Contents

1. Introduction

FDS2FEM is a command-line tool for coupling the computational fluid dynamics (CFD) programme Fire Dynamics Simulator (FDS) and the general purpose finite element method (FEM) programme ABAQUS to perform thermal-structural analyses. FDS is available in public domain. Its source code is maintained under SVN version control at Google Code (http://code.google.com/p/fds-smv/). ABAQUS is a commercial software product of Dassault Systèmes.

The coupling established by FDS2FEM is sequential and one-directional. The results of an FDS simulation are used as time-dependent boundary conditions for a subsequent ABAQUS analysis with no feedback from ABAQUS to FDS. This approach is also known as "forcing". Quantities available for transfer include surface temperature and heat flux in the form of adiabatic surface temperature.

Why do we need an external tool to establish this kind of coupling? First of all, there is a need to get two independent simulation software, FDS and ABAQUS, to communicate with each other. The output data from FDS has to be converted into a form that ABAQUS can accept as input. Secondly, the data cannot be transferred "as is". In most cases, there will be differences between the geometries, mesh types and mesh resolutions of an FDS and an ABAQUS model — even if they represent the same object. There is a need to create a mapping between the representations.

A schematic diagram of a coupled FDS-ABAQUS simulation is shown in Figure 1. The procedure begins with the creation of input files for both models. FDS2FEM has some special requirements concerning the structure and content of the input files. These are discussed in Chapter 5. In the next phase, the FDS simulation is run and the user-requested output files are created. At this point FDS2FEM is ready to be run. It is operated using a configuration file based user interface. FDS2FEM generates a mapping between the FDS and ABAQUS model geometries and transforms FDS output data into a thermal boundary condition for the ABAQUS model. This boundary condition is stored in the ABAQUS input file and two external data files. Available mapping methods are discussed in Chapter 7. After the FDS2FEM run is completed, the ABAQUS model is ready to be run.

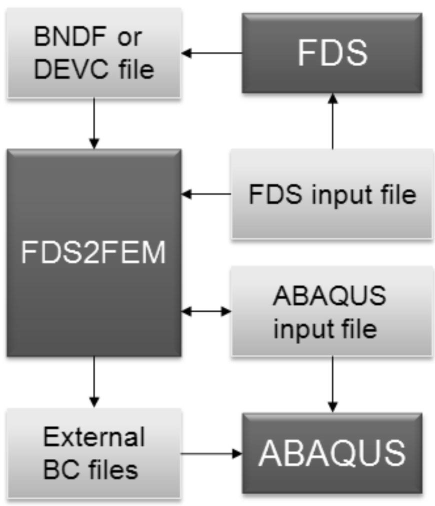

Figure 1. Overview of the FDS-ABAQUS coupling scheme

2. Limitations

The current version of FDS2FEM has some major limitations. They are listed in what follows.

- **FDS: unsupported features.** The following FDS features are not supported.
  - Multiple-mesh geometries (when NSET connectivities are given)
  - Non-uniform meshes
  - Creation and removal of obstructions during simulation
  - Device arrays
  - Making fuels disappear ( )
- **FDS: boundary patch identification.** Boundary file output stored by FDS is grouped into patches. Associating obstruction surfaces in FDS with node sets in ABAQUS is done by referring to patch numbers and node set names. Numbering of the FDS patches is not straightforward. This is because solid obstructions defined in an FDS input file are processed by FDS to different, usually smaller, pieces of solid obstructions internally. This internal numbering is used to identify the patches. If and/or keywords are used in FDS input, the numbering of patches might become ambiguous.
- **ABAQUS: input file format.** The ABAQUS input file has to conform to the part-part instance-assembly structure. Continuation lines are not allowed.
- **ABAQUS: unsupported keywords and parameters.** The following ABAQUS keywords are not supported: , , and . In addition, the following keyword-parameter pairs are not supported: - and - . The program will output an error message and stop execution when unsupported keywords and/or parameters are encountered.
- **ABAQUS: supported element types.** Only some of the element types available in ABAQUS are supported. These include the following linear elements: DC3D8, DCC3D8, DCC3D8D, DC3D4, DS3 and DS4.
- **ABAQUS: coordinate system.** Only Cartesian coordinate system is supported.
- **ABAQUS: node set definitions.** ABAQUS node sets that are defined outside part and part instance levels are not recognized.
- **ABAQUS: model size.** FDS2FEM memory and CPU time requirements increase drastically as the number of elements in the model is increased.
- **Geometry differences between FDS and ABAQUS models.** These are unavoidable due to the different mesh resolutions and ways to represent geometries. The user should take this into account when building the models and deciding which data from FDS is imported into ABAQUS and how the mapping should be done. The coupling between the two programmes is quite weak and approximate. Also, the results of the fire simulation are subject to many uncertainties due to the nature of fire phenomena.
- **Forcing of thermal boundary conditions.** Using surface temperature calculated in FDS as a boundary condition in ABAQUS has one obvious problem. We are forcing the ABAQUS heat conduction solver to (partially) follow the results given by the less accurate FDS heat conduction solver. The reliability of the coupled analysis becomes strongly dependent on the performance of the latter. Due to this, the material model in FDS should be adequately defined and the user should be familiar with the limitations of the one-dimensional solid phase solver. When tranferring net heat flux using adiabatic surface temperature, the coupled analysis is much less sensitive to the performance of the FDS solid phase solver.

- **Research software.** FDS2FEM is a useful tool for an engineer who knows what (s)he is doing. One is required to have good knowledge of CFD fire simulations and FEM modelling. One should use expert judgement on how and which information to transfer from CFD to FEM and how accurate results one might anticipate. The geometry limitations in FDS make it impossible to model shadowing effects of thermal radiation for fine structures like steel trusses. Even structural elements as simple as I-beams pose problems, because the flanges and webs of the beams are not resolved in the FDS simulation.

3. Disclaimer

Users are warned that FDS2FEM is intended for use only by those competent in the fields of fire and structural simulation, and is intended only to supplement the informed judgement of the qualified user. The software packages (FDS, ABAQUS) that are used in context of FDS2FEM are computer models that may or may not have predictive capability when applied to a specific set of factual circumstances. Even if the fire and structural models might be appropriate, the use of FDS2FEM to transfer the fire related information to the structural model might not be appropriate. Lack of accurate predictions by the model could lead to erroneous conclusions with regard to life safety. All results should be evaluated by an informed user.

4. Using FDS2FEM

4.1 Basics

FDS2FEM is operated using a configuration file based user interface. A configuration file consists of any number of keyword-value pairs in arbitrary order. Recognized keywords are listed in Table 1. A short description of the keywords and their valid values is also given. A more thorough description is given in the command reference section (Chapter 6). Default values are shown in bold face font, when applicable. If no default value is listed, the user is required to provide the value. These keyword-value pairs are obligatory for FDS2FEM to run. Optional keywords are listed using slanted font. If the same keyword appears many times in the configuration file then the last value in the file is used.

Table 1. Recognized keyword-value pairs

Keyword	**Values (default as bold face)**	**Description**
fds_input	string	FDS input file name
fem_input	string	ABAQUS input file name
quantity	wall_temperature / adiabatic_surface_temperature	Quantity to be transfered
fds_output	devc / bndf	FDS data file type: DEVC or BNDF
nset_input	string (default **no file given**)	Connectivity list file name
dump_fds_xyz	xyz / vtk / **off**	FDS node coordinate dump
dump_fds_data	sdf / vtk / **off**	FDS data dump
dump_fem_xyz	xyz / vtk / **off**	ABAQUS node coordinate dump
dump_fem_data	sdf / vtk / **off**	ABAQUS data dump
fds_statistics	on / **off**	Calculate FDS node set statistics
fem_statistics	on / **off**	Calculate ABAQUS node set statistics
match_translate	manual / automatic / **off**	Translate coordinates before mapping
match_rotate	manual / automatic / **off**	Rotate coordinates before mapping
origin_fds	3× real (default **0.0 0.0 0.0**)	FDS origin
origin_abqs	3× real (default **0.0 0.0 0.0**)	ABAQUS origin
euler_alpha	real (default **0.0**)	Euler angle α
euler_beta	real (default **0.0**)	Euler angle β
euler_gamma	real (default **0.0**)	Euler angle γ
mapping	**nearest** / devc_to_nset / off	Mapping method selection
mp_n	integer (default **4**)	Number of NNs used in mapping
mp_nmx	integer (default **8**)	Maximum number of NNs used in mapping
mp_cut	real (default **0.0**)	Cut-off radius for mapping
mp_del	real (default **0.1**)	Cut-off radius tolerance factor
mp_deg	real (default **2.0**)	Power of the distance decay of the weights

FDS2FEM is a command-line application for both Windows and Linux operating systems. It is run by a simple command

where is the name of the configuration file. It should be noted that the FDS2FEM executable should be located in a directory that is included in the environment variable . The configuration file is a plain text (ASCII) file. It should be located in the same directory as the FDS and ABAQUS input files, the FDS output files and the optional NSET connectivity file, i.e., all files should be located in the same directory. A good practice is to make a new directory for the fds2fem project, where the FDS model is run so that all FDS output files are there. Another way is to run the FDS model in some directory and copy the associated files to the FDS2FEM project directory.

4.2 Preparing an FDS input file

FDS2FEM is used to transfer thermal boundary conditions (surface temperature or heat flux in the form of adiabatic surface temperature) from FDS to ABAQUS. Before it can be run, some preparations have to be made in the FDS input file.

Firstly, the user should make sure that only supported FDS features are being used (see Chapter 2). Multiple mesh geometries, non-uniform meshes and device arrays are, for example, not supported. Secondly, the user should request device or boundary file output of the desired quantity, whether it be or . It is advisable to remove all unnecessary output requests of the desired quantity. In the case of boundary file output this can be done by using the option of the MISC-namelist. The user shoud then explicitly define the surfaces from which output data is collected by using and options of the -namelist. Finally, the user should also restrict the amount of data dumps by using the option of the -namelist. This will reduce the size of the FDS2FEM output files considerably.

It should be noted that entries with output are not supported and that the obstructions should be stationary, i.e., they should not be created or removed during the simulation. The -namelist can be used, but it is still a fragile construction in FDS2FEM. If s are used, the user should carefully check the mapped data (s should also be stationary, i.e., applied before the beginning of the simulation). The user should note that each “thick” obstruction will have six faces, so that it is convenient to request the boundary output using option of the -namelist.

When using device output (-namelist), only the coordinates of the device and the output quantity have to be given. The output time interval is dictated by the option of the -namelist.

The user should keep in mind that and boundary file and device requests can coexist in an FDS model. In this way, a single FDS run can produce data for many different mappins. The FDS2FEM configuration file can be used to select the desired output type and quantity. Below is an example of an FDS model of a room with three doors. of each of these doors is requested as output using both boundary file and devices. FDS input file features required by FDS2FEM are shown in boldface font.

```
      DT_BNDF=5.0,
      DT_DEVC=5.0

      BNDF_DEFAULT=.FALSE.


      BNDF_FACE(1)=.TRUE.


      BNDF_FACE(-1)=.TRUE.


      BNDF_FACE(2)=.TRUE.
```

```
&BNDF QUANTITY='WALL TEMPERATURE' /

&DEVC
      QUANTITY='WALL TEMPERATURE',


&DEVC
      QUANTITY='WALL TEMPERATURE'


&DEVC
      QUANTITY='WALL TEMPERATURE'
```

4.3 Preparing an ABAQUS input file

FDS2FEM has some requirements and restrictions concerning the ABAQUS input file. The input file has to conform to the part-part instance-assembly structure. Keyword continuation lines are not allowed. In addition, some ABAQUS keywords and many element types are not supported. These were discussed in Chapter 2.

FDS2FEM modifies the existing ABAQUS input file by adding Include-keywords that point to external data files. These data files contain the generated boundary condition. The user defines target surfaces for the boundary condition by using node sets (Nset-keyword). By default, FDS2FEM parses the ABAQUS input file and includes all available node set definitions in the target surface. Most of the time, this is not what the user wants. It is better that the boundary condition is applied only to a few node sets. Target node sets can be selected by creating a specific NSET connectivity file. The NSET connectivity file and its contents are discussed in Chapter 5.3.

Below are some examples of what FDS2FEM might add to the Abaqus input file. These lines should not be edited or removed.

Most of the additions are Include-keywords that point to external files. These external files contain , , and keyword lines that are used to assign a boundary condition to the target node sets. When surface temperature is used as the transfer quantity, and keywords are used. On the other hand, when net heat flux is transferred, , and keywords are used. In this case, also a keyword line is added in the ABAQUS input file.

4.4 Preparing an NSET connectivity file

Node sets defined in an ABAQUS input file can be connected to surface patches or devices in FDS. This is achieved by using an NSET connectivity file. The format of the connectivity file is simple: on each line a single ABAQUS node set name followed by a space-delimited list of FDS surface patch or device numbers (or IDs). This way, each of the node sets receives a boundary condition from selected surface patches or devices. An FDS surface patch or device may provide data for multiple ABAQUS node sets, i.e. the same patch or device number can appear on more than one line in the connectivity file. An ABAQUS node set name can appear only once. An example of the contents of an NSET connectivity file is given below.

Here we assign boundary condition data from FDS surface patches (or devices) one to three to ABAQUS part instances , , and . Each part instance has a node set named The first part instances listed in the NSET connectivity file, , receives data only from surface patch number one. The second part instance, , receives data from surface patches one and two, etc.

For devices, the ordinal numbers are easily deduced. Device number one is the first one defined in the FDS input file, device number two is the second one defined, and so on. For surface patches (boundary file output), the situation is more complicated. FDS obstructions have six faces. In boundary file output, each face is treated as a separate surface patch. Thus for each obstruction, boundary file output can be requested from zero to six patches. If the FDS model consists of a single obstruction and boundary file output is requested from all its faces, the boundary file will contain data for six patches - numbers one to six. If output is requested only from three of the faces, the boundary file will contain data for three patches – numbers one to three. In summary, each FDS obstruction can provide boundary output from zero to six patches and the patches are numbered in the order they appear in the FDS input file. If or namelists are used, things get even more complicated.

In the complicated cases, the only straightforward way to create an NSET connectivity map is by trial and error. The strategy is to give just one line in the connectivity file and there one FDS patch number at a time and visualize the imported nodes in some external viewer. This can be done by giving FDS2FEM the commands and . This way the user can figure out the correspondence between FDS surface patches and their ordinal numbers. This is a quite laborious approach, but at the present the only one.

To avoid trouble with FDS patch numbers, only the ABAQUS node set names can be listed in the NSET connectivity file. In this case, all available FDS data is used in the mapping. This approach should produce good results if the nearest neighbour mapping algorithm is given sensible parameter values. To increase the robustness of this approach, the user is advised to give the optional radius argument to the nearest neighbour mapping method, so that “wrong” surface patches are not used as a source for some node sets. One should be especially careful when there are objects that are thin, and the mapping algorithm might use FDS data from both sides of the object.

When using direct device-to-node set mapping ( ), data from FDS devices is treated in a special way. If a single device is assigned to a single node set, the data is transferred “as is”. If, on the other hand, multiple devices are assigned to a single node set, an average value is used. This is a simple arithmetic average (i.e. non-weighted).

4.5 Preparing a configuration file

FDS2FEM uses a configuration file based user interface. The configuration file is an arbitrarily named plain text (ASCII) file with keyword-value pairs on consecutive lines. These keyword-value pairs can be given in any order. Recognized keyword-value pairs were listed in the beginning of the chapter (

) and their meaning are more thoroughly explained in Chapter 6. Only lines beginning with a recognized keyword are read in. The configuration file can also include empty lines as well as comment lines beginning with the hash character (#). If the same keyword exists many times in the file then the last occurrence of the keyword is used.

Below, an example of the contents of a (minimal) configuration file is given. An easy way to get acquainted with the user interface is by studying the example cases included in the installation package.

5. FDS2FEM configuration file keyword reference

FDS2FEM uses a configuration file based user interface where commands are given using keyword-value pairs. Mandatory keywords and their arguments are described in what follows.

- **`fds_input`**
  This is a mandatory keyword. The argument, complete name of the FDS input file, should also be given.

- **`fem_input`**
  This is a mandatory keyword. The argument, complete name of the ABAQUS input file, should also be given. Note that this option need not be given if FDS2FEM is only used for FDS node coordinate and data dumps (i.e. .

- **`quantity`**
  This is a mandatory keyword that defines the quantity that is extracted from FDS to be used as a boundary condition in ABAQUS. Either of the two possible values should be given.

- **`fds_output`**
  This is a mandatory keyword that defines what type of FDS output is read. The arguments refer to device (DEVC) and boundary files (BNDF).

Optional keywords and their arguments are described below. These are used to control the model matching and mapping algorithms and to request output.

- **`nset_input`**
  With this keyword, the user can provide FDS2FEM with a connectivity map associating FDS devices or boundary patches with ABAQUS node sets. This connectivity map is given in a separate file. The name of this file is a mandatory argument for this keyword. Each line of the connectivity file should begin with the complete name of an ABAQUS node set (i.e. containing both the part instance name and the node set name separated by a period). After the node set name, there should be a space-delimited list of FDS device names (either device ID or number) or boundary patch numbers. Each ABAQUS node set will then receive boundary condition data from the corresponding FDS devices or patches. If no devices or patches are listed, all available FDS data is used.
- **`dump_fds_xyz`**
  The default value of this keyword is , i.e., the coordinates of the FDS nodes are not dumped to the hard drive using either XYZ or VTK-format.
  `dump_fem_xyz`
  The default value of this keyword is , i.e., the coordinates of the ABAQUS nodes are not dumped to the hard drive using either XYZ or VTK-format.
  `dump_fds_data`
  The default value of this keyword is , i.e., relevant FDS data is not dumped to the hard drive using either SDF or VTK-format.
- **`dump_fem_data`**
  The default value of this keyword is , i.e., relevant ABAQUS data is not dumped to the hard drive using either SDF or VTK-format.
  `fds_statistics`

The default value of this keyword is , i.e., no information on the FDS node set statistics is printed. This statistics might be useful when setting the mapping method parameters. Information on the average nearest neighbour distances and FDS node set coordinate bounds are given.

- **`fem_statistics`**
  The default value of this keyword is , i.e., no information on the ABAQUS node set statistics is printed. This statistics might be useful when setting the mapping method parameters. Information on the average nearest neighbour distances and ABAQUS node set coordinate bounds are given.
- **`match_translate`**
  The default value of this keyword is , i.e., no translation of the node coordinates is done before the mapping. The translation is performed before the (optional) rotation. If the argument is given, the origins of both the FDS and ABAQUS model are translated to user-given locations. If the argument is given, the model matching algorithm translates the center of mass of node coordinates of both models to the origin of the global coordinate system.
- **`match_rotate`**
  The default value of this keyword is , i.e., no rotation of the ABAQUS node coordinates is done before the mapping. The rotation is performed after the (optional) translation. If the argument is given, the ABAQUS node coordinates are rotated based on user-given Euler angles. If the argument is given, the model matching algorithm tries to match the node coordinate sets by rotating the ABAQUS node coordinates around the z-axis of the global coordinate system. It is assumed that the positive z-axis of the Cartesian coordinate system of both FDS and ABAQUS models is pointing at the same direction. It should be noted that in any case only the ABAQUS node coordinates are rotated.
- **`origin_fds`**
  Reference point for the FDS geometry. Default value is the origin of the global coordinate system. If this keyword is given, the three real arguments are also needed. The manual translation operation sets this point as the new origin of the FDS model.
- **`origin_abqs`**
  Reference point for the ABAQUS geometry. Default value is the origin of the global coordinate system. If this keyword is given, the three real arguments are also needed. The manual translation operation sets this point as the new origin of the ABAQUS model. The optional rotation operation is done with respect to the global coordinate system.
- **`euler_alpha`**
  Euler angle α for the manual rotation of ABAQUS node coordinates. Angle α is given in degrees. The default value is zero degrees.
- **`euler_beta`**
  Euler angle β for the manual rotation of ABAQUS node coordinates. Angle β is given in degrees. The default is value is zero degrees.
- **`euler_gamma`**
  Euler angle γ for the manual rotation of ABAQUS node coordinates. Angle γ is given in degrees. The default value is zero degrees.

- **`mapping {`**
  This keyword is used to choose the desired mapping method or to set mapping off. If mapping is omitted, FDS2FEM can still be used to dump FDS node coordinates, FDS output data and ABAQUS

node coordinates in a selected output format (e.g. VTK-format). The different mapping methods are explained in a separate section below. Only a short description of the mapping-related keywords is given in this list.

- **`mp_deg`**
  The inverse power (real number, default value is 2.0) of the distance dependence of the weights of the FDS node coordinates used in the interpolation algorithm. The nearest neighbour FDS node coordinates have weights decaying as the inverse power of the distance to the ABAQUS node coordinate, i.e., the weights are proportional to 1/(distance)^power. If just a simple arithmetic mean of the neighbouring FDS data points is wanted, the power should be set equal to zero.
- **`mp_n`**
  Number of nearest neighbour FDS nodes (an integer) of an ABAQUS node used in both k-nearest mapping algorithms. For a simple radius mapping algorithm this keyword is not used. The default number of nodes is four. At least this many nearest neighbour FDS nodes are tried to be used by the mapping algorithm. In some cases it might be that there are not this many applicable nodes available and in such a case the mapping is done with fewer points.
- **`mp_nmx`**
  Maximum number of nearest neighbour FDS nodes (an integer) of an ABAQUS node used in both k-nearest mapping algorithms. For a simple radius mapping algorithm this keyword is not used. The default maximum number of nodes is eight. At most this many nearest neighbour FDS nodes are used by the mapping algorithm. The mapping uses more than nodes if there are nodes that are at an equal distance or at a slightly longer distance than the node furthest of the nodes from the ABAQUS node. The slightly longer measure is given by the keyword .
- **`mp_cut`**
  This is used to give either the cut-off radius for the simple radius mapping or the optional cut-off radius for the k-nearest mapping. Radius is given as a real number in metres in both cases and the default value is zero, which means that no cut-off radius is used in the k-nearest mapping method. For the simple radius mapping a positive real number must be given and all the FDS data within this radius is used in data mapping. The k-nearest mapping method uses also the parameter to extend the nearest neighbour search radius a little bit to find close outlier points that are also included in the mapping. The actual cut-off radius will be shorter than the given if there are or more points found within the given radius, i.e., is the maximum cut-off radius.
- **`mp_del`**
  A tolerance criterion (a real number, default is 0.1) for the radius of the furthest nearest neighbour node used in the k-nearest mapping algorithms. For a simple radius mapping algorithm this keyword is not used. The search radius is extended by this factor, i.e, a new search radius is (1 +) × the old search radius that is decided using and information. See also: .

6. Mapping methods

6.1 Overview

FDS2FEM is used to transfer thermal boundary conditions (surface temperature or heat flux in the form of adiabatic surface temperature) from FDS to ABAQUS. There are three main ways to generate ABAQUS boundary conditions from FDS output data. These are called *mapping methods*. They are set up using the keyword.

- **`mapping off`:** No mapping. This option can be used e.g. when the user wants to dump FDS node coordinates or output data in some other format than the FDS default (e.g. VTK-format). ABAQUS input can be omitted.
- **`mapping devc_to_nset`:** Direct device-to-nset mapping. In this mapping method, one or more ABAQUS node sets receive data each from one or more FDS devices. If only one device is connected to a node set, the data is transferred "as is". Otherwise, an arithmetic mean is used. All of the nodes in a node set receive the same data. Device-to-nset mapping requires the use of a connectivity file (described in Chapter 5).
- **`mapping nearest`:** Nearest neighbour mapping. This is the default mapping method used in FDS2FEM. In this method, each ABAQUS node (target) receives data from one or more FDS nodes (sources). Each source node can be given a weight based on the source-target distance. The number of source nodes can be controlled based on number and distance limits. In essence, this is a weighted average method. However, it allows for considerable tweaking using the five control parameters (, , , and). Nearest neighbour mapping can be used with both device and boundary file output from FDS.

6.2 Nearest neighbour mapping

The mapping method can be done as a global mapping or as a ABAQUS surface set based local mapping. If no surface set connectivity is defined (no keyword given) then a global mapping is done. For each ABAQUS node point that is demanding input data from the FDS calculation, all FDS data points are considered, when nearest points are search for. If the user has defined NSET to DEVC/BNDF connectivity using to give the connectivity file name then just those FDS data points are used that are connected to the current surface set (NSET). Below three different interpolation methods that can be used in the mapping are described. This description is valid both for the global and local mapping, if you keep in mind that for local mapping just those FDS data points are included that correspond to the surface set in question.

There exists three different ways to define the mapping neighbourhood for the mapping method "nearest". All of these ways decide for each ABAQUS node point the corresponding FDS node points, whose data will be used in the data interpolation. The three different ways are doing the data interpolation similarly, but the FDS data points used in the interpolation will vary. The interpolation is just a distance weighted average, where the distance dependence is given as a power law:

$$w_{\mathrm{FDS}}^{i} = \frac{c}{\left|r_{\mathrm{FDS}}^{i} - r_{\mathrm{FEM}}^{j}\right|^{p}} ,$$

where c is a normalization constant and p is the power given with the option and

$$\sum_{i=1}^{n} w_{\mathrm{FDS}}^{i} = 1 .$$

Just those FDS points (i=1,…,n) are used that belong to the mapping neighbourhood. Vector $\boldsymbol{r}^{j}_{\mathrm{FEM}}$ indicates the location the j^{th} ABAQUS node, where a surface temperature or adiabatic surface temperature is interpolated using the data on the FDS node points, whose coordinates are $\boldsymbol{r}^{i}_{\mathrm{FDS}}$. There are three different types of neighbourhood methods available to map the neighbouring FDS node points that are used in the interpolation for each FEM node point. These methods are

- **Simple inside-radius mapping:** This method is chosen, if the keyword has a value of zero or less. For positive values, some other type of mapping is used. Here a user given radius, , is used to decide if a FDS node is included or excluded when doing the interpolation. Only points within the specified radius are used. When using this method, only the keywords and affect the interpolation. The other mapping-related keywords are ignored. If no FDS nodes are found within the given cut-off radius, then the programs stops and prints an error message. It is the user's responsibility to give a large enough cut-off radius.
- **k-nearest mapping:** Here the number of nearest neighbour FDS nodes, , used in the interpolation is given as user input. This parameter dictates that at least the closest neighbours are used in the interpolation, but there might be up to nodes used in the interpolation, if these additional points are just a little bit further away than the last "good" neighbour, which is the neighbour number . The "little bit further" means that these close outliers are within a sphere that has (1 +)*(radial distance of the neighbour point) radius around the considered FEM node. The default value for is 0.1. This mapping method will always find at least one nearest neighbour so it is quite robust. (Well, at least neighbours are found if there are at least so many FDS nodes.)
- **k-nearest mapping with a cut-off radius:** Like above, but the user given cut-off radius is also used when looking for the neighbouring FDS nodes. The parameters and behave in a similar way as in the k-nearest mapping method. If or more FDS nodes are found within the given radius, then this method is same as the k-nearest mapping method described above. If there are fewer than neighbours within the cut-off radius , then some close outliers are tried to find up to the radial distance (1 +)* . The cut-off radius might be so small that no neighbours are found, but in this case the programme uses the closest neighbour data even though it might be far away in order to make the method suitable for use with batch jobs, so that the programme does not stop.

The default values of the mapping-related parameters are given in Table 1. It should be noted, that by default a k-nearest mapping without cut-off radius is performed using four nearest neighbours and a close outlier factor of 0.1. Also, at most eight points are used in total (the neighbours and the close outliers).

The close outlier factor is used in the mapping method, because quite often there are many neighbouring points at the same radial distance from the FEM node, especially when using uniform meshes. Without the extension some of these equal neighbours would be included in and some excluded from the interpolation. The default values given in Table 1 are not the best ones to be used. Finding optimal (or good enough) parameters for the mapping method requires some thought and experimenting from the user.

The inverse power dependence of the weights means that if a FEM node is exactly at the same location as an FDS node, no interpolation is done, the value of this FDS data point is used directly due to the mathematical form of the weight. If this kind of behaviour is not wanted then a zero power could be given and then the interpolation is transformed into a simple arithmetic mean of the neighbouring FDS nodes. In future versions of this programme, a user input should be added that would limit the weight a single point can have.

A schematic diagram of the mapping method is shown in Figure 2. If the simple radius mapping method is used, just the (eight) nodes within the solid circle are used. If the k-nearest mapping with a cut-off radius r is used with, e.g., and , then all points inside the dashed circle are used ( ). The radius of the solid circle is the smaller of the two: user given cut-off radius or the distance of the closest neighbouring node. The additional search for neighbours within $r + dr$ is used, because otherwise some of the nodes at the solid circle might not be included in the interpolation depending on the number of neighbours used. The k-nearest neighbours without a cut-off radius is similar, there just the radius of the closest neighbour is used always. The additional dr search could be avoided by giving .

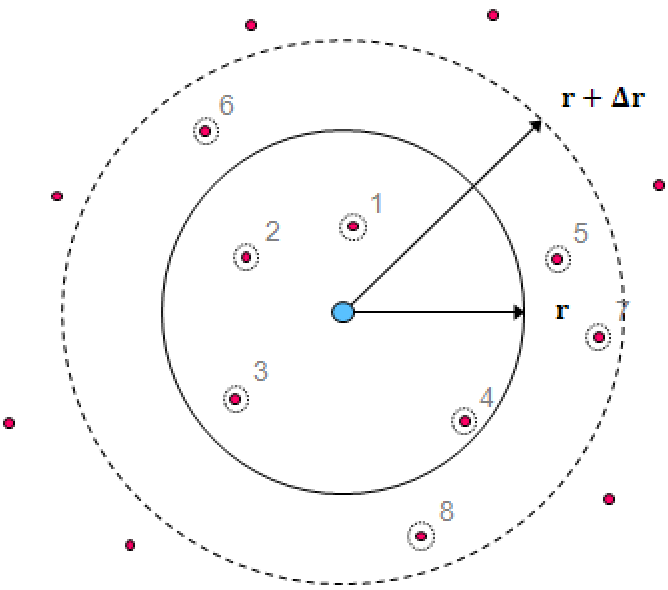

Figure 2. Schematic diagram of the nearest neighbour mapping method

Which method to use? This depends on how well your FDS geometry matches your FEM geometry. If the geometries match well, e.g., the shape and dimensions of the objects are nearly identical, then a k-nearest mapping with or without a cut-off radius with should work quite well. The cut-off radius could be used if there are different objects close by, whose data is mapped. The parameter should be larger than the number of nearest neighbours in the FDS mesh, i.e., larger or equal to four (a plane is mapped). This way, just the points directly neighbouring the FEM nodes are used in the interpolation. If there are considerable deviations between the FDS and FEM models, more averaging could be desirable so that a larger neighbourhood should be defined and an optional cut-off radius could be used so that the interpolation does not use too distant neighbours, which could be the case for a corner point, which has just a few neighbours close by.

FDS uses a simple meshing strategy with rectilinear cells and many non-overlapping meshes may be used, e.g. a finer mesh close to the fire and coarser meshes further away. The present version of fds2fem supports multiple FDS meshes for BNDF data only when no connectivity file is given (all FDS points are used for each ABAQUS node point) or the BNDF output is generated just within the first mesh defined in the FDS input file. The usage of the DEVC output does not have any FDS meshing restrictions. The practical cell sizes that are used in cases, where a fire-related structural analysis would be done, are in the order of 10 cm to 20 cm close to the fire origin and can be much larger, e.g., 1.0 m, due to limitations in computational resources. The FEM model of the studied structural element will probably have much finer computational mesh and the shape of the element may be quite different from what is possible to represent using the relatively coarse rectangular mesh cells that are used in the FDS model. Even in this case, fds2fem might be useful, because usually far away from the fire source the hot smoky gas layer is heating the structures and a structural member (or a part of it) could be imagined to be in a spatially homogeneous surrounding when the adiabatic surface temperature is not depending much on the fine details of the geometry.

7. Example cases

The installation package contains example cases of the FDS-ABAQUS coupling procedure. These include the FDS and ABAQUS input files as well as the corresponding FDS2FEM configuration file. Each example case also contains instructions (file *info.txt*) on how to run the case. By reading the comments in the FDS input files and the configuration files the user should learn basic usage of FDS2FEM quite easily.

The current version includes five example cases that are described below. The user is encouraged to run all the example cases, make modifications to the mapping options and examine the results to get acquainted with the subtleties of mesh mapping. FDS and ABAQUS data can be dumped using the keywords and . VTK-format output together with some scientific visualization software, e.g. ParaView (http://www.paraview.org), can be used to visualize the data prior to and after mapping.

The examples described below use surface temperature as the thermal boundary condition. It would be straightforward for the user to modify the cases to use adiabatic surface temperature instead. This would allow for an interesting comparison between results obtained with and without the effects of the FDS heat conduction solver.

7.1 Case 1 (case_1)

FDS model. A simple oblong room (like a train car) with doors at both ends and at one side. The steel doors are heated by a radiator plate on the floor. FDS measures and outputs WALL TEMPERATURE and ADIABATIC SURFACE TEMPERATURE using both devices and boundary files.

ABAQUS model. A simple steel door made of linear brick elements. The same FEM model can be used for all of the three doors with different thermal loads.

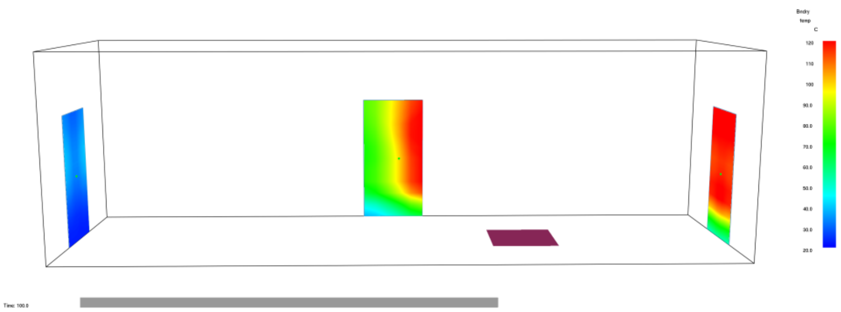

Figure 3. Snapshot of case_1 FDS wall temperature data

7.2 Case 2 (case_tc1_1)

FDS and ABAQUS models. A metal-insulator chess board that is heated with a 100 kW/m^2 burner centrally located. There are two different mapping cases, where one maps each metal and insulator patch in FDS model to the corresponding FEM surface set. This way the mapping algorithm keeps the metal and insulator data separate, when the surface temperatures are mapped. The other mapping case does not use the information on the patches so all FDS data points are used so that the points close to the metal-insulator boundary will use both the insulator and metal data during the mapping, so the sharp surface temperature boundaries in the FDS model are a little bit smeared (averaged) out in the mapping process. Similar effects are not so pronounced when transferring the adiabatic surface temperatures due to their nature. The adiabatic surface temperature is not explicitly dependent on the properties of the surface material like the surface temperature. But there might still be sharp and steep gradients in the adiabatic surface temperature output of FDS calculation due to the radiation, there might be sharp shadows and the mapping might smear out these sharp gradients due to the averaging nature of the mapping process. The user can control the amount of averaging done by the mapping algorithm by changing the corresponding parameters, e.g., making the mapping region smaller and/or increaseing the distance mapping power.

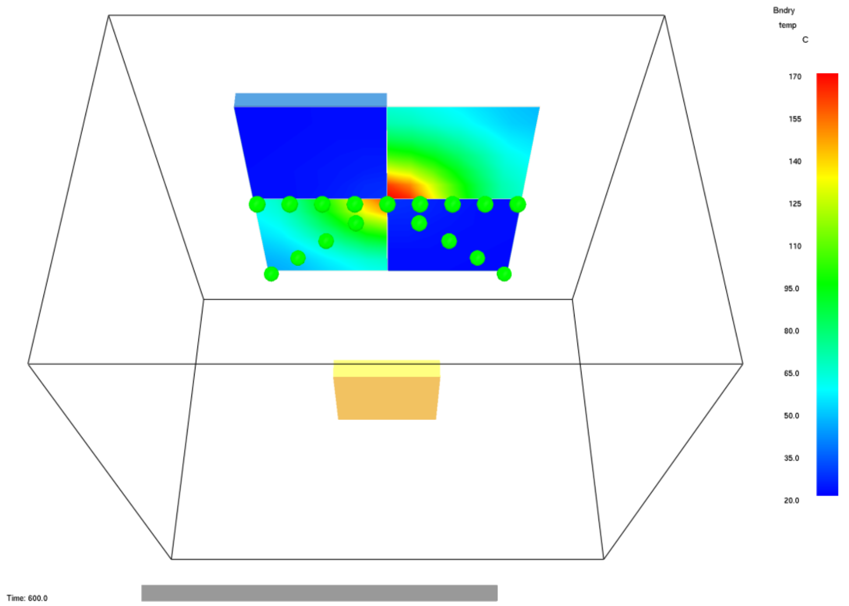

Figure 3. Snapshot of case_tc1_1 FDS wall temperature data

7.3 Case 3 (case_tc3_2)

FDS and ABAQUS models. A solid plate with a square hole. The plate is centrally heated by radiation from a hot plate. The FEM mesh is not symmetric. In this example case, the coordinates of the FDS and FEM models do not match and manual model matching is performed by giving the translational and rotational information that is needed to match the coordinates.

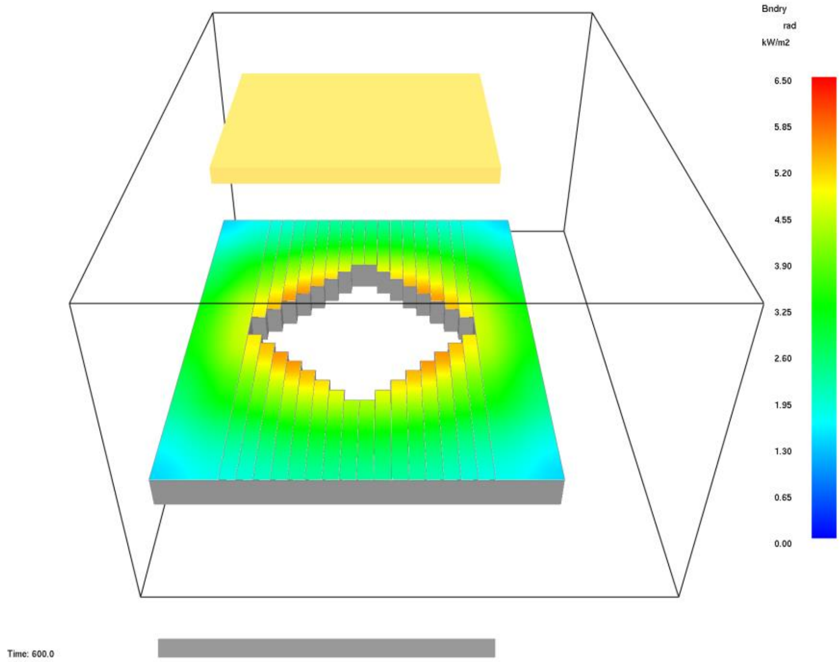

Figure 4. Snapshot of case_tc3_2 FDS wall temperature data

7.4 Case 4 (case_tc4_2)

FDS and ABAQUS models. A square hollow section of a steel column (0.3 m × 0.3 m × 3.0 m) heated by radiation from a hot surface. The correct node set of the FEM model should be given to FDS2FEM so that just the outside surface of the FEM model gets the boundary condition from the FDS calculation. The FEM input file defines all four faces of the column as a single node set, so a mapping from a certain face to the corresponding face cannot be made. If a better mapping would be needed then the FEM model input should be made so that the different faces will have different node sets defined.

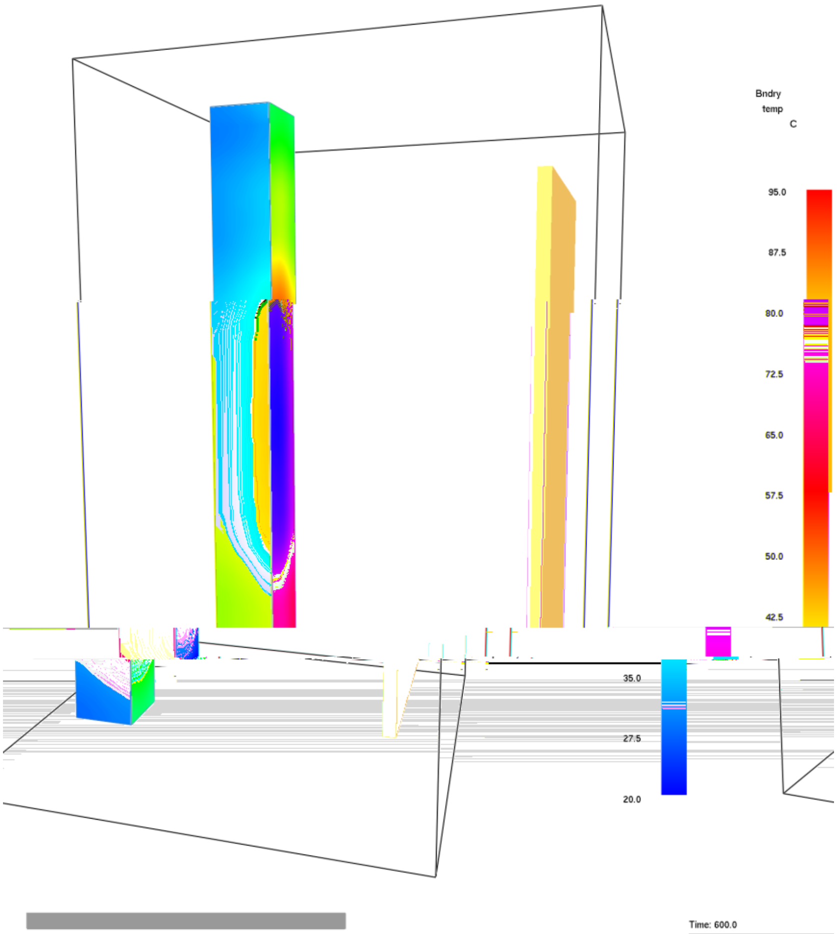

Figure 5. Snapshot of case_tc4_2 and case_tc5_2 FDS wall temperature data

7.5 Case 5 (case_tc5_2)

FDS and ABAQUS models. Similar to the previous case, but now the FEM model is a steel column with a circular cross section and the FDS model is a square column. The FDS part of this case is exactly the same as in the case_tc4_2, because the cross sections of the beams are modelled by single cells that are square shaped in FDS due to the rectilinear shape of the FDS meshes and obstructions.

8. Source code

A brief introduction to the FDS2FEM source code is given in this chapter. Below is the list of the source code files and a short description of their contents. The most important file for the user is . It countains the module that is used to read user input and to set default values for configuration variables. The mesh mapping module can be found in file The programming language is free format Fortran 90/95.

fem_dump.f	ABAQUS data dump
fem_output.f	modification of ABAQUS input file
fem_reader.f	parsing ABAQUS input file
fem_stats.f	calculating ABAQUS node set statistics
cfg_reader.f	parsing configuration file
error.f	general error messages
fds_dump.f	FDS data dump
fds_reader.f	parsing FDS input and output files
fds_stats.f	calculating FDS node set statistics
global.f	global constants, variables and arrays
main.f	main program
mapping.f	mapping data between FDS and ABAQUS node sets
matching.f	matching model coordinates
math.f	general mathematics
misc.f	miscellaneous
string.f	general string handling