

TempMap v1.2.0 User Guide

TempMap is a temperature mapping program developed by ATA Engineering, Inc. For Windows, the user can start the temperature mapping program by typing *TempMap* in the **Command Prompt** window or by double-clicking on the **TempMap.exe** icon.

When the user runs TempMap, the following screen will appear. The various TempMap fields are numbered, and a description of each is provided below.

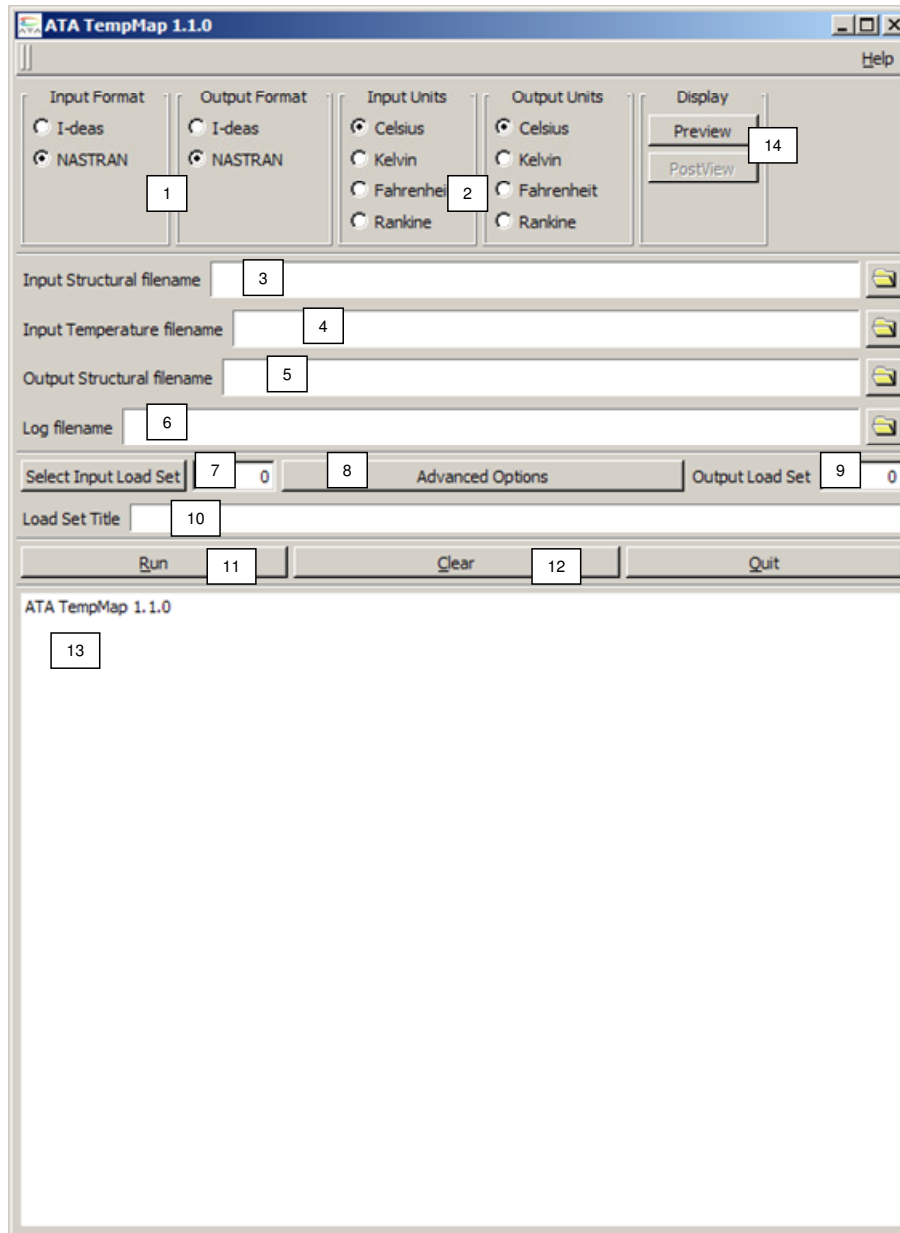


Figure 1. TempMap interface showing the key input and output fields.

[1] Input/Output Formats

These toggles allow the user to define the type of structural file that will be used for input and output. Currently, TempMap can read and write I-deas universal and NASTRAN input files.

[2] Input/Output Units

These toggles allow the user to define the input and output units. This allows the user to convert the temperatures to the appropriate units. When mapping to I-deas, a units data set is written to the output universal file. When mapping to NASTRAN, the units used are written to a comment in the output file.

[3] Input Structural Filename

This is the finite element model (FEM) that contains the target nodes for the temperature mapping. It must also include a dummy temperature load definition to identify which nodes will received the mapped temperatures.

From I-deas, the user will need to write out a universal file of the FEM. This FEM must contain at least one boundary condition load set containing nodal temperatures.

For NASTRAN, the model must contain loads written as TEMP cards. No other temperature cards are currently recognized by TempMap. In addition to loads and grids, coordinate systems are read, and the grid coordinates are transformed to Coordinate System 0.

For any input model type, the input structural model must use the same length units as the input temperature data, i.e., if the temperature model has length in inches, the model file must also be in inches.

[4] Input Temperature File Name

This is the file containing the temperature data the user wants to map to the input structural FEM. The format and contents of this file are defined in Appendix A.

The name of the input file will be used to seed the default output structural file and log file names.

[5] Output Structural Filename

This file contains the mapped temperature load set data. The name must not be the same as that of the input structural or temperature file.

I-deas loads will be written out to data sets 792 and 2414. NASTRAN loads will be written as TEMP cards.

[6] Log Filename

This file contains the log of all processing messages.

[7] Select Input Load Set

The user selects a boundary condition set containing the dummy nodal temperatures that define the nodes that will have temperatures mapped to them. If the user presses this button, TempMap will determine all of the load sets in the structural model and give the user a list from which to choose. The user can also enter the load set ID manually.

[8] Advanced Options

By clicking on this button, TempMap brings up the **Advanced Options** window, shown below in Figure 2.

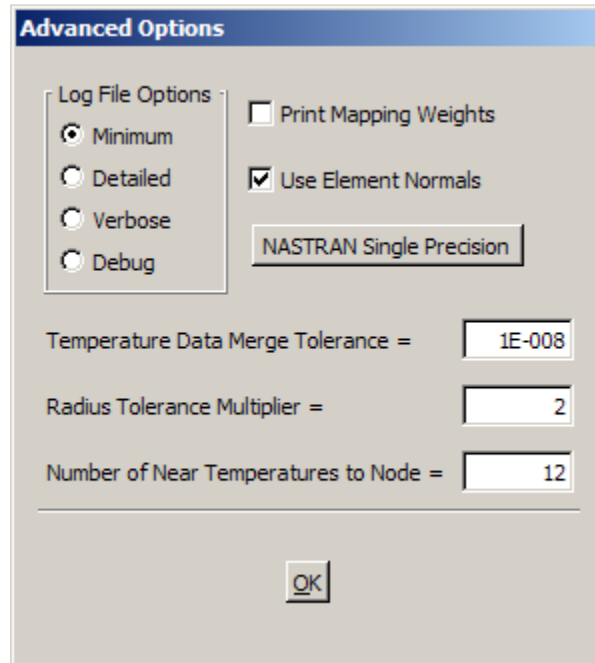


Figure 2. **Advanced Options** window.

Log File Options adjusts the amount of information written to the log file. Use **Minimum** for most cases.

The **Print Mapping Weights** toggle allows the user to write out the mapping weights for the mapping from the thermal to structural model. The file is named the same as the log file, except the file extension will be *.map*, i.e., for a log file, *example.log*, the map file will be *example.map*. This option is toggled off by default.

The format of the file is the structural grid ID, then the number of thermal grids used to map the temperature, then the thermal grid ID and weight for each of the thermal grids used in the mapping. Fields are separated by tabs. An example is shown below:

```
53    2    15    0.993109    35    0.006891
```

The **Use Element Normals** option creates a local normal at a node based on the average of the element normals. This is used for projection of 3-D temperature data onto a 2-D

surface local to the grid point. By default, this option is toggled off. In some cases, the temperature mapping can be improved by turning this option on.

The **NASTRAN Single Precision/NASTRAN Double Precision** toggle allows the user to toggle the output format as either **NASTRAN Single Precision** or **Double Precision**. The default format is **Single Precision**.

The **Temperature Data Merge Tolerance** option sets the radius at which two temperature data points are considered to be coincident. The default is 1E-08, and the minimum is 0.0. With this option, if one temperature point lies within the radius of another temperature point, it is removed from the list of temperature points and the temperature at the remaining temperature point is the average of the two points. If N points are within the radius of another temperature point, the N points are removed, and the temperature at the remaining point is the average of the N+1 points.

The **Radius Tolerance Multiplier** option allows the user to select a multiplier on the radius used for finding the nearest temperature point. This option is useful if there is significant space between the temperature data and the representative structure. For example, if there is a nonstructural thermal protective coating on a structure, the temperature mesh may be on the outer surface of the protective coating and the structure may be on the inner mold line. The user should change this value only if TempMap issues warning about not finding temperatures for nodes. The default is 2.

The **Number of Near Temperatures to Node** option allows the user to select the maximum number of temperature points near a node that are used to calculate the temperature mapping. Increasing the number of nodes helps in computing accurately mapped temperatures for data that are densely distributed in one direction but sparsely distributed in another direction, as shown in the figure below. The default is 12 temperature points near the node. Please note that increasing the number of temperature points used to create the mapping will cause TempMap to run longer.

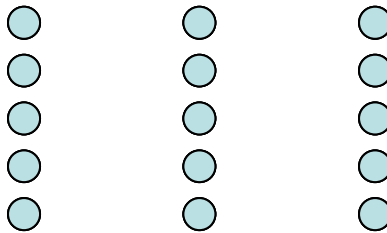


Figure 3. Example temperature distributions where increasing the number of nodes will improve the temperature mapping.

[9] Output Load Set

I-deas requires a unique load set number for each load set in a FEM. The program automatically creates a unique default load set number based on the input structural file when the user presses the **Select Input Load Set** button. If the user is performing multiple temperature mappings, care should be taken to make sure that all of the user load set IDs are unique.

For the I-deas output option, TempMap also creates a result set containing the computed temperatures so that the user can view contour plots of the data in postprocessing. The result set has the same name as the load set, and its data type is *temperature*.

[10] Load Set Title

The title is used in the output file as a description of the model. If the temperature file has a title, this will be the default.

[11] Run

Pressing this button begins the mapping process.

[12] Clear

This will clear all input.

[13] Output window

This window will contain the output messages from the mapping procedure. Typical output from the program is shown below.

```
Running ATA TempMap 1.0.0...
```

```
* Scanning universal file ...
  - 173 Nodes in universal file
  - 151 Elements in universal file
  - Mapping data on Temperature Set:
    1 - TEMPERATURE SET 1
    Output Load Set and Results Set will be:
    2 - NONE

* Reading Nodes from universal file ...
  - 173 Nodes stored

* Reading Temperatures from universal file ...
  - 173 Temperatures stored

* Reading Temperature data ...
  - 4 Temperature data points read

* Computing bounding boxes for structural and temperature models ...
  - 100% of FEM within temperature model boundary

* Sorting Temperature data ...
  - Looking for duplicates
  - No coincident Temperature Data Points found
  - Temperature data sorted

* Mapping Temperatures to FEM ...
  - 100 Mapped node temperatures computed
  - 173 Mapped node temperatures computed
```

```
* Writing mapped temperatures ...
  - 173 Mapped node temperatures written to output file

* 0 Warnings

* 0 Errors

* TempMap completed successfully
```

If less than 95% of the FEM lies within the temperature model, the program issues a warning but continues mapping. If the overlap is less than 1%, it stops processing and shows the user an error message (see below). This error typically occurs when the coordinate transformation definition (Appendix A) is not correct. TempMap writes full details of the two bounding boxes to the log file.

```
* Computing bounding boxes for structural and temperature models ...
  - 0% of FEM within temperature model boundary

* 1 WARNINGS - SEE LOG FILE

* 1 ERRORS - SEE LOG FILE

* TempMap COMPLETED WITH ERRORS
```

Finally, while TempMap is designed to be a robust tool to map temperatures from a temperature model to a finite element model, the user should look at the results of the temperature mapping to ensure that the data are correct.

[14] Preview and PostView

The **Preview** and **PostView** buttons allow the user to display the model and the temperature point cloud. **Preview** can be used to display just the structural model, just the temperature point cloud, or both. **PostView** is disabled until a mapping is completed, at which point the user can display the mapped temperatures. The temperatures are shown on a relative scale, with the blue end of the color spectrum representing the lower temperatures and the red end representing the higher temperatures. However, no actual scale is given for the temperature magnitudes.

The view screen, shown in Figures 4 and 5 below, operates the same for both views. The left mouse button rotates the model, the center mouse button zooms the model in and out, and the right mouse button moves the model up and down. In addition, there are icons for fixed standard orientations. Also, in the **Controls** window, there is an **Angles** tab that gives the user fine control over the view orientation. Finally, the user can toggle on or off perspective.

With the **Controls** window, the user can turn on and off the point cloud, elements, and/or grids. The sizes of the point cloud and grids can also be adjusted here.

If there are multiple time sets, the user can view any of these sets. A **Time Step Selection** tab will appear under the **Controls** menu, and the user selects from a pull-down menu (Figure 6).

When the user is done with the viewer, the window can be closed by clicking on the window X button.

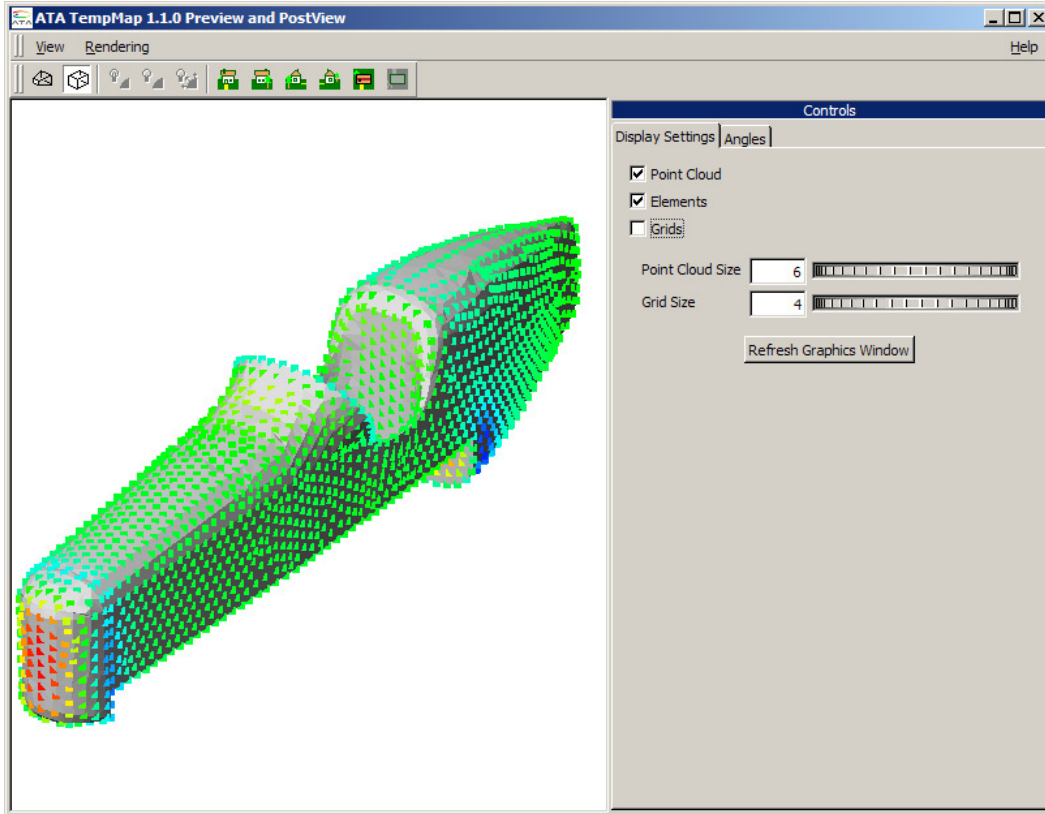


Figure 4. TempMap Preview display.

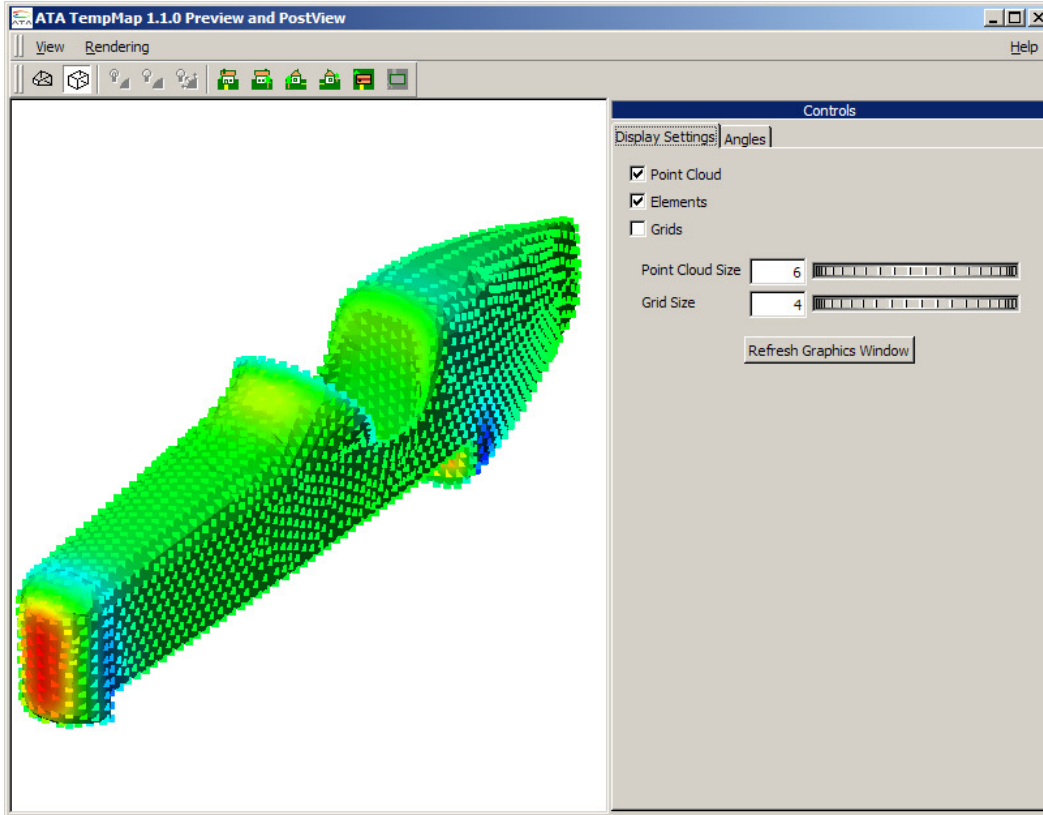


Figure 5. TempMap PostView display.

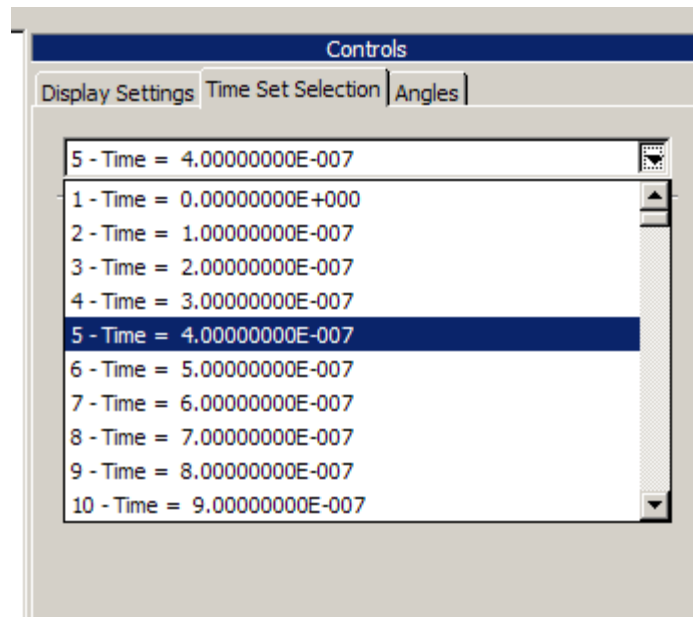


Figure 6. Selection of time set for cases with multiple time sets.

TempMap License

TempMap uses the Sentinel RMS server for licensing. This is the current license server for ATA products. For more details on this server, please go to <http://www.ata-e.com/software/license-server-ata-software>.

TempMap Revision History

Version 1.2.0

- Bug fix in triangulation that caused fatal error
- Mapping improvements for quad shapes
- Mapping improvements for near edge
- Use element normal is now “off” by default

Version 1.1.0

- Added Preview and PostView capability
- Added additional advanced options, allowing more flexibility in the mapping

Version 1.0.4

- Fixed 3-D to local 2-D projection error

Version 1.0.3

- Improved mapping algorithm for sparse temperature data by including local 2-D Delaunay triangulation of nearby temperatures

Version 1.0.2

- Releases license by pressing **X** on GUI window
- Prints only first instance of node with no mapped temperature, which prevents the GUI text window from being overloaded with messages and crashing

Version 1.0.1

- Included a check for duplicate grid ID numbers when reading temperature data
- Fixed documentation errors

Version 1.0.0

- Original release of TempMap

Appendix A: Temperature File Contents and Format

The temperature data file is delimited into blocks of data by keywords. Not all of the keywords are necessary, but if included, the keywords must be in the order shown below.

1. **TITLE.** Optional. If present, must be first record in file.
2. **COORD.** Optional. Defines transformation from the temperature model coordinate system to FEM coordinate system. If not present, the mapping program assumes that the temperature model and FEM coordinate systems are identical.
3. **SYMM.** Optional. Allows the temperature data to be reflected through a symmetry plane.
4. **TIMES.** Optional (*required for multiple times steps*). Allows for temperatures at multiple time steps or cases.
5. **OUTTIMES.** Optional. Allows for multiple output time steps.
6. **DYNAMIC.** Optional. Prints out data for a transient analysis. Only the keyword **DYNAMIC** is needed. The keyword **DYNAMIC** is ignored for I-deas.
7. **TEMPERATURE.** Required. Coordinates of the points and the temperature at these points. At least one point is required.
8. **CYL.** Optional. Sets the coordinate system as cylindrical. Otherwise, the data are in Cartesian coordinates. If this card is used, it must be on the next non-comment line following **TEMPERATURE**.

An alternate format of the temperature data file replaces the **TEMPERATURE** keywords with **GRIDS** and **TEMPS**. This will be discussed later.

The format of the temperature data file is as follows:

1. For comments, use a \$. Any information after the first \$ will be ignored:
For example:

```
$ This entire line is a comment
3.4 2.7 4.5 0.1 $ all data after the first "$" are ignored.
```

2. For a title, two lines are necessary. The first line has the keyword **TITLE** as its first 5 characters, and the second line contains the title. For example:

```
TITLE
The next uncommented line is the title for the results
```

3. For a coordinate transformation, 4 lines are necessary. The first line contains the keyword **COORD** as its first 5 characters, and the next 3 lines contain the definition of the coordinate system transformation, as shown below. The data lines are free-format:

```
COORD
X0 Y0 Z0 $ The offset from TM origin to FEM origin in the FEM CS.
X1 Y1 Z1 $ TM X-axis vector in FEM coordinate system (FEM CS).
X2 Y2 Z2 $ A vector defining the TM XZ plane in the FEM CS.
```

The transformation from the temperature model to FEM is shown by the following equations:

$$\vec{X}_{FEM} = \begin{bmatrix} \vec{X} \\ \vec{Y} \\ \vec{Z} \end{bmatrix}^T \vec{X}_{TEMP} + \vec{X0}$$

where :

\vec{X}_{TEMP} = coordinates in the temperature model coordinate system

\vec{X}_{FEM} = coordinates in the FEM coordinate system

$$\vec{X} = \frac{\vec{X1}}{|\vec{X1}|}, \quad \vec{Y} = \frac{(\vec{X2} \times \vec{X1})}{|\vec{X2} \times \vec{X1}|}, \quad \vec{Z} = \vec{X} \times \vec{Y}$$

$$\vec{X0} = (X0, Y0, Z0), \quad \vec{X1} = (X1, Y1, Z1), \quad \vec{X2} = (X2, Y2, Z2)$$

4. For the **SYMM** block, 3 lines are necessary. The first line contains the keyword **SYMM** as its first 4 characters, and the next 2 lines contain the definition of a symmetry plane in the FEM coordinate system, as shown below. The data lines are free-format:

```
SYMM
X0 Y0 Z0 $ A point on the symmetry plane in the FEM CS.
X1 Y1 Z1 $ The normal of the symmetry plane in the FEM CS.
```

For each body point not on the symmetry plane, an additional temperature point location is created by the following equations:

$$\vec{X}_{PSYMM} = \vec{X}_P - 2[(\vec{X}_P - \vec{X0}) \cdot \hat{X1}] \hat{X1}$$

where :

\vec{X}_P = original temperature point coordinates.

\vec{X}_{PSYMM} = coordinates of symmetric temperature point.

$$\hat{X1} = \frac{\vec{X1}}{|\vec{X1}|}, \quad \vec{X0} = (X0, Y0, Z0), \quad \vec{X1} = (X1, Y1, Z1)$$

5. For the input time data, at least 2 lines are needed. The first line contains the keyword **TIMES** as its first 5 characters, the next line contains the number of time values, and the third line and beyond contain the time values in ascending order, with 4 values per line, except for the last line, which can have 1 to 4 values.

```
TIMES
N          $ Number of times
T1 T2 T3 T4 $ Time values
...
TN-1 TN    $ Final row can have less than 4 values
```

If only 2 lines are input, the time values will default to 1, 2, 3, etc. For instance:

```
TIMES
10
```

will generate times 1 to 10.

- For the output time data, at least 3 lines are needed. The first line contains the keyword **OUTTIMES** as its first 8 characters, the next line contains the number of time values, and the third line and beyond contain the time values in ascending order, with 4 values per line, except for the last line, which can have 1 to 4 values.

```
OUTTIMES
N          $ Number of times
T1 T2 T3 T4 $ Time values
T5 T6 T7 T8 $ Additional Time values
...
TN-1 TN    $ Final row can have less than 4 values
```

If input times are defined but no output times are defined, the output times will be the same as the input times. Also, if output times are between input times, TempMap will interpolate the temperatures between two input times. Any output times greater or less than the input times will be ignored.

- For the temperature data, at least two lines are needed. The first contains the keyword **TEMPERATURE** as its first 11 characters, and each line that follows contains the coordinates of a grid point with the temperature at that point. The data lines are free-format. An example is shown below:

```
TEMPERATURE
X1 Y1 Z1 T1 $ Point and temperature
X2 Y2 Z2 T2
X3 Y3 Z3 T3
```

If the data are in cylindrical coordinates, the temperature data will be as follows:

```
TEMPERATURE
CYL
R1 TH1 Z1 T1 $ Point and temperature
R2 TH2 Z2 T2
R3 TH3 Z3 T3
```

where TH_i are theta angles in degrees. The cylindrical coordinates will be read, converted to Cartesian coordinates, and stored in memory.

If input times are defined, additional lines are needed for each temperature, with 4 values per line, except for the last line, which can have 1 to 4 values, as shown below:

```
TIMES
6
T1 T2 T3 T4
T5 T6
TEMPERATURE
X1      Y1      Z1      T1_1 $ Point and temperature
T1_2    T1_3    T1_4    T1_5 $ Additional Temperatures
T1_6                                $ Additional Temperatures
X2      Y2      Z2      T2_1
T2_2    T2_3    T2_4    T2_5
T2_6
```

For cylindrical coordinates:

```
TIMES
6
T1 T2 T3 T4
T5 T6
TEMPERATURE
CYL
R1      TH1     Z1      T1_1 $ Point and temperature
T1_2    T1_3    T1_4    T1_5 $ Additional Temperatures
T1_6                                $ Additional Temperatures
R2      TH2     Z2      T2_1
T2_2    T2_3    T2_4    T2_5
T2_6
```

The following are valid temperature files:

```

TITLE
EXAMPLE TEMPERATURES
COORD
  9.30000E+01    0.00000E+00    0.00000E+00
  1.00000E+00    0.00000E+00    0.00000E+00
  0.00000E+00    0.00000E+00    1.00000E+00
$
SYMM
  9.30000E+01    0.00000E+00    0.00000E+00
  0.00000E+00    1.00000E+00    0.00000E+00
$
$   X, inch      Y, inch      Z, inch      T
$
TEMPERATURE
  1.14044E+01    -8.17367E+00    -2.34699E+01    24.1345
  6.42327E+00    -6.33882E+00    -2.34646E+01    11.4598
  1.14044E+01    -1.83098E+00    -3.27500E+01    10.2369
  6.46169E+02    -2.91713E+01    -3.26565E+01    -24.8273
  1.97866E+02    -2.94464E+01    -3.26963E+01    -48.8448
  2.35048E+02    -3.51424E+01    -2.62059E+01    -12.9118

```

and

```

TITLE
EXAMPLE TEMPERATURES, MULTIPLE TIMES
COORD
  9.30000E+01    0.00000E+00    0.00000E+00
  1.00000E+00    0.00000E+00    0.00000E+00
  0.00000E+00    0.00000E+00    1.00000E+00
SYMM
  9.30000E+01    0.00000E+00    0.00000E+00
  0.00000E+00    1.00000E+00    0.00000E+00
TIMES
3
0.0  1.0  2.0
OUTTIMES
4
0.0  0.5  1.0  1.5
DYNAMIC
$   X, inch      Y, inch      Z, inch      T1
$   T2          T3 ...
TEMPERATURE
  1.14044E+01    -8.17367E+00    -2.34699E+01    24.1345
  24.1345        24.1345
  6.42327E+00    -6.33882E+00    -2.34646E+01    11.4598
  11.4598        11.4598
  1.14044E+01    -1.83098E+00    -3.27500E+01    10.2369
  10.2369        10.2369

```

Keep in mind that the records are free-format; the indentations and adherence to columns shown here are used for legibility, not because they are required.

The alternate temperature data file is delimited into blocks of data by keywords. Not all of the keywords are necessary, but if included, the keywords must be in the order shown below.

1. **TITLE.** Optional. If present, must be the first record in the file.
2. **COORD.** Optional. Defines transformation from the temperature model coordinate system to FEM coordinate system. If not present, the mapping program assumes that the temperature model and FEM coordinate systems are identical.
3. **SYMM.** Optional. Allows the temperature data to be reflected through a symmetry plane.
4. **TIMES.** Optional (*required for multiple times steps*). Allows for temperatures at multiple time steps or cases.
5. **OUTTIMES.** Optional. Allows for multiple output time steps.
6. **DYNAMIC.** Optional. Prints out data for a transient analysis. Only the keyword **DYNAMIC** is needed. The keyword **DYNAMIC** is ignored for I-deas.
7. **GRIDS.** Required. Defines point IDs and coordinates of the points. At least one point is required.
8. **CYL.** Optional. Sets the coordinate system as cylindrical. Otherwise, the data are in Cartesian coordinates. If this card is used, it must be on the first non-comment line following **GRIDS**.
9. **TEMPS.** Required. Defines point IDs and the temperature(s) at these points. At least one is required. The numbers and values of all of the point IDs must match for each point ID defined by the **GRIDS** keywords.

For the **GRIDS** and **TEMPS** data, at least two lines are needed for each. The first contains the keyword **GRIDS** as its first 5 characters, and each line that follows contains the point ID for a grid point and the coordinates of the grid point.

The **GRIDS** are followed by the **TEMPS**. The first contains the keyword **TEMPS** as its first 5 characters, and each line that follows contains the point ID for a grid point with the temperature at that point.

The data lines are free-format. An example is shown below:

```
GRIDS
GR1 X1 Y1 Z1 $ Point ID and coordinates
GR2 X2 Y2 Z2
GR3 X3 Y3 Z3
TEMPS
GR1 T1 $ Point ID and temperature
GR2 T2
GR3 T3
```

If the data are in cylindrical coordinates, the temperature data will be as follows:

```
GRIDS
CYL
GR1 R1 TH1 Z1 $ Point ID and coordinates
GR2 R2 TH2 Z2
GR3 R3 TH3 Z3
TEMPS
GR1 T1 $ Point ID and temperature
GR2 T2
GR3 T3
```

where THi are theta angles in degrees. The cylindrical coordinates will be read, converted to Cartesian coordinates, and stored in memory.

If input times are defined, additional lines are needed for each temperature, with 4 values per line, except for the last line, which can have 1 to 4 values, as shown below:

```
TIMES
6
T1 T2 T3 T4
T5 T6
GRIDS
GR1 X1 Y1 Z1 $ Point ID and coordinates
GR2 X2 Y2 Z2
TEMPS
GR1 T1_1 T1_2 T1_3 $ Point ID and temperature
T1_4 T1_5 T1_6 $ Additional Temperatures
GR2 T2_1 T2_2 T2_3
T2_4 T2_5 T2_6
```

For cylindrical coordinates:

```
TIMES
6
T1 T2 T3 T4
T5 T6
GRIDS
CYL
GR1 R1 TH1 Z1 $ Point ID and coordinates
GR2 R2 TH2 Z2
TEMPS
GR1 T1_1 T1_2 T1_3 $ Point ID and temperature
T1_4 T1_5 T1_6 $ Additional Temperatures
GR2 T2_1 T2_2 T2_3
T2_4 T2_5 T2_6
```

The following are valid alternate format temperature files:

```
TITLE
EXAMPLE TEMPERATURES
COORD
  9.30000E+01    0.00000E+00    0.00000E+00
  1.00000E+00    0.00000E+00    0.00000E+00
  0.00000E+00    0.00000E+00    1.00000E+00
$
SYMM
  9.30000E+01    0.00000E+00    0.00000E+00
  0.00000E+00    1.00000E+00    0.00000E+00
GRIDS
  1    1.14044E+01    -8.17367E+00    -2.34699E+01
  2    6.42327E+00    -6.33882E+00    -2.34646E+01
  3    1.14044E+01    -1.83098E+00    -3.27500E+01
TEMPS
  1    24.1345
  2    11.4598
  3    10.2369
```

and

```
TITLE
EXAMPLE TEMPERATURES, MULTIPLE TIMES
COORD
  9.30000E+01    0.00000E+00    0.00000E+00
  1.00000E+00    0.00000E+00    0.00000E+00
  0.00000E+00    0.00000E+00    1.00000E+00
SYMM
  9.30000E+01    0.00000E+00    0.00000E+00
  0.00000E+00    1.00000E+00    0.00000E+00
TIMES
5
0.0  0.5  1.0  1.5  2.0
OUTTIMES
4
0.0  0.5  1.0  1.5
DYNAMIC
GRIDS
  1    1.14044E+01    -8.17367E+00    -2.34699E+01
  2    6.42327E+00    -6.33882E+00    -2.34646E+01
  3    1.14044E+01    -1.83098E+00    -3.27500E+01
TEMPS
  1    24.1345    24.1345    24.1345
  24.1345    24.1345
  2    11.4598    11.4598    11.4598
  11.4598    11.4598
  3    10.2369    10.2369    10.2369
  10.2369    10.2369
```

TempMap also allows the user to include additional files by use of a **#include** statement. If the user has an input file with the following:

```
TITLE
EXAMPLE TEMPERATURES
COORD
  9.30000E+01    0.00000E+00    0.00000E+00
  1.00000E+00    0.00000E+00    0.00000E+00
  0.00000E+00    0.00000E+00    1.00000E+00
$
SYMM
  9.30000E+01    0.00000E+00    0.00000E+00
  0.00000E+00    1.00000E+00    0.00000E+00
$
#include "file2.dat"
```

and a **file2.dat** that has the lines:

```
$
$   X, inch          Y, inch          Z, inch          T
$
TEMPERATURE
  1.14044E+01    -8.17367E+00    -2.34699E+01    24.1345
  6.42327E+00    -6.33882E+00    -2.34646E+01    11.4598
  1.14044E+01    -1.83098E+00    -3.27500E+01    10.2369
  6.46169E+02    -2.91713E+01    -3.26565E+01    -24.8273
  1.97866E+02    -2.94464E+01    -3.26963E+01    -48.8448
  2.35048E+02    -3.51424E+01    -2.62059E+01    -12.9118
```

then TempMap will read both files and concatenate them together.