E+03
NODE =    4
T = 0.320056946E+03
ELEMENT =          7
NODE =    1
T = 0.300041534E+03
NODE =    2
T = 0.300027405E+03
NODE =    3
T = 0.320030182E+03
NODE =    4
T = 0.320045715E+03

............

ELEMENT =         38
NODE =    1
T = 0.320051636E+03
NODE =    2
T = 0.320033600E+03
NODE =    3
T = 0.340036682E+03
NODE =    4
T = 0.340056305E+03
ELEMENT =         39
NODE =    1
T = 0.320033600E+03
NODE =    2
T = 0.320011841E+03
NODE =    3
T = 0.340012939E+03
NODE =    4
T = 0.340036682E+03
ELEMENT =         40
NODE =    1
T = 0.320011841E+03
NODE =    2
T = 0.320000000E+03
NODE =    3
T = 0.340000000E+03
NODE =    4
T = 0.340012939E+03

...........

ELEMENT =         51
NODE =    1
T = 0.340000031E+03
NODE =    2
T = 0.340004333E+03
NODE =    3
T = 0.360004730E+03
NODE =    4
T = 0.360000031E+03

...........

ELEMENT =         72
```

```
NODE =    1
T = 0.360004730E+03
NODE =    2
T = 0.360017670E+03
NODE =    3
T = 0.380019012E+03
NODE =    4
T = 0.380005035E+03
ELEMENT =        73
NODE =    1
T = 0.360017670E+03
NODE =    2
T = 0.360035706E+03
NODE =    3
T = 0.380038422E+03
NODE =    4
T = 0.380019012E+03

........

ELEMENT =        94
NODE =    1
T = 0.380038422E+03
NODE =    2
T = 0.380058197E+03
NODE =    3
T = 0.400062317E+03
NODE =    4
T = 0.400041138E+03

.......

ELEMENT =       295
NODE =    1
T = 0.580099609E+03
NODE =    2
T = 0.580124084E+03
NODE =    3
T = 0.600129333E+03
NODE =    4
T = 0.600103821E+03

......

ELEMENT =       296
NODE =    1
T = 0.580124084E+03
NODE =    2
T = 0.580130127E+03
NODE =    3
T = 0.600135559E+03
NODE =    4
T = 0.600129333E+03

......

ELEMENT =       397
NODE =    1
T = 0.680157288E+03
NODE =    2
T = 0.680135925E+03
NODE =    3
T = 0.700140686E+03
NODE =    4
T = 0.700162781E+03
ELEMENT =       398
NODE =    1
T = 0.680135925E+03
NODE =    2
T = 0.680088501E+03
NODE =    3
T = 0.700091553E+03
NODE =    4
T = 0.700140686E+03
ELEMENT =       399
```

```
 . NODE =    1
 . T = 0.680088501E+03
   NODE =    2
   T = 0.680031189E+03
   NODE =    3
   T = 0.700032349E+03
   NODE =    4
   T = 0.700091553E+03
   ELEMENT =       400
   NODE =    1
   T = 0.680031189E+03
   NODE =    2
   T = 0.680000000E+03
   NODE =    3
   T = 0.700000000E+03
   NODE =    4
   T = 0.700032349E+03
  END_DEFINE_TEMPERATURE
*END
*PROCEDURE PT2T_AREF
 ALTERNATE REFERENCE FRAMES
     2 1           0.0 2          0.0 3          0.0          0.0          0.0          0.0
*END
*PROCEDURE PT2T_JREF
*END
*PROCEDURE PT2T_BEAM
*END
*PROCEDURE PT2T_RIGID
*END
*PROCEDURE PT2T_OTHER
*END
*PROCEDURE PT2T_CONN               . QUAD4    ELEMENT TYPE
<ES_EXPE_CMD>
     NMAT=       1
     NSECT=      1
      1      2     23     22                        . PATRAN ELEMENT # 1
      2      3     24     23                        . PATRAN ELEMENT # 2
      3      4     25     24                        . PATRAN ELEMENT # 3
      4      5     26     25                        . PATRAN ELEMENT # 4
      5      6     27     26                        . PATRAN ELEMENT # 5
      6      7     28     27                        . PATRAN ELEMENT # 6
      7      8     29     28                        . PATRAN ELEMENT # 7
      8      9     30     29                        . PATRAN ELEMENT # 8
      9     10     31     30                        . PATRAN ELEMENT # 9
     10     11     32     31                        . PATRAN ELEMENT # 10

     ............

    116    117    138    137                        . PATRAN ELEMENT # 111
    117    118    139    138                        . PATRAN ELEMENT # 112
    118    119    140    139                        . PATRAN ELEMENT # 113
    119    120    141    140                        . PATRAN ELEMENT # 114
    120    121    142    141                        . PATRAN ELEMENT # 115

     .......

    195    196    217    216                        . PATRAN ELEMENT # 186
    196    197    218    217                        . PATRAN ELEMENT # 187

     ......

    407    408    429    428                        . PATRAN ELEMENT # 388
    408    409    430    429                        . PATRAN ELEMENT # 389
    409    410    431    430                        . PATRAN ELEMENT # 390
    410    411    432    431                        . PATRAN ELEMENT # 391
    411    412    433    432                        . PATRAN ELEMENT # 392
    412    413    434    433                        . PATRAN ELEMENT # 393
    413    414    435    434                        . PATRAN ELEMENT # 394
    414    415    436    435                        . PATRAN ELEMENT # 395
    415    416    437    436                        . PATRAN ELEMENT # 396
    416    417    438    437                        . PATRAN ELEMENT # 397
    417    418    439    438                        . PATRAN ELEMENT # 398
    418    419    440    439                        . PATRAN ELEMENT # 399
    419    420    441    440                        . PATRAN ELEMENT # 400
 *END
```

```
*PROCEDURE PT2T_BC
CONSTRAINT DEFINITION  1
    ZERO       2
     1
    ZERO       1
    21
    ZERO       2
    22
    ZERO       1
    42
    ZERO       2
    43
    ZERO       1
    63
    ZERO       2
    64
    ZERO       1
    84
    ZERO       2
    85
    ZERO       1
   105
    ZERO       2
   106
    ZERO       1
   126
    ZERO       2
   127
    ZERO       1
   147
    ZERO       2
   148
    ZERO       1
   168
    ZERO       2
   169
    ZERO       1
   189
    ZERO       2
   190
    ZERO       1
   210
    ZERO       2
   211
    ZERO       1
   231
    ZERO       2
   232
    ZERO       1
   252
    ZERO       2
   253
    ZERO       1
   273
    ZERO       2
   274
    ZERO       1
   294
    ZERO       2
   295
    ZERO       1
   315
    ZERO       2
   316
    ZERO       1
   336
    ZERO       2
   337
    ZERO       1
   357
    ZERO       2
   358
    ZERO       1
   378
    ZERO       2
```

```
379
 ZERO      1
399
 ZERO      2
400
 ZERO      1
420
 ZERO      2
421
 ZERO      1
441
*END
*PROCEDURE PT2T_MODEL
*CALL ES ( FUNCTION = 'DEFINE ELEMENTS' ; --
           ES_NAME = <ES_NAME>          ; --
           ES_PROC = <ES_PROC>          ; --
           ES_PARS = <ES_PARS>          )
[XQT TAB
*CALL PT2T_START
*CALL PT2T_TITLE
*CALL PT2T_JLOC
*CALL PT2T_BC
STOP
*CALL MATDAT  . USER DEFINED SECTION PROPERTIES.
[XQT LAU
[XQT ELD
*CALL PT2T_CONN
*CALL ES ( FUNCTION = 'DEFINE FREEDOMS' )
[XQT AUS
*CALL PT2T_TEMP
*END
```

II.6

Subroutine GCPTEMP

```
      SUBROUTINE GCPTEMP(TEMPN, NOID, LDATA, NENS, NGR, NEL, NISID, IT)
      COMMON /FILES/  NINP,NOUT,NPRT,NEUF,NCOM,NVES,NRPT,NLOD,NE43,NUMD,
     +                NLD1,NLD2,NLD3
      COMMON/HEDATA/ ITC, NID, ISID
      COMMON /STEVE/ LFKEY,JBC,JAD,JAF,JAT
      COMMON /CSTEVE/ KFKEY
      CHARACTER KFKEY*40
      PARAMETER (NN=30000, NNO=4000)
      DIMENSION NENS(3000), TNO(32), NISID(NN), STEMP(NNO), LDATA(39,*)
      DIMENSION TEMPN(NN), NOID(NN)
      DIMENSION TNO2(20,32),STEMP2(20,NNO),JIT(NN,3),MAXNO(NNO)
      DIMENSION J1NOID(NNO),J1TNO2(32)
      CALL CARDOT('*PROCEDURE '//KFKEY(1:LFKEY)//'_TEMP')
        JAT=1
      WRITE(NCOM, '(A)') '[XQT <ES_PROC>'
      WRITE(NCOM, '(A)') '  *def/a ES_NAME = <ES_NAME>'
      IF (NN.LT.IT) WRITE(NCOM,'(A,I20)') 'NN IS TOO SMALL, SHOULD > '
     $,IT
      IF(NNO.LT.NGR) WRITE(NCOM,'(A,I20)') 'NNO IS TOO SMALL, SHOULD >'
     $,NGR
      IMN=0
      DO 2 I=1,IT
      GJNN=NISID(I)/100.0
      JNN=INT(GJNN)
      JN3=NISID(I)-JNN*100
      GJN1=JNN/100.0
      JN1=INT(GJN1)
      JN2=JNN-JN1*100
      JIT(I,1)=JN1
      JIT(I,2)=JN2
      JIT(I,3)=JN3
      IMN=IMN+JN1
 2       CONTINUE
      IF(IMN.EQ.0) GOTO 1000
cc
  15  DO 16 I=1, IT
  16  IF (NISID(I).GT.0) GOTO 17
      GOTO 1100
  17  JJ= JIT(I,3)
      WRITE(NCOM,'(A,I5)') '  *def/i es_load_set = ', JJ
      WRITE(NCOM,'(A)') '   DEFINE TEMPERATURE'
      DO 110 I=1, NGR
      MAXNO(I)=0
      J1NOID(I)=0.0
      DO 110 J=1, 20
 110  STEMP2(J,I) = 0.0
      DO 120 I=1, IT
      JJ1=JIT(I,1)
      JJ2=JIT(I,2)
        IF(JJ.NE.JIT(I,3)) GOTO 120
      STEMP2(JJ2,NOID(I)) = TEMPN(I)
      J1NOID(NOID(I))=JJ1
      MAXNO(NOID(I))=MAXNO(NOID(I))+1
      NISID(I)=-NISID(I)
 120  CONTINUE
      DO 130 NE=1, NEL
      NJ1T=0
      DO 140 I=1, NENS(NE)
      J1TNO2(I)=J1NOID(LDATA(6+I,NE))
      NJ1T=NJ1T+J1TNO2(I)
 140  CONTINUE
      IF(NJ1T.EQ.0) GOTO 130
      WRITE(NCOM, '(A,I8)') '    ELEMENT =', NE
      DO 145 I=1, NENS(NE)
      J=1
```

```
      IF(J1TNO2(I).EQ.0) GOTO 145
      WRITE(NCOM,'(A,I3)') '     NODE = ', I
155   TNO2(J,I) = STEMP2(J,LDATA(6+I,NE))
      IF(J.GE.MAXNO(LDATA(6+I,NE))) GOTO 160
      J=J+1
      GOTO 155
160   GOTO(210,220,230,240,250,260,270,280) J1TNO2(I)
      PRINT *,'INVALID TEMPRATURE DATA CARD TYPE FOUND ON NEUTRAL FILE'
      GOTO 290
210   WRITE(NCOM,'(A,E16.8)') '  T = ',TNO2(1,I)
      GOTO 290
220   WRITE(NCOM,'(A,2(E16.8,2x))') '  T/LINEAR = ',(TNO2(K,I),K=1,2)
      GOTO 290
230   WRITE(NCOM,'(A,3(E16.8,2x))') '  T/LINEAR = ',(TNO2(K,I),K=1,3)
      GOTO 290
240   WRITE(NCOM,'(A,E16.8,A)') '  T/PWLIN = ',TNO2(1,I),'  ++'
      DO 241 K=2,MAXNO(LDATA(6+I,NE))
241   WRITE(NCOM,'(A,E16.8,A)') '             ',TNO2(K,I),'  ++'
      WRITE(NCOM,*) '  :'
      GOTO 290
250   WRITE(NCOM,'(A,3(E16.8,2x))') '  T/QUADRATIC = ',(TNO2(K,I),K=1,3)
      GOTO 290
260   WRITE(NCOM,'(A,4E16.8)') '  T/CUBIC = ',(TNO2(K,I),K=1,4)
      GOTO 290
270   WRITE(NCOM,'(A,3E16.8)') '  T/EXPON = ',(TNO2(K,I),K=1,3)
      GOTO 290
280   WRITE(NCOM,'(A,2E16.8,A)') '  T/TABLE = ',(TNO2(K,I),K=1,2),'  ++'
      DO 281 K=3,MAXNO(LDATA(6+I,NE)),2
281   WRITE(NCOM,'(12x,2E16.8,A)') TNO2(K,I),TNO2(K+1,I),'  ++'
      WRITE(NCOM,*) '  :'
290   CONTINUE
145   CONTINUE
130   CONTINUE
       WRITE(NCOM,'(A)') '   END_DEFINE_TEMPERATURE'
      GOTO 15
1100   WRITE(NCOM,'(A)') '*END'
      GOTO 1200
1000  CONTINUE
  5   DO 6 I=1, IT
  6   IF (NISID(I).GT.0) GOTO 7
      GOTO 100
  7   JJ= NISID(I)
C     WRITE(NCOM,'(A,I5)') 'NGR=    ', NGR
      WRITE(NCOM,'(A,I5)') '  *def/i es_load_set = ', JJ
      WRITE(NCOM,'(A)') '   DEFINE TEMPERATURE'
      DO 10 I=1, NGR
 10   STEMP(I) = 0.0
      DO 20 I=1, IT
        IF(JJ.NE.NISID(I)) GOTO 20
      STEMP(NOID(I)) = TEMPN(I)
      NISID(I)=-NISID(I)
 20   CONTINUE
      DO 30 NE=1, NEL
      TTNO=0.0
      DO 40 I=1, NENS(NE)
C       WRITE(NCOM,'(A,I3)') 'LDATA=',LDATA(6+I,NE)
C       WRITE(NCOM,'(A,E16.9)') 'STEMP(LDATA)=',STEMP(LDATA(6+I,NE))
      TNO(I) = STEMP(LDATA(6+I,NE))
 40   TTNO=TTNO+ABS(TNO(I))
      IF(TTNO.EQ.0.0) GOTO 30
      WRITE(NCOM, '(A,I8)') '     ELEMENT =', NE
      DO 45 I=1, NENS(NE)
      IF(TNO(I).EQ.0.0) GOTO 45
      WRITE(NCOM,'(A,I3)') '     NODE = ', I
      WRITE(NCOM, '(A,E16.9)') '     T =', TNO(I)
 45   CONTINUE
```

```
 30  CONTINUE
      WRITE(NCOM,'(A)') '    END_DEFINE_TEMPERATURE'
     GOTO 5
 100  WRITE(NCOM,'(A)') '*END'
1200 RETURN
       END
```

Appendix III Integrated Thermal / Structural Analysis

This Appendix shows the COMET runstream of an example of integrated thermal/structural analysis in COMET. The SPAR element was used in thermal analysis and ES1 element was applied in structural analysis. Different meshes were used in thermal and structural analyses as shown in Fig. 2. The results of a temperature distribution, a stress distribution and the plate buckling sensitivity parameter are also shown in Fig. 2. Detailed results are published in the proceedings of 35th SDM conference (AIAA Paper No. 94-1594).

COMET subroutine temptran.ams which transfers temperature between different meshes is also listed in this appendix for future reference.

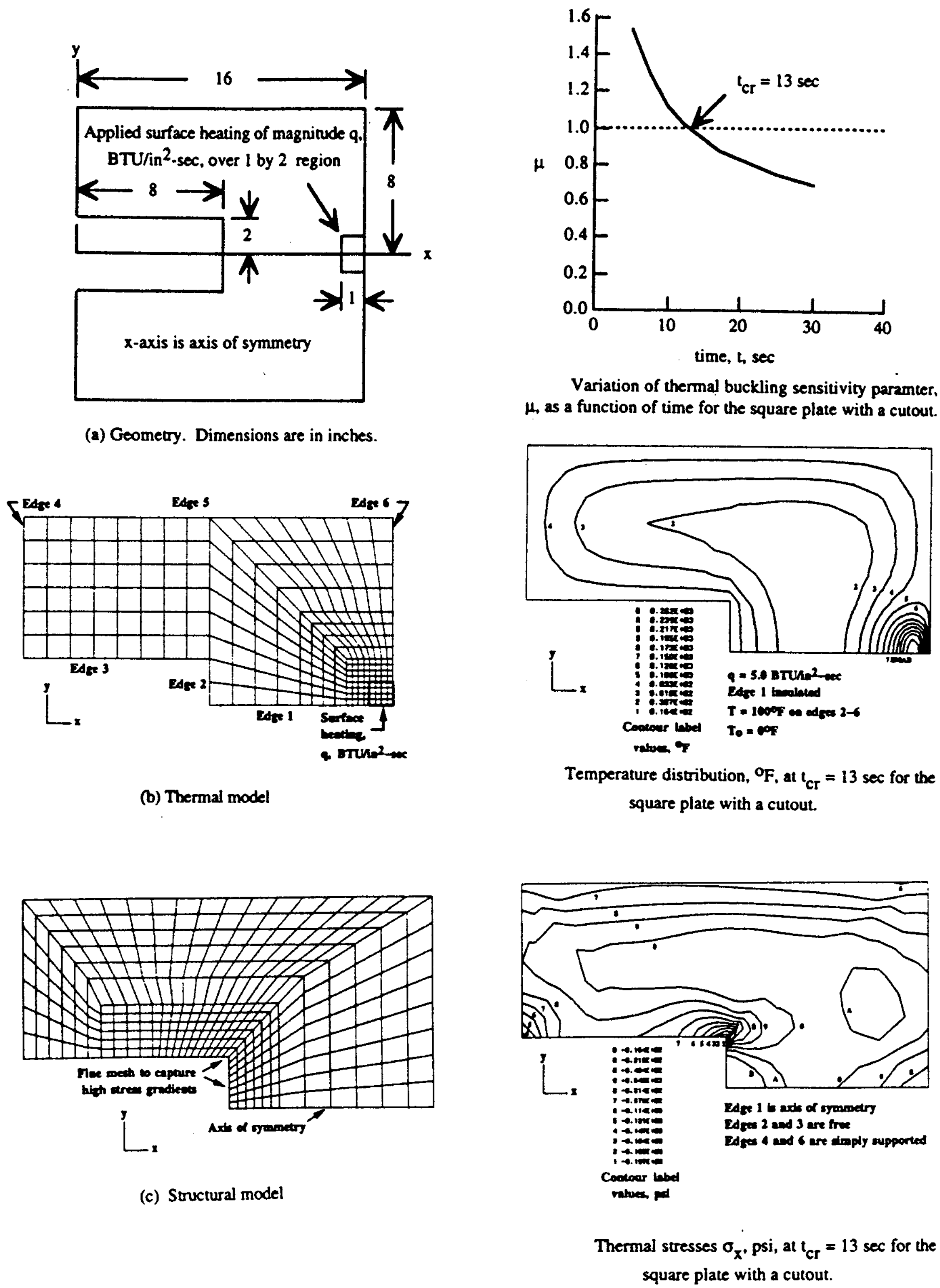


Fig. 2 Thermal / structural analysis meshes and results

COMET Runstream

(Thermal / Structural Analysis)

```
cp /csm/prc/proclib.gal  .
chmod +w proclib.gal
time testbed << \endinput
*open 28, 'proclib.gal'
*set plib = 28
*set echo=off
*open/new 1,new.DBC
*add sim.clp
*call PLATE
*toc 1
. * set echo on, ma, md
*open/new 2, bbb.DBC
*copy 2 =1,NODA.TEMP.1.1
*copy 2 =1,DEF.K41.3.4
*copy 2,XYZ.K41.1.1 =1,JLOC.BTAB.2.5
*open/new 1, TEST.DBC
*copy 1 =2,NODA.TEMP.1.1
*copy 1 =2,DEF.K41.3.4
*copy 1 =2,XYZ.K41.1.1
*toc 2
*close 2
*add simply3.clp
*toc 1
*def/i ES_GCP=<true>
*call BUCK
*toc 1
[XQT EXIT
\eof
```

```
*procedure PLATE
*set echo=off
*def time=27.
*def/i tp1=23
*def/i tp2=28
*def temp=20.
[XQT AUS
*def kx=<<226./3600.>/12.>
 TABLE(NI=9,NJ=1):COND PROP 1:I=1 2 3 4 5 6
. SANDY's Properties
  J=1:70. .323 0.0906 .006122 .006122 .006122
  TABLE(NI=8,NJ=1):COND COEF 1:I=1 2 3 4
   J=1: 0.323 0.0 0.0 0.0
  TABLE(NI=1,NJ=1): K THIC:J=1:1.
  TABLE(NI=1,NJ=1): K    AREA : J=1:   1.0
.  TABLE(NI=1,NJ=32): SOUR K21 1
  TABLE(NI=1,NJ=16): SOUR K41 2
   j=1,16:5.0
. J=1,4: 0.
. J=5,8: 10.0
. J=9,12: 0.0
. J=13,16: 10.0
. J=17,20: 0.0
. J=21,24: 10.0
. J=25,28: 0.0
. J=29,32: 10.0
 TABLE(NI=1,NJ=49):TEMP NODE: DDATA=9.:J=1,9:9.
                              DDATA=8.:J=10,13:125.
                              DDATA=8.:J=14,17:193.
                              DDATA=1.:J=18,24:267.
                              DDATA=7.:J=25,31:224.
                              DDATA=1.:J=32,38:210.
                              J=39:182.
. TABLE(NI=1,NJ=17):TEMP NODE: DDATA=1.:J=1,7:267.
                               DDATA=7.:J=40,46:218.
                               DDATA=4.:J=47,49:150.
  TABLE(NI=1,NJ=49):APPL TEMP:J=1,9:50.0
                              J=10,13:50.0
                              J=14,17:50.0
                              J=18,24:50.0
                              J=25,31:50.0
                              J=32,38:50.0
                              J=39:50.0
. TABLE(NI=1,NJ=17):APPL TEMP:J=1,7:50.0
                              J=40,46:50.0
                              J=47,49:50.0
 *call MESH
 [XQT TGEO
 [XQT TRTB
  RESET  T1=0.  T2=<time> DT=0.1 $ BETA=0.666667
  RESET mxndt=5000
    TEMP=0.0
    TSAVE=1.0
 [xqt aus
     define t=1 tran temp 1 1 <tp1> <tp2>
      noda temp=union(t)
 [XQT DCU
     PRINT 1 SOUR K21
     PRINT 1 NODA TEMP 1 1
     TOC 1
```

```
*end
*procedure MESH
. Initial Model Def. Procedure
   [xqt TAB
     START 273 6
     JLOC
      1 14. 0. 0. 16. 0. 0. 9 1 9
      9 14. 2. 0. 16. 2. 0.
      82 12. 0. 0. 13.5 0. 0. 4 1 9
      4 12. 4. 0. 13.5 2.5 0.
      118 14. 2.5 0. 16. 2.5 0. 8 1 4
      8 12.5 4. 0. 16. 4. 0.
      150 8. 0. 0. 11. 0. 0. 4 1 9
      4 8. 8. 0. 11. 5. 0.
      186 12. 5. 0. 16. 5. 0. 8 1 4
      8 9. 8. 0. 16. 8. 0.
      218 7. 2. 0. 7. 8. 0. 7 1 8
      7 0. 2. 0. 0. 8. 0.
 .
 [xqt ELD
     RESET NUTED=1
     K41
     MATERIAL=PROP
     NMAT=1
     NSECT = 1
     GROUP 1
     1 2 11 10 1 4 8
     GROUP 2
     5 6 15 14 1 4 4
     GROUP 3
     41 42 51 50 1 4 4
     82 83 87 86 1 3 8
     85 1 10 89
     89 10 19 93
     93 19 28 97
     97 28 37 101
     101 37 46 105
     105 46 55 109
     109 55 64 113
     113 64 73 117
     73 74 118 117 1 8 1
     117 118 126 116
     118 119 127 126 1 7 3
     116 126 134 115
     115 134 142 114
     150 151 155 154 1 3 8
     153 82 86 157
     157 86 90 161
     161 90 94 165
     165 94 98 169
     169 98 102 173
     173 102 106 177
     177 106 110 181
     181 110 114 185
     114 142 186 185
     142 143 187 186 1 7 1
     185 186 194 184
     184 194 202 183
     183 202 210 182
     186 187 195 194 1 7 3
```

```
    158 162 219 218
    162 166 220 219
    166 170 221 220
    170 174 222 221
    174 178 223 222
    178 182 224 223
    218 219 226 225 1 6 7
*end
```

```
*procedure BUCK
. Initial Model Def. Procedure
*call ES (function = 'DEFINE ELEMENTS'; es_proc = ES1; es_name = EX47)
   [xqt TAB
    START 361 6
  JLOC
    1 8. 0. 0. 10. 0. 0. 7 1 9
    7 8. 2. 0. 10. 4. 0.
    64 7.75 2. 0. 9.5 4. 0. 7 1 12
    7 5. 2. 0. 5. 4. 0.
    148 11. 0. 0. 16. 0. 0. 6 1 9
    6 11. 5. 0. 16. 8. 0.
    202 3. 2. 0. 4.5 2. 0. 4 1 7
    4 3. 4. 0. 4.5 4. 0.
    230 2.5 5. 0. 10.5 5. 0. 16 1 6
    16 0. 8. 0. 15. 8. 0.
    326 2.5 2. 0. 2.5 4. 0. 6 1 6
    6 0. 2. 0. 0. 7. 0.
 .
 [xqt GCP
     fabrication
       shell
         fabid = 1
         nlayers = 1
         matid = 1
         thick = 0.1
         intpts = 2
         angle = 0.0
       end
     endfab
     material
       isoel 1 1 1
       1.0E7 0.3 0.322 1.0E-6 0.0 0.0 0.0
     endmat
 .
 [xqt ELD
  <ES_EXPE_CMD>
    NSECT = 1
    1 2 9 8 1 6 20
    148 149 155 154 1 5 8
    7 148 154 14
    14 154 160 21
    21 160 166 28
    28 166 172 35
    35 172 178 42
    42 178 184 49
    49 184 190 56
    56 190 196 63
    63 196 245 70
    70 245 244 77
    77 244 243 84
    84 243 242 91
    91 242 241 98
    98 241 240 105
    105 240 239 112
    112 239 238 119
    119 238 237 126
    126 237 236 133
    133 236 235 140
    140 235 234 147
```

```
230 231 247 246 1 15 5
245 196 197 261
261 197 198 277
277 198 199 293
293 199 200 309
309 200 201 325
202 203 207 206 1 3 6
205 141 142 209
209 142 143 213
213 143 144 217
217 144 145 221
221 145 146 225
225 146 147 229
226 227 231 230 1 3 1
229 147 234 233
326 202 206 327
327 206 210 328
328 210 214 329
329 214 218 330
330 218 222 331
326 327 333 332 1 5 5
226 230 331 222
331 230 246 337
337 246 262 343
343 262 278 349
349 278 294 355
355 294 310 361

[xqt TAB
    CON 1
    ZERO 6: 1,289
    ZERO 1 2 3 4 : 356,361,1
    ZERO 1 2 3 5: 311,325,1:
    ZERO 1 2 3 4 5: 310: 201
    ZERO 1 2 3 4 : 153,195,6
    ZERO 2 4: 1,7,1
    ZERO 2 4: 148,152,1
stop
  [XQT temptran3
   *open 1,TEST.DBC
   *toc 1
[xqt ES1
   *set echo on, ma, md
  *def/a es_name = EX47
   *do $k = 1,6
   *def/i itime=<$k>
   *def/i ES_LOAD_SET=<itime>
    DEFINE TEMPERATURE
  *find 1, DEF.EX47.* /seq=dsn
  *rem DEF.EX47.* : dsn=<dsn>
  *find 1, NODE.TEMP  /seq=dsntemp
  *rem NODE.TEMP : dsnjloc=<dsntemp>
  *find 1, TRAN.TEMP  /seq=dsntemp
  *rem TRAN.TEMP : dsnjloc=<dsntemp>
        *do $j=1, 324              .   324 Elements
   ELEMENT = <$j>
    *do $i = 12, 15        .    Node #1 - #4 in DEF.EX47.
     *def/i ioff1 = <$i>
     *gal2mac  /Name=nnode /Ioff=<ioff1> --
              /Maxn=1 1, <dsn> DATA.<$j>
```

```
   NODE = <<$i>-11>
     *def/i ioff2 = <<nnode>-1>
       *gal2mac /Name=temp  /Ioff=<ioff2> /Maxn=1 1, <dsntemp> --
             DATA.<<itime>>
          T=<temp>
.         T=<<temp>-70.0>
     *enddo
  *enddo
    END_DEFINE_TEMPERATURE
   *enddo
   .
  [xqt DCU
.       print 1 MATL.DATA
stop
.
*toc 1
   *do $i=6,6
*call L_STATIC (location=centroids; nval_meth=1; stress=<true>;  --
                print=<true>; load_set=<$i>)
*call L_STABIL_2 (function='eigen'; cons_set=1; bcon_set=1; n_modes=3; --
                print=<true>; load_set=<$i>)
*toc 1
   *enddo
*end
```

Subroutine TRANS

(Transfer temperature between different meshes)

```
C$Header: /csm/sam/mods/RCS/temptran.ams, 93/09 maring Exp $
C=DECK TRANS
C=BLOCK FORTRAN
C=AUTHOR     David Xue
C=DATE       September, 93
C=PURPOSE    Transfer node tempretures between different meshes
C=IF EXTP
      PROGRAM TEMPMESH
      CALL TRANS
      STOP
      END
C=ENDIF
       subroutine TRANS
c N1=NODE NUMBER OF MESH1, N2=NODE NUMBER OF MESH2, N3=ELEMENT NUMBER OF MESH2
c Transfer temperature from mesh2 to mesh1
       PARAMETER(N1=361,N2=273,N3=240,NELN=4)
       COMMON/MAT/A1(3,N1),A2(3,N2),A3(N3,8),NA4(4,N3),INODEN(N1),
     $ TA1(N1,6),TA2(N2,6)
       DIMENSION NA5(16,N3)
       CHARACTER*40 ds1,ds2,ds3,ds4,rn1,rn2,rn3,rn4,rn5
C Read temperature from NODA.TEMP.1.1 (6 records) to TA2(N2,6)
c      lib=1
       lib=LMOPEN ('COLD', 0, 'TEST.DBC', 0, 2000)
       CALL GMCODN(ds1,'NODA ','TEMP ',1,1,0)
       iesn=LMFIND(lib,ds1,1800)
       CALL GMCORN(rn1,'DATA ',1,6)
       CALL GMGETN('R',lib,iesn,rn1,'S',TA2,N,0,0,0,1800)
c      CALL GMGETN('R',lib,iesn,rn1,'S',TA2,-N2,N2,0,0,1801)
cc      print *,'TA2(N2,6)='
c      do 30 I=1,N2
c 30   print *,'N2=',I,' *',(TA2(I,J),J=1,6)
C Read node coordinates of K41 from XYZ.K41.1.1 which is copied from old
C JLOC.BTAB.2.5 of K41 element meshes.
       CALL GMCODN(ds2,'XYZ ','K41 ',1,1,0)
       iesn2=LMFIND(lib,ds2,1700)
       CALL GMCORN(rn2,'DATA ',1,0)
       CALL GMGETN('R',lib,iesn2,rn2,'S',A2,N,0,0,0,1701)
       print *,'OLD K41 MESH NODE COORD. X, Y, Z, A2(3,N2)='
       do 35 I=1,N2
  35   print *,'N2=',I,' *',(A2(J,I),J=1,3)
C Read node coordinates of EX47 from JLOC.BTAB.2.5
       CALL GMCODN(ds3,'JLOC ','BTAB ',2,5,0)
       iesn3=LMFIND(lib,ds3,1600)
       CALL GMCORN(rn3,'DATA ',1,0)
       CALL GMGETN('R',lib,iesn3,rn3,'S',A1,N,0,0,0,1601)
       print *,'NEW EX47 MESH NODE COORD. X, Y, Z, A1(3,N1)='
       do 33 I=1,N1
  33   print *,'N1=',I,' *',(A1(J,I),J=1,3)
C Read element-node numbers of K41 from DEF.K41.3.4
       CALL GMCODN(ds4,'DEF ','K41 ',3,4,0)
       iesn4=LMFIND(lib,ds4,1500)
       CALL GMCORN(rn4,'DATA ',1,N3)
       CALL GMGETN('R',lib,iesn4,rn4,'I',NA5,N,0,0,0,1501)
       DO 41 J1=1,N3
       DO 41 J2=1,4
  41   NA4(J2,J1)=NA5(J2+12,J1)
c      CALL GMGETN('R/L',lib,iesn4,rn4,'I',NA4,16,0,0,12,1501)
C
       print *,'NA5.1=',(NA5(J,1),J=1,16)
       print *,'NA5.2=',(NA5(J,2),J=1,16)
```

```
      print *,'NA4(4,N3)= ELEMENT_NODES'
      do 45 I=1,N3
 45   print *,'ELEMENT=',I,' *',(NA4(J,I),J=1,4)
      CALL TTT
C Create new temperature NODE.TEMP for EX47 meshes
      CALL GMPUNT(lib,'NODE.TEMP ',idsn,4800,2500)
c     CALL GMCORN(rn5,'DATA ',1,1)
      CALL GMCORN(rn5,'DATA ',1,6)
c     CALL GMPUTN('W',1,idsn,rn5,'S',TA1,-N1,1,0,0,2501)
      CALL GMPUTN('W',1,idsn,rn5,'S',TA1,-N1,6,0,0,2501)
      CALL GMCLOS(lib,0,2600)
c     lib=LMOPEN ('COLD', 0, 'TEST.DBC', 0, 2000)
      RETURN
      END
C=END FORTRAN
c
C=DECK TTT
C=BLOCK FORTRAN
      SUBROUTINE TTT
c     PARAMETER(N1=121,N2=81,N3=64,NELN=4)
      PARAMETER(N1=361,N2=273,N3=240,NELN=4)
      COMMON/MAT/A1(3,N1),A2(3,N2),A3(N3,8),NA4(4,N3),INODEN(N1),
     $ TA1(N1,6),TA2(N2,6)
      DIMENSION EXMAX(N3),EXMIN(N3),EYMAX(N3),EYMIN(N3)
C A3(N3,8)=x1,x2,x3,x4,y1,y2,y3,y4
      DO 2 I=1,N3
       DO 5 J=1,NELN
      A3(I,J)=A2(1,NA4(J,I))
      A3(I,J+NELN)=A2(2,NA4(J,I))
 5    CONTINUE
 2    CONTINUE
c     print *,'A3(N3,8)= ELEMENT COORDINATES'
c     do 7 I=1,N3
c 7   print *,'I=',I,' *',(A3(I,J),J=1,8)
      DO 10 I=1,N1
 10   INODEN(I)=0
      NTOTAL=0
C Find all common nodes both meshes have
      DO 20 I=1,N1
          X1=A1(1,I)
          Y1=A1(2,I)
          Z1=A1(3,I)
        DO 25 J=1,N2
          X2=A2(1,J)
          Y2=A2(2,J)
          Z2=A2(3,J)
          XR1=ABS(X1-X2)
          XR2=0.01*X1
          IF(XR1.GT.XR2)GOTO 25
          YR1=ABS(Y1-Y2)
          YR2=0.01*Y1
          IF(YR1.GT.YR2)GOTO 25
          DO 27 K=1,6
          TA1(I,K)=TA2(J,K)
c     print *,'K=',K,' *I=',I,'*J=',J,'   TA1(I,K)=',TA1(I,K)
 27   CONTINUE
C INODEN(I)=1, Ith node temperature has been evalueted.
          INODEN(I)=1
          NTOTAL=NTOTAL+1
          IF(NTOTAL.EQ.N1)GOTO 100
```

```
 25      CONTINUE
 20    CONTINUE
       print *, 'Numbers of equare nodes are: ',NTOTAL
C Find Xmax, Xmin, Ymax and Ymin for each element
       DO 30 I=1,N3
           XMAX=0.0
           XMIN=1.0E20
           YMAX=0.0
           YMIN=1.0E20
         DO 35 J=1,NELN
           X=A3(I,J)
           Y=A3(I,NELN+J)
           IF(XMAX.LT.X) XMAX=X
           IF(XMIN.GT.X) XMIN=X
           IF(YMAX.LT.Y) YMAX=Y
           IF(YMIN.GT.Y) YMIN=Y
 35      CONTINUE
           EXMAX(I)=XMAX
           EXMIN(I)=XMIN
           EYMAX(I)=YMAX
           EYMIN(I)=YMIN
c          print *,'ELEMENT=',I,'XMAXMIN,YMAXMIN=',XMAX,XMIN,YMAX,YMIN
 30    CONTINUE
C Find if Jth node is in Ith element
       DO 40 I=1,N3
           AX1=A3(I,1)
           AX2=A3(I,2)
           AX3=A3(I,3)
           AX4=A3(I,4)
           AY1=A3(I,5)
           AY2=A3(I,6)
           AY3=A3(I,7)
           AY4=A3(I,8)
           CALL AAR(AREA12,AX1,AX2,AX3,AY1,AY2,AY3)
            CALL AAR(AREA34,AX1,AX3,AX4,AY1,AY3,AY4)
           AREA=AREA12+AREA34
           XMAX=EXMAX(I)
           XMIN=EXMIN(I)
           YMAX=EYMAX(I)
           YMIN=EYMIN(I)
           DO 45 J=1,N1
              IF(INODEN(J).EQ.1)GOTO 45
              X=A1(1,J)
              Y=A1(2,J)
         IF(X.GT.XMAX.OR.X.LT.XMIN.OR.Y.GT.YMAX.OR.Y.LT.YMIN)GOTO 45
              CALL AAR(AREA1,X,AX1,AX2,Y,AY1,AY2)
              IF(AREA1.GT.AREA)GOTO 45
              CALL AAR(AREA2,X,AX2,AX3,Y,AY2,AY3)
              AX12=AREA1+AREA2
              IF(AX12.GT.AREA)GOTO 45
              CALL AAR(AREA3,X,AX3,AX4,Y,AY3,AY4)
              AX123=AX12+AREA3
              IF(AX123.GT.AREA)GOTO 45
              CALL AAR(AREA4,X,AX4,AX1,Y,AY4,AY1)
              AX1234=AX123+AREA4
              ERROR=ABS((AX1234-AREA)/AREA)
c             PRINT *,'ERROR=',ERROR
              IF(ERROR.GT.0.01)GOTO 45
C Find the Jth node is in triangle 1-2-3 or 3-4-1
         IF(AX12.GT.AREA12) THEN
```

```
              DO 50 K=1,6
              T1=TA2(NA4(1,I),K)
              T2=TA2(NA4(2,I),K)
              T3=TA2(NA4(3,I),K)
              T4=TA2(NA4(4,I),K)
        CALL INTEP(AREA34,T3,T4,T1,AX3,AX4,AX1,AY3,AY4,AY1,TEMP,X,Y)
              TA1(J,K)=TEMP
 50     CONTINUE
        GOTO 54
        END IF
              DO 53 K=1,6
              T1=TA2(NA4(1,I),K)
              T2=TA2(NA4(2,I),K)
              T3=TA2(NA4(3,I),K)
              T4=TA2(NA4(4,I),K)
        CALL INTEP(AREA12,T1,T2,T3,AX1,AX2,AX3,AY1,AY2,AY3,TEMP,X,Y)
              TA1(J,K)=TEMP
 53     CONTINUE
 54       INODEN(J)=1
              NTOTAL=NTOTAL+1
              print *,'new node ',J,'  is in old element ',I
               IF(NTOTAL.EQ.N1)GOTO 100
 45         CONTINUE
 40        CONTINUE
 100          CONTINUE
c ******************************************
c FOR PATRAN INPUT RESULT FILE (SEE PATRAN MANUAL 27.7.2.1)
          NNODES1=N1
          NNODES2=N2
          MAXNOD=0
          DEFMAX=0.0
          NDMAX=0
          NWIDTH=1
c ******************************************
            DO 70 K=6,6
               print *,'K=',K,' * OLD TEMPERATURE IS'
               print *,(TA2(I,K),I=1,N2)
               print *,'K=',K,' * NEW TEMPERATURE IS'
               print *,(TA1(I,K),I=1,N1)
 70         CONTINUE
               open (unit=9,file='TEMP.NEW',status='old')
               write(9,*)'TEMPERATURE PATRAN RESULT FILE'
               write(9,7111)NNODES1,MAXNOD,DEFMAX,NDMAX,NWIDTH
 7111       FORMAT(2I9,E15.6,2I9)
               write(9,*)'IN PATRAN FORMAT'
               write(9,*)'NEW NODE TEMPERATURE'
             DO 72 I=1,N1
               write(9,7222)I,TA1(I,6)
 7222       FORMAT(I8,E13.7)
 72          CONTINUE
                close(unit=9)
             open (unit=9,file='TEMP.OLD',status='old')
             write(9,*)'TEMPERATURE PATRAN RESULT FILE'
             write(9,7111)NNODES2,MAXNOD,DEFMAX,NDMAX,NWIDTH
             write(9,*)'IN PATRAN FORMAT'
             write(9,*)'OLD NODE TEMPERATURE'
             DO 74 I=1,N2
             write(9,7222)I,TA2(I,6)
 74         CONTINUE
             close(unit=9)
```

```
                return
                end
C=END FORTRAN
c
C=DECK AAR
C=BLOCK FORTRAN
           SUBROUTINE AAR(A,X1,X2,X3,Y1,Y2,Y3)
           A=0.5*((X3-X1)*(Y1-Y2)-(X1-X2)*(Y3-Y1))
           A=ABS(A)
           RETURN
           END
C=END FORTRAN
c
C=DECK INTEP
C=BLOCK FORTRAN
         SUBROUTINE INTEP(A,T1,T2,T3,X1,X2,X3,Y1,Y2,Y3,T,X,Y)
         A1=Y2-Y3
         B1=X3-X2
         C1=X2*Y3-Y2*X3
         XN1=(A1*X+B1*Y+C1)*0.5/A
         A2=Y3-Y1
         B2=X1-X3
         C2=X3*Y1-Y3*X1
         XN2=(A2*X+B2*Y+C2)*0.5/A
         A3=Y1-Y2
         B3=X2-X1
         C3=X1*Y2-Y1*X2
         XN3=(A3*X+B3*Y+C3)*0.5/A
         T=XN1*T1+XN2*T2+XN3*T3
           RETURN
           END
C=END FORTRAN
```