FDS2FTMI User's Guide

An automated code to one-way coupling between FDS and FEM using FTMI

Julio Cesar Silva

April 18, 2022
FDS Version
Revision: FDS6.7.7-902-g0957e283f

Contents

Chapter 1

Introduction

This document explains the developed strategy to evaluate the behavior of structures under fire conditions by a coupled analysis including Fire Dynamics Simulator (FDS[**?**]) and Finite Element Method (FEM) codes. Common practice is to extract the results from a fire simulation, and apply those as boundary conditions at the exposed surfaces for the thermomechanical analysis. This kind of approach is commonly referred as one-way coupling. In a two-way coupling strategy, the thermomechanical results, e.g., displacements, collapses, etc. are transposed back to the fire simulation. The two-way approach can lead to a more complex simulation, increasing the amount of data to be transferred between the models. Its advantages are related to cases where displacements can change the fire source or the fluid flow pattern, creating a different fire scenario. Using one-way coupling it is possible to establish each model separately, which means that the same fire simulation results can be used on slightly different thermomechanical models, contributing to development of optimized structures.

The Fire-ThermoMechanical Interface [**?**] is a one-way coupling procedure that uses the Adiabatic Surface Temperature (T_{AST}) concept [**?**, **?**] to establish an interface between FDS and FEM codes. The procedure recreates the *net* heat flux evaluated in FDS, in the FEM environment by transferring T_{AST} and the heat transfer coefficient (h) from FDS results and adopting the wall temperature (T_{w}) provided by the FEM code at each time step (h can also be adopted as a constant). This heat flux can be calculated by FEM over the elements faces and applied into each node of the exposed surface. FDS2FTMI is a code created to perform these tasks in an automated way, allowing the simulation of the behavior of global structures, discretized with shell and/or solid elements, exposed to fire conditions. FTMI can be used with any FEM code, at this time, FDS2FTMI is implemented between FDS and ANSYS or between FDS and LS-DYNA [1].

The code FDS2FTMI is included in the FDS-SVM repository [**?**] under `Utilities/Structural_Interaction/fds2ftmi`. A makefile is provided to help in the compilation process with different operational systems and compilers. A shell and a bat scripts (`scripts/build_fds2ftmi.sh`, `scripts/build_fds2ftmi.bat`) are also provided to build this manual, run the Verification cases and check the results [2].

Some validation work done with this methodology is provided at Zhang *et. al* [**?**].

[1] 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 author or 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.

[2] Two environment variables are needed to run these scripts: FDS_GITROOT, which is the path for fds-smv git folder; and ANSYS, which is the path to the ANSYS executable (including the executable).

Chapter 2

Getting Started

Heat can be transferred from flames and hot gases to structures surfaces by radiation and convection. The total heat flux ($\dot{q}''_{\text{tot}}$) is defined by the sum of these two parcels:

$$\dot{q}''_{\text{tot}} = \dot{q}''_{\text{r}} + \dot{q}''_{\text{c}} = \varepsilon \left(e_{\text{inc}} - \sigma T_{\text{w}}^4\right) + h(T_{\text{g}} - T_{\text{w}}) \tag{2.1}$$

where $\dot{q}''_{\text{r}}$ is the radiative heat flux, $\dot{q}''_{\text{c}}$ is the convective heat flux, ε is the emissivity, e_{inc} is the radiative thermal energy incident on the exposed surfaces, σ is the Stefan-Boltzmann constant, T_{w} is the wall temperature and T_{g} is the gas temperature.

Advanced fire simulation models provide results that characterize the three dimensional evolution of the fire, the radiative thermal energy incident on the exposed surfaces and the gas temperatures. However, the models are not capable of accurately characterizing the temperature distribution on solids, and thus the surface temperature. Consequently, the total heat flux, as presented on Eq. 2.1, cannot be precisely calculated at the end of the fire simulation. Therefore, an additional approach is necessary to cross this barrier.

2.1 Adiabatic Surface Temperature

In order to establish an accurate interface between the fire simulation and the thermomechanical analysis, the Adiabatic Surface Temperature concept is used [**?**]. A real surface can be replaced by a perfectly insulated surface exposed to the same heating conditions; the total heat flux to this ideal surface is by definition zero :

$$\varepsilon \left(e_{\text{inc}} - \sigma T_{\text{AST}}^4\right) + h(T_{\text{g}} - T_{\text{AST}}) = 0 \tag{2.2}$$

and the temperature of this idealized surface can be obtained as an output from the fire simulation.

Since the real and the hypothetical surfaces are exposed to the same heating conditions, the total heat flux that will be applied to the thermomechanical model can be obtained by the solution of the system formed by Eqs. (2.1) and (2.2):

$$\dot{q}''_{\text{tot}} = \varepsilon\sigma \left(T_{\text{AST}}^4 - T_{\text{w}}^4\right) + h(T_{\text{AST}} - T_{\text{w}}) \tag{2.3}$$

This single variable (T_{AST}) is considered capable of reducing the complexity of the fire simulation into one simple scalar [**?**, **?**, **?**]. To achieve a correct definition of the total heat flux, accounting correctly both the convective and radiative heat fluxes, it is also necessary to add the convective heat transfer coefficient (h).

During the FDS simulation, T_{AST} and h can be recorded via `DEVC` or `BNDF` outputs. To use `DEVC`s, the points (or nodes) where the information will be collected need to be determined in advance, before the fire simulation. On other hand, `BNDF` generates outputs at all exposed surfaces (boundary cells marking the interface between gas and solid) into the domain. FDS2FTMI extracts information stored in `BNDF` files, so the coupling procedure can be achieved for different discretization levels (at FEM). Small modifications or dimensioning in FEM does not imply restarting the fire simulation.

2.2 Capturing the Mesh Information from FEM

In FDS, the domain is divided into orthogonal cells in a rectilinear grid and geometries are approximated by blocks (OBSTs). For FEM thermomechanical analysis, the structure is usually discretized with solid (tetrahedrons or hexahedrons) or shell (3D plane) elements. Also, the element size needed at the thermomechanical model can lead to an unfeasible fire simulation. Even for modest structures, with simple geometries, the models generated for each side of this interface may not match perfectly. Therefore, the communication between the two models is based on the spatial position of the nodes on the exposed surfaces (FEM) and the cell faces (FDS). The geometries created for both models need to share a common Cartesian coordinate system, i.e., they need to have the same origin and they need to be consistent. FDS2FTMI main function is to read the BNDF files and extract T_{AST} and h at the exposed surfaces. This code also generates the files that need to be read by the FEM code to apply correct boundary conditions. The source code and a makefile can be found at `Utilities/Structural_Interaction/fds2ftmi/scripts`.

The input parameters for FDS2FTMI are the spatial position of the exposed surfaces at FEM model. These parameters are divided in 2 files, “nodes.dat” and “elements.dat”, and include the node positions (Cartesian coordinates) and the node numbers at the exposed faces ordered using right-hand rule with normal pointing outwards.

ANSYS

The interface with ANSYS is developed using surface effect elements, so the information presented in the files “nodes.dat” and “elements.dat” are related to those surface effect elements. To help in the generation of the surface effect elements and these two input files, some ANSYS macros were created. It is possible to export the mesh information using the macros “ansys2ftmi.geom” or “ansys2ftmi_shell.geom”. The first can be used for solid elements and the second one is devoted to shell elements. The shell element file creates interfaces on both sides (top and bottom layers) of the shell elements [1]. These macros were designed to create interface elements attached to the boundary element faces between solid and gas. To select just the exposed surfaces related to a specific scenario, those surfaces can be selected in ANSYS before running the macros (Select Entities > Areas). The macros are available at `Utilities/Structural_Interaction/fds2ftmi/scrips`.

LS-DYNA

To create this interface with LS-DYNA, only the information of the actual mesh needs to be presented in the files “nodes.dat” and “elements.dat”.

Examples of these two files are given below:

nodes.dat

The first line of this file has the number of nodes and the highest node number in the FEM model. All other lines have just the node number and the Cartesian coordinates (x, y and z).

```
100   2400
1    1.000    0.000    0.000
```

[1] To help convert an ANSYS model geometry into OBST, an ANSYS macro (“ansys2fds_areas_obst.geom”) that convert areas into OBST is available at scripts.

```
   2    1.000    0.100    0.000
   3    1.000    0.200    0.000
   4    1.000    0.300    0.000
   .
   .
   .
   100  1.000    1.000    1.000
```

In this example, the exposed surface has 100 nodes and the highest number at the FEM model is 2400. So, when the code generates nodes for interface elements it starts in 2401. All additional lines have the nodes coordinates.

elements.dat

The first line of this file has the number of interface elements that compose the exposed surface, and 3 additional flags. First flag says the type of element used in FEM model: 0 means solid elements and 1 means shell elements. If the thermomechanical model uses shell elements, the interface is defined in both top and bottom layers. The second flag defines in which layer the heat flux is prescribed: 1 means top layer and 2 means bottom layer. The last flag defines the number of the interface element type in ANSYS.

```
   200     0     0     2
   801   120     2   148   117   121   149   139   118
   802   126   128   130   151   127   144   125   150
   803   118   120   117   135   119   118   157   136
   804   128   130   128   126   129   145   127   144
   .
   .
   .
  1000   178   163    41    69   179   164    10    80
  END
```

In this example, there are 200 interface elements in this model, they are attached to solid elements (so the first and second flag are 0) and they correspond to the element type 2 in the FEM model. The code FDS2FTMI can work with bilinear or linear elements, which have a different number of nodes. In the case of using linear elements, the non-used positions are filled with zeros.

In the next example there are 72 interface elements attached to top layer of shell elements (from 73 to 144) and another 72 applying the heat flux to bottom layer of the shell elements (145-216). The input file structure remains the same, but it is split into two parts, each one with its own flags.

```
    72     1     1     2
    73     1     8    10     2     0     0     0     0
    74     2    10     9     3     0     0     0     0
    75     4    11    12     6     0     0     0     0
    76     2    10    11     4     0     0     0     0
   .
   .
   .
   144    82    91    90    84     0     0     0     0
    72     1     2     3
   145     8     1     2    10     0     0     0     0
```

```
146    10     2     3     9     0     0     0     0
147    11     4     6    12     0     0     0     0
148    10     2     4    11     0     0     0     0
 .
 .
 .
216    91    82    84    90     0     0     0     0
END
```

2.3 Executing the code

After the definition of the files nodes.dat and elements.dat, the code FDS2FTMI is executed to extract information from FDS boundary files, this code requires some information about the FDS model:

CHID `CHID` drives the naming convention of FDS output files. It is needed to give the path to the code get information from .smv and .bf files.

Cell size This is used to create a virtual cell over the node position to search for results in `BNDF` files. It is generally adopted as the same value as the cell size in FDS simulation.

Number of variables The number of variables (1 or 2) is used to determine if the interface will get h from FDS simulation or use a constant value. If the number of variables is defines as 2, the code will extract both T_{AST} and h from FDS simulation. If the number of variables is defined as 1, the code will get just the results for T_{AST} and a constant value of h is prescribed in ANSYS. So, just in this case, the next value on this same line is h.

Time To address the time boundary, the code needs the starting, ending time and the time interval. The time interval that the results are written into `BNDF` files can be controlled by `DT_BNDF` at `DUMP` line, the default is given by $2\Delta t/\text{NFRAMES}$.

Number of average points In some scenarios, a constant exposure condition can be assumed. To calculate a constant steady-state exposure condition from a short time history, a number of average points should be provided. If NO average is wanted, 0 (zero) should be prescribed.

Time average The time step at the FEM model can be bigger than the time interval when FDS write results. In this case, an average needs to be performed to calculate the correct thermal exposure of the surfaces. If NO average is wanted, 0 (zero) should be prescribed.

FEM Code This is used to tell FDS2FTMI which FEM code will be used in the thermomechanical analysis. At this moment, ANSYS and LS-DYNA is supported, and this variable should be equal to 1 for ANSYS and 2 for LS-DYNA.

Output file The code will generate an output file containing all the commands that need to be read by ANSYS (File > Read input from). This ouput file will create all elements, parameters and loads necessary to input the heat flux from the flames and hot gases into the nodes of the exposed surfaces.

These information can be passed to the code via arguments (command line), input file or just typing as the code steps through execution. As arguments or input file, the parameters need to be given at the same order they are presented above. For example:

```
fds2ftmi_linux_64 simple_panel_hot 0.05 2 0 600 2 0 0 1 simple_panel
```

or

```
fds2ftmi_linux_64 simple_panel_hot.fds2ftmi
```

with simple_fire_hot.fds2ftmi:

```
simple_panel_hot
0.05
2
0 600 2
0 0
1
simple_panel
```

start the code with simple_panel_hot as `CHID`, a cell size equal to 0.05, 2 variables, starting at 0s to 600s with 2s as time interval, with no average points, no time average, exporting the results in ANSYS format and writing the results at “simple_panel.dat”.

The code will search into the `BNDF` files for the results. Searching over every surface in the FDS simulation can slow down this process. To reduce the size of the `BNDF` files, add `BNDF_DEFAULT=.FALSE.` in `MISC` line and the add `BNDF_OBST=.TRUE.` at the exposed surfaces `OBST` lines.

Chapter 3

Verification Cases

This chapter contains the Verification cases that test the FDS2FTMI code. The SVN number on the figures is related to the FDS repository [?] subversion control number used to compile the codes and run the cases.

3.1 Simple panel exposed to radiation and convection (simple_panel_hot)

This example verifies the FDS2FTMI ability to translate the heat transfer between solid and gas phase at fire simulation into boundary conditions for a FEM model (ANSYS). It also demonstrates that the code can extract correctly the results from BNDF files. This model is composed by 2 panels, one has a constant temperature of $T_s = 1000$ °C and the other one has an initial temperature of $T_s = 20$ °C. The temperature evolution at the later panel is the target of this exercise (the input files are provided in examples/simple_panel_hot). The case is set to be 2D, with only one cell at y axis. These panels are 1 m height (z axis) and they are 1 m from each other. Three DEVCs were setup to compare the results, they are placed at 0.775 m (1), 0.525 m (2) and 0.275 m (3) in z axis. After the fire simulation, the boundary conditions are translated to ANSYS by FTMI. A comparison between the Adiabatic Surface Temperature T_{AST} and the surface temperature T_s obtained by FTMI and FDS is provided in Fig. 3.1.

Figure 3.1: Comparison of the results obtained by FTMI (with ANSYS) and FDS.

3.2 H profile column exposed to a localizes fire (h_profile)

In this case a simple supported H-profile column is exposed to an adjacent hot panel ($T_s = 1000$ °C). The steel column is 3 m tall (z axis). The cross section is a 0.3 m (flange, parallel to x) x 0.4 m (web, parallel to y) with a 12.5 mm thick web and 16 mm thick flanges. The hot panel is parallel to the web and localized 30 cm from the web in x direction (or 15 cm from the end of the flange). The column is discretized with shell elements (FEM) to verify the use of the code to model shell structures. Each side of these plane elements will have a different thermal exposure, incident radiation and gas temperature around the surface. FDS2FTMI is designed to prescribe the correspondent heat flux at the shell elements top and bottom layers. Points A, B and C are located at the column middle span. A and C are located at the end of the flange (closest points to hot panel), B is located at the middle of the web.

The target of this qualitative verification exercise is to check the thermal gradient due to heat fluxes applied in top and bottom layers, the spatial temperature distribution and the thermomechanical behavior due to heating. The evolution of the surface temperature is presented in Fig. 3.2. As the hot panel is parallel to the web, the heat flux incident to its surface will be higher than the heat flux incident to the flanges, so the surface temperature at A is higher than at B. Even with the high conductivity of steel is possible to observe a little gradient between the exposed and back surfaces at A and B. The spatial distribution also presented in Fig. 3.2 shows the expected profile. This profile is not exactly symmetric through column mid-span because of the energy reflected by the ground and the variation on the gas velocities around the column (which changes the heat transfer coefficient - h), so the temperature is lower at the top of the column.

Figure 3.2: Evolution of surface temperature and spatial distribution at 60 min.

This simple supported column is subjected to a vertical load of 325 kN, which correspond to 1/50 of the Euler's buckling critical load. Under these loading conditions, the column will be close to a uniform stress state and remain straight at the beginning of the simulation, then the horizontal displacements generated will be due to the temperature distribution. The evolution of horizontal displacements is presented in Fig. 3.3. As expected, the horizontal displacements in x axis are equal at A and C and represents the maximum achieved at the column, as can be observed at the lateral view. In y direction the value of the displacement at these two points are the same, but as column is expanding and they are moving away from each other the signal of these displacements are opposite.

Figure 3.3: Evolution of horizontal displacements in x and y axis and a lateral view of the plane xz (displacements amplified by x15).

3.3 Summary of Verification Results

This section summarizes the results from Verification cases with the objective of evaluate the output predicted by FDS2FTMI code. The presented results can help identify errors in the source code.

Case Name	Expected	Predicted	Type of Error	Error	Error	Within
	Metric	Metric			Tolerance	Tolerance

Bibliography

[1] K. McGrattan, S. Hostikka, R. McDermott, J. Floyd, C. Weinschenk, and K. Overholt. *Fire Dynamics Simulator, User's Guide*. National Institute of Standards and Technology, Gaithersburg, Maryland, USA, and VTT Technical Research Centre of Finland, Espoo, Finland, sixth edition, September 2013.

[2] J.C. Silva, A. Landesmann, and F.L. Ribeiro. Fire-Thermomechanical Interface Model to Performance-based Analysis of Structures Exposed to Fire. *Fire Safety Journal*, 83:66–78, 2016.

[3] U. Wickström, D. Duthinh, and K.B. McGrattan. Adiabatic Surface Temperature for Calculating Heat Transfer to Fire Exposed Structures. In *Proceedings of the Eleventh International Interflam Conference*. Interscience Communications, London, 2007.

[4] U. Wickström, R. Jansson, and H. Tuovinen. Verification fire tests on using the adiabatic surface temperature for predicting heat transfer. Technical Report 2009:19, SP Technical Research Institute of Sweden, Bo̊ras, Sweden, 2009.

[5] `https://github.com/firemodels/fds`.

[6] C. Zhang, J.C. Silva, C. Weinschenk, D. Kamikawa, and Y. Hasemi. Simulation Methodology for Coupled Fire-Structure Analysis: Modeling localized fire tests on a steel column. *Fire Technology*, 52(1):239–262, 2016.

[7] D. Duthinh, K. McGrattan, and A. Khaskiac. Recent advances in fire structure analysis. *Fire Safety Journal*, 43:161–167, 2008.

[8] J. Sandström, U. Wickström, and M. Velkovic. Adiabatic Surface Temperature: A sufficient Input Data for a Thermal Model. In *Proceedings of Application of Structural Fire Engineering*, 2009.

[9] U. Wickström, A. Robbins, and G. Baker. The Use of Adiabatic Surface Temperature to Design Structures for Fire. In *Proceedings of Structures in Fire (SIF)*, 2010.