

NIST Technical Note 1828

Software Independent Data Mapping Tool for Structural Fire Analysis

Dilip K. Banerjee

<http://dx.doi.org/10.6028/NIST.TN.1828>

NIST Technical Note 1828

Software Independent Data Mapping Tool for Structural Fire Analysis

Dilip K. Banerjee
*Materials Science and Engineering Division
Materials Measurement Laboratory*

<http://dx.doi.org/10.6028/NIST.TN.1828>

February 2014



U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Under Secretary of Commerce for Standards and Technology and Director

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

National Institute of Standards and Technology Technical Note 1828
Natl. Inst. Stand. Technol. Tech. Note 1828, 45 pages (February 2014)

CODEN: NTNOEF

<http://dx.doi.org/10.6028/NIST.TN.1828>

Software Independent Data Mapping Tool for Structural Fire Analysis

Dilip K. Banerjee

Abstract

In order to model the structural behavior under fire, three separate analyses need to be conducted: a) fire propagation and growth (fire modeling), b) transient heat transfer in structural members due to fire, and c) structural analysis to account for both thermal and mechanical load. Advanced methods for fire modeling are based on computational fluid dynamics (CFD) approach. Typical heat transfer analysis is conducted using finite element analysis (FEA) approach employing solid 3-D or 2-D shell elements. Structural analysis is often conducted using a FEA approach, but employing beam and shell elements, especially for large structures. While both thermal and structural analyses are conducted using finite elements, fire modeling is often carried out using finite difference method (FDM), where the computational domain is discretized using rectangular grids. The computational domain is made up of right parallelepipedic volumes called meshes. Fire modeling is also conducted with the finite element method in some cases.

Typical outputs of a fire model are heat flux and gas temperature field, which are inputs to the heat transfer model as thermal boundary conditions. Subsequently, the transient temperatures computed by the heat transfer analysis in the entire computational domain comprising structural members are inputs to the structural analysis model. However, this transfer of data is rather complicated because of the difference in the level and type of discretization used in each of these analyses. As mentioned earlier, fire modeling using CFD approach uses FDM, while both heat transfer and structural analyses use FEA approach. Since the type of discretization is different, a mapping tool is therefore required to transfer temperatures and heat flux from fire model to a heat transfer model. This is because the coordinates of nodes or grid points in both models may vary. On the other hand, the transfer of data from heat transfer model to structural analysis model poses additional challenges as different types of elements are used in each analysis (e.g., solid elements in thermal model while beam and shell elements in structural model). A mapping tool is also required to transfer the thermal data from heat transfer model to structural analysis model.

This document describes the approach developed for both types of data transfer. Transfer of data from fire model to heat transfer model is discussed first. Then the transfer of data from heat transfer model to structural analysis model is discussed.

Table of Contents

1. Introduction	3
2. Data Transfer from Fire Model to Heat Transfer Model.....	4
2.1. Run FDS2ASCII from command line:	7
2.2. Strip header and text information in FDS output file	8
2.3. Create a FE model for thermal modeling:.....	9
2.4. MATLAB data mapping script:.....	9
2.5 Data Formats:.....	14
3. Thermal Model to Structural Model Data Transfer	15
3.1 Theory	15
3.1.1 Beam or Line Elements	15
3.1.2 Shell Elements	17
3.2 Temperature Mapping Methodology:	18
3.3 Data Mapping Script:	21
3.4 Verification	23
3.4.1 Fireproofed Steel Beam Temperatures:	23
3.4.2 Unprotected Steel Beam Temperatures:	27
3.4.3 Unprotected Steel Beam Deflections:.....	28
3.4.4 Cardington Test Results:	29
3.4.5 Complete Fire to Thermal to Structural Data Transfer	35
3.4.6 Data Formats:.....	39
4. Summary	41
5. References	41
6. Appendix 1 (Time-temperature curves used for Figure 21)	42

1. Introduction

To predict structural behavior under fire, both spatial and temporal variations of temperatures in structural members need to be accurately determined. The ability to predict with high confidence time-varying temperature profiles in structural members is necessary for a robust performance-based approach to the fire resistance design of structures. In most cases, computational modeling approaches are used to obtain structural temperatures based on exposure to a particular fire scenario. Structural analyses are then conducted with these temperatures obtained at specific times into the fire event. Typically, a comprehensive modeling approach requires three separate analyses: a) fire dynamics modeling, b) heat transfer analysis, and c) structural analysis. Each of these analyses typically involves different computational approaches such that seamless transfer of thermal data between models is difficult. For example, fire dynamics is typically modeled with computational fluids dynamics (CFD) approach using finite difference method (FDM). On the other hand, heat transfer (thermal) analysis and structural analysis are conducted using finite element (FE) method. Most heat transfer analysis employ solid hexahedral or shell elements while structural models use beam and shell elements. Transfer of data between FDM and FE meshes or FE to FE meshes (fire to thermal) and FE thermal and structural models (thermal to structural) are not straightforward and require tools for mapping data appropriately. The present technical note describes the development of two MATLAB¹ scripts for transferring thermal data between fire and thermal and thermal and structural models (Fig. 1).

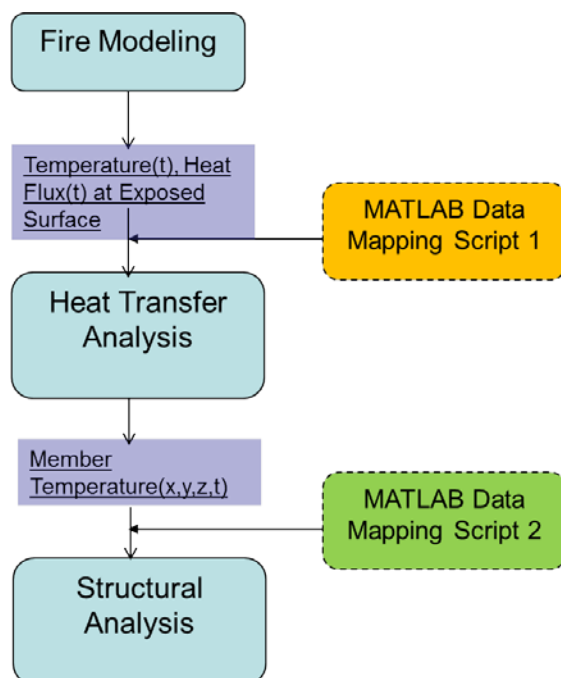


Fig. 1. A schematic representation of the data flow in fire-thermal-structural analysis.

¹ Certain commercial software or materials are identified to describe a procedure or concept adequately. Such identification is not intended to imply recommendation, endorsement, or implication by NIST that the software or materials are necessarily the best available for the purpose.

2. Data Transfer from Fire Model to Heat Transfer Model

In this section, the transfer of relevant data from a computational fluid dynamics (CFD) model for fire dynamics to a finite element model for heat transfer in structural members is discussed. CFD model provides the most comprehensive modeling approach for simulation of fire initiation, propagation, and growth. NIST Fire Dynamics Simulator (FDS) is one such software [1]. In the following section, a procedure is discussed for transferring thermal data from a FDS model to a finite element analysis (FEA) model of structural members for subsequent heat transfer calculations.

FDS numerically solves the spatially filtered form of the Navier-Stokes equations appropriate for incompressible flow using a technique known as large-eddy simulation (LES). The mass, momentum, and energy equations are discretized by finite differences and a numerical solution is obtained as a function of time on a 3-D rectilinear grid. All FDS calculations must be performed within a domain that is made up of right parallelepipedic volumes called meshes. Each mesh is divided into rectilinear volumes or cells. The number of cells depends on the desired resolution of the computed flow dynamics calculations.

Typical output quantities from FDS that are of interest here include a) exposed surface heat flux as a function of time and b) Adiabatic Surface Temperature (AST) as a function of time. As a first example, the transfer of adiabatic surface temperatures from a FDS model to a FEA heat transfer model is discussed.

AST is defined as the temperature of a surface which cannot absorb or lose heat to the environment, i.e. a perfect insulator. Typically, it is a weighted mean temperature of the radiation temperature and the gas temperature depending on the heat transfer coefficients.

The first step is to develop an FDS input file, which describes the computational domain and defines the structural member(s) as obstruction(s) [3]. FDS can be instructed to output AST on exposed surface as boundary files. This means that the AST data computed on all grid points at the boundary of structural members are written to a binary output file. The data mapping between FDS model and FEA thermal model is challenging because computational meshes do not match as shown in Fig. 2.

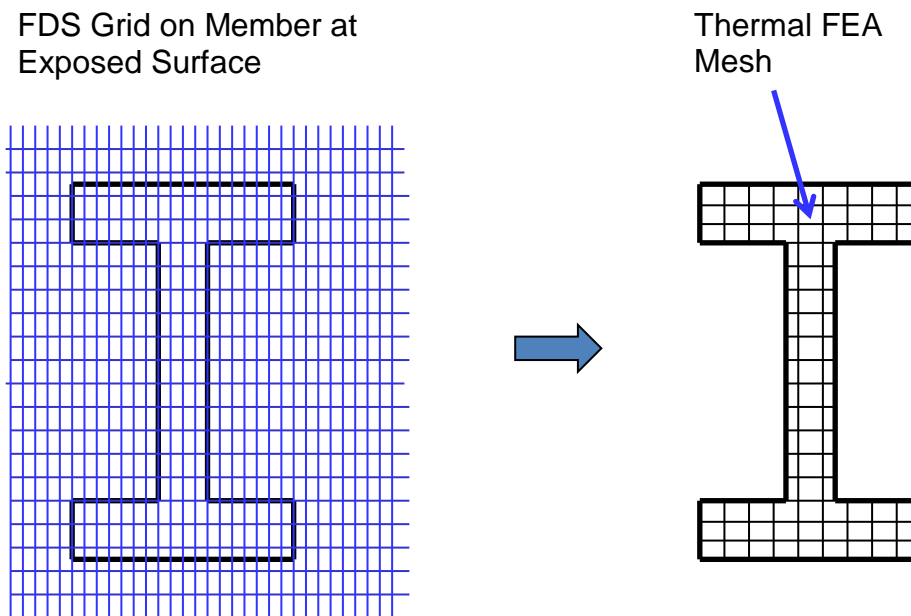


Fig. 2. A schematic representation of computation grid in FDS and a typical FEA mesh for thermal analysis.

Selected portion from a typical FDS input file is shown in Fig. 3. Note that the six lines starting with “&OBST” in Fig. 3 define top flange, web, and bottom flange of a girder as obstructions. The last line starting with “&BNDF” in Fig. 3 instructs FDS to write AST data on grid points located on the surface of the girder. Such data are written by FDS as “*.bf” files [1]. Since these data are written in binary format, they need to be converted to ASCII format for subsequent transfer to a FEA heat transfer model. Since FDS generates a large amount of computational data, a major challenge is to extract only relevant numeric data needed for subsequent heat transfer analysis. Parsing of computational output data is needed for a software-independent tool to work. Therefore, there is a need to create unformatted data files free of any header or text information as typically written out by FDS. The sequence of steps needed for data mapping from FDS to a FEA heat transfer model is shown in the flowchart in Fig. 4.


```

&HEAD CHID='case_5_v1_test',TITLE='FDS to ANSYS mapping Exercise, FDS Version 5' /
&MESH IJK=240,200,160,XB=0.0,6.0,0.0,5.0,0.0,4.0 /
&TIME TWFN=30.0 /
&MISC SURF_DEFAULT='GYPSUM BOARD', BDNF_DEFAULT=.FALSE /

.....
&SURF ID      = 'INSULATED BEAM FL VERT'
COLOR         = 'ANTIQUE WHITE'
BACKING       = 'EXPOSED'
MATL_ID(1:2,1) = 'INSULATION','STEEL'
THICKNESS(1:2) = 0.05,0.0375 /
.....

W33x130 girder Example

&OBST XB=3,3.4,0.5,4.5,3.1,3.1,SURF_ID='INSULATED BEAM FL VERT', BDNF_OBST=.TRUE/ top surf of top flange
&OBST XB=3,3.4,0.5,4.5,2.925,2.925,SURF_ID='INSULATED BEAM FL VERT', BDNF_OBST=.TRUE/ bot surf of top flange
&OBST XB=3,3,0.5,4.5,2.925,3.1,SURF_ID='INSULATED BEAM FL HOR', BDNF_OBST=.TRUE/ left side of top flange
&OBST XB=3,4,3.4,0.5,4.5,2.925,3.1,SURF_ID='INSULATED BEAM FL HOR', BDNF_OBST=.TRUE/ right side of top flange
&OBST XB=3,3.4,0.5,0.5,2.925,3.1,SURF_ID='INSULATED BEAM FL EDGE', BDNF_OBST=.TRUE/ front side of top flange
&OBST XB=3,3.4,4.5,4.5,2.925,3.1,SURF_ID='INSULATED BEAM FL EDGE', BDNF_OBST=.TRUE/ back side of top flange

.....
&BNDF QUANTITY='GAUGE_HEAT_FLUX',CELL_CENTERED=.TRUE/
&BNDF QUANTITY='WALL_TEMPERATURE',CELL_CENTERED=.TRUE/
&BNDF QUANTITY='ADIABATIC_SURFACE_TEMPERATURE',CELL_CENTERED=.TRUE/

```

Fig. 3. A portion of a typical FDS input file.

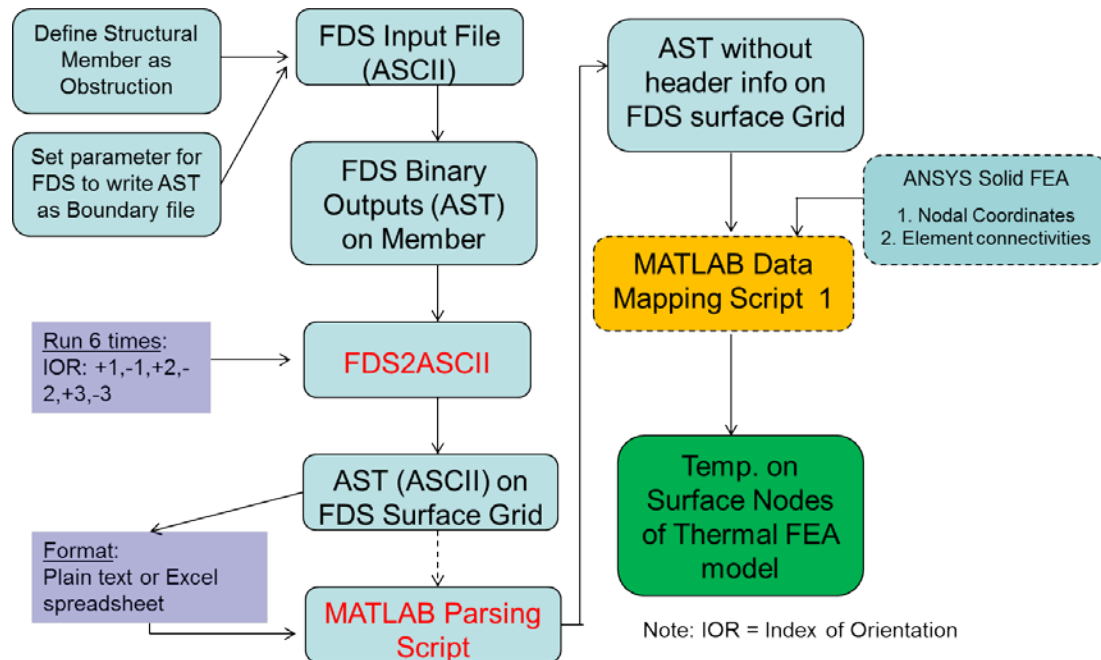


Fig. 4. A flowchart showing sequence of steps for transferring FDS temperatures to exposed surface of structural members in a heat transfer model.

A program called “FDS2ASCII” (which comes with the standard installation of the FDS software) can be run to convert the AST boundary file data from binary to ASCII format [3]. FDS2ASCII can be run interactively or in batch mode. A typical output from an interactive FDS2ASCII run session is shown in Fig. 5.

```
[dilip@bfr1104166 case_5_v1_test_fds_thermal]$ fds2ascii
Enter Job ID string (CHID):
case_5_v1_test
What type of file to parse?
PL3D file? Enter 1
SLCF file? Enter 2
BNDF file? Enter 3
3
Enter Sampling Factor for Data?
(1 for all data, 2 for every other point, etc.)
1
Domain selection:
y - domain size is limited
n - domain size is not limited
z - domain size is not limited and z levels are offset
ya, na or za - slice files are selected based on type and
location.
The y, n, z prefix are defined as before.
n
1 MESH 1, GAUGE HEAT FLUX
2 MESH 1, WALL TEMPERATURE
3 MESH 1, ADIABATIC SURFACE TEMPERATURE
Enter starting and ending time for averaging (s)
7.9 8.0
Enter orientation: (plus or minus 1, 2 or 3)
1
Enter number of variables
3
Enter boundary file index for variable 1
1
Enter boundary file index for variable 2
2
Enter boundary file index for variable 3
3
Enter output file name:
case_5_v1_test_xp_alt
Writing to file... case_5_v1_test_xp_alt
```

Fig. 5. A typical output from an interactive “FDS2ASCII” run.

Since it is cumbersome to run FDS2ASCII in interactive mode, an input script has been developed to run FDS2ASCII from command line (Fig. 6). FDS2ASCII writes output files with header information and text information, which needs to be stripped. The sequence of steps is written below:

2.1. Run FDS2ASCII from command line:

FDS2ASCII needs to be run 6 times to write the entire boundary file data (e.g., in +X, -X, +Y, -Y, +Z and -Z directions) for all six directions. An input script (Fig. 6) has been developed for each of the six directions. A typical file, called “case_5_v1_test_xm_input”, is shown in Fig. 6. Note that the file has all the information a user will enter in an interactive session as shown in Fig. 5. The last line shows the name of the output file of the program. The program can then be executed as follows.

Run as: `$ fds2ascii < case_5_v1_test_xm_input [for -X direction]`

This creates an ASCII output file (Fig. 7) called “case_5_v1_test_xm” listing x, y, z coordinates and boundary file quantities, e.g., AST, Heat flux etc.

```
case_5_v1_test
3
1
n
13.2 13.3
-1
3
1
2
3
case_5_v1_test_xm
```

Fig. 6. A typical input file for running FDS2ASCII program in batch mode.

2.2. Strip header and text information in FDS output file

Fig. 7 shows first few lines of the typical output file as written out by the FDS2ASCII program:

```
Patch 5 3.00<x< 3.00, 0.50<y< 4.50, 2.92<z< 3.10
X,Y,Z,GAUGE HEAT FLUX,WALL TEMPERATURE,ADIABATIC SURFACE TEMPERATURE
m,m,m,kW/m2,C,C
0.30000E+01, 0.51250E+00, 0.29375E+01, 0.66834E+02, 0.64825E+03, 0.73548E+03
0.30000E+01, 0.53750E+00, 0.29375E+01, 0.66814E+02, 0.64902E+03, 0.73529E+03
0.30000E+01, 0.56250E+00, 0.29375E+01, 0.66724E+02, 0.64968E+03, 0.73493E+03
0.30000E+01, 0.58750E+00, 0.29375E+01, 0.66688E+02, 0.65032E+03, 0.73478E+03
0.30000E+01, 0.61250E+00, 0.29375E+01, 0.66606E+02, 0.65090E+03, 0.73447E+03
0.30000E+01, 0.63750E+00, 0.29375E+01, 0.66535E+02, 0.65134E+03, 0.73423E+03
0.30000E+01, 0.66250E+00, 0.29375E+01, 0.66464E+02, 0.65169E+03, 0.73401E+03
0.30000E+01, 0.68750E+00, 0.29375E+01, 0.66305E+02, 0.65183E+03, 0.73345E+03
0.30000E+01, 0.71250E+00, 0.29375E+01, 0.66178E+02, 0.65190E+03, 0.73294E+03
0.30000E+01, 0.73750E+00, 0.29375E+01, 0.65949E+02, 0.65186E+03, 0.73206E+03
```

Fig. 7. First few lines of a typical boundary output file written out by the FDS2ASCII program.

Note that the FDS output file containing boundary data has text information as header lines and also at several locations in the output file. A MATLAB script has been written which can read the above FDS ASCII output data as shown in the above figure in either Microsoft Excel format or as text file and can remove all the text information from the file and create an ASCII file (comma separated) with X, Y, Z coordinates of grid points and corresponding values of output quantities (heat flux, wall temperature, and AST) as shown in Fig. 8.

3.0000	0.5125	2.9375	66.8340	648.2500	735.4800
3.0000	0.5375	2.9375	66.8140	649.0200	735.2900
3.0000	0.5625	2.9375	66.7240	649.6800	734.9300
3.0000	0.5875	2.9375	66.6880	650.3200	734.7800
3.0000	0.6125	2.9375	66.6060	650.9000	734.4700
3.0000	0.6375	2.9375	66.5350	651.3400	734.2300
3.0000	0.6625	2.9375	66.4640	651.6900	734.0100
3.0000	0.6875	2.9375	66.3050	651.8300	733.4500
3.0000	0.7125	2.9375	66.1780	651.9000	732.9400
3.0000	0.7375	2.9375	65.9490	651.8600	732.0600

Fig. 8. First few lines of a typical FDS boundary output file after running MATLAB parsing script on data shown in Fig. 7.

Note that FDS writes X, Y, Z coordinates of each grid point on the boundary and corresponding output quantities as requested in the FDS input file. The parsed output file contains all the data from the above figure except the header and text lines. Five other files can be written by the MATLAB program (in other 5 directions using input scripts similar to the one in Fig. 6). These six files contain output (wall temperature, heat flux, and AST in this example) data for all grid points on the surface of a structural member at a given time into the fire simulation.

2.3. Create a FE model for thermal modeling:

A finite element model of structural members needs to be created in any FEA software. The geometry needs to be discretized using shell or solid hexahedral elements. The geometry of the structural member needs to have the same dimensions as those defined in FDS. Also, the global coordinates and origin in both the FEA heat transfer model and FDS need to match.

2.4. MATLAB data mapping script:

A MATLAB script has been developed to perform the following:

- Identify the nodes that are on the boundary of the structural member as defined in the FE heat transfer model. It reads the FE nodal coordinates and element connectivity data (from the heat transfer model) and by using a unique algorithm, it identifies the surface nodes in the FE model and collects their nodal coordinates into a matrix (see description below). Create a 3-D uniform, orthogonal grid in MATLAB using minimum and maximum x, y, z values in the FE model. This is the target matrix for the data transfer.

- b) Read all six FDS boundary ASCII output files (from section 2.2 above) and assemble them into a matrix form in “X, Y, Z, AST” format. This matrix is the parent matrix for data transfer.
- c) Use MATLAB utility called “Griddata” to map the AST from **step b** into target matrix given in **step a**. This MATLAB function is used to map FDS AST data into the FE 3-D grid (heat transfer model).
- d) Write the nodal (mapped) AST data (Node #, X, Y, Z, AST) in ASCII format, which can be read by any FEA heat transfer software.

The algorithm for finding nodes on the surface of the FE model described in step (a) above works as follows. Using element connectivity data in the FE (heat transfer) model, the data mapping script obtains (1) a column vector containing the number of elements that use each node, (2) a column vector pointing to the beginning of the element list for each node, and (3) finally, a column vector containing a list of elements surrounding each node.

Next, nodes on faces of each element are compared with nodes of other elements that are connected to the same nodes to determine if all nodes along the faces of each pair of elements match. This is possible since an element list was already obtained that contains the elements that are connected to each node. If all nodes along common faces of a pair of elements match, then these nodes are not on free surface. When this match is not found, then nodes of the element face are on free surface and these nodes are written to a column vector. A unique list of such nodes can be obtained using a simple MATLAB functionality that removes any duplication of surface nodes in the column vector containing surface nodes.

As an example, a W33x130 Girder was chosen as a test problem (with 0.5 inch uniform fireproofing) as shown in Fig. 9. The girder is placed in a compartment (Fig. 10) and was exposed to a fire from a burner located at the floor. The girder was defined as an “obstruction” in the FDS input model. AST data on the exposed boundary of the girder were written to output files. These temperatures were then mapped to the boundary of an equivalent ANSYS FE thermal model using the MATLAB Data Mapping Script.

Mapped temperatures at boundary nodes are written to an ASCII file and can be read in any FE heat transfer (HT) software for subsequent thermal analysis. Fig. 10 and 11 compare temperatures on exposed surface of the girder as written by FDS with those mapped on an ANSYS FE model using the MATLAB data mapping script (see quantitative comparisons in Fig. 10). Fig. 12, 13, 14 show three different views of exposed surface temperatures in FDS and corresponding mapped temperatures on the exposed surface of the girder using MATLAB “slice” plots. Mapped temperatures at surface nodes are written in an ASCII “.csv” file. Overall, the mapped temperatures on the exposed surface agree very well with those written out by FDS.

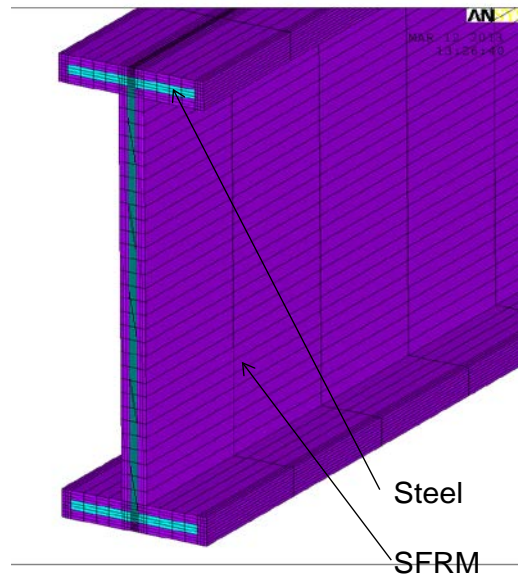


Fig. 9. A FEA model of a fireproofed steel beam.

The following two plots (Fig. 10 and 11) show a comparison of AST data in FDS/Smokeview and mapped AST nodal data in ANSYS as written out by the MATLAB script.

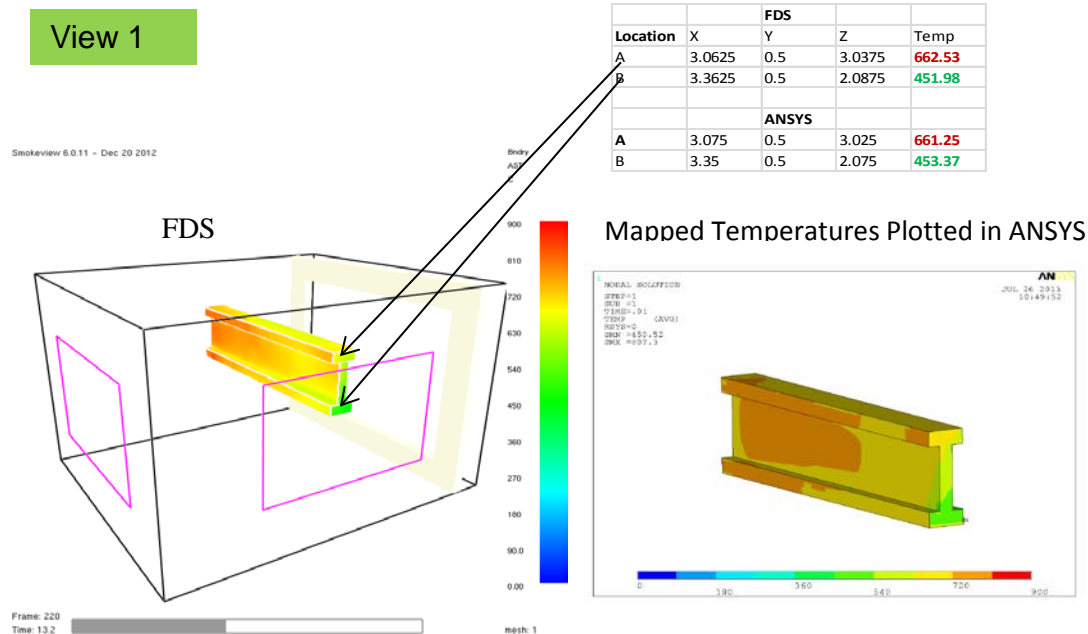


Fig. 10. A comparison of exposed surface temperatures in FDS and those mapped by the MATLAB script and subsequently read in ANSYS (view 1).

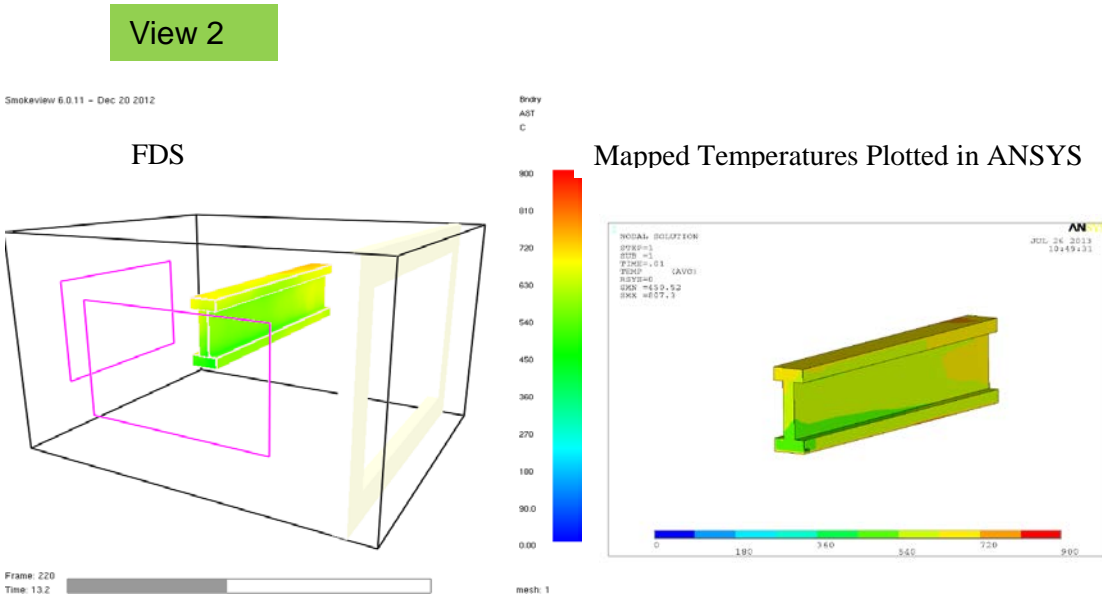


Fig. 11. A comparison of exposed surface temperatures in FDS and those mapped by the MATLAB script and subsequently read in ANSYS (view 2).

A comparison of plots of the adiabatic surface temperature in FDS/Smokeview and mapped temperatures on FE surface nodes using MATLAB script (displayed using MATLAB “slice plot”) is shown in the following three figures (Fig. 12, 13, and 14).

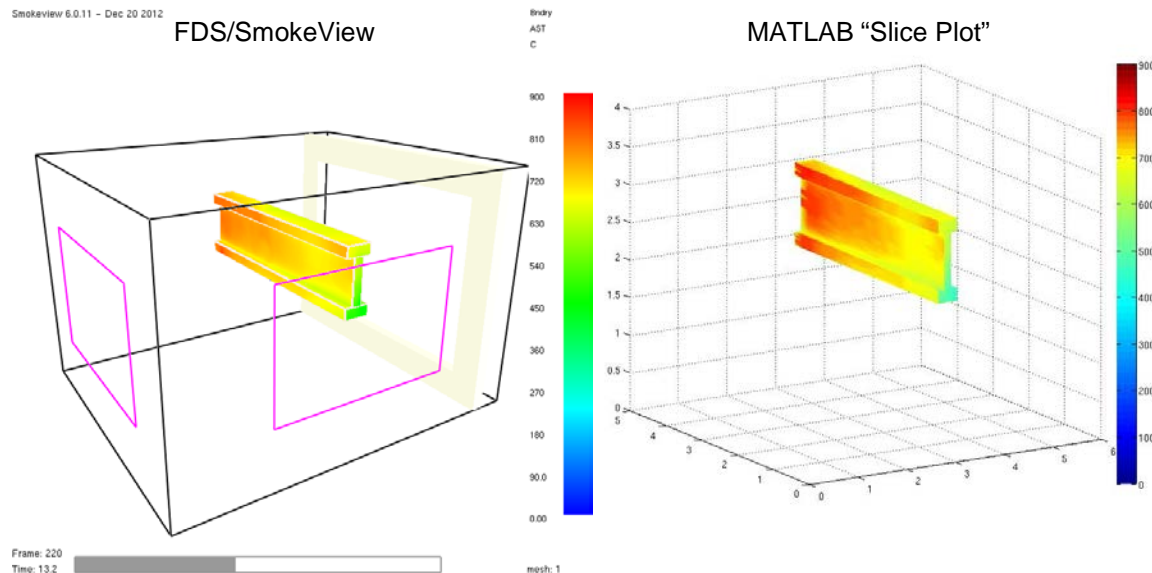


Fig. 12. A comparison of exposed surface temperatures in FDS and those mapped by the MATLAB script and displayed using MATLAB “slice” plot (view 1).

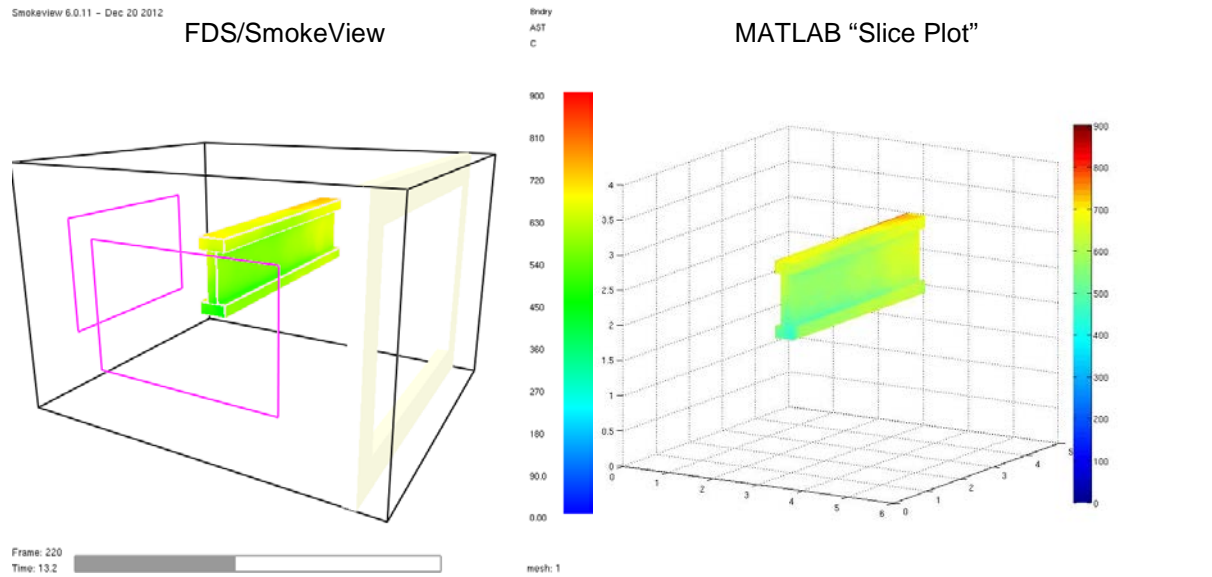


Fig. 13. A comparison of exposed surface temperatures in FDS and those mapped by the MATLAB script and displayed using MATLAB "slice" plot (view 2).

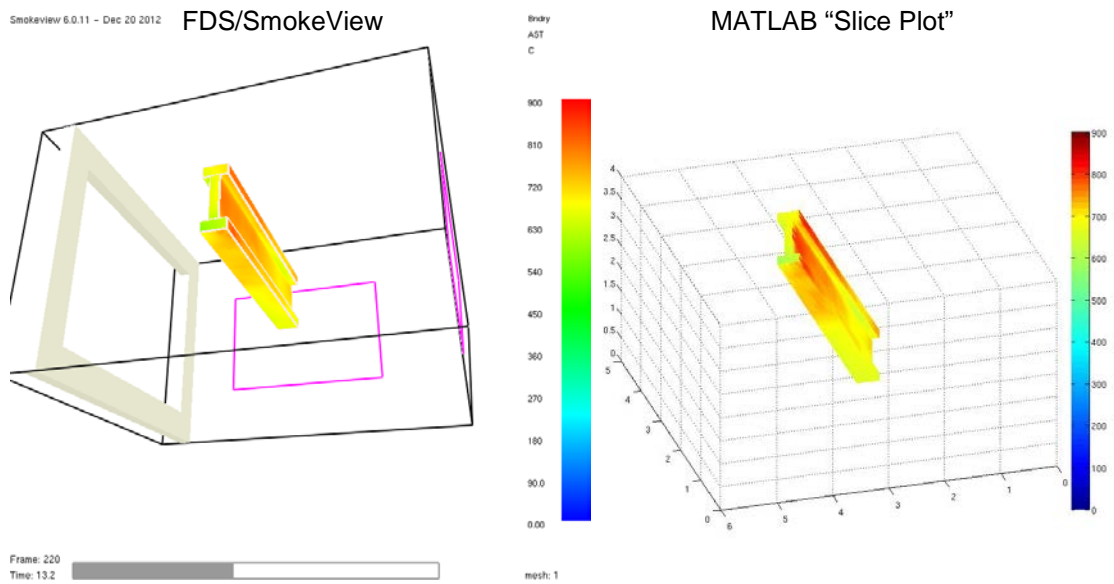


Fig. 14. A comparison of exposed surface temperatures in FDS and those mapped by the MATLAB script and displayed using MATLAB "slice" plot (view 3).

2.5 Data Formats:

The input data to the MATLAB data transfer script is the FDS boundary file data in ASCII format, which were obtained by running the “FDS2ASCII” program six times (+x, -x, +y, -y, +z, -z directions). The header information and texts inside these data files are removed by the MATLAB parsing script.

The input data (x, y, z, AST for each FDS grid point on the member boundary) from FDS after running the MATLAB parsing script are in the following format:

X	Y	Z	AST
3.0000	0.5125	2.9375	735.4800
3.0000	0.5375	2.9375	735.2900
3.0000	0.5625	2.9375	734.9300
3.0000	0.5875	2.9375	734.7800
3.0000	0.6125	2.9375	734.4700
3.0000	0.6375	2.9375	734.2300
3.0000	0.6625	2.9375	734.0100
3.0000	0.6875	2.9375	733.4500

The MATLAB data mapping script maps these data to the FE transfer model nodes on the FEA model boundary. The format of the output file obtained after running the MATLAB data mapping script is:

Node #	x-coordinate	y-coordinate	z-coordinate	“Mapped AST”
--------	--------------	--------------	--------------	--------------

These data can be easily read by any FEA software.

3. Thermal Model to Structural Model Data Transfer

The transfer of computed thermal results from a heat transfer analysis modeled with solid or shell elements to a structural analysis model comprising beam and shell elements is described here. Transfer of thermal results can be easily accomplished in traditional FEA (finite element analysis) software when compatible elements of the same FE mesh are used for both heat transfer and structural analysis calculations (e.g., solid to solid or shell to shell etc.). Such an approach is practical when the structural model is simple and when the model can be described with relatively small number of elements. However, for larger structures the use of beam and/or shell elements is often necessary to ensure that a structural model of a reasonable size is developed. Currently, there is no commercial software available to map thermal data computed in solid thermal FE model to a beam and/or shell FE structural model. Therefore, a general purpose and an efficient approach are needed to map thermal results from a solid thermal FE model to an equivalent structural FE model comprising line and/or shell elements. Accurate mapping of thermal results in structural analysis model is needed to account for the effects of thermal expansion/contraction and temperature dependent changes in material properties associated with fire exposure.

3.1 Theory

In the solid thermal FE model, the transient temperatures are computed at each nodal point. For the structural analysis models these temperatures are transferred as thermal loads. They are transferred as nodal quantities and are applied as body loads. Depending on the FEA software and types of elements used, the thermal loads can be entered differently. For example, for most shell elements in the ANSYS structural model, temperatures are input as thermal loads at the corners of the outside faces of the element and at corners of interfaces between layers. In typical ANSYS beam element model, temperatures are input at each node of a beam element as a mean temperature and two temperature gradients in the element section X and Y directions, assuming the Z direction being along the beam direction. Note that the temperature variation in the beam axial direction is addressed by proper discretization. The following sections describe data mapping strategies for beam and shell elements.

3.1.1 Beam or Line Elements

In order to compute the mean temperatures and the temperature gradients for the structural nodes in the beam element as required in the ANSYS beam element model, a number of transfer points need to be defined on the beam section. Temperature results for these transfer points are

determined by interrogating the thermal results obtained using the solid thermal analysis model. Using temperatures of the transfer points, the mean temperatures and temperature gradients at each structural model node can then be computed easily (see Fig. 15).

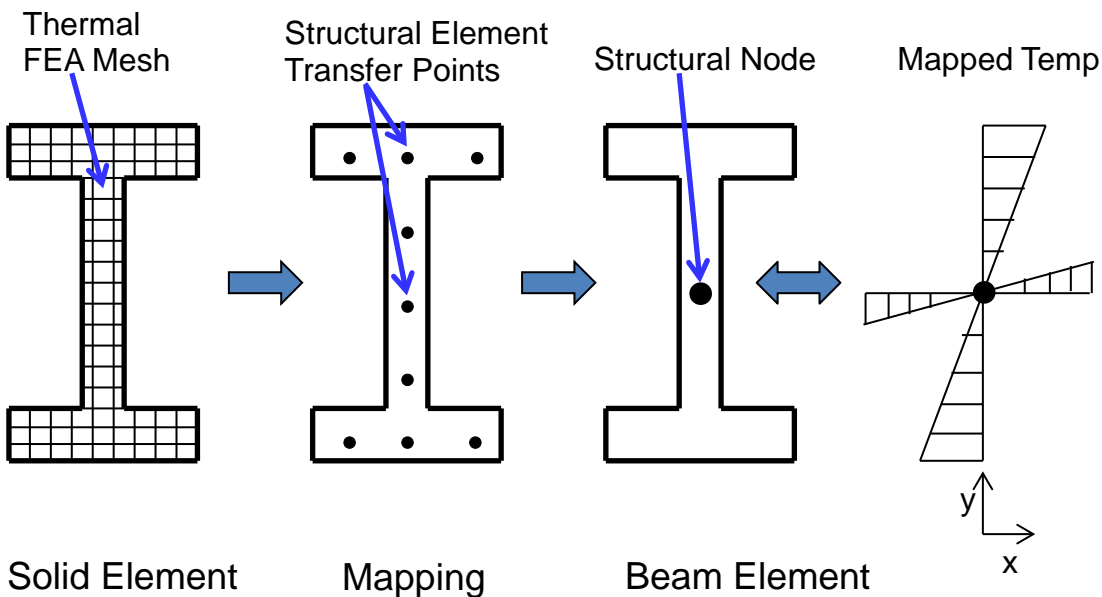


Fig. 15. A schematic of temperature transfer from thermal to beam model in ANSYS.

For ABAQUS, temperatures input at specific transfer points along beam sections are mapped by the software to appropriate integration or Gauss points. (Fig. 16) [3].

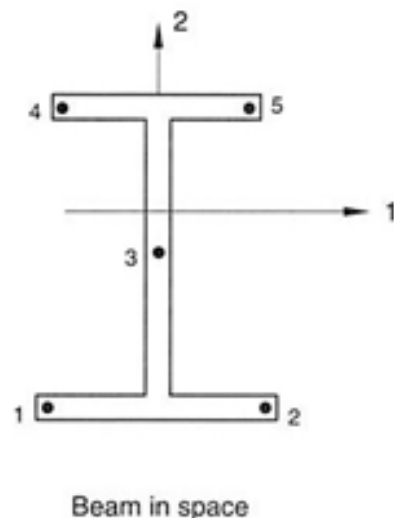


Fig. 16. A schematic of locations where temperatures are specified in ABAQUS beam element.

3.1.2 Shell Elements

For the structural shell element model (typically used for concrete slab or steel beam), the coordinates of the structural nodes are used as transfer point coordinates and an average temperature is obtained by querying the thermal solid element model results. Shell elements can have layers with varying material properties and number of integration points through the thickness of the layers. For layered shell elements, temperature transfer points corresponding to each structural node will have varying height for each layer but have same coordinates in the other two directions as the nodes themselves. The height for each layer should correspond to the type of offset used (e.g., mid-plane etc. as shown in Fig. 17). Fig. 17 shows a structural layered shell model for a concrete slab comprising 3 layers with mid-plane offset. In this case, each structural element is a 4-node quadrilateral element with three layers. For each element, the temperatures are computed by (a) specifying coordinates for 4 nodes in a given layer (with proper layer height) and (b) then using these coordinates as transfer point coordinates and interrogating the thermal solid element results.

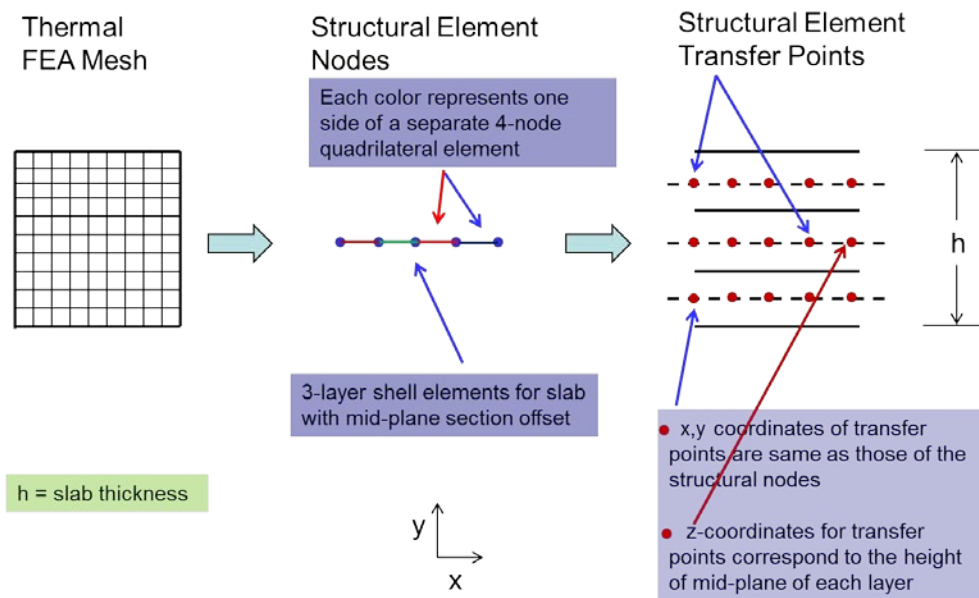


Fig. 17. A view of a solid element mesh of a thermal model and corresponding structural 3-layer shell element model (mid-plane offset) with locations of transfer points.

3.2 Temperature Mapping Methodology:

Temperature at each transfer point corresponding to a node of a structural element in the structural model is obtained by (a) determining the exact solid element to which the transfer point belongs and (b) computing an average temperature of the transfer point by using that solid element shape function and transfer point coordinates. The first goal is to determine the solid finite element that contains the transfer point. Assuming an 8-node brick solid element, the following three functions (e.g., f , g , and h) can be written:

$$\begin{aligned} f(x) &= x_{pt} - \sum_{i=1}^8 N_i x_i \\ g(y) &= y_{pt} - \sum_{i=1}^8 N_i y_i \\ h(z) &= z_{pt} - \sum_{i=1}^8 N_i z_i \end{aligned} \quad (1)$$

Where, N_i is the element shape function in local or natural coordinates and x_i , y_i , and z_i are the global coordinates of nodes of an 8-noded brick element and x_{pt} , y_{pt} , and z_{pt} are the coordinates of the transfer point. Note that a local coordinate system is one which is located on or within the boundary of an element. Shape functions are listed for common element types in any standard finite element text book. For eight-noded linear brick elements, the shape functions are given as shown in equation (2) using natural coordinates ξ , η , and ζ (Fig. 18) for the i th node:

$$N_i = \frac{1}{8} (1 + \xi \xi_i) (1 + \eta \eta_i) (1 + \zeta \zeta_i), \quad \text{for } i = 1, 2, \dots, 8 \quad (2)$$

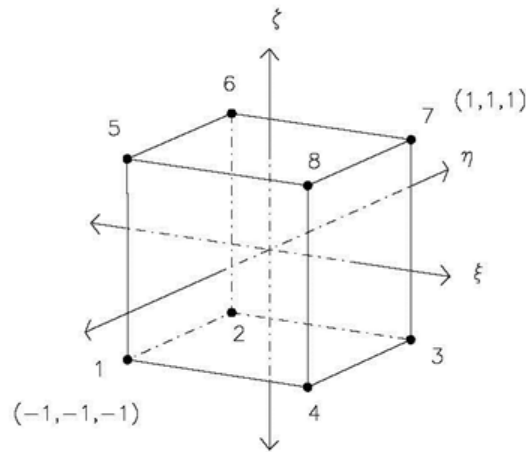


Fig. 18. A sketch of a solid brick element showing natural coordinates (ξ , η , and ζ).

Note that ξ_i, η_i , and ζ_i values of the eight corner nodes are either -1 or +1 in the above equation. Any arbitrary point within this element will have ξ_i, η_i , and ζ_i values between -1 and +1 range. Since the shape function, N , is a function of ξ , η , and ζ , a first order Taylor expansion can be written as follows:

$$\begin{aligned}
 f(x + \Delta x) &= f(x) + \frac{\partial f}{\partial \xi} \Delta \xi + \frac{\partial f}{\partial \eta} \Delta \eta + \frac{\partial f}{\partial \zeta} \Delta \zeta \\
 g(y + \Delta y) &= g(y) + \frac{\partial g}{\partial \xi} \Delta \xi + \frac{\partial g}{\partial \eta} \Delta \eta + \frac{\partial g}{\partial \zeta} \Delta \zeta \\
 h(z + \Delta z) &= h(z) + \frac{\partial h}{\partial \xi} \Delta \xi + \frac{\partial h}{\partial \eta} \Delta \eta + \frac{\partial h}{\partial \zeta} \Delta \zeta
 \end{aligned} \tag{3}$$

The objective of finding the transfer point within an element can be obtained by:

$$f(x + \Delta x) = g(y + \Delta y) = h(z + \Delta z) = 0 \tag{4}$$

Zeroing out the left side of each Taylor expansion equations in equation (3) yields,

$$\begin{aligned}
 \frac{\partial f}{\partial \xi} \Delta \xi + \frac{\partial f}{\partial \eta} \Delta \eta + \frac{\partial f}{\partial \zeta} \Delta \zeta &= -f(x) \\
 \frac{\partial g}{\partial \xi} \Delta \xi + \frac{\partial g}{\partial \eta} \Delta \eta + \frac{\partial g}{\partial \zeta} \Delta \zeta &= -g(y) \\
 \frac{\partial h}{\partial \xi} \Delta \xi + \frac{\partial h}{\partial \eta} \Delta \eta + \frac{\partial h}{\partial \zeta} \Delta \zeta &= -h(z)
 \end{aligned} \tag{5}$$

Equation (5) can be written in matrix form as shown below:

$$\begin{pmatrix} \frac{\partial f}{\partial \xi} & \frac{\partial f}{\partial \eta} & \frac{\partial f}{\partial \zeta} \\ \frac{\partial g}{\partial \xi} & \frac{\partial g}{\partial \eta} & \frac{\partial g}{\partial \zeta} \\ \frac{\partial h}{\partial \xi} & \frac{\partial h}{\partial \eta} & \frac{\partial h}{\partial \zeta} \end{pmatrix} \begin{pmatrix} \Delta \xi \\ \Delta \eta \\ \Delta \zeta \end{pmatrix} = \begin{pmatrix} -f(x) \\ -g(y) \\ -h(z) \end{pmatrix} \quad (6)$$

The left hand side matrix in equation (6) containing the derivatives can be easily obtained. The system of equations are then solved using Cramer's rule for the $\Delta \xi$, $\Delta \eta$, and $\Delta \zeta$ values.

Updated ξ , η , and ζ values are checked to see if they fall within -1 to +1 range. If this becomes true, then the point is contained within the particular element. Equation (6) can be solved iteratively to ensure that the final values of $\Delta \xi$, $\Delta \eta$, and $\Delta \zeta$ do not change further between iterations. For each point, the looping can be done over a thermal solid element set that is in the vicinity of the transfer point. This set of thermal solid elements (that are located near the transfer point) can be determined by knowing the coordinates of the centroids of all the solid elements.

When it is found that the point is contained within a given element, the temperature at the transfer point is then determined by the following equation:

$$T_j = \sum_{i=1}^8 N_i T_i \quad (7)$$

where, T_j denotes computed temperatures at each transfer point and i varies from 1 to 8 for the hexahedral solid element that contains the transfer point. Note that the values of N_i will change based on values of the natural coordinates of the transfer point. The sequence of steps that is followed to obtain the temperature of a transfer point is given in the following flowchart (Fig. 19):

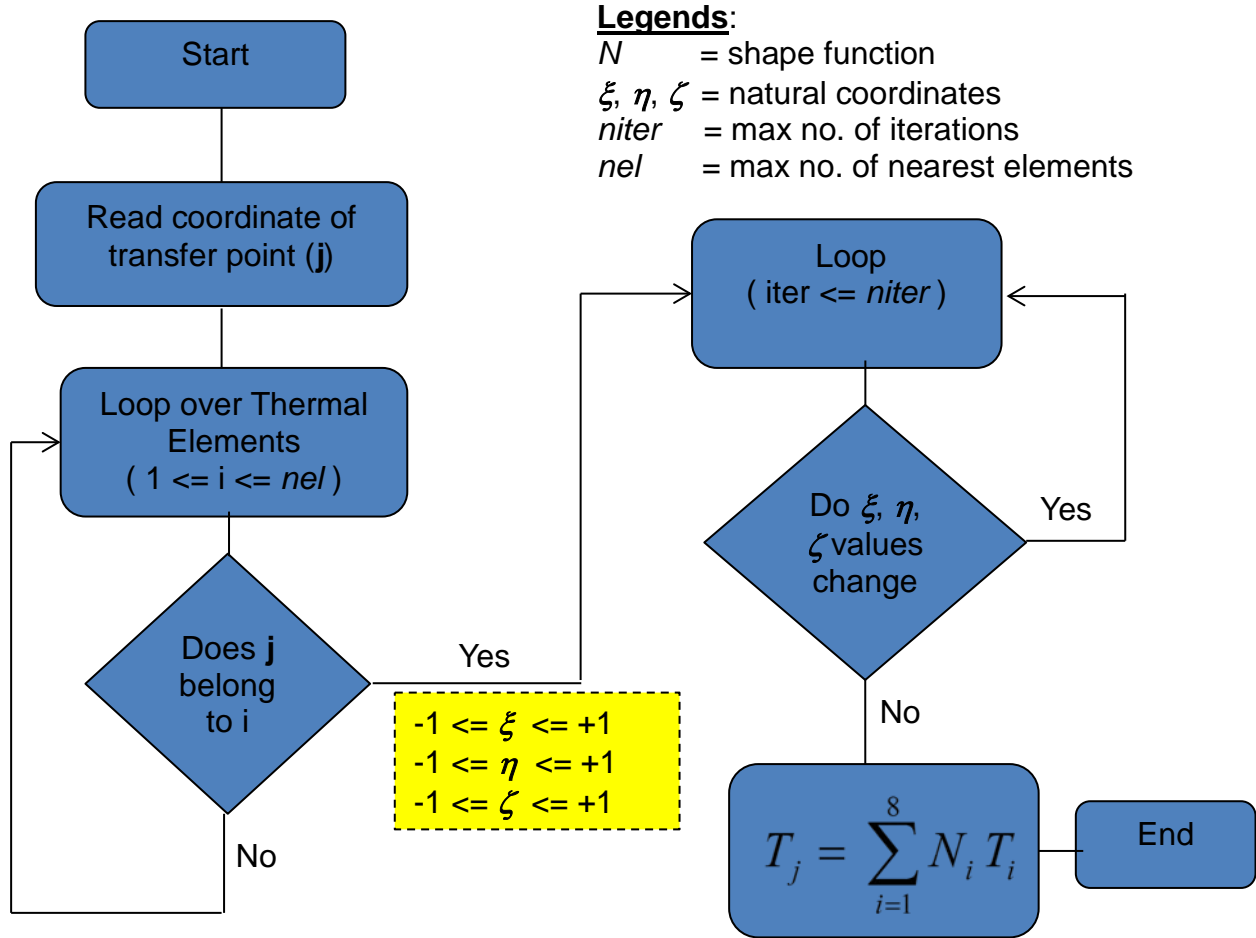


Fig. 19. A flowchart showing the sequence of steps used to compute temperatures at a transfer point.

Note that the loop over elements shown in Fig. 19 can be chosen such that only those elements, whose centroids lie within a user chosen distance away from the coordinate of the transfer point, can be searched. This way the computation time will be manageable especially for a large model.

3.3 Data Mapping Script:

The data mapping script has been developed in the MATLAB software. The data flow and framework for this script is shown in Fig.20.

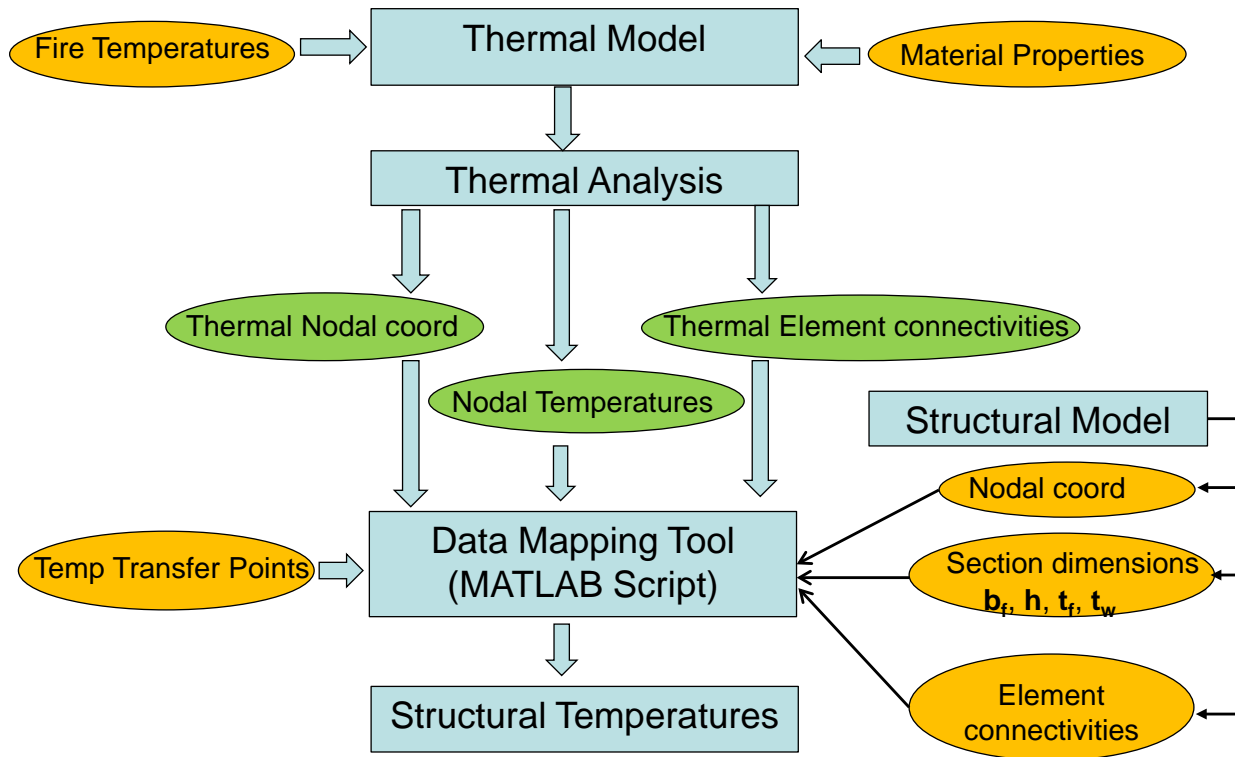


Fig. 20. A flow chart of the Data Mapping Script and its framework.

As Fig. 20 shows, a thermal analysis is first conducted using solid brick elements with any FE software. The inputs to the thermal model are fire boundary conditions at the exposed surface(s), appropriate initial conditions, and thermal properties of the materials included in the model. A successful thermal analysis will generate time dependent nodal temperatures at all nodes in the model. The user needs to write nodal temperatures at specific time intervals (e.g., half hour, 1 hour etc.) into a comma separated variable (.csv) file. Additionally, two other files need to be written in .csv format: a) nodal coordinates of all nodes (e.g., node number, x, y, z) and element connectivities (e.g., element number, node 1, node 2,....., node 8) in the solid element model. The Data Mapping Script reads these three files and also reads three additional files: a) list of transfer points along with their coordinates, b) nodal coordinates in the structural model, and c) structural model element connectivities. Note that the transfer point coordinates can be computed automatically using beam section dimension or a user can input these transfer point coordinates using a “*.csv” file. The script then computes temperatures at each of the transfer points. These nodal temperatures are used to define element body loads. Currently, the script outputs temperatures in ANSYS format for all elements in the structural model. Work is ongoing to develop the capability where the user will be given a choice to choose the software format for output of the mapped temperatures.

3.4 Verification

The verification cases focus on the accuracy of structural temperatures obtained for the following cases: a) a protected member exposed to a fire, b) an unprotected member exposed to a fire, c) a structural assembly exposed to a fire using condition similar to the one in the Cardington Test 1.

3.4.1 Fireproofed Steel Beam Temperatures:

In the first verification example, a single W24X55 Beam with 0.5 inch uniform fireproofing was chosen (Figure 9). The beam was discretized with solid 8-node hexahedral elements. The beam was exposed to an oil-fired fire. Time varying boundary conditions were artificially altered in order to develop non-uniform, varying time-temperature profile for beam web, top flange, and bottom flange as shown in Figure 21. This was done by modifying the fire curve (e.g., Fire Curve (A), Fire Curve (B), and Fire Curve (C)) and using these curves at different locations (Fig. 21). Time-temperature values for each of these curves are listed in Appendix 1. Note the values of temperatures were arbitrarily chosen in order to generate some variation of temperature along beam web and in axial directions.

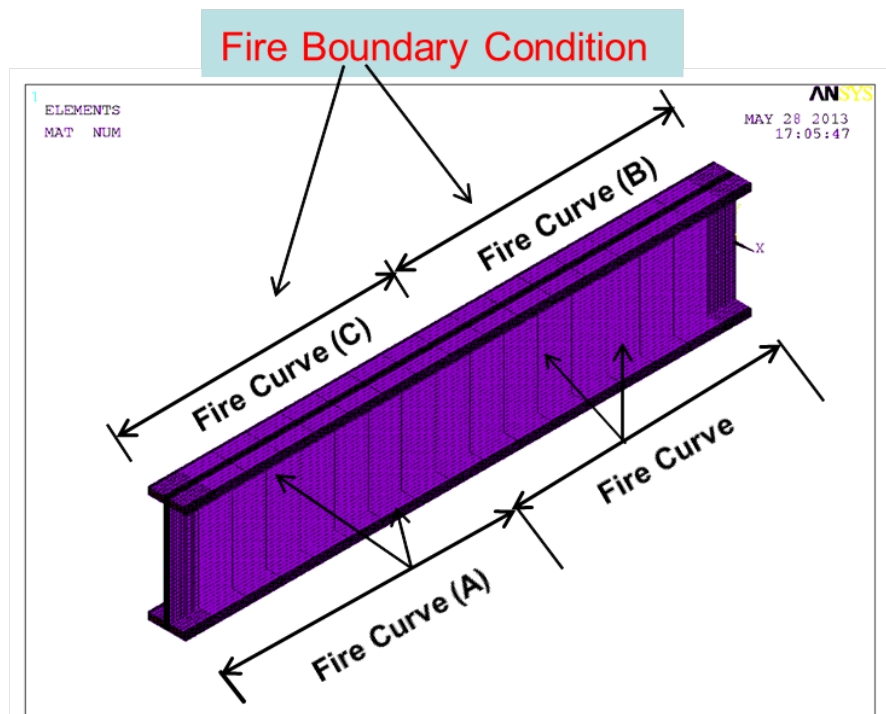


Fig. 21. Artificial non-uniform fire conditions applied to the exposed surfaces of the solid thermal model. See Appendix 1 for the time-temperature data.

Next, a structural FEA model was developed using beam elements in ANSYS for the steel beam only. The coordinates and dimensions of the beam in the structural and thermal model were the same. Note that the number of transfer points per node is chosen by the user. A large number of transfer points will increase the computational time, while too few transfer points will result in possibly inaccurate temperatures for the structural node. In this data mapping exercise, a set of fifteen temperature transfer points was chosen for each node in the structural analysis model (Fig. 22).

The data mapping script reads the coordinates of these transfer points after reading (a) thermal solid model nodal coordinates, (b) thermal solid model element connectivities, (c) thermal solid model nodal temperatures, (d) structural model nodal coordinates, and (e) structural model element connectivities. Next, the data mapping script probes the solid element analysis results for each transfer point to determine the equivalent temperature at these transfer points.

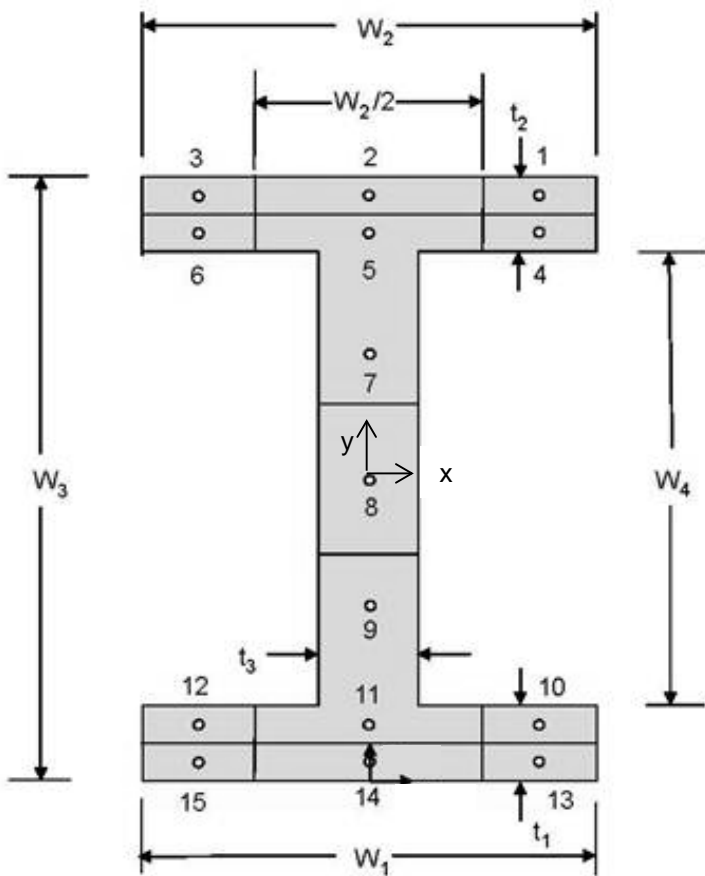


Fig. 22. Temperature transfer points along the beam section at each structural node.

A simple example is given for computing nodal mean temperatures and two temperature gradients for each structural node (as required by ANSYS) following ref [4] in the following section (T_x and T_y are the temperature gradients in the x and y directions):

$$\text{Mean temperature} = \frac{\sum w_i T_i}{A} \quad (8a)$$

$$T_x = \frac{2t_2(T_1 - T_3 + T_4 - T_6)}{3(W_1 t_1 + W_2 t_2)} - \frac{2t_1(T_{10} - T_{12} + T_{13} - T_{15})}{3(W_1 t_1 + W_2 t_2)} \quad (8b)$$

$$T_y = \frac{T_1 + T_3 + T_4 + T_6 + 2(T_2 + T_5)}{4(W_3 + W_4)} - \frac{T_{10} + T_{12} + T_{13} + T_{15} + 2(T_{11} + T_{14})}{4(W_3 + W_4)} \quad (8c)$$

where, A is the total cross section area, w_i is the area weight at each transfer point, T_i is the temperature at each transfer point as computed by the data mapping script. The beam node is located at mid web in the origin (point 8 in Fig. 22). Areas around each point have area weightages as shown in Fig. 22. For example, the area weightage at each web node is 1/3 of the total web area. Using these area weightages, equation 8 has been empirically formulated [4].

As a first case for verification of this data mapping script, the following exercise is conducted. For the fireproofed beam shown in Fig. 9, three locations were arbitrarily chosen along beam length (Fig. 23). At each location, three node points were chosen in the solid model (one each at top flange, mid-web, and bottom flange). Temperatures obtained at each of these nine points from solid thermal model are compared with interpolated results from MATLAB script using exact coordinates of these points. Results are compared for two different time steps (Table 1).

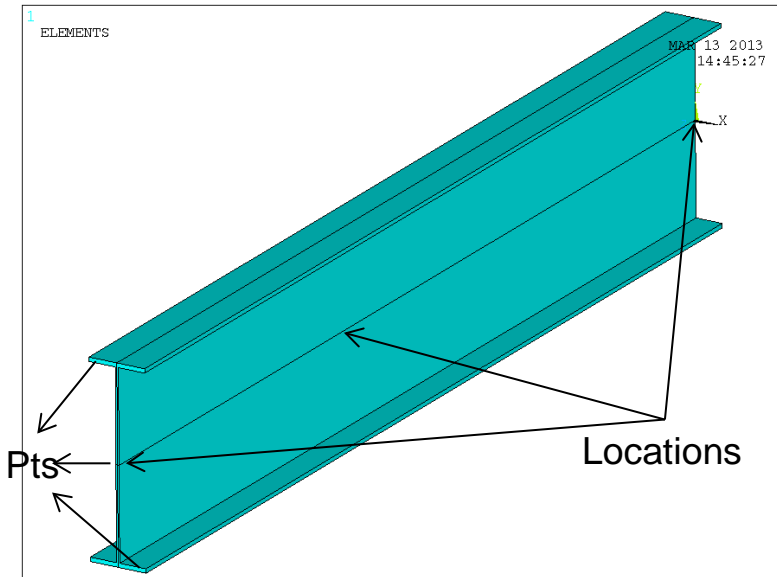


Fig. 23. Locations where ANSYS solid model outputs are compared with mapped results.

Table 1: Comparisons of temperatures (in degree Celsius) at specified locations obtained with ANSYS solid thermal model and with the MATLAB data mapping script.

Time = 20 min							
Location	Top Flange		Mid Web		Bottom Flange		
	ANSYS Thermal	MATLAB	ANSYS Thermal	MATLAB	ANSYS Thermal	MATLAB	
z = 0 in	48.99	48.98	79.70	79.70	74.24	74.24	
z = 80 in	42.60	42.60	69.73	69.73	65.16	65.16	
z = 120 in	42.58	42.58	69.72	69.71	65.14	65.14	
Time = 120 min							
z = 0 in	457.85	457.85	651.42	651.42	650.59	650.59	
z = 80 in	423.42	423.42	613.81	613.81	615.09	615.09	
z = 120 in	423.05	423.06	613.63	613.62	614.94	614.94	

Table 1 shows an almost exact agreement between the nine ANSYS thermal solid element model nodal temperatures and those computed with the data mapping script using coordinates of these nodes and probing the ANSYS thermal results. Fig. 24 and 25 show temperature profile in the steel beam at 1 hour and 2 hours into the fire as computed in the ANSYS thermal model and mapped in the structural model.

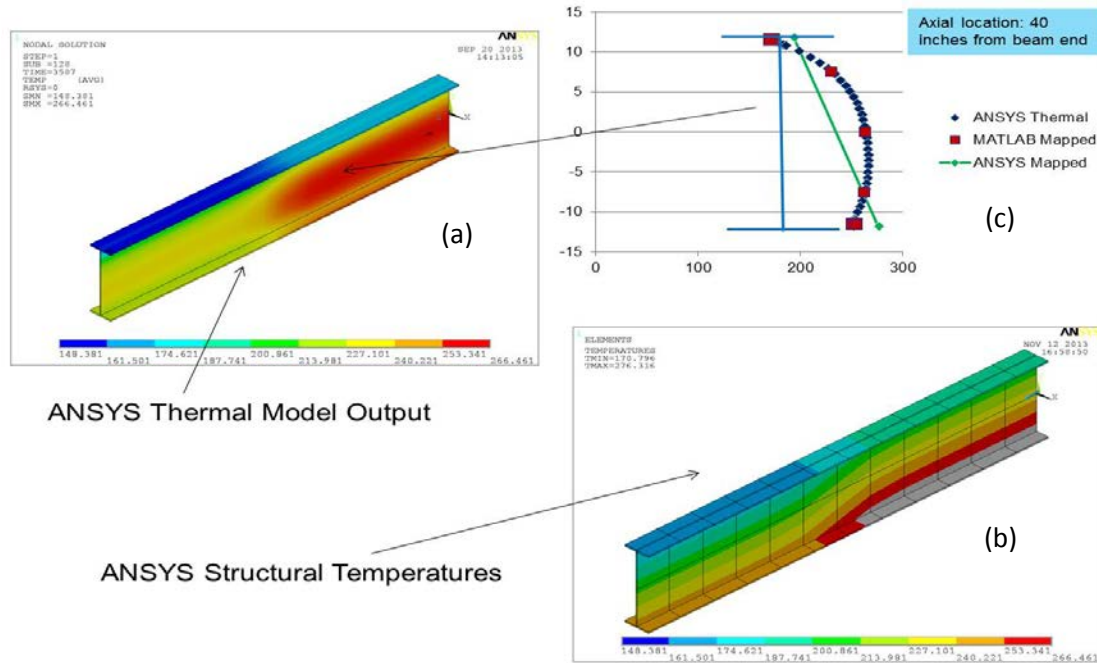
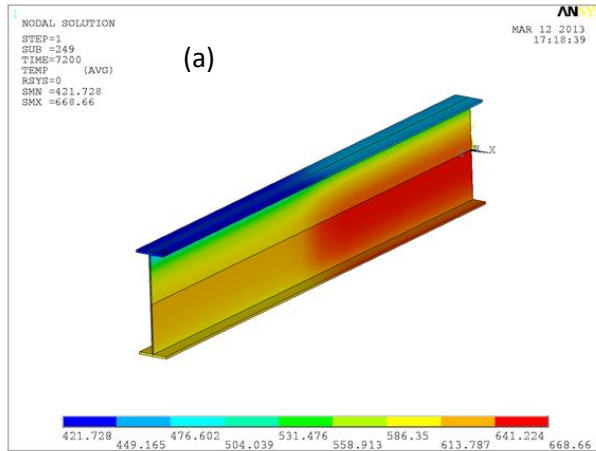
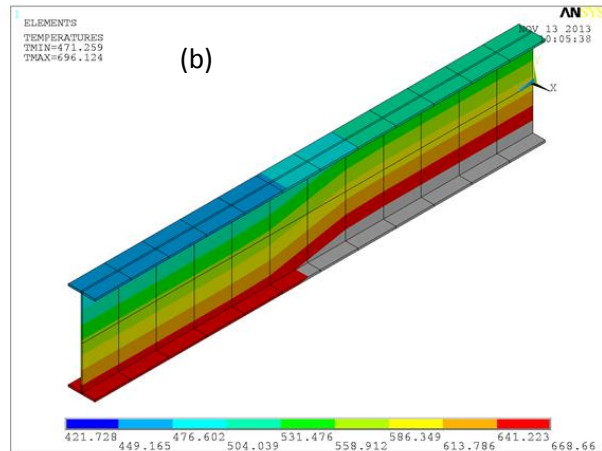


Fig. 24. Temperatures in the steel beam at 1 hour: (a) as computed in the ANSYS thermal solid model, (b) as mapped in the structural beam element model using the data mapping script, (c) comparison of temperature profiles along the web (at a location along the beam length): ANSYS thermal: temperatures from heat transfer mode: MATLAB Mapped: temperatures at transfer points along the web; ANSYS Mapped: structural temperatures obtained with equation (8).



ANSYS Thermal Model Output



ANSYS Structural Temperatures

Fig. 25. Temperatures in the steel beam at 2 hours: (a) as computed in the ANSYS thermal solid model, (b) as mapped in the structural beam element model using the data mapping script.

Note that there are some differences in temperatures in the ANSYS solid model and those mapped on the ANSYS beam element model using the data mapping script. This discrepancy between the ANSYS thermal and Structural mapped results is due to the use of linear temperature gradients at each structural beam node (e.g., mean temperature and two temperature gradients at each structural node). Note that the temperature scale for the structural temperatures in Fig. 24(b) and 25(b) was adjusted to match with those of the corresponding thermal model results (Fig. 24(a) and 25(a)). As a result, temperatures of certain regions were outside this scale and they are marked in gray color. Since ANSYS allows only three temperature inputs at each structural node point, the temperature averaging scheme used needs to be properly developed and validated with real data for each case. Note that beam elements with multiple integration points can also be used as ANSYS will map temperatures to integration points appropriately. Other software such as ABAQUS allows more locations for input of mapped temperatures (see Fig. 16). Fig. 24 (c) shows temperatures along beam height at particular axial location of the beam. It is evident that ANSYS structural temperatures show linear profiles. However, it can be noticed that transfer point temperatures obtained with MATLAB script fall exactly on the ANSYS thermal results indicating that the data mapping algorithm computed temperatures correctly.

3.4.2 Unprotected Steel Beam Temperatures:

In order to verify that the mapped temperature field is accurate, an example was created where the heat transfer in an unprotected beam was modeled with fixed temperatures at the bottom of lower flange (600°C) and the top of the upper flange (100°C). Fig. 26(a) displays the computed

temperature field at 1 hour into the simulation as computed by the ANSYS solid element model. Note that there is no heat flow in the beam axial direction. Mapped temperatures as computed using the MATLAB script are displayed in Fig. 26(b), where mean temperatures and two temperature gradients at each structural node are computed using fifteen transfer points as discussed above.

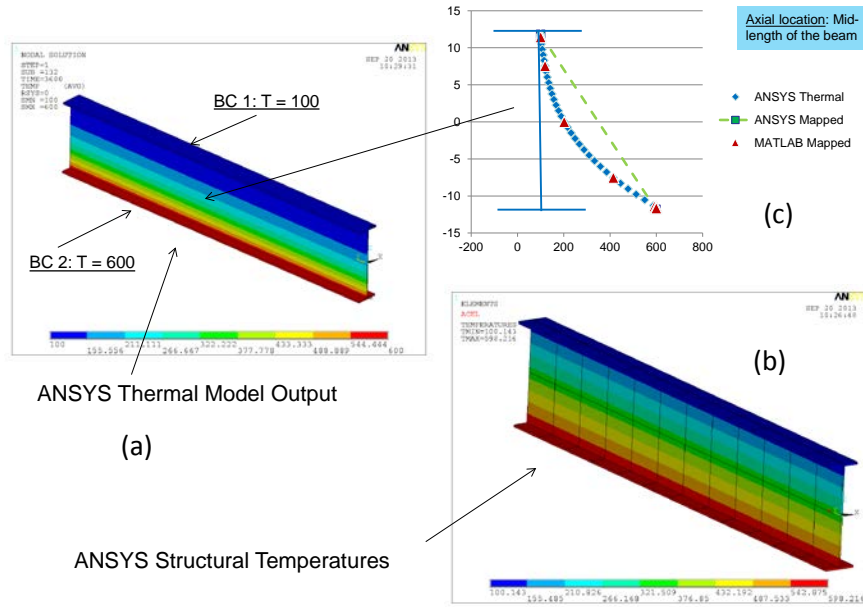


Fig. 26. Temperatures in an unprotected steel beam at 1 hour: (a) as computed in the ANSYS thermal solid model, (b) as mapped in the structural beam element model using the data mapping script, (c) comparison of temperature profiles along the web height (at a location along the beam length) among ANSYS thermal, ANSYS structural results along with computed temperatures at transfer points.

Note that there is a very good agreement between Fig. 26(a) and 26(b) in terms of the trend of the variation of temperature profile across the beam section. However, because of limitation of the ANSYS structural body load input format, the minimum and maximum values of the temperatures in the structural model do not exactly match those in the ANSYS solid thermal model. As mentioned for Fig. 24, Fig. 26 (c) shows temperatures along beam height at particular axial location of the beam. Again, transfer point temperatures along the web height obtained with the MATLAB script lie exactly on the ANSYS thermal results verifying that the data mapping algorithm computed temperatures correctly.

3.4.3 Unprotected Steel Beam Deflections:

Another verification case that is of interest to structural engineers includes an analysis of the mechanical behavior of a beam as a result of exposure to elevated temperatures. In this section, a static, linear structural analysis simulation is conducted using thermal loading obtained for an unprotected beam (as discussed in Section 3.4.2). Two simulations were run: (1) ANSYS structural simulation using computed temperatures shown in Fig. 26(a) with solid hexahedral elements and (2) ANSYS structural simulation using computed temperatures at 1 hour as shown in Fig. 26(b) with beam elements. No mechanical loading is considered. The beams are pinned at either end. ASTM A572 Grade 50 steel properties are used for structural analysis. Identical properties were used for both of these simulations. Fig. 27(a) and 27(b) show vertical deflections computed in both of these models.

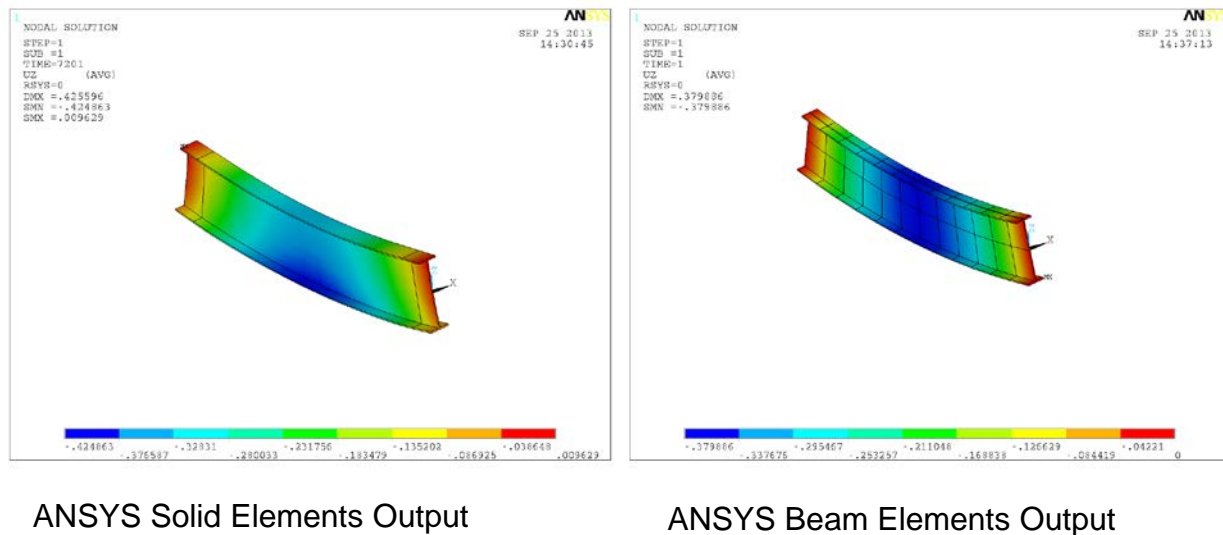


Fig. 27. Vertical displacements in inches at 1 hour in an unprotected steel beam: (a) as computed in the ANSYS structural solid element model using temperatures in Fig. 26(a), (b) as computed with ANSYS structural beam element model with mapped temperatures shown in Fig. 26(b).

The maximum deflection of 0.425 inches obtained with ANSYS solid elements model is reasonably close to the maximum deflection of 0.387 inches obtained with the ANSYS beam element model. In addition, the displacement profile in the beam for both of these cases agrees reasonably well.

3.4.4 Cardington Test Results:

Another verification case that was chosen is the data mapping for the Cardington Test 1 [5]. The Cardington test was conducted on a composite floor assembly from the seventh floor of an 8-

story frame. A single UB305x165x40 restrained beam and the surrounding concrete floor spanning about 29.5 ft. (9 m) between a pair of columns were heated using a gas-fired furnace. Lightweight concrete was used for the slab. The gas-fired furnace was constructed on the sixth floor to heat the specified structure between two columns. The dimensions of the furnace used are: 26.2 ft. (8 m) in length, 9.8 ft. (3 m) in width, and 6.6 ft. (2 m) in height. The columns and end connections were not subjected to heating. The steel was unprotected. During the fire test, the beam was heated over the middle 26.2 ft. (8m) of its 29.5 ft. (9m) length at a rate ranging from 3 °C/minute to 10 °C/minute until a temperature in the range of 800 °C to 900 °C was reached through the profile of the beam section. A structural analysis model was developed using dimensions used in the Cardington test (see Fig. 28). The beam was modeled using 1-D line elements and the slab was developed using 3-layer, 4-node quadrilateral elements (Fig. 28 (a)). Using same dimensions, an equivalent thermal model was developed using solid elements (shear studs were not included) (Fig. 28 (b)). An enclosure was built around the composite floor assembly in the thermal model. The exterior surface nodes of the enclosure were heated as a function of time using the furnace heating data available from the Cardington test. The exposed surface of the composite floor assembly and interior surface of the enclosure were assigned emissivity data using known values in the literature. ANSYS radiosity method [2] was used to generate transient heating in the composite floor assembly for a 2 hour fire. After simulation, nodal temperatures were written at six different times (20, 40, 60, 80, 100, and 120) min to a file in ASCII format. A sketch of the elevation view of the composite floor model is shown in Fig. 29.

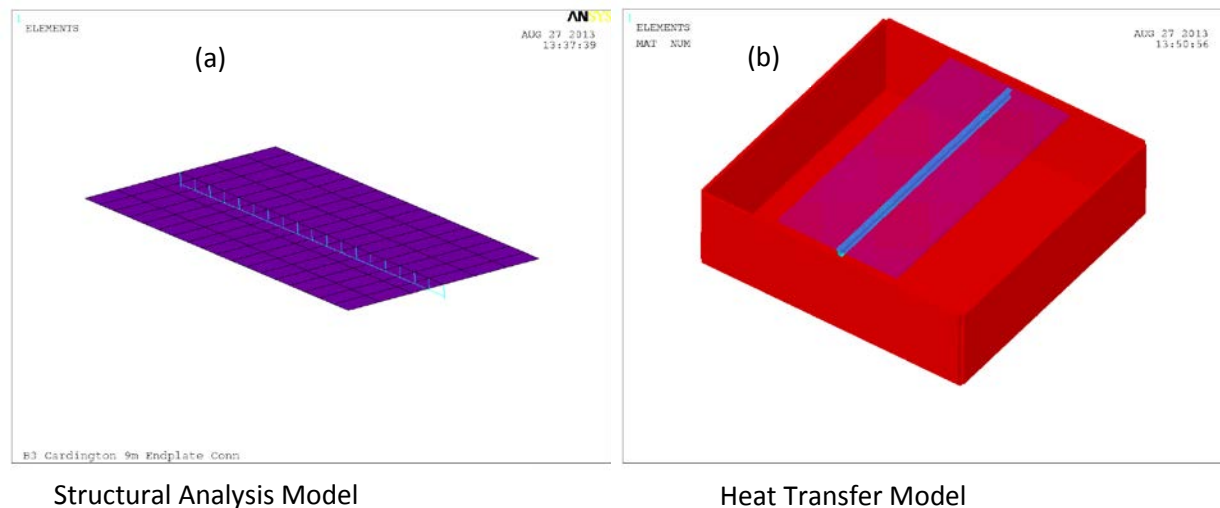


Fig. 28 A FEA model for structural analysis and an equivalent FEA heat transfer model for the composite floor system in Cardington test 1.

As discussed earlier, fifteen transfer points corresponding to each node in a structural beam element were developed (Fig.22). The MATLAB script was queried using these transfer point

coordinates against six sets of thermal results (at different times into the fire) to obtain temperatures at these transfer points for the beam element nodes. For the concrete slab, nodal coordinates of each element (adjusting for the layer height) were used to query the six sets of thermal results using the MATLAB script to obtain the corresponding temperatures. Output structural temperatures in the beam and the slab in ANSYS format at 1 hour and 2 hour into the simulation are shown below (Fig. 30 and 31). Also shown are temperatures from the heat transfer simulation using solid elements. A good agreement was noticed between ANSYS thermal model temperatures and mapped structural temperatures. Note that in this example, the layer height for the transfer points in the slab was chosen such that the first layer was very close to exposed surface height in the thermal model. This allows for ease in comparison of the mapped temperatures of the exposed slab surface in the structural model with the temperatures in the heat transfer model. In reality, the transfer points for the shell nodes for the first layer should be at the height corresponding to that of the mid-plane of the first layer above (see Fig. 17). This is why there is a small discrepancy in temperatures of the slab near the unexposed surface (see Fig. 32).

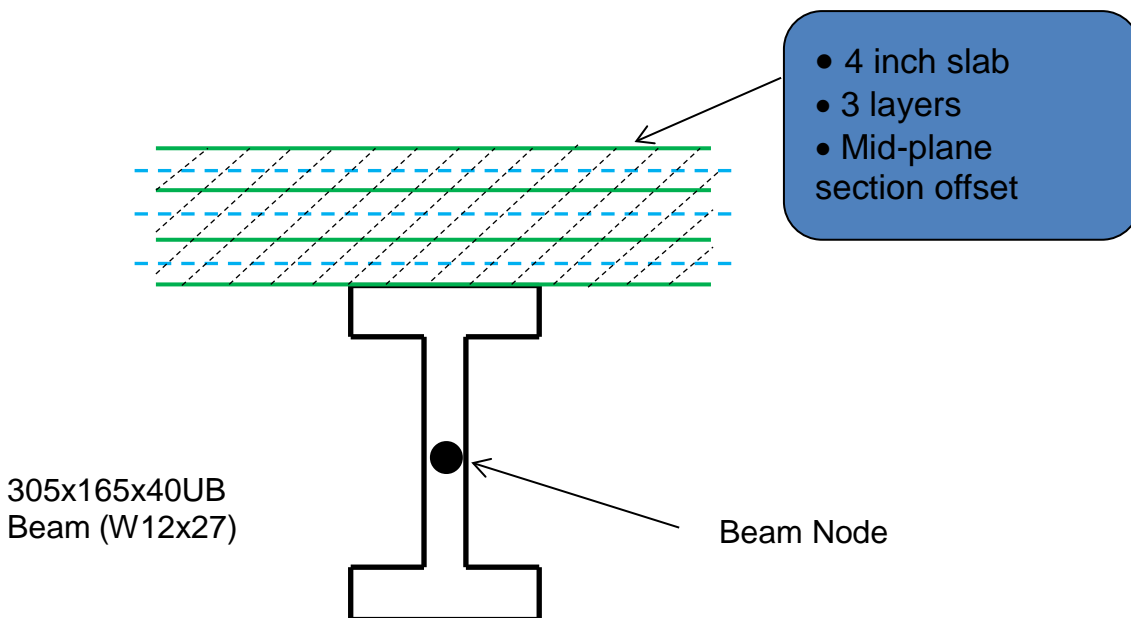
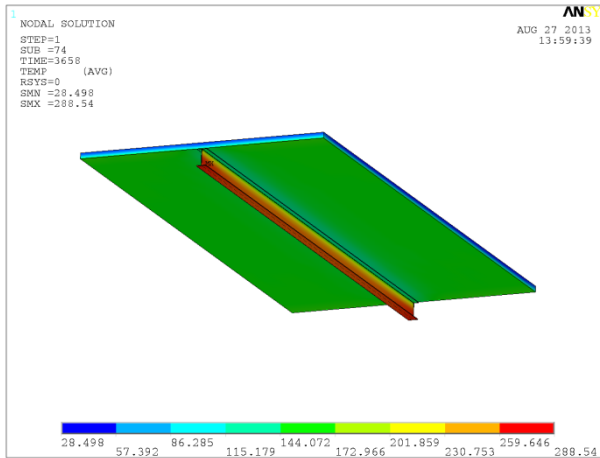


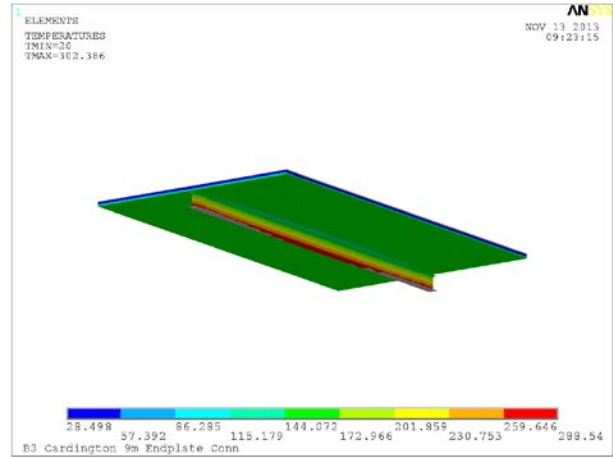
Fig. 29. A sketch of the elevation view of the composite floor model in Cardington Test 1.

ANSYS Temperatures (solid elements)

Mapped Structural Temperatures



(a)

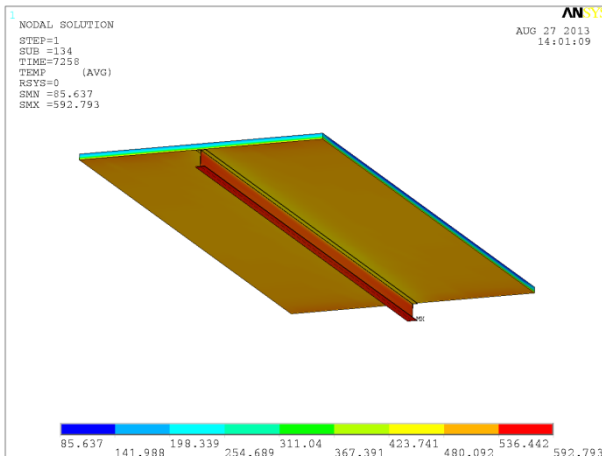


(b)

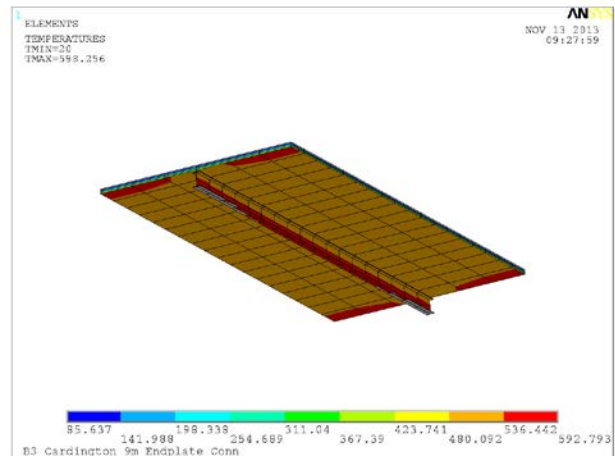
Fig. 30. A plot temperatures in the composite floor model in Cardington test 1 at 60 minutes into the fire: (a) heat transfer FEA results and (b) Mapped temperatures in ANSYS structural model.

ANSYS Temperatures (solid elements)

Mapped Structural Temperatures



(a)



(b)

Fig. 31. A plot temperatures in the composite floor model in Cardington test 1 at 120 minutes into the fire: (a) heat transfer FEA results and (b) Mapped temperatures in ANSYS structural model.

ANSYS Temperatures (solid elements)

Mapped Structural Temperatures

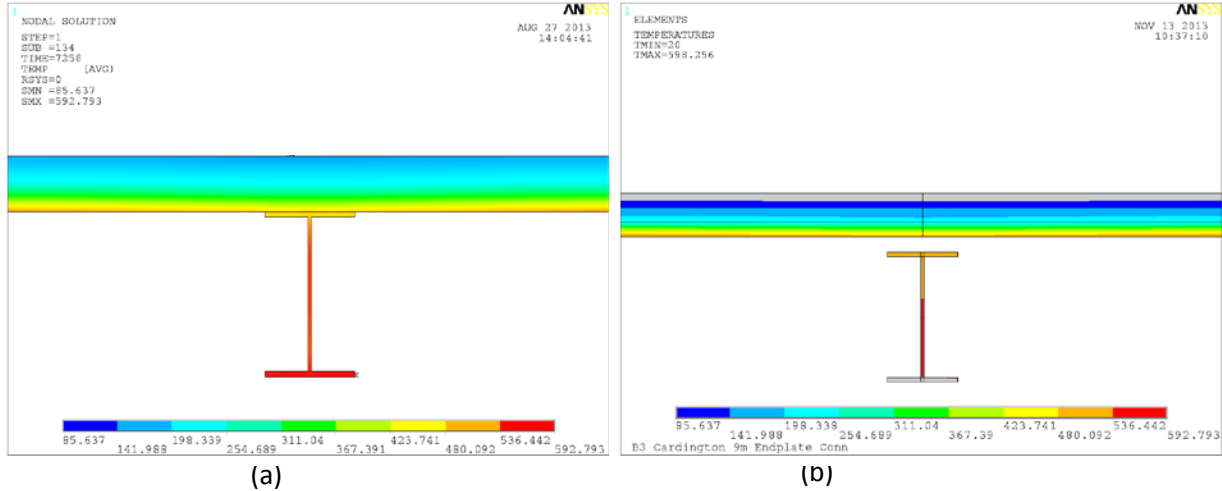


Fig. 32. A plot temperatures in the composite floor model in Cardington test 1 at 120 minutes into the fire: (a) heat transfer FEA results and (b) Mapped temperatures in ANSYS structural model.

Note that in the structural model shear studs connect top of the beam with the slab. The grooved metal deck is not modeled in the structural model. Shear studs are not shown in Fig. 32 and they are not included in the heat transfer model.

3.4.4.1 Cardington Test Slab Temperature Verification:

In this section, slab temperatures as mapped by the MATLAB script are compared with nodal temperatures of nearest nodes in the heat transfer model. Fig. 33 shows a plot of shell elements for the concrete slab in Cardington Test 1. Each slab element has 4 nodes and the elements have three layers. MATLAB script mapped temperatures of nodes of two different elements at 2 hour into the fire are compared with those obtained with the ANSYS heat transfer model (see two inset plots where nodal temperatures at 3 layers are compared in Fig. 33). A good agreement was obtained.

Mapped temperatures of transfer points (at 2 hour into the fire) at three layers for each node belonging to these two elements are listed in the spreadsheet shown in Fig. 34(a). Also shown are the temperatures of the nearest node in the ANSYS heat transfer model. Temperature variation along the slab height for 2 transfer points obtained with the MATLAB script are plotted along with those of two nearest nodes from the ANSYS heat transfer model in Fig. 34(b). Again, very good agreement was obtained.

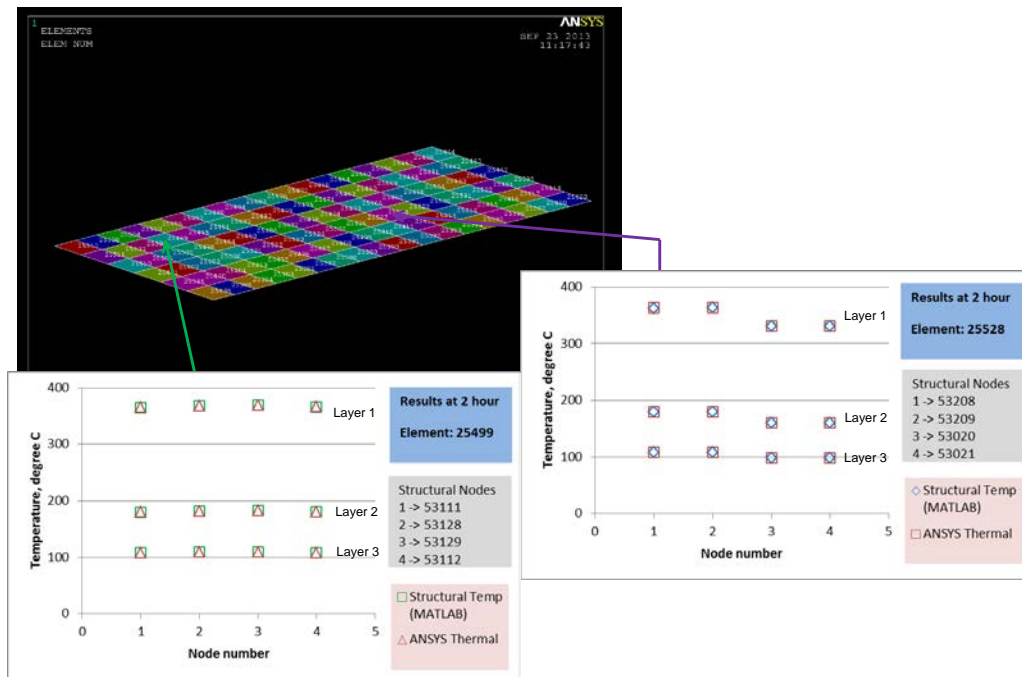
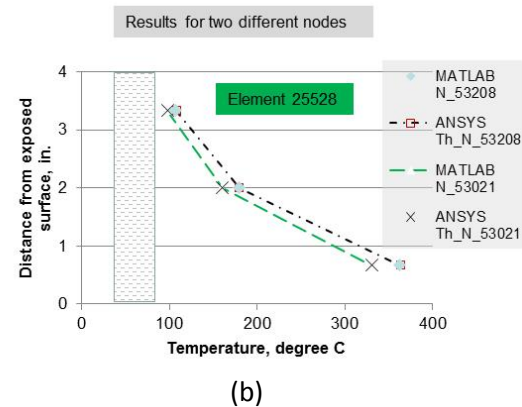


Fig. 33. A color plot of shell elements for the concrete slab in Cardington Test 1. MATLAB script mapped temperatures of nodes of two elements at 2 hour into the fire are compared with those obtained with the ANSYS heat transfer model (see two inset plots where nodal temperatures at 3 layers are compared).

Results at 2 hour for the slab in Cardington Test 1					
Structural Element	Structural Node	Layer Number	MATLAB Temp	Nearest Thermal node	ANSYS Thermal Temp
25528	53208	layer1	362.50	6603	363.05
		layer2	178.94	6605	179.31
		layer3	107.50	6607	107.72
	53209	layer1	362.87	6608	363.39
		layer2	179.06	6610	179.44
		layer3	107.54	6612	107.76
	53020	layer1	330.82	3903	330.75
		layer2	160.26	3905	160.23
		layer3	97.68	3907	97.669
	53021	layer1	330.57	3898	330.52
		layer2	160.16	3900	160.14
		layer3	97.66	3902	97.651
25499	53111	layer1	364.58	14060	365.63
		layer2	179.82	14062	180.41
		layer3	107.87	14064	108.19
	53128	layer1	368.32	14535	368.55
		layer2	182.11	14537	182.22
		layer3	109.17	14539	109.23
	53129	layer1	369.08	14530	369.32
		layer2	182.41	14532	182.52
		layer3	109.31	14534	109.36
	53112	layer1	365.56	14055	366.64
		layer2	180.20	14057	180.79
		layer3	108.03	14059	108.35



(a)

Fig. 34. (a) A spreadsheet showing MATLAB script mapped transfer point temperatures (at 2 hour into the fire) at 3 layers for two different shell elements for the slab in the Cardington Test 1 along with temperatures of the nearest node in the ANSYS heat transfer model and (b) Temperature variation along the slab height for 2 transfer points obtained with the MATLAB script is shown along with those of two nearest nodes from the ANSYS heat transfer model.

3.4.5 Complete Fire to Thermal to Structural Data Transfer

As a case of a complete fire to thermal to structural data transfer example, a protected steel girder was subjected to a fire in a compartment. The fireproofed girder was specified as an obstruction in the FDS input file. A heat transfer model of the girder was created in ANSYS using 8-node solid hexahedral elements. Finite element structural analysis model of the girder was created in ANSYS using (a) solid elements and (b) beam elements. The exposed surface temperature at about 24 seconds into the fire is shown in Fig. 35(a). These temperatures were mapped to the exposed surface nodes of the FE heat transfer model using the MATLAB fire/thermal data mapping script. Fig. 35(b) shows a plot of these temperatures. A visual comparison of Fig. 35(a) and 35(b) shows a reasonably good agreement.

Subsequently, a FE heat transfer analysis was conducted in ANSYS for 2 hours. The MATLAB thermal/structural data mapping script was used to map these temperatures (at 1 hour and 2 hour into the thermal simulation) to an equivalent structural model of the steel girder composed of 1-D beam or line elements. Fig. 36 (a) shows temperatures in the beam at 1 hour as computed by the ANSYS heat transfer model while Fig. 36(b) displays the temperatures as mapped in the structural beam element model using the MATLAB data mapping script. Note that the agreement between Fig. 36(a) and 36(b) is reasonable, although the maximum temperature in Fig. 36(b) is different from those in Fig. 36(a) due to the linear temperature gradient approach available with the ANSYS beam elements. Note that the temperature scale of structural model results and thermal model results was not artificially made equal. Similar agreement was obtained with temperatures at 2 hour (not shown here for brevity).

Fig. 37 shows vertical displacements under gravity loads (e.g., self-weight or the weight of the structure itself) only for the girder which has pinned support at either end (at the centroid of the web) for both solid element and beam element models. Although the deformation behavior of the beam for both the models is similar, the maximum displacements differ (e.g., 0.221×10^{-4} m in the solid model as opposed to 0.236×10^{-4} m in the beam element model). Note that linear structural analysis was conducted for all cases.

Fig. 38 and 39 show plots of vertical deflections due to the combined gravity and thermal loads (obtained using the MATLAB data mapping script as shown in Fig. 36(b) for the 1 hour case). Again, the maximum displacements differ among the solid element and beam element models, with lower maximum vertical displacements for the beam element model. However, the deformation behavior among the models is similar, with both models showing out of plane displacements due to non-uniform thermal gradients.

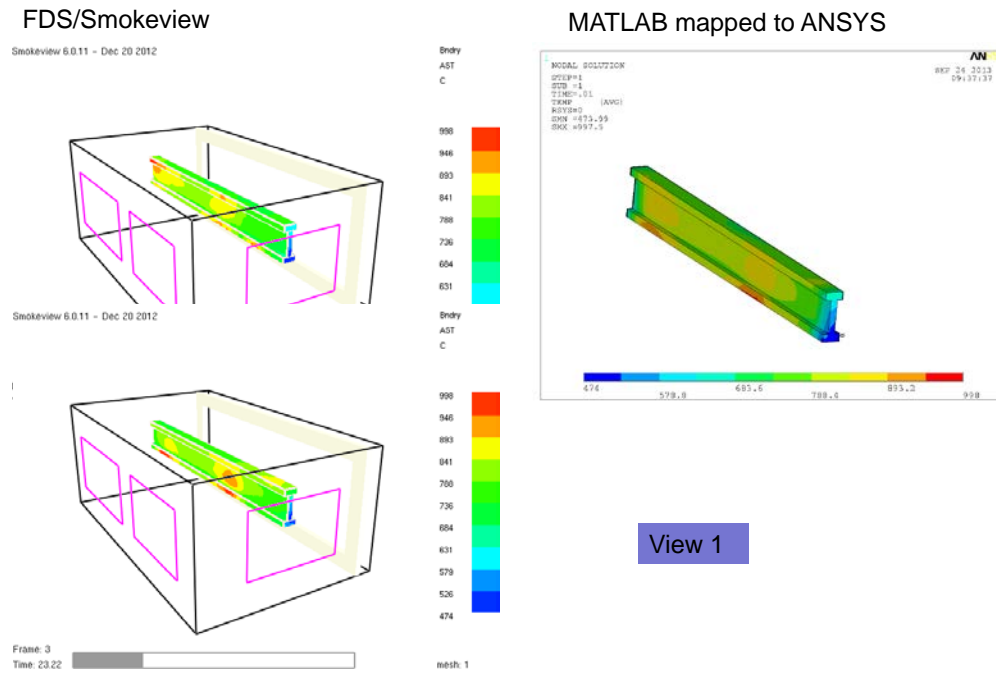


Fig. 35. A comparison of exposed surface temperatures in FDS and those mapped by the MATLAB script and read into the ANSYS heat transfer model.

ANSYS Thermal

1 hour

View 1

MATLAB Mapped Structural

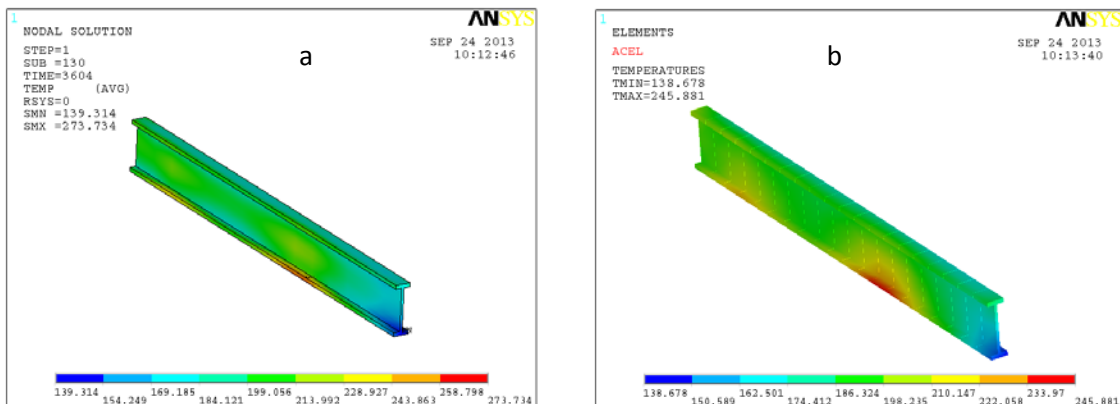


Fig. 36. (a) Temperatures in the beam at 1 hour as computed by the ANSYS heat transfer model; (b) as mapped in the structural beam element model using the MATLAB data mapping script.

ANSYS Solid Elements

Gravity Loads only

ANSYS Beam Elements

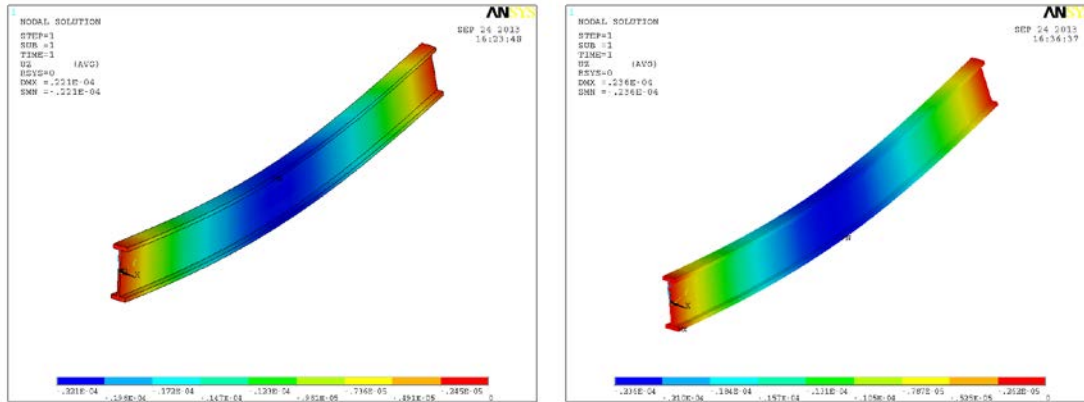
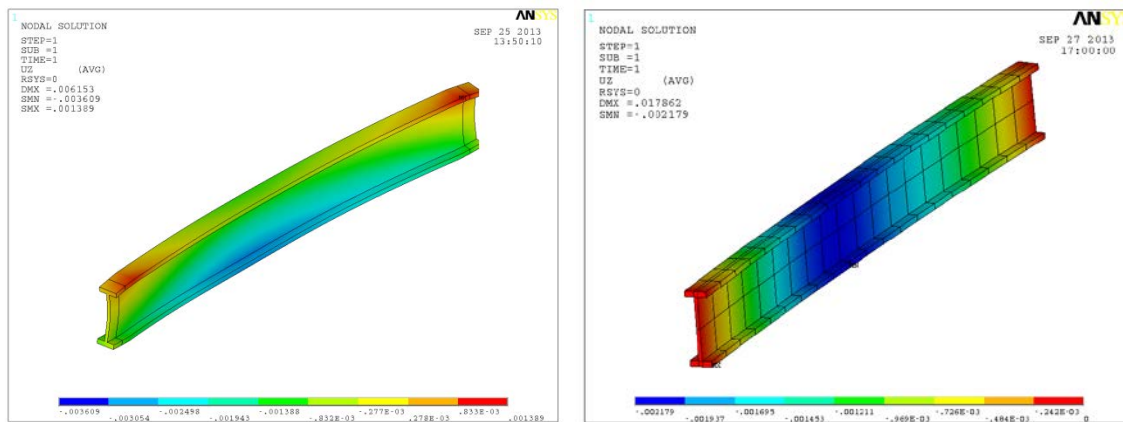


Fig. 37. Vertical displacements due to gravity loads only: (a) as computed in the ANSYS structural solid element model; (b) as computed with ANSYS structural beam element model.

Gravity + Thermal (1 hour)

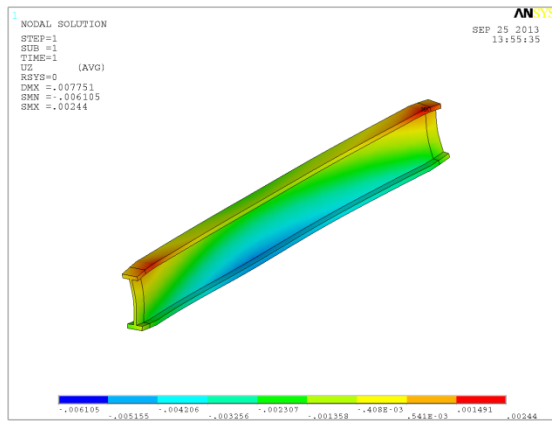


Solid Elements

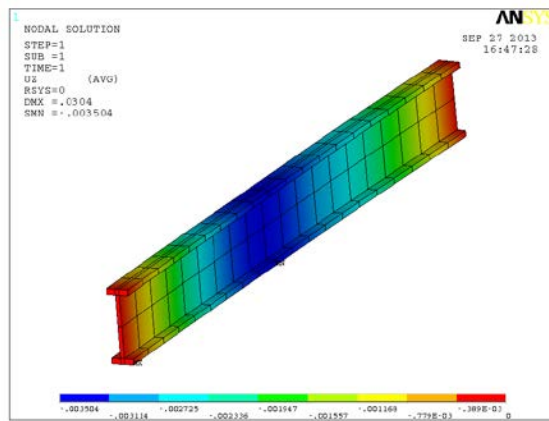
Beam Elements

Fig. 38. Vertical displacements at 1 hour in an unprotected steel beam: (a) as computed in the ANSYS structural solid element model using temperatures as shown in Fig. 36(a); (b) as computed with ANSYS structural beam element model with mapped temperatures shown in Fig. 36(b).

Gravity + Thermal (2 hour)



Solid Elements



Beam Elements

Fig. 39. Vertical displacements at 2 hour in an unprotected steel beam: (a) as computed in the ANSYS structural solid element model using temperatures as shown in Fig. 36(a); (b) as computed with ANSYS structural beam element model with mapped temperatures.

3.4.6 Data Formats:

The flowchart in Fig. 6 shows the files that the MATLAB data transfer script reads. In this section format of these files is described. No header or text information should be included in any of these ASCII files. Sample data formats are shown below with examples.

3.4.6.1 Thermal Nodal Coordinates (solid heat transfer FEA model):

<u>Node #</u>	<u>x-coordinate</u>	<u>y-coordinate</u>	<u>z-coordinate</u>
1	-386.9	-2.0	-6.0
2	-386.9	-2.0	-6.2
3	-386.9	-2.0	-6.0
4	-386.9	-2.0	-6.1
5	-386.9	-1.5	-6.2

3.4.6.2 Thermal Element Connectivity (solid heat transfer FEA model):

<u>Element #</u>	<u>Node 1</u>	<u>Node 2</u>	<u>Node 3</u>	<u>Node 4</u>	<u>Node 5</u>	<u>Node 6</u>	<u>Node 7</u>	<u>Node 8</u>
1	1	3	13	12	24	25	57	49
2	3	4	15	13	25	27	61	57
3	4	2	6	15	27	18	34	61
4	12	13	14	11	49	57	59	51
5	13	15	16	14	57	61	63	59

3.4.6.3 Nodal Temperatures (solid heat transfer FEA model) for six time steps:

<u>Node #</u>	<u>Temp, Step 1</u>	<u>Temp, Step 2</u>	<u>Temp, Step 3</u>	<u>Temp, Step 4</u>	<u>Temp, Step 5</u>	<u>Temp, Step 6</u>
1	80.4	150.8	255.8	414.1	546.2	648.2
2	50.5	96.0	165.3	270.1	373.1	465.6
3	80.3	150.7	255.6	413.8	545.7	647.8
4	80.1	150.1	254.8	412.6	544.3	646.5
5	79.8	149.2	253.5	410.7	541.9	644.2

3.4.6.4 Element Connectivity (Structural Model: beam or shell):

Example is shown for a beam model comprising 2-node line elements:

<u>Element #</u>	<u>Node 1</u>	<u>Node 2</u>
21580	48889	48872
21579	48888	48889
21578	48887	48888
21577	48886	48887
21576	48885	48886

3.4.6.5 Nodal Coordinates (Structural Model: beam or shell)

<u>Node #</u>	<u>x-coordinate</u>	<u>y-coordinate</u>	<u>z-coordinate</u>
48872	3300.6	1448.6	1858.6
48889	3320.3	1448.6	1858.6
48888	3340.0	1448.6	1858.6
48887	3359.7	1448.6	1858.6
48886	3379.4	1448.6	1858.6

Note that x-coordinates represent coordinates for the axial direction of the beam.

3.4.6.6 Nodal Temperature Transfer Points

3.4.6.6.1 Beam Elements:

The transfer points are to be defined by the user. Fig. 9 shows 15 transfer points for each beam node, whose temperatures are computed by the MATLAB script by interrogating the thermal results. The axial or x-coordinate of each transfer point associated with a node is the nodal x-coordinate, while y and z coordinates are determined by knowing the beam cross section dimensions (beam section width, beam section height, beam flange thickness, and beam web thickness). The y and z coordinates of all transfer points associated with each node are the same, only the axial coordinates of transfer points will vary from node to node. The user will be given a choice to enter transfer point coordinates in y and z directions by providing a file name.

3.4.6.6.2 Shell Elements:

The transfer points associated with shell nodes of a given element need to be defined for each layer of the shell element. The coordinates of the transfer points associated with nodes for each layer are the same as the nodal coordinates except that the coordinates in the vertical direction (e.g., slab height) will vary according to the vertical coordinates of each layer (e.g., the vertical coordinate for mid-plane offset will be the vertical coordinate of the mid-plane of each layer, see Fig. 3 or 17). The user will be asked to define slab thickness and number of layers. The coordinates of transfer points are computed from these data.

4. Summary

This report provides a summary of a stand-alone data mapping tool/software that was developed for transferring computed temperatures from a fire dynamics model to a FE heat transfer model and from a FE heat transfer model to a FE structural analysis model. Such a data mapping tool is needed as traditionally incompatible discretization schemes are used for each type of analyses. Therefore, this tool is appropriate in a sequential fire modeling / thermal analysis / structural analysis approach for analyzing the behavior of structures under fire.

FDS was used as the fire modeling software in this study. The transfer of fire modeling results from a FDS model to a heat transfer model comprises three tasks: a) parsing ASCII boundary file data obtained from a FDS analysis and storing AST data on grid points located on exposed surface of structures in the FDS model in matrix form, b) development of a unique computer algorithm for locating the nodes that are situated on the exposed surface of the FE heat transfer model, and c) mapping the temperature results in FDS grid (step a) to the FE surface nodes using a built-in MATLAB function.

The algorithm that was developed for transferring computed results from a FE solid element based heat transfer model to an equivalent FE structural model (using line and shell elements) relies on the fundamental concepts of finite elements pertaining to element shape functions and the use of local or natural coordinates.

The present tool has been implemented in MATLAB and can be used to interface with results from any commercial FEA software as long as ASCII data files containing nodal coordinates, element connectivities, and nodal temperatures are obtained from relevant FE models in ASCII format.

5. References

1. McGrattan, K., Hostikka, S., and Floyd J. (2010), “Fire Dynamics Simulator (Version 5) Technical Reference Guide”, NIST Special Publication 1018-5, Gaithersburg, MD.
2. ANSYS Mechanical Release 12.1 (2011), ANSYS Inc., Southpointe, 275 Technology Drive, Canonsburg, PA.
3. Dassault Systemes (2008), ABAQUS Version 6.8, User’s manual.
<http://www.simulia.com>.
4. Duthinh, D., McGrattan, K., and Khaskia, A. (2008), “Recent Advances in Fire-Structure Analysis”, Fire Safety Journal 43, pp. 161-167.
5. “The behavior of multi-story steel framed buildings in fire” (1999), British Steel plc., Swinden Technology Institute, Rotherham, UK.

6. Appendix 1 (Time-temperature curves used for Figure 21)

Time (s)	Fire Curve Temperature (°C)	Fire Curve (A) Temperature (°C)	Fire Curve (B) Temperature (°C)	Fire Curve (C) Temperature (°C)
0.0	20.0	20.0	20.0	20.0
93.5	56.2	26.2	26.2	26.2
200.0	124.6	94.6	84.6	44.6
293.0	189.5	159.5	139.5	99.5
318.6	224.5	194.5	174.5	134.5
730.2	265.3	235.3	215.3	175.3
1197.5	295.6	265.6	245.6	205.6
1748.2	313.8	283.8	263.8	223.8
2161.5	323.0	293.0	273.0	233.0
2381.0	344.3	314.3	294.3	254.3
2407.7	360.1	330.1	300.1	260.1
2709.0	399.0	369.0	349.0	309.0
3051.3	444.9	414.9	394.9	354.9
3146.0	478.3	448.3	428.3	388.3
3392.4	513.6	483.6	463.6	423.6
3501.9	527.7	497.7	467.7	427.7
3611.5	541.9	511.9	481.9	441.9
3982.7	563.3	533.3	503.3	463.3
4092.8	568.7	538.7	508.7	468.7
4198.8	645.9	615.9	565.9	525.9
4308.4	660.0	630.0	580.0	540.0
4570.3	663.8	633.8	583.8	543.8
4735.5	669.3	639.3	589.3	549.3
4983.5	674.8	644.8	574.8	534.8
5424.0	689.3	659.3	589.3	549.3
5974.2	718.0	688.0	618.0	578.0
6909.7	762.9	732.9	662.9	622.9
7225.8	782.5	752.5	672.5	632.5