SAFIR

Users manual

Version 2011.b.0

This document explains the introduction of data for the software SAFIR developed at the University of Liege for the simulation of building structures subjected to fire.

Only the thermal part of SAFIR is considered in this document (still under construction).

This document is organized in the manner specified in the standard ISO/FDIS 16730:2008(E)

"*Fire safety engineering - Assessment, verification and validation of calculation methods*"

To date (23 September 2011), this document has not been through third party auditing

A. Description of SAFIR

A.1 Description of the model

The underlying model is that in which heat is distributed in the structure essentially by conduction because most of construction elements are made of solid. Radiation and convection are the heat transfer modes in internal cavities such as those present, for example, in hollow core slabs. Convection and radiation are the heat transfer mode at the boundaries between the object analyzed and the environment, i.e. the fire.

A.2 Description of the basic processing tasks performed - calculation methods and procedures performed

SAFIR first reads the whole input file and copies it in the output file.

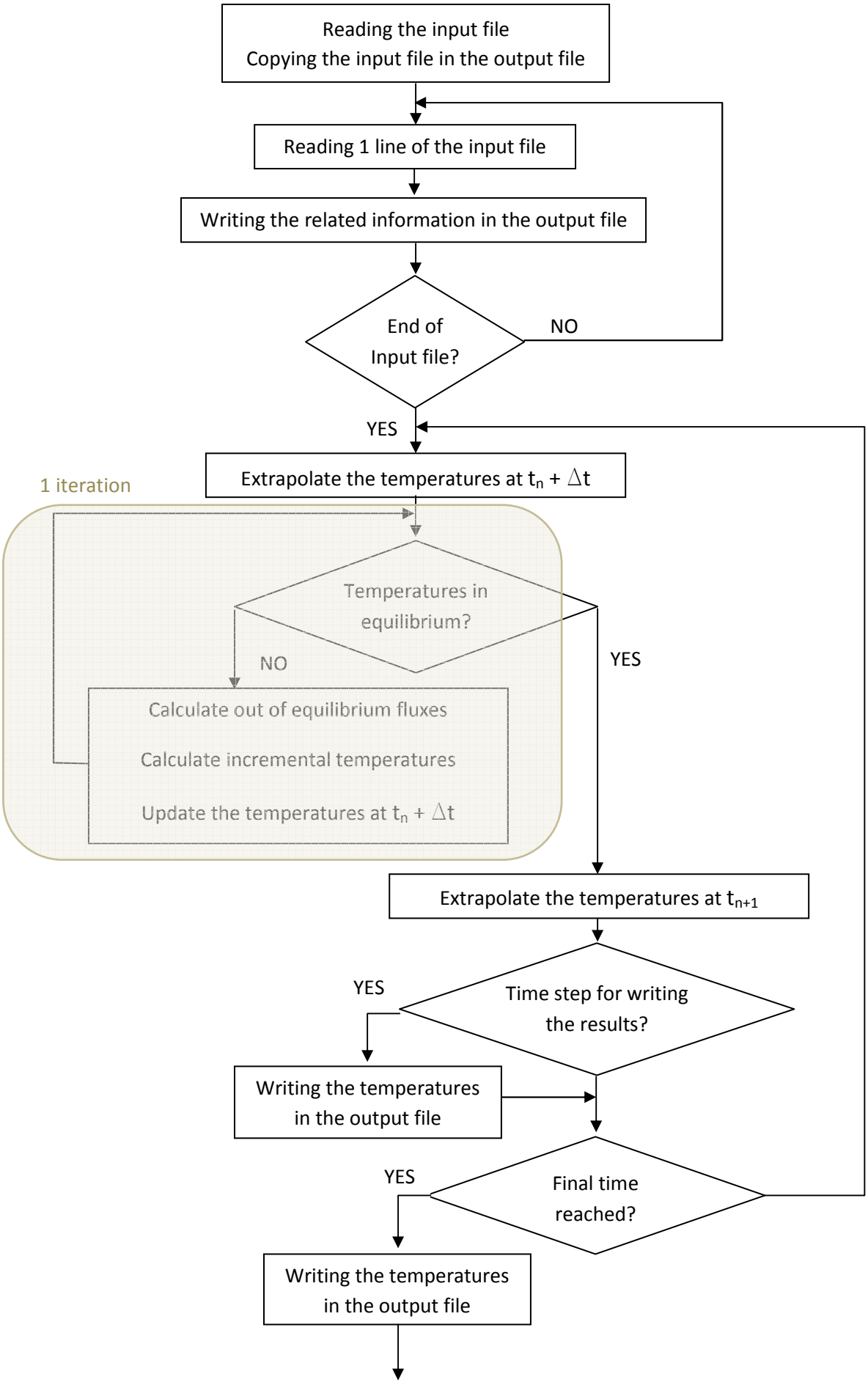
Then it reads all lines of the input file again, one by one, tries to interpret the data that are found in each line and prints the interpreted results in the output file before going to the next line.

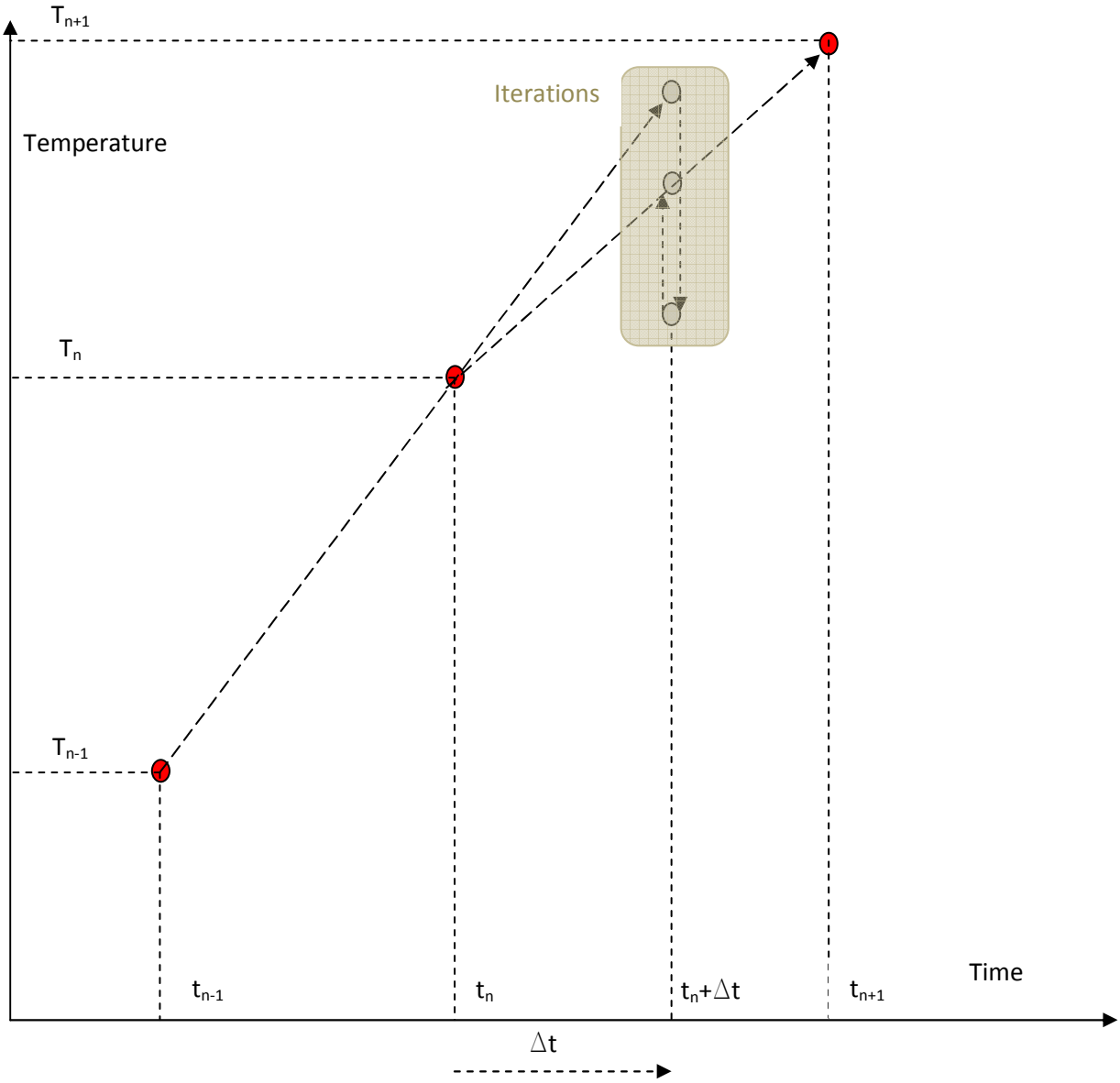
Then SAFIR will calculate the temperature in the structure at the successive time steps chosen by the user. For each time step:

- SAFIR first extrapolates the temperatures from two the previous time steps, t_{n-1} and t_n, to a first approximation at a given time $t_n + \Delta t$[1] comprised between t_n and t_{n+1}. The temperatures at this stage are not in equilibrium.
- Then SAFIR refines this first approximation of the temperatures at $t_n + \Delta t$ until thermal equilibrium is restored. Several iterations are required to restore the equilibrium up to an accepted degree of precision[2].
- SAFIR finally extrapolates the temperatures at t_{n+1} from the values at t_n and $t_n + \Delta t$.
- If the time step is one of the time steps defined for printing the results, see TIMEPRINT command, the temperatures at the nodes are printed in the output file. Temperatures are also eventually written in the .TEM or .TSH file to be used in a subsequent analysis.

[1] The value of Δt is governed in the data by the parameter TETA = $\Delta t/(t_{n+1}-t_n)$

[2] The degree of precision required before thermal equilibrium is considered as acceptable is governed in the data by the parameter PRECISION.


Reading the input file
Copying the input file in the output file
Reading 1 line of the input file
Writing the related information in the output file
End of
Input file?
NO
YES
Extrapolate the temperatures at $t_n + \Delta t$
1 iteration
Temperatures in
equilibrium?
NO
YES
Calculate out of equilibrium fluxes
Calculate incremental temperatures
Update the temperatures at $t_n + \Delta t$
Extrapolate the temperatures at t_{n+1}
YES
Time step for writing
the results?
Writing the temperatures
in the output file
YES
Final time
reached?
Writing the temperatures
in the output file


T_{n+1}
Iterations
Temperature
T_n
T_{n-1}
t_{n-1}
t_n
$t_n+\Delta t$
t_{n+1}
Time
Δt

A. 3 Description of the types of skills required to execute typical runs

The user should be able to operate a P.C. computer, either a laptop or a desktop.

The user should be familiar with the Windows operating system.

The user should be familiar with the file naming convention based on a file name and a file type. He should be able to use the Explorer of Windows to move from one folder to another, to copy, delete, or rename files.

The user should be able to open files with a text editor, in order to prepare or modify input files and to examine the results produced with SAFIR.

In order to execute typical runs, the user should be familiar with the basic mechanisms of heat transfer such as conduction, convection and radiation. He should also be familiar with the finite element technique applied to solving heat transfer problems.

In order to judge from the relevant character of the results the user should be familiar with the behavior of structures subjected to fire. For example, a user that has all skills and experience mentioned above would be able to perform a numerical simulation of a section protected by a gypsum plaster board and subjected to a fire during 4 hours, only missing the fact that no gypsum plaster board can stay in place for such a long duration; it would disintegrate long before.

B. Installations and operating instructions

B.1 Minimum hardware configuration required

SAFIR can run on a P.C with a single processor. 1.6 GHz.

1 Go of RAM

120 Go Hard Disk

This has been tested on an ACER Aspire One ZG5 notebook.

SAFIR could **not** run on a Pentium II, 350 MHz, 384 Mo RAM, 37,2 Go ROM.

B.2 Computers on which SAFIR has been executed successfully

- Laptop HP Pavilion Entertainment PC
  Intel Corel2 CPU, 1.66 GHz, 2 Go RAM, 232 Go Hard Disk
- Dell Precision T5400
  Intel Xeon CPU E5420, 2.50 GHz,8 Go RAM, 300 Go Hard Disk
- Asus under Windows XP Professional Version 2002, Service Pack 3
  Intel Pentium 4, 2.60 GHz, 1 Go RAM, 29 Go Hard Disk
- ACER Aspire One ZG5
  Intel Atom 1.6 GHz, 1 Go RAM, 120 Go Hard Disk

B.3 Programming language, software operating system and version in use

SAFIR has been in written in FORTRAN:
Intel Visual Fortran Compiler Integration for Microsoft Visual Studio 2005, 11.1.3466.2005 under Microsoft Visual Studio 2005, version 8.0.50727.762

Some routines, written at the early days of development of the code, are in FORTRAN 77 while the most recent subroutines are in FORTRAN 90.

B.4 Instructions for installing the program

SAFIR is delivered in the form of an executable file, for example "SAFIR2009.exe", and a security file name “identity.key”. These files have to be copied on the device on which SAFIR will be found by the C.P.U of the computer during execution. This could be, for example, a USB key, an external hard disk or, more likely, one of the hard disk of the P.C. Both files must be located on the same device.

The file “identity.key” has to be located in the root directory of the device.

The file “SAFIRxxxx.exe” can be located in the folder of choice of the user. Two different options are possible:

1. Users normally organize their input files in different folders; one folder may for example be used for each different project. The executable can be copied in each of these folders, see Figure 1.

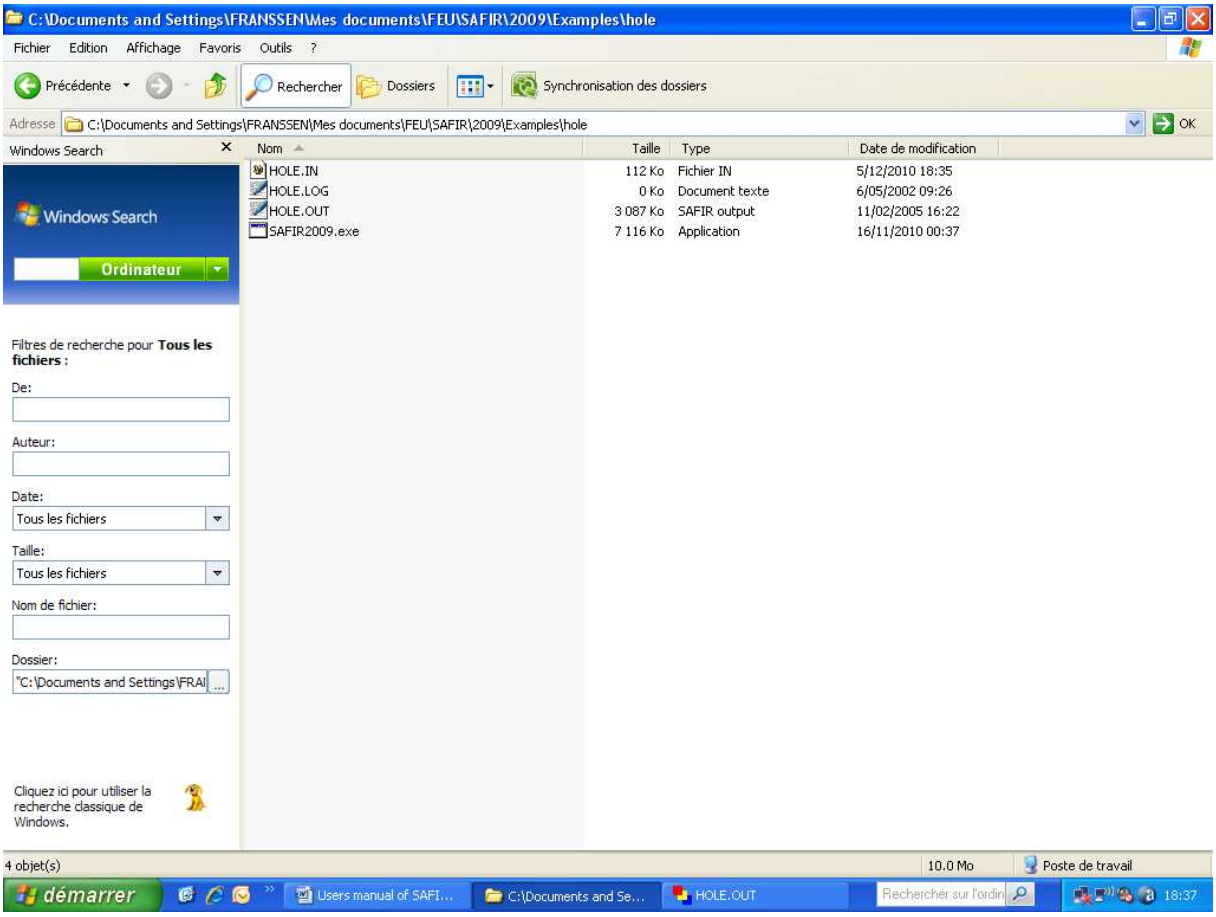


Figure 1: folder with the executable

To run SAFIR from Internet Explorer, double click on the executable file. SAFIR will open a DOS window in which the user is requested to type the file name of the input file ("`filename`" from the complete name "`filename.IN`"), see Figure 2.

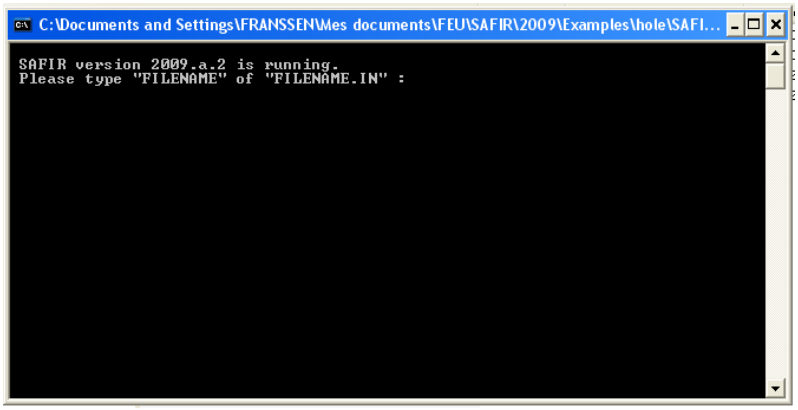


Figure 2: DOS window with the request for the file name

The user has to type the filename, see Figure 3, and press the ENTER key on the keyboard.

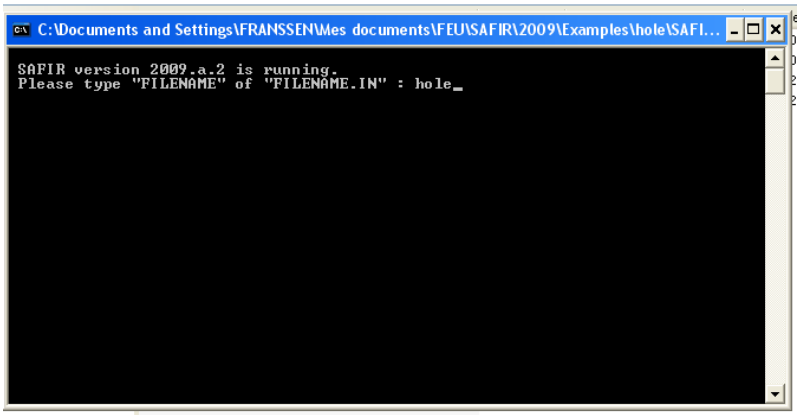


Figure 3: DOS window with the file name

SAFIR starts to run and some messages are written in the DOS window, which allows seeing the progress of the simulation, see Figure 4. When the simulation stops, the window is closed immediately[3].

[3] If an error message is written as the last line in the window, it is normally not possible to read it before the window closes. The same message is normally written at the end of the output file.

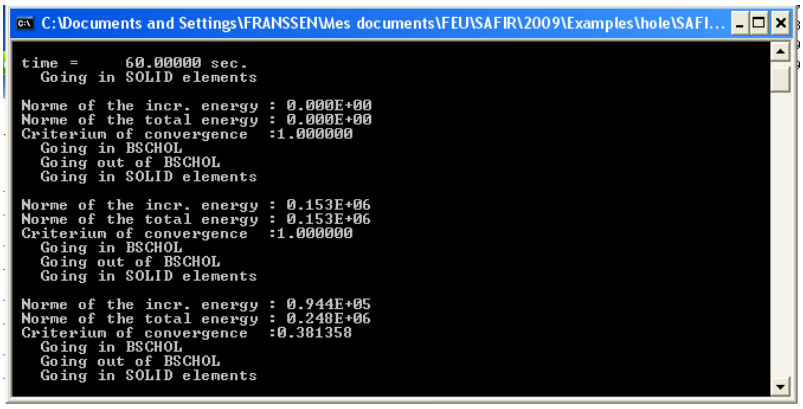


Figure 4: DOS window with the messages

2. SAFIR is copied in one single folder of the hard disk, for example in the folder "c:\program files\SAFIR\2009". SAFIR will then have to be run with the tool called "SafirShell". This tool is installed by execution of "SafirShell.msi" from the CD disk provided with SAFIR. When this tool is started, a window is opened, see Figure 5.

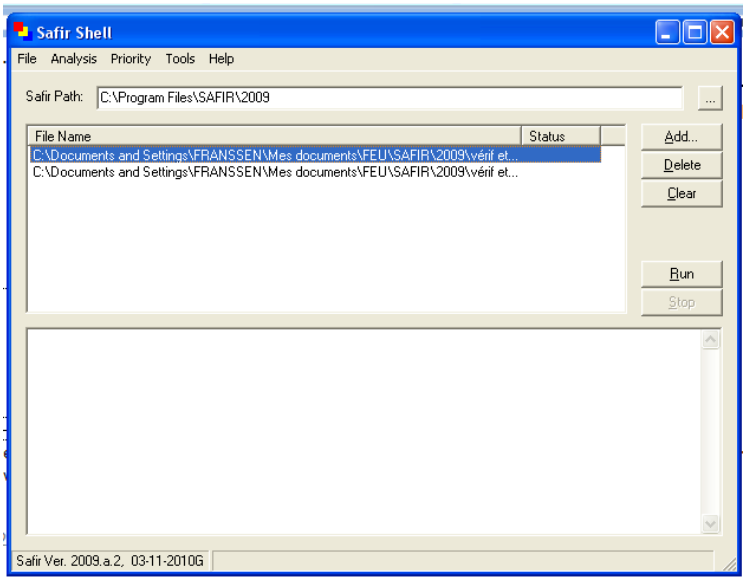


Figure 5: window of the SafirShell

The folder where the executable file of SAFIR is located is indicated in the "Safir Path" window. This folder can be selected by the user with the […] button on the top right.

The input files that will be treated by SAFIR can be chosen by the user using the [Add..] button. SAFIR will start running as soon as the [Run] button is clicked on.

The advantages of using the SafirShell tool are:

- The executable of SAFIR is present only in one folder on the hard disk.
- Several input files can be chosen and treated in a batch mode, one after the other. If one runs fails for any reason, the next input files will be treated.
- The messages that were written in the DOS window will now be written in the bottom window of the SafirShell and will remain there to be read even after the runs are finished.

- The priority allocated by the Operating System to the SAFIR process can be selected in the Priority scroll down menu. For processors with a single core, it is recommended to choose "below normal" because this will hardly affect the velocity of SAFIR but will allow time enough to be allocated for performing usual office tasks.

B.5 Typical personal time and setup time to perform a typical run

The typical personal time and setup time to perform a typical run (not counting the time required to gather all information about the structure to be analyzed is in the order of half a day for a 2D analysis and one day for a 3D analysis.

B.6 Estimation of the computer execution time for typical applications

The computer execution times given here have been taken from runs on a Laptop HP Pavilion Entertainment PC, Intel Corel2 CPU, 1.66 GHz, 2 Go RAM, 232 Go Hard Disk. The time step used for the simulations was 15 seconds. Much faster runs could be performed with little loss of precision if longer time steps would be used.

The duration of calculation mentioned are taken from the clock, i.e. difference between the time when the calculation is finished and the time when it started.

For 2D analyses.

One quarter of a reinforced concrete section, Figure 6, 144 nodes, 2 hours of fire

=> 3 seconds of simulation.

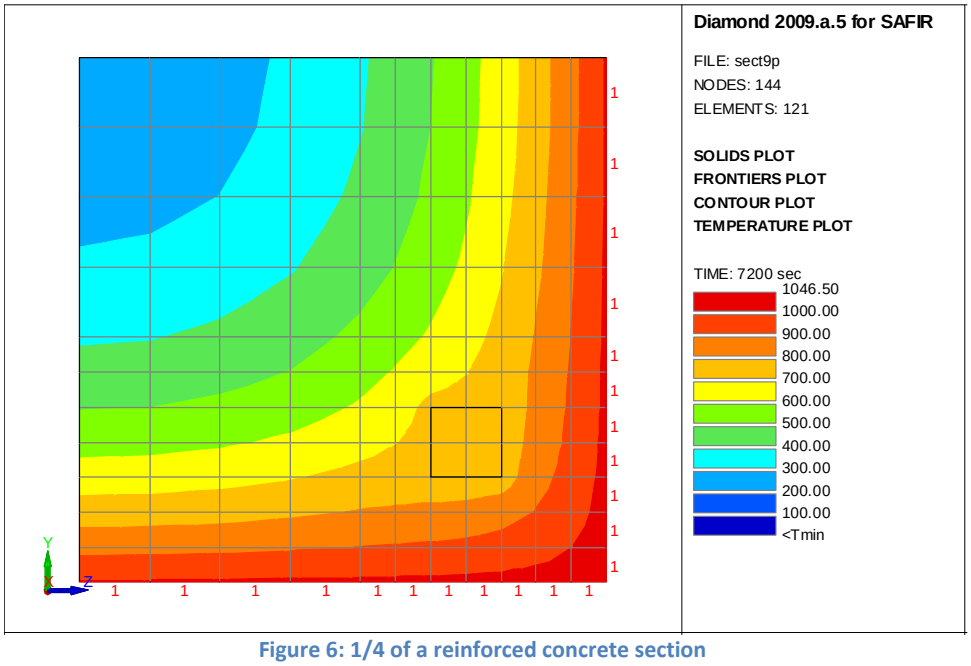


Figure 6: 1/4 of a reinforced concrete section

Composite steel-concrete beam, Figure 7, 932 nodes, 2 hours of fire

=> 21 seconds of simulation

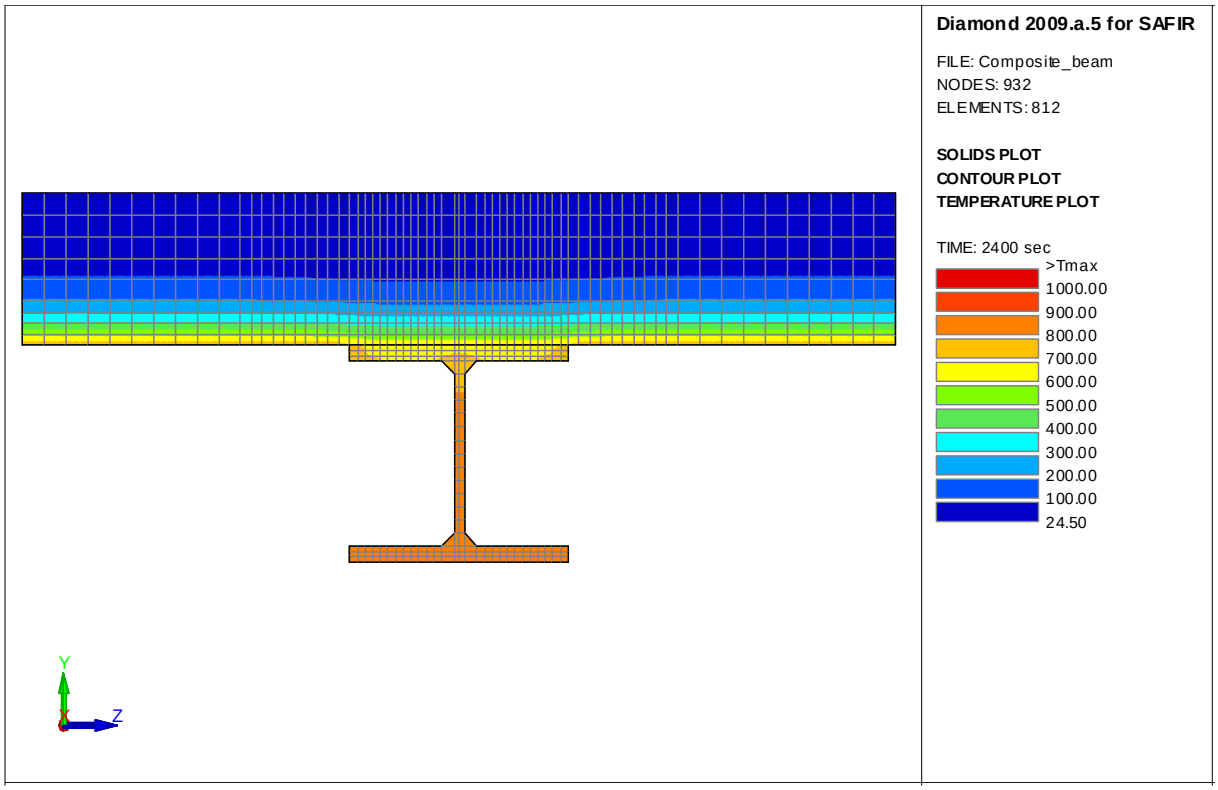


Figure 7: composite steel-concrete beam

Circular concrete section[4], Figure 6, 1051 nodes, 2 hours of fire

=> 33 seconds of simulation.

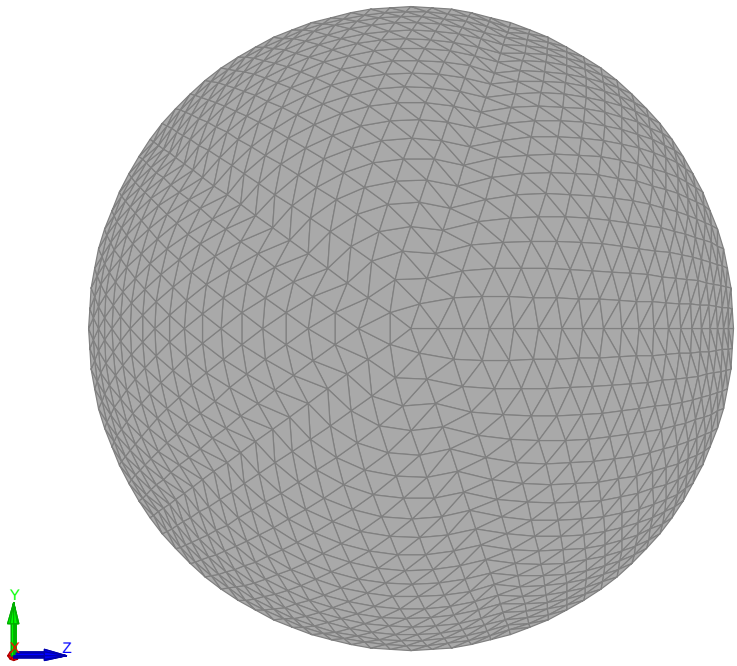


Figure 8: circular concrete section

Concrete bridge deck[5], Figure 9, with 24108 nodes, 120 minutes of fire

=> 9 minutes 7 seconds of simulation.

Figure 9: bridge deck

[4] Model created with GID by Eric Tonicello, MP Ingénieurs Conseils
[5] Model created with GID by Eric Tonicello, MP Ingénieurs Conseils

For 3D analyses.

Composite joint with a crude discretisation, Figure 10, 2915 nodes, 120 minutes of fire => 1 minute 5 seconds of simulation

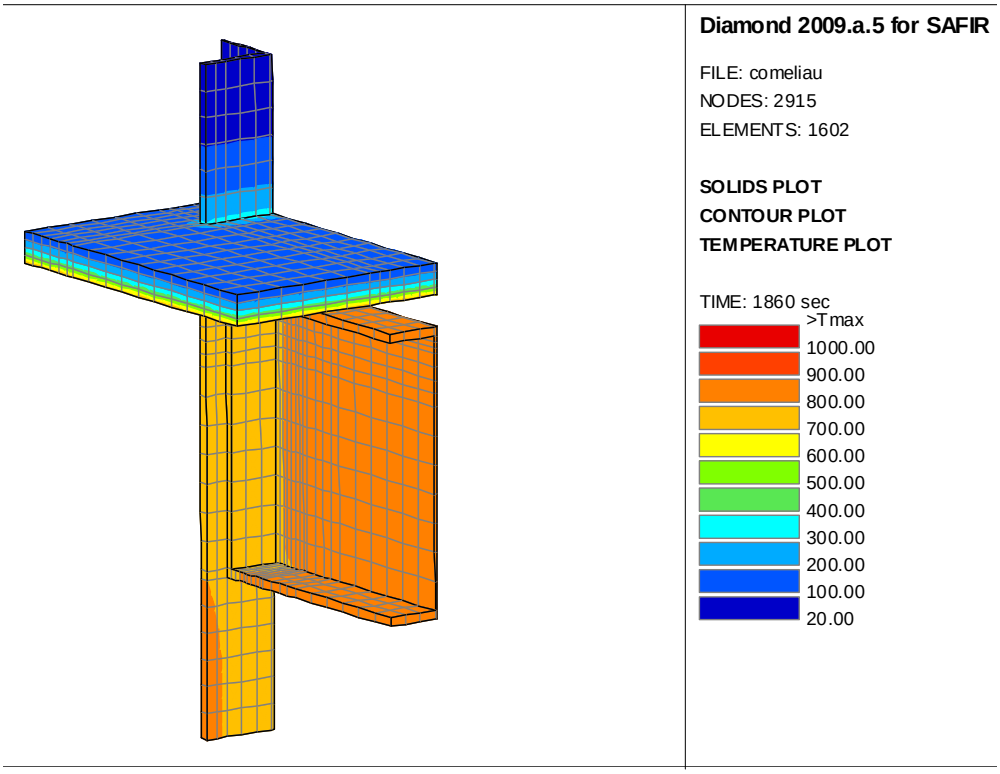


Figure 10: composite joint (crude mesh)

Concrete beam with a circular cavity in the web, 6183 nodes, 120 minutes of fire

=> 7 minutes 31 seconds of simulation

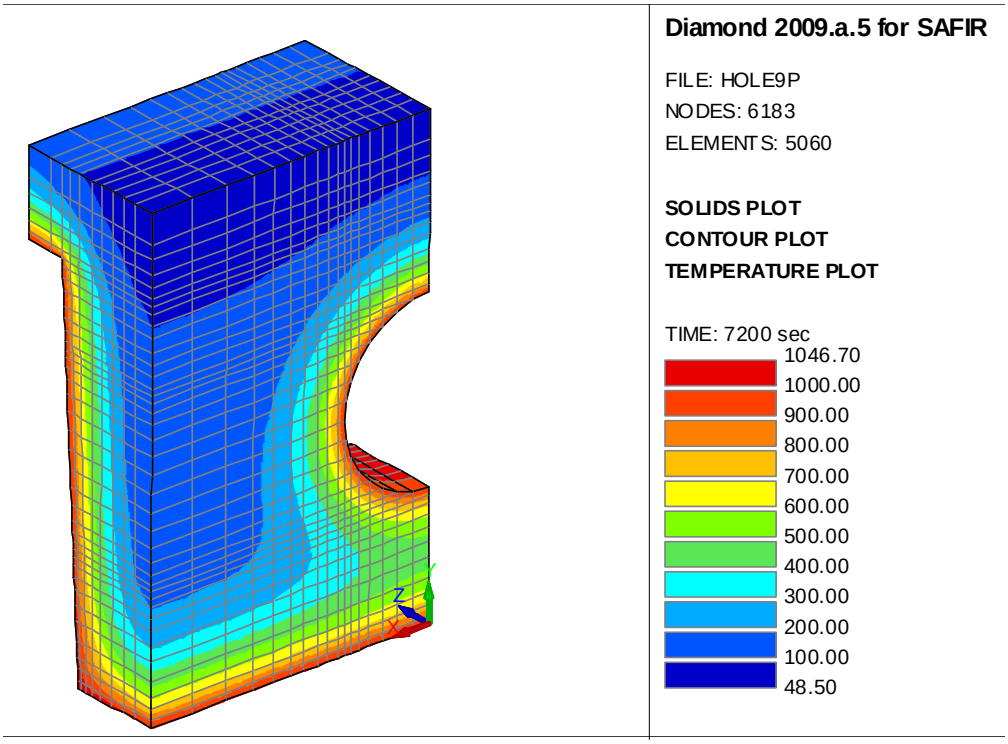


Figure 11: concrete beam with a circular cavity in the web

Composite joint[6], Figure 10, 31502 nodes, 120 minutes of fire

=> 48 minutes 47 seconds of simulation

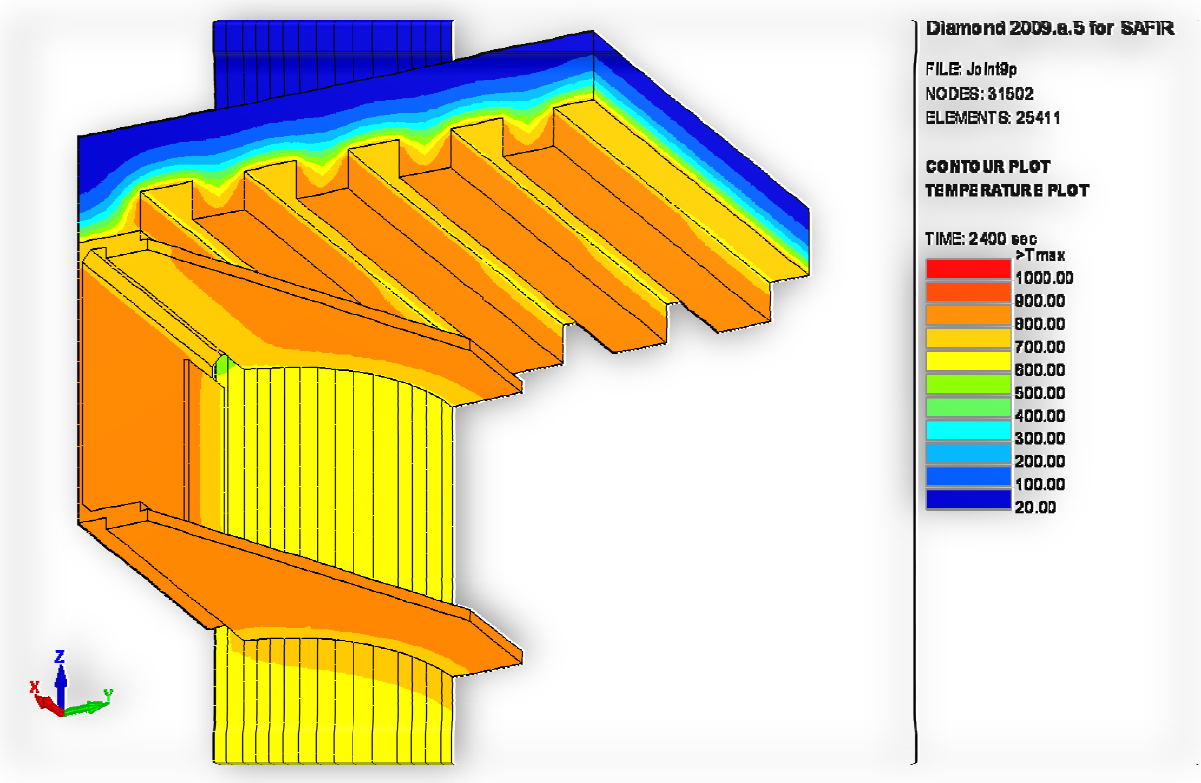


Figure 12: composite steel-concrete joint

[6] Model by Elisabetta Alderighi, Univ. of Pisa.

Figure 13 shows the CPU time on the computer utilized, as a function of the number of nodes, for 120 minutes of simulation when a time step of 15 seconds is used. Shorter times could be obtained by reducing the fire duration or using longer time steps.

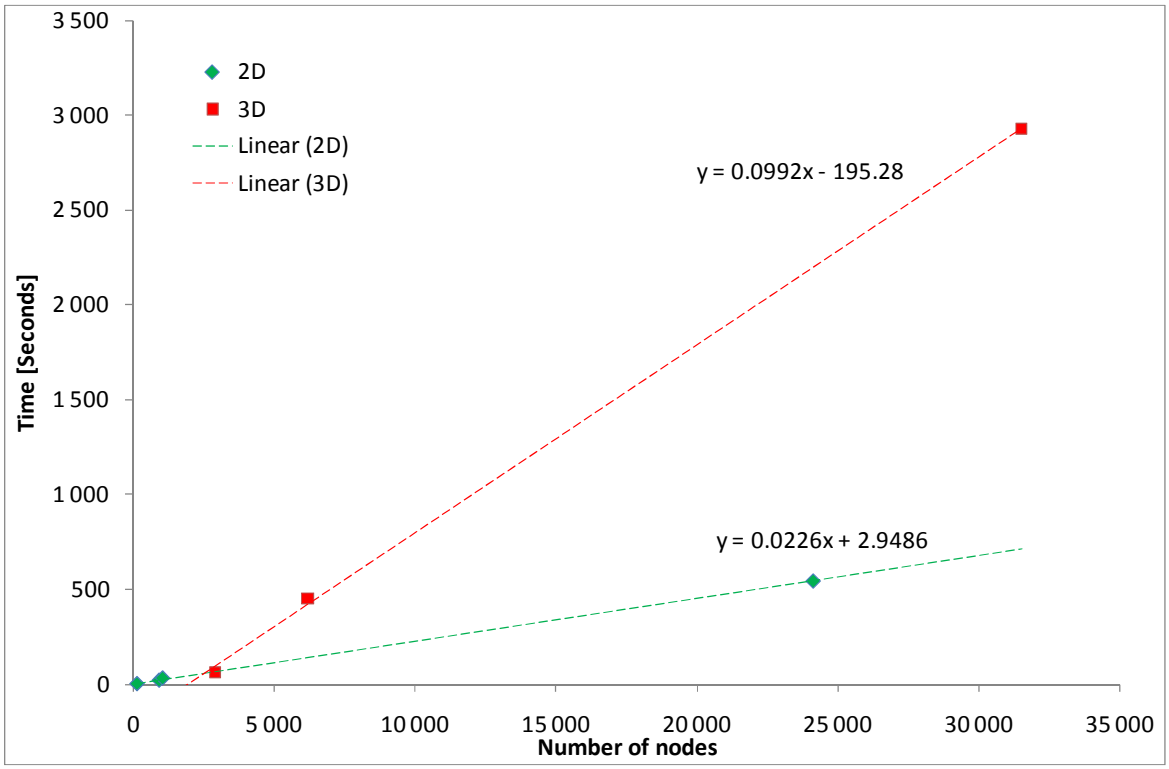


Figure 13: CPU time as a function of nodes

C. Program considerations

C.1 Major options available

The first option that the user has to take depends on the type of structural calculation that will be performed subsequently after the thermal calculation.

1) A 3D thermal calculation is typically performed to establish the temperature distribution in a detail of a construction, for example a connection, see Figure 10, Figure 11 or Figure 12. Such a 3D thermal calculation cannot be followed by a structural calculation made by SAFIR. The temperature can eventually be used in a simple structural calculation model such as, for example, the component method in steel connections or the strut and tie model in concrete elements, in which the influence of the calculated temperatures on the materials strength can be taken into account.
2) A 2D thermal calculation can be performed on the cross section of a beam-column finite element, see Figure 6 to Figure 9. This means that the discretisation made for the thermal calculation will also be used to calculate the mechanical properties of the section in the subsequent structural calculation.
3) A 2D thermal calculation can be performed for evaluating the temperature distribution across the thickness of a shell finite element. Figure 14, for example, shows a 14 cm thick concrete slab protected on the underneath by a 2 cm thick layer of protection material. The discretisation has to comply with the following rules:
    - An arbitrary width of the shell is modeled. In Figure 14, for example, a width of 10 cm has been modeled, but it could be as well 1 cm or 1 meter. Also the horizontal position of the section in the local system of coordinates, i.e. in the direction of **z**, is not relevant.
    - The section is modeled by rectangular finite elements, the width of which is the same as the width of the discretised section. The number of elements on the thickness of the section, N, is chosen by the user (N = 18 on Figure 14)
    - The nodes are numbered from 1 in the bottom left corner to N+1 on the top left corner, and from N+2 to 2 N + 2 on the right side of the section.
    - In the local system of coordinates, the structural material of the slab is located from -*t*/2 to +*t*/2, in the direction of **y**, with *t* the thickness of the structural part of the slab (from node 5 to node 19 on Figure 14).
    - There is only one structural material represented in the thermal analysis of the slab (material `SILCONC_EN` in Figure 14). If re-bars are present in the slab, they are not considered in the thermal analysis but will be added in the .TSH file. Layers of other materials, typically insulating products, can be present outside the structural part of the shell (material `INSULATION` in Figure 14).

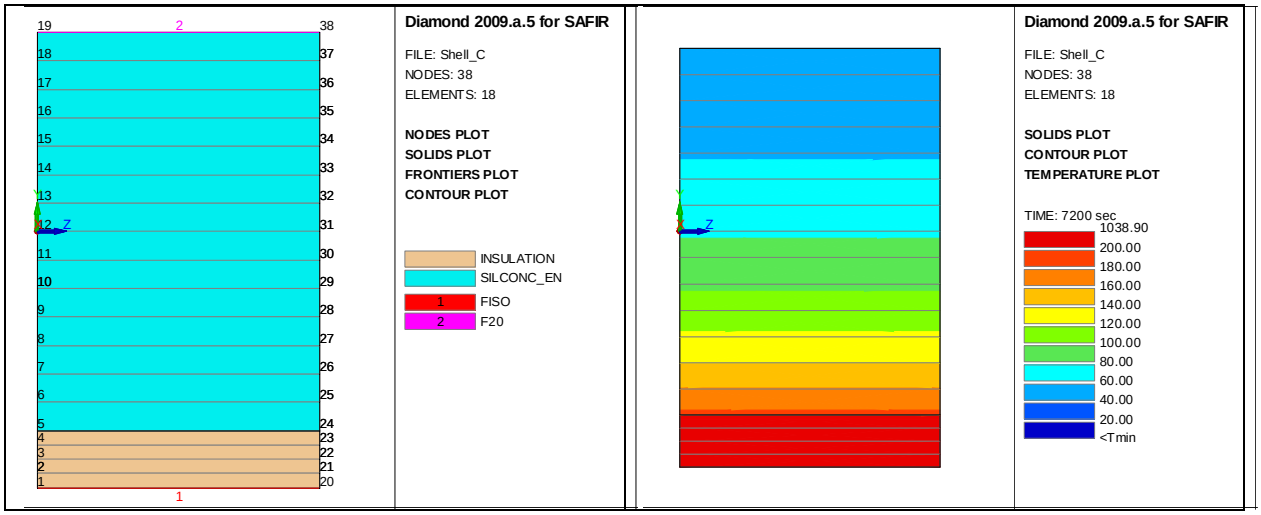


Figure 14: concrete shell protected on the bottom face

Figure 15 shows a 12 mm steel plate exposed to the fire on both sides and modeled by a single finite element. A width of 100 mm has been considered here, but this value does not influence the temperature distribution across the thickness and is thus not relevant.

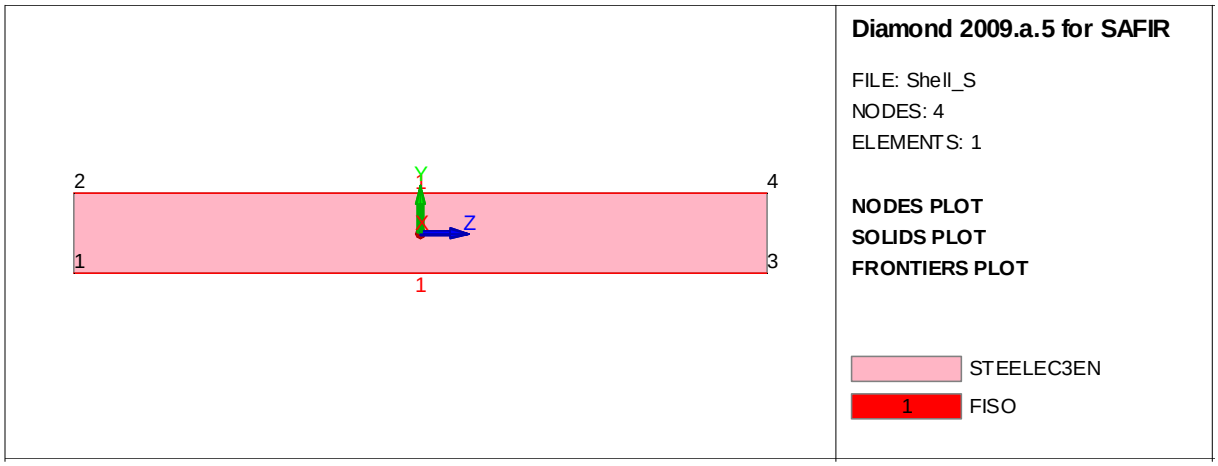


Figure 15: steel plate

The second option to be decided by the user is whether the fire is considered as a localized fire or not.

1) Very often, the fire is not localized. Large parts of the building are considered to be characterized by a uniform situation. For example, one room of the building is characterized by a post flash-over fire curve, while the other rooms are not subjected to the fire. Or a fire compartment is divided in an upper zone that is characterized by an elevated but uniform temperature and a lower zone in which the temperature is lower, but also uniform. In these cases, several finite elements are subjected to the same attack from the fire. It is therefore sufficient to determine the temperature in one of all the elements of the same type which are subjected to the same attack. For example, if a steel beam located in a post flash-over compartment is represented by shell finite

elements, it is sufficient to determine the temperature distribution once for the flanges and once for the web[7]. In order to perform these two thermal calculations, it is sufficient to know the level of the fire attack (the fire curve, for example). It is not necessary to know the position in the compartment of the element for which the temperature is being determined (because all elements of the same type in the compartment have the same temperature distribution).

2) In some case, the fire is a localized fire, which means that its dimensions are small compared to the dimensions of the compartment. This could be the case, for example, of one car burning in a car park that is so large that it may contain tens of cars. In this case, all finite elements in the compartment are subjected to a different attack from the fire, essentially because the radiative heat transfer from the localized fire to the elements depends on the distance between both and on the angle between the surface of the element and the direction of the fire. It will then be necessary to perform a thermal calculation in a lot of different positions of the structure in order to represent the fact that the temperature distribution varies from places to places. Typically, a uniaxial temperature distribution will be determined across the thickness in 4 different positions of each shell finite element and a 2D temperature distribution will be determined in each of the longitudinal point of integration of each beam-column finite element. In order to perform this series of independent thermal calculation, SAFIR must know where each section is located in the compartment with respect to the fire. This means that the structural model must have been created and the position of characteristics of the fire must have been positioned in the compartment before the thermal calculations can be performed.

C.2 Limits of applicability

This section presents the range of scenarios over which the underlying theory is known or believed to be valid and the range of input data over which the calculation method was tested.

- ✓ The scenario that have been considered here are fire scenarios in building structures, with a duration that extends typically between some minutes and a few hours. Extremely fast heating rates such as those that may occur, for example, in the launch pad of Ariane rockets, are not within the scope of the calculation method used in SAFIR. On the other end of the range, applications that involve very long exposure to elevated temperatures such as, for example, those that may be encountered in some industrial processes, are also not within the scope of SAFIR.

[7] Assuming that the flanges and the web have different thicknessess.

- ✓ The fire scenario may include, after the heating phase, a so called cooling phase leading to decrease of the temperature in the materials. Such a cooling phase can be considered by SAFIR as far as material properties during cooling are known.
- ✓ The range of temperature is from 0°C to 1200°C. The effect of freezing of free water that may be present in the materials is indeed not taken into account. On the other end of the temperature range, the material properties for fire applications are not known beyond 1200°C.

C.3 Restrictions and limitations of SAFIR

In this section are listed the restrictions and limitations of SAFIR, including appropriate data range and SAFIR's behavior when the ranges are exceeded.

Appropriate data range	Behavior when the ranges are exceeded
SAFIR can treat 2D or 3D models	If *ndim* is not 2 or 3, a message is written in the output file and SAFIR stops execution.
Number of nodes must be at least 3	If *nnode* is less than 2, a message is written in the output file and SAFIR stops execution.
teta must be greater than 0 and not greater than 1	If *teta* is out of scope, a message is written in the output file and SAFIR stops execution.
The initial temperature should be greater than 0 and not greater than 1200	If *tinitial* is out of scope, a warning message is written in the output file, but the run continues
The number of cores of the computer should be greater than 0	If *ncores* is less than 1, it is changed to 1 by SAFIR.
The number of cores mentioned in the input file should not be larger than the number of cores of the computer	If *ncores* is larger than the number of cores of the computer, the run continues
The number of materials must be greater than 0	If *nmat* is less than 1, , a message is written in the output file and SAFIR stops execution.
The number of solid elements must be greater than 0.	If *nsolid* is less than 1, a message is written in the output file and SAFIR stops execution.
The number of integration points in each direction must be 1, 2 or 3.	If *ng* is not 1, 2 or 3, a message is written in the output file and SAFIR stops execution.
The number of frontiers in a cavity must be at least 2.	If *nfrontiervoid* is less than 2, a message is written in the output file and SAFIR stops execution.

D. Input data description

D.1 Input variables

D.1.1 General

The data to be used by SAFIR in a thermal calculation must be placed in a txt file[8].

The name of the file, *filename*, is chosen by the user and must conform to the rules of Windows. For example, some characters cannot be used in the name of the file. *filename* is limited to 70 characters.

The file type of the input file must be IN. The complete name of the input file in the folder where it is present will thus be *filename*.IN. On the hard disk of a particular P.C., the full name could be, for example if example1 has been chosen as the file name by the user, c:/my files/projects/SAFIR/UsersManual/example1.IN. This input file will be referred to in this manual as the .IN file.

In general, each line of the .IN file starts with a command; it is a chain of characters typed in capital letters, with a maximum length of 10 characters. This command indicates to SAFIR the meaning of the numbers that are present on the same line or in the subsequent lines after the command. The commands are also useful for the user who reads the file; they make it quite easy to locate particular information in the input file compared to a situation where only numbers would be present in the file.

The format for writing the input of each card is free. This means that the real values can be written in any form. “235.E6”, “235000000.” and “235.00E+06”, for example, are equally valid and express the same value. Subsequent values must be separated by any number of spaces, or by a comma. The values that are described as belonging to a card in the manual can be split on several lines. Additional information (such as characters used to comment the file) can be written after the last information of a card, provided that it is written on the same line as the last information. The following examples are all valid examples of the same card.

```
NNODE  1   2.   3.
```

```
NNODE, 1, 2., 3.
```

[8] Also called an ASCII file.

```
NNODE  1
        2.
        3.
```

```
NNODE  1
      2.   3.   These are the coordinates of node 1
```

The order in which the data are introduced in the input file is mandatory. All data must be introduced in the order that is mentioned hereafter.

Blank card can be introduced in the file between the cards mentioned hereafter.

Distances are in meter. Temperatures are in degree Celsius.

D.1.2 Detailed description of the input file

SERIES 1: COMMENTS

a) Multiple cards (can be 0)

As many lines as required may be introduced by the user at the beginning of the file in order to comment the content of the file. These comments will not influence in any manner the results of the calculation. They will help understand what the file is about, which may be particularly useful if the file is found and opened several years after it has been created. Experience shows that the users tend to neglect the comments, relying on the characters that are used in *filename* in order to differentiate several calculations made for a project. Using extensively the possibility of introducing detailed comments in the file is nevertheless highly preferable because it allows reporting much more information.

b) 1 card

1 blank line to indicate that the comments are finished. [-]

SERIES 2: NUMBER OF NODES

a) 1 card

NNODE, *nnode*

- NNODE [A5]

  Command

- *nnode* [Integer]

  Number of nodes of the model. Minimum value is 3. There is no maximum value.

```
This file has been created as an example for the users manual of SAFIR
Date: 16/12/2010
Author: Jean-Marc Franssen

     NNODE   16
```

SERIES 3: NUMBER OF DIMENSIONS

a) 1 card

NDIM, *ndim*

- NDIM [A4]

  Command

- *ndim* [Integer]

  Number of dimensions of the model.

  ndim = 2 for 2D models

  ndim = 3 for 3D models

SERIES 4: DEGREES OF FREEDOM

a) 1 card

NDOFMAX, *1*

- NDOFMAX [A10]

  Command

- *1* [Integer]

  This card indicates that the maximum number of degrees of freedom for the nodes is 1 in a thermal analysis.

b) 1 card

EVERY_NODE, *1*

- EVERY_NODE [A10]

  Command

- *1* [Integer]

  This card allocates 1 degree of freedom to all nodes of the model.

c) multiple cards

FROM, *nno1*, TO, *nno2*, STEP, *dnno*, NDOF, *0*

- FROM [A4]

  Command

- *nno1* [Integer]

  First node of the series.

- TO [A2]

  Command

- *nno2* [Integer]

  Last node of the series.

- STEP [A4]

  Command

- *dnno* [Integer]

Step in the series of nodes

- NDOF [A4]

  Command

- *0* [Integer]

  This card allocates 0 degree of freedom to the nodes *nno1, nno1+dnno, nno1+2*dnno, … , nno2-dnno, nno2*

This card supersedes the effects of the card EVERY_NODE that has been executed earlier. It is used mainly when the input file has been created with a text editor and, in order not to disturb the automatic generation of nodes by irregularities, some nodes have been introduced that are not linked to any element. SAFIR can thus not calculate the temperature at these nodes. One possible solution is to declare 0 degree of freedom to these nodes with the FROM card[9]. The following lines are all valid examples of the FROM card.

```
FROM  789  TO  789  STEP  1  NDOF  0
FROM   90  TO  100  STEP  1  NDOF  0
FROM  200  TO  300  STEP 10  NDOF  0
```

d) 1 card

- END_NDOF [A8]

  Command

This card indicates the end of the series on degrees of freedoms.

In most cases, series 4 will comprise only 3 cards. The input file may now look like this:

```
This file has been created as an example for the users manual of SAFIR
Date: 16/12/2010
Author: Jean-Marc Franssen

     NNODE   16
      NDIM    2
   NDOFMAX    1
EVERY_NODE    1
  END_NDOF
```

[9] Another solution is to fix the value of the temperature at these nodes with a BLOCK card, see series xxx.

SERIES 5 (optional): MATRIX SOLVER

a) 1 card (optional)

SOLVER, CHOLESKY

- SOLVER [A6]

  command

- CHOLESKY [A8]

  command used to force SAFIR to solve the system of equations by the method of Cholesky with a storage of the matrix by a skyline method. This method is not as efficient as the default method of Pardiso based on a sparse matrix solver. It is thus not recommended to use this card. The possibility has been given as a safety measure because Pardiso has been introduced recently in the code; this card allows going back to the previous but outdated method in case any problem would appear with the new method.

  If this card is used, the next card on NCORES cannot be used, because Cholesky systematically uses only one core of the computer.

b) 1 card (optional)

NCORES, *ncores*

- NCORES [A6]

  command

- *ncores* [integer]

  Number of cores of the CPU of the computer used by matrix solver. The default value is 1, in which case this card may be omitted. This card can be used to force SAFIR to use more than 1 core, if present on the computer. Recent experience has shown that using more than 1 core hardly reduces the time of the runs with the present version of Pardiso and this card can thus be omitted as a common practice. The possibility of using the card has nevertheless been given in order to allow users to perform their own test on their particular system, and in order to offer the possibility of working with more than 1 core in the future if new releases of Pardiso show to exploit several cores more efficiently.

  This card cannot be used if the previous card of CHOLESKY has been used, because Cholesky systematically uses only one core of the computer.

SERIES 6: THERMAL CALCULATION

a) 1 card

- COMMAND [A8]

  COMMAND = TEMPERAT is used for a normal thermal calculation from time t = 0 to a final time to be defined by the user.

  COMMAND = RESTARTT is used if a previous thermal analysis has been made and a new thermal analysis has to be restarted from a restart time t_i (to be defined by the user) up to a final time. The temperature field at time t_i is taken from the previous analysis. The new analysis can be performed either on the same structure as the previous analysis or on a new structure that is only one part of the previous structure. The first possibility is used if the structural analysis shows that the thermal analysis had been stopped too early. The second possibility is used, for example, to take into account the fact that some layers of concrete or of protective material have fallen off the structure (what part has fallen off and at what time has to be decided by the user).

The next 3 cards, b), c) and d) are present only if RESTARTT has been entered in card a). If TEMPERAT has been entered in card a), then go directly to card e).

b) 1 card

FIRSTFILE, *filename*

- FIRSTFILE [A9]

  Command

- *filename* [A20]

  Name (comprising the file name and the file type) of the output file were the results of the previous thermal analysis had been written. This file must be present in the same folder as the input file of the thermal analysis

c) 1 card

FIRSTTIME, *time*

- FIRSTTIME [A9]

  Command

- *time* [Real]

  Time (in seconds) when the second analysis will start. The temperatures will be read from the first analysis and used as initial values for the second analysis.

d) 1 card

- COMMAND [A10]

  COMMAND = MATCHNODES is used if the second structure is exactly the same as the first one or if its nodes were the first nodes of the first structure.

In that case, the initial temperature at each node in the second structure is taken as the temperature at the same node in the first structure.

COMMAND = MATCHCOORD allows considering that the initial temperature at each node in the second structure is taken as the temperature of the node of the first structure with same coordinates. This possibility is used if some among the first nodes of the first structure have been deleted when creating the second structure.

e) 1 card

TETA, *teta*

- TETA [A4]

  Command

- *teta* [Real]

  Parameter for the time integration. 0 < *teta* ≤ 1.

  Values of *teta* close to 0 will force SAFIR to perform the time integration in a nearly explicit manner, whereas *teta* = 1 will lead to a fully implicit integration. *teta* = 0.9 has been used consistently by the developers without any problem.

f) 1 card

TINITIAL, *tinitial*

- TINITIAL [A8]

  Command

- *tinitial* [Real]

  Temperature (in degree Celsius) in the model at time t = 0, usually taken as 20°C.

e) 1 card (optional)

- COMMAND [A8 or A10]

  COMMAND = MAKE.TEM stores the average temperature of the elements for a subsequent structural analysis. The section analysed here is the cross section of a beam.

  COMMAND = MAKE.TSH Stores the temperature of the first NNODE/2 nodes for a subsequent structural analysis. The section analysed describes the temperature evolution across the thickness of a shell element, see Figure 14 and Figure 15.

  COMMAND = MAKE.TEMHA is similar to MAKE.TEM, but the fire is a localised fire as described in Annex C of EN 1991-1-2 (equation C.4, flame impacting the ceiling – Hasemi fire).

  COMMAND = MAKE.TSHHA is similar to MAKE.TSH, but the fire is a localised fire as described in Annex C of EN 1991-1-2 (equation C.4, flame impacting the ceiling – Hasemi fire).

If MAKE . TEMHA or MAKE . TSHHA has been entered in card e), then:

1) The next 2 cards, f) and g) must be present.
2) A file called *hasemi.txt* must be present in the same folder as the input file. This file describes the position of the fire in the structure and its thermal characteristics, such as the rate of heat release. The structure of this file is described in D.1.3.
   Note: the name of the file is *hasemi.txt*, even if the function HASEMI_FR is used in series 14.

f) 1 card (optional)

- *filename* [A20]

  This is the complete name of the input file of the structural analysis (maximum 20 characters).

  Note: the input file describing the structural analysis must be present in the same folder as the input file for the thermal analysis when the thermal analysis is run. If the structural analysis is 2D, gravity must be pointing downward the Y axis (i.e. Y is vertical upward). If the structural analysis is 3D, gravity must be pointing downward the Z axis (i.e. Z is vertical upward).

g) 1 card (optional)

COMMAND, *ielemtype*

- COMMAND [A9 or A10]

  COMMAND = BEAM_TYPE if the temperatures are determined in a beam section.

  COMMAND = SHELL_TYPE if the temperatures are determined in a shell section.

- *ielemtype* [integer]

  is the number in the structural input file of the section type (beam or shell) treated in this thermal analysis.

SERIES 7 (optional): EQUATION RENUMBERING

One (and only one) of the following cards may be used. The card may be omitted if the sparse matrix solver has been used (which is the default option), because renumbering is perform by Pardiso and the renumbering made by SAFIR does not accelerate the runs; the time spent for renumbering is thus wasted. One of these cards should be used only of CHOLESKY has been chosen as the matrix solver, see series 5.

a) 1 card (optional)

- NORENUM [A7]

  Command. No renumbering of the equations will be performed. Introducing this card has the same effect as omitting this card. The possibility is given to introduce this card for compatibility with earlier version of the GID-SAFIR interface.

b) 1 card (optional)

- RENUMPERM [A9]

  Command. Renumbering of equations by logical permutations.

c) 1 card (optional)

RENUMGEO, *nno1*

- RENUMGEO [A9]

  Command. Renumbering of equations by geometrical method.

- *nno1* [integer]

  Number of the node where geometrical renumbering will start. If *nno1* = 0, then renumbering starts successively from all nodes (this process can take a long time).

  RENUMGEO, *nnoi* with *nnoi* being the number of a node located geometrically in a corner of the model is probably a good compromise between the time spent for renumbering and the acceleration provided to the run in the method of CHOLESKY.

d) 1 card (optional)

- RENUM [A4]

  Command. Has the same effect as RENUMGEO, *1* + RENUMPERM. Produces then most efficient renumbering, but can take a very long time.

e) 1 card (optional)

- READRENUM [A8]

  Command. The renumbering has been done in a previous run. The match between equation numbers before renumbering and equation numbers after renumbering is read from the .REN file that had been created during the renumbering. This card can be used only if the typology of the model has not been modified since the renumbering has been made.

SERIES 8: MATERIALS

a) 1 card (optional)
NMAT, *nmat*

- NMAT [A4]

  Command

- *nmat* [integer]

  Number of different materials. If two materials have the same name but different thermal properties (such as, e.g., the emissivity) or different mechanical properties (such as, e.g., the yield strength), this makes two different materials.

SERIES 9: ELEMENTS (general)

a) 1 card

- ELEMENTS [A8]

  Command.

b) 1 card
SOLID, *nsolid*

- SOLID [A5]

  Command

- *nsolid* [integer]

  Number of SOLID elements in the model.

c) 1 card
NG, *ng*

- NG [A2]

  Command

- *ng* [integer]

  Number of integration points in each direction in the element. Not less than 1, not greater than 3. 2 has been used consistently by the developers without any problem.

d) 1 card
NVOID, *nvoid*

- NVOID [A5]

  Command

- *nvoid* [integer]

  Number of internal voids, also called internal cavities, i.e. cavities in the section where there is no material and heat transfer is by convection and, essentially, by radiation. 0 must be typed if there is no cavity in the model. In

the present version of SAFIR (2011), internal cavities can be present only in 2D models.

e) 1 card (optional)
This card is present only if *nvoid* ≠ 0
FRTIERVOID, *nfrontiervoid*

- FRTIERVOID [A10]
  Command
- *nfrontiervoid* [integer]
  Number of maximum (for all the voids of the section) number of surfaces (i.e. sides of elements) enclosing the internal voids. For example, if a section has 2 cavities and one of them is enclosed by 50 surfaces, whereas the second one is enclosed by 30 surfaces, then *nfrontiervoid* is equal to 50.

f) 1 card

- END_ELEM [A8]
  Command.

The input file, starting after the comments, may now look like this.

```
     NNODE   16
      NDIM    2

   NDOFMAX    1
EVERY_NODE    1
  END_NDOF

  TEMPERAT
      TETA       0.9
  TINITIAL      20.0

      NMAT    2

  ELEMENTS
     SOLID    9
        NG    2
     NVOID    0
  END_ELEM
```

SERIES 10: NODES

a) 1 card

- COMMAND [A5 or A9]

  COMMAND = NODES. The position of the nodes will be given in a Cartesian system of coordinates

  COMMAND = NODES_CYL. The position of the nodes will be given in a cylindrical system of coordinates. Cylindrical coordinates are transformed for the internal solution process by the following equations:

  $(r,\theta) => (Y,Z) = (r\cos\theta, r\sin\theta)$ if *ndim* = 2

  $(r,\theta,Z) => (X,Y,Z) = (r\cos\theta, r\sin\theta, Z)$ if *ndim* = 3

  Note that θ is in degrees. The transformation is made after all nodes have been read and the automatic generation and repeat have been made, see card c) and d).

Several cards are then written, following the format of b), c) or d), until all *nnode* nodes have been given their position.

b) 1 card for 1 node

NODE, *nno*, *rcoordg*(1,*nno*), *rcoordg*(2,*nno*), *rcoordg*(3,*nno*)

- NODE [A5]

  Command

- *nno* [integer]

  Number of the node

- *rcoordg*(1,*nno*) [real]

  First coordinate (in meter) of node *nno*.

- *rcoordg*(2,*nno*) [real]

  Second coordinate (in meter or in degree) of node *nno*.

- *rcoordg*(3,*nno*) [real]

  Third coordinate (in meter) of node *nno*. Present only if *ndim* = 3.

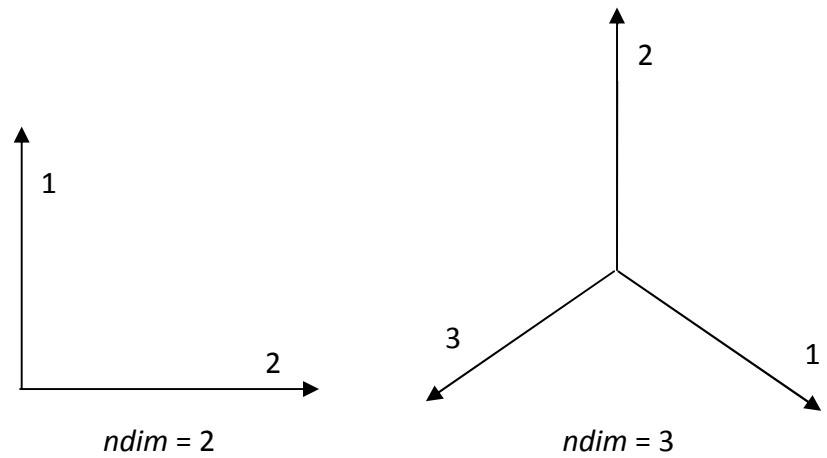


Figure 16: global axes in a Cartesian system of coordinates

c) 1 card (optional) for automatic generation

GNODE, *nno*, *rcoordg*(1,*nno*), *rcoordg*(2,*nno*), *rcoordg*(3,*nno*)

- GNODE [A5]

  This command is used for automatic equidistant generation of nodes between the previously defined node and the node *nno*.

- *nno* [integer]

  Number of the node

- *rcoordg*(1,*nno*) [real]

  First coordinate (in meter) of node *nno*.

- *rcoordg*(2,*nno*) [real]

  Second coordinate (in meter or in degree) of node *nno*.

- *rcoordg*(3,*nno*) [real]

  Third coordinate (in meter) of node *nno*. Present only if *ndim* = 3.

d) 1 card (optional) for repeating a series of nodes

REPEAT, *nno*, *delta*(1), *delta*(2), *delta*(3), *kgene*

- REPEAT [A5]

  This command is used to repeat the definition of nno previously defined nodes with a increment in coordinates

- *nno* [integer]

  Number of nodes to be repeated

- *delta*(1) [real]

  Increment for the first coordinate (in meter).

- *delta*(2) [real]

  Increment for the second coordinate (in meter or in degree).

- *delta*(3) [real]

  Increment for the third coordinate (in meter). Present only if *ndim* = 3.

- kgene [integer]

  Number of times that the previous *nno* defined nodes have to be repeated

For example, the following series of cards:

```
NODES
      NODE    1     0.000     0.000
     GNODE    4     0.000     0.120
    REPEAT    4     0.025     0.010    3
```

is equivalent to:

```
NODES
      NODE    1     0.000     0.000
      NODE    2     0.000     0.040
      NODE    3     0.000     0.080
      NODE    4     0.000     0.120
      NODE    5     0.025     0.010
      NODE    6     0.025     0.050
      NODE    7     0.025     0.090
      NODE    8     0.025     0.130
      NODE    9     0.050     0.020
```

```
NODE    10      0.050     0.060
NODE    11      0.050     0.100
NODE    12      0.050     0.140
NODE    13      0.075     0.030
NODE    14      0.075     0.070
NODE    15      0.075     0.110
NODE    16      0.075     0.150
```

The nodes that have been generated are shown on Figure 17.

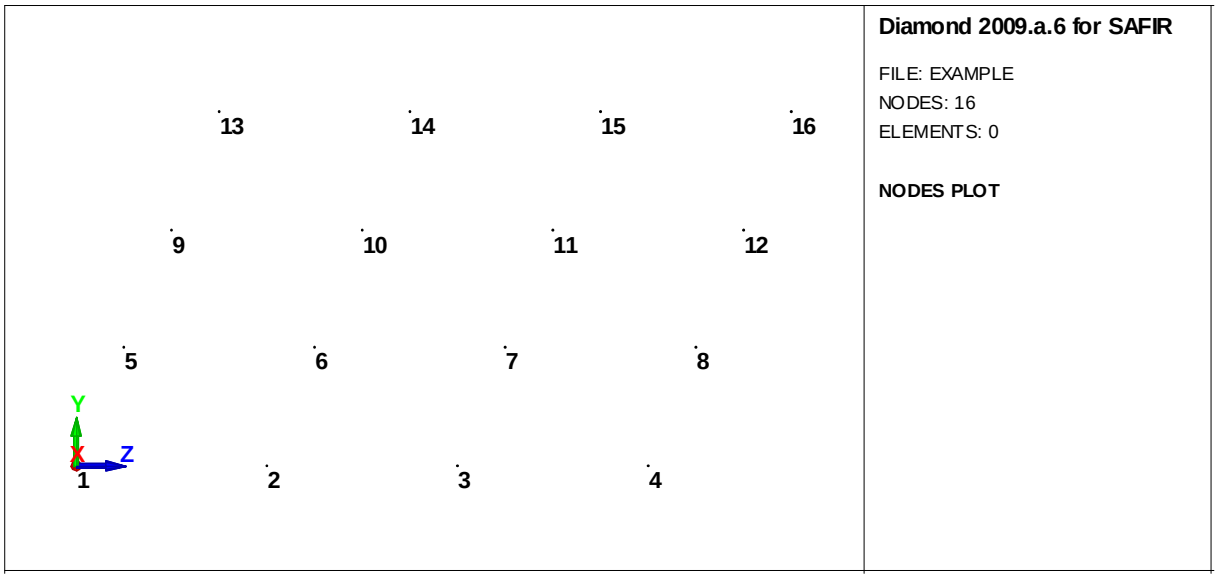


Figure 17: nodes created by GNODE and REPEAT commands

SERIES 11: NODE LINE

This series is only present if MAKE.TEM or MAKE.TEMHA has been given in series 6, e).

The values entered in this series will not influence the values of the temperatures that will be calculated. They are relevant only for the structural calculation using the beam elements the section of which is described in the discretisation made for the thermal analysis, see [1], section xxx.

a) 1 card

NODELINE, Y_0, Z_0

- NODELINE [A8]

  Command

- Y_0 [real]

  First coordinate of the node line

- Z_0 [real]

  Second coordinate of the node line

b) 1 card

YC_ZC, Y_C, Z_C

- YC_ZC [A5]

  Command

- Y_C [real]

  First coordinate of the centre of rotation

- Z_C [real]

  Second coordinate of the centre of rotation

SERIES 12: IMPOSED TEMPERATURES

a) 1 card

- FIXATIONS [A9]

  Command

b) 1 card (optional) at each node where the evolution of the temperature as a function of time is imposed by the user.

BLOCK, *nno*, *cblock*(*nno*)

- BLOCK [A5]

  Command

- *nno* [integer]

  Number of the node where the evolution of the temperature is imposed

- *cblock*(*nno*) [A10]

  Name of the function describing the evolution of the temperature at this node.

c) 1 card (optional) for each master-slave relationship.

SAME, *nno1*, *nno2*, YES

- SAME [A4]

  Command

- *nno1* [integer]

  Number of the slave node (its temperature is forced to be equal to the temperature of the master node).

- *nno2* [integer]

  Number of the master node (a master node can have several slave nodes. A SAME cards must be introduced for each slave).

- YES [A3]

  Command

d) 1 card (optional) to repeat the previous SAME command.

REPEAT, *n*, *incr*, YES

- REPEAT [A6]

  Command

- *n* [integer]

  Number of times that the previous SAME command has to be repeated.

- *incr* [integer]

  Increment on the node numbers *nno1* and *nno2* of the previous SAME command.

- YES [A3]

  Command

e) 1 card (optional) to create master-slave relationship between all nodes that have the same coordinates.

SAMEALL, YES

- SAMEALL [A7]

  Command. All the nodes of the model that have the same coordinates (with a precision of 0.316 mm) will automatically be attributed a master-slave relationship.

- YES [A3]

  Command

f) 1 card

- END_FIX [A7]

  Command indicating the end of the series.

SERIES 13: SOLID ELEMENTS

a) 1 card

- NODOFSOLID [A10]

  Command

Several cards are then written, following the format of b), c) or d), until all *nsolid* elements have been described.

b) 1 card for 1 element

ELEM, *nsol, nodesofsolid*(1,*nsol*), *nodesofsolid*(2,*nsol*), … ,
nodesofsolid(*numberofnodesinsolid*,*nno*), *matsolid*(*nsol*), *epsrsolid*(*nsol*)

- ELEM [A4]

  Command

- *nsol* [integer]

  Number of the element

- *nodesofsolid*(1,*nsol*) [integer]

  First node of element *nsol*.

- *nodesofsolid*(2,*nsol*) [integer]

  Second node of element *nsol*.

  …

- *nodesofsolid*(numberofnodesinsolid,*nsol*) [integer]

  Last node of element *nsol*.

- *matsolid*(*nsol*) [integer]

  Material number of the element *nsol.* The materials are numbered from 1 to nmat according to the sequence appearing in series xxx.

- *epsrsolid(nsol)* [real]

Residual stress (in Pa or N/m²) in element *nsol*. The values entered for *epsrsolid* will not influence the values of the temperatures that will be calculated. They are relevant only for the structural calculation using the beam elements the section of which is described in the discretisation made for the thermal analysis see [1], section xxx. Residual stresses are relevant when *ndim* = 2 and MAKE . TEM or MAKE . TEMHA have been given.

If *ndim* = 2, *numberofnodesinsolid* = 4. For triangular elements, *nodesofsolid*(4,*nsol*) = 0

If *ndim* = 3, *numberofnodesinsolid* = 8. For elements which have a triangular base, *nodesofsolid*(7,*nsol*) = 0, *nodesofsolid*(8,*nsol*) = 0.

In 2D elements, the nodes have to be given in counterclockwise order, see Figure 18.

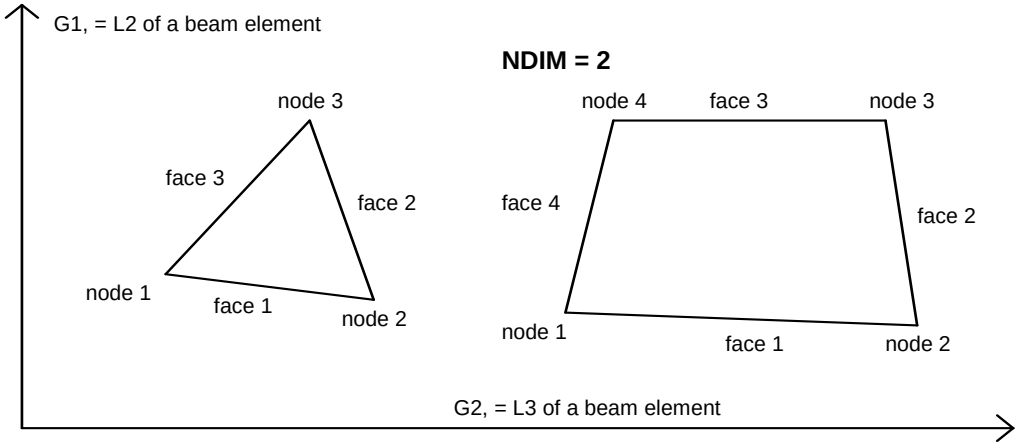


Figure 18: nodes and faces of elements for 2D analyses

In 3D elements with 6 nodes, nodes 1, 2, 3 must belong to one of the triangular frontiers and nodes 4, 5, 6 to the other one, with the order 1, 2, 3 drilling into the element and 4, 5, 6 drilling out of the element. Node 4 must be on the same edge as node 1, and thus 5 with 2 and 6 with 3, see Figure 19.

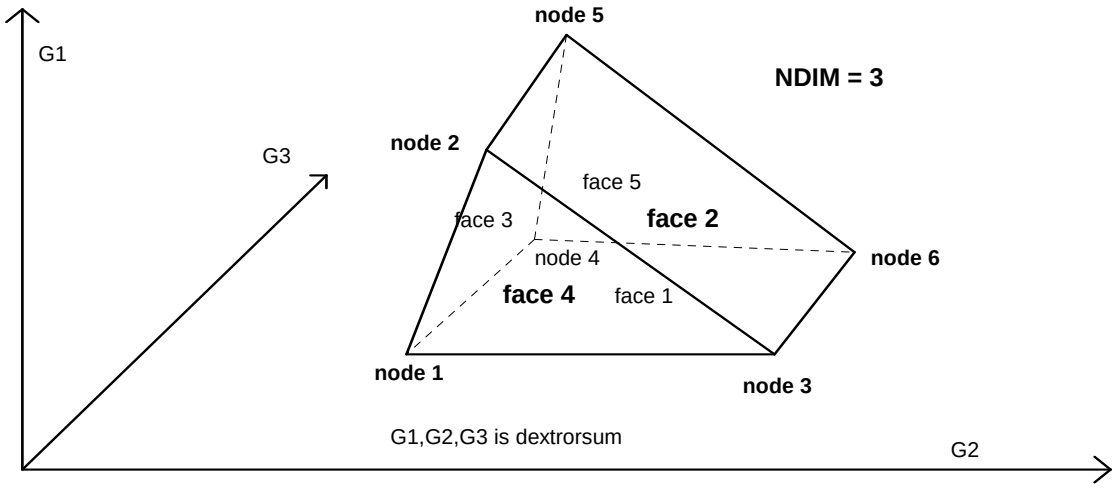


Figure 19: nodes and faces of elements with 6 nodes for 3D analyses

In 3D elements with 8 nodes, nodes 1, 2, 3, 4 must belong to one frontier and nodes 5, 6, 7, 8 to the opposite one, with the order 1, 2, 3, 4 drilling into the element 5, 6, 7, 8 drilling out of the element, see Figure 20. Node 5 must be on the same edge as node 1, and thus 6 with 2, etc.

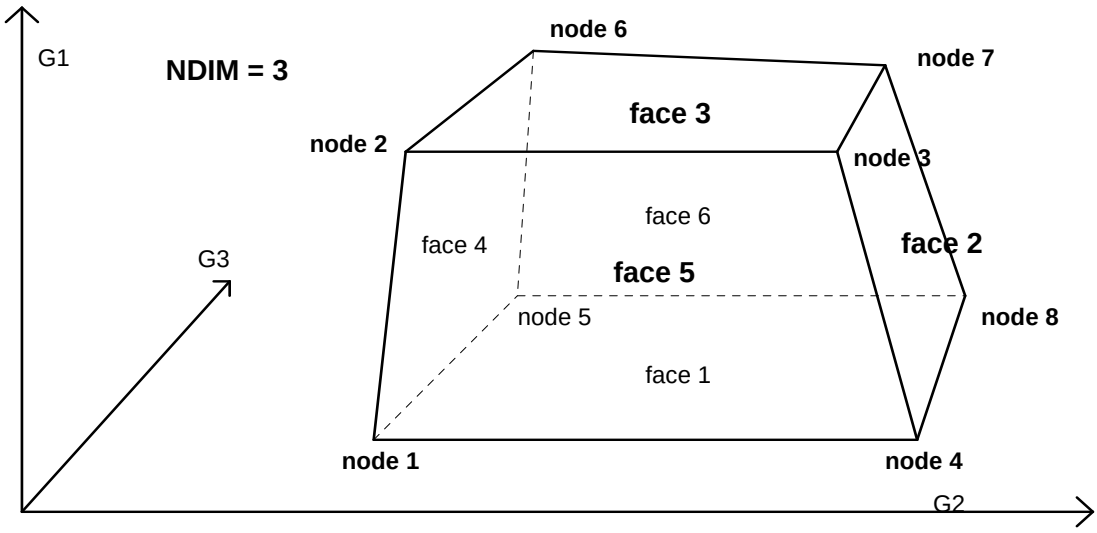


Figure 20: nodes and faces of elements with 8 nodes for 3D analyses

c) 1 card (optional) for automatic generation

GELEM, *nsol*, *nodesofsolid*(1,*nsol*), *nodesofsolid*(2,*nsol*), … ,
nodesofsolid(*numberofnodesinsolid*,*nno*), *matsolid*(*nsol*), *epsrsolid*(*nsol*), *kgene*

- GELEM [A5]
  Command
- *nsol* [integer]
  Number of the element
- *nodesofsolid*(1,*nsol*) [integer]
  First node of element *nsol*.
- *nodesofsolid*(2,*nsol*) [integer]
  Second node of element *nsol*.

  …
- *nodesofsolid*(numberofnodesinsolid,*nsol*) [integer]
  Last node of element *nsol*.
- *matsolid*(*nsol*) [integer]
  Material number of the element *nsol.* The materials are numbered from 1 to nmat according to the sequence appearing in series xxx.
- *epsrsolid(nsol)* [real]
  Residual stress (in Pa or N/m²) in element *nsol*.
- *kgene* [integer]

  The elements from the previously defined one up to this one will be generated automatically. *kgene* is the increment of the nodes from one element to the next one.

d) 1 card (optional) for repeating a series of elements

REPEAT, *n*, *incr*, *kgene*

- REPEAT [A6]

  Command
- *n* [integer]

  Number of elements to be repeated
- *incr* [integer]

  Increment in the node numbers.
- *kgene* [integer]

  Number of times that the *n* elements have to be repeated.

For example, the following series of cards:

```
 FIXATIONS
   END_FIX

NODOFSOLID
      ELEM    1    1    2    6    5    1   0.
     GELEM    3    9   10   14   13    1   0.    4
    REPEAT    3    1                             2
```

is equivalent to:

```
FIXATIONS
   END_FIX

NODOFSOLID
      ELEM    1    1    2    6    5    1  0.0000E+00
      ELEM    2    5    6   10    9    1  0.0000E+00
      ELEM    3    9   10   14   13    1  0.0000E+00
      ELEM    4    2    3    7    6    1  0.0000E+00
      ELEM    5    6    7   11   10    1  0.0000E+00
      ELEM    6   10   11   15   14    1  0.0000E+00
      ELEM    7    3    4    8    7    1  0.0000E+00
      ELEM    8    7    8   12   11    1  0.0000E+00
      ELEM    9   11   12   16   15    1  0.0000E+00
```

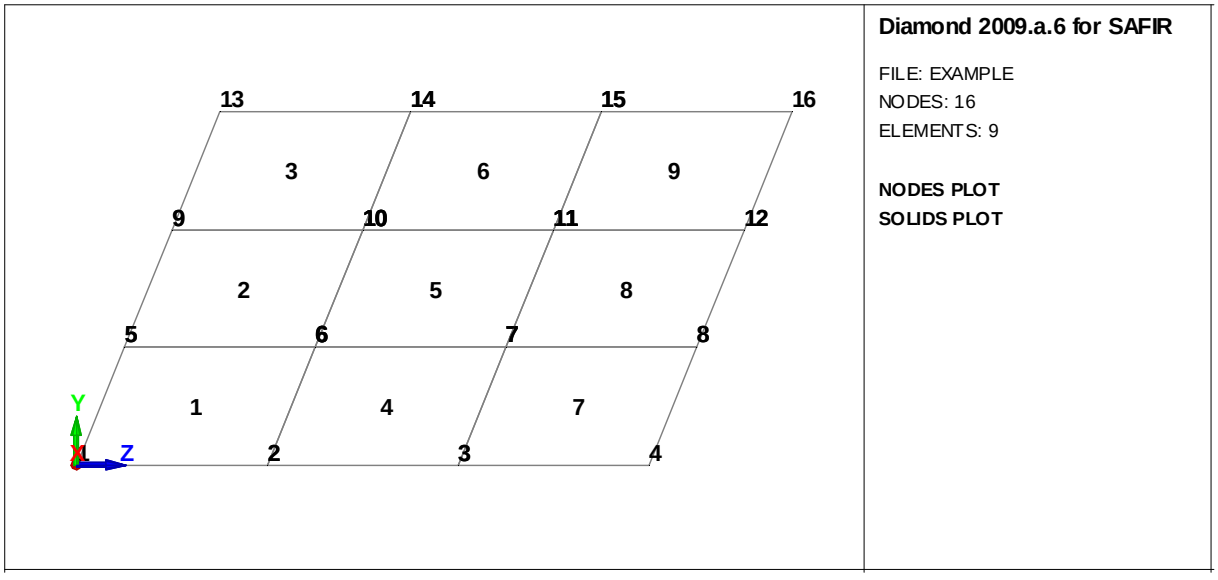


Figure 21: elements created by GELEM and REPEAT commands

When all elements have been defined, it is possible to change the material of one or several elements. This can be useful if the input file has been created with a text editor and extensive utilization has been made of the GELEM and REPEAT commands[10].

e) 1 card (optional) for each element in which the material has to be changed.

NEW_MAT, *nsol*, *matsolid*(*nsol*)

- NEW_MAT [A7]

  Command

- *nsol* [integer]

  Number of the element in which the material has to be changed

- *matsolid*(*nsol*) [integer]

  Number of the material to be given to the element *nsol*

For example, the following series of cards:

```
NODOFSOLID
      ELEM    1    1    2    6    5    1   0.
     GELEM    3    9   10   14   13    1   0.    4
    REPEAT    3    1                            2
   NEW_MAT    5    2
```

will allocate the materials to the elements as shown on Figure 22 if STEELEC3EN is the first material name and CALCONC_EN is the second one, see series 18. Without the NEW_MAT command, the description of the elements with the appropriate materials would require 7 cards, instead of 4 cards in the example shown here. The utilisation of the REPEAT command would indeed not be possible.

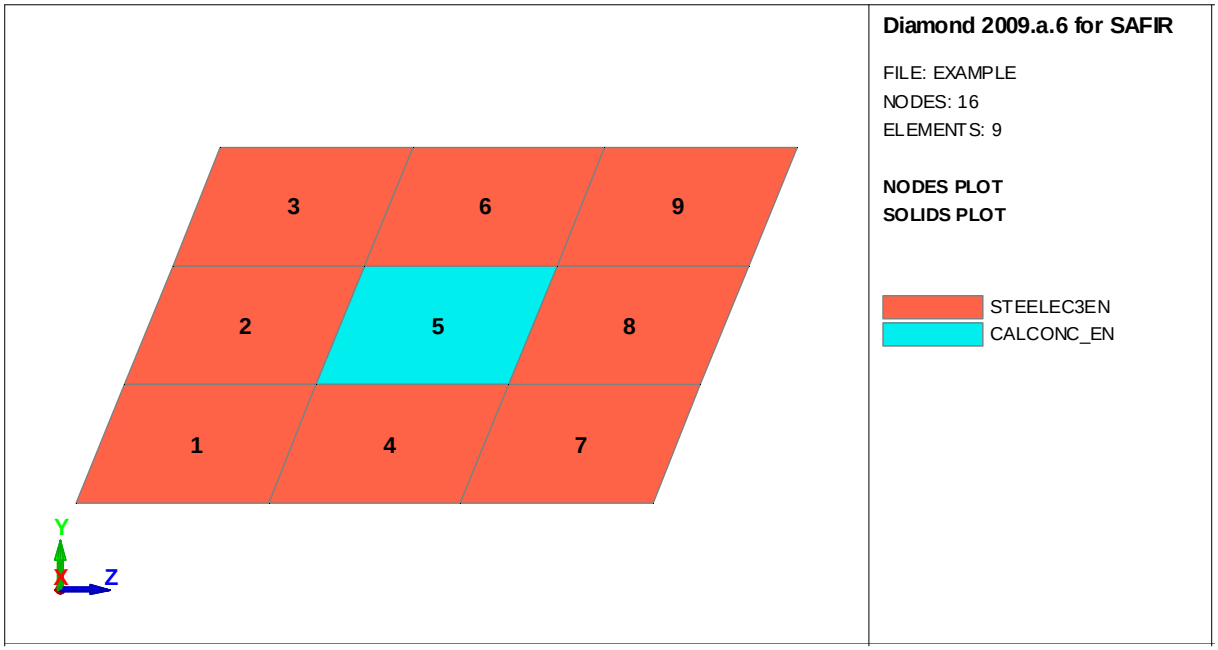


Figure 22: result of the NEW_MAT command

[10] NEW_MAT is typically used to create a steel rebar in a concrete section.

SERIES 14: FRONTIERS

a) 1 card

- FRONTIER [A8]

  Command

Different cards can be used in order to allocate boundary conditions to selected surfaces of some elements. Two different boundary conditions can be used: either a time-temperature fire curve describes the environment around the model (commands F and GF) or the incipient flux on the model is prescribed (commands FLUX and GFLUX).

b) 1 card for each element that has at least one surface exposed to a fire curve.

F, *nsol, cfrontiersolid*(1,*nsol*), … ,*cfrontiersolid*(*numberoffrontier*,*nsol*),

- F [A1]

  Command

- *nsol* [integer]

  Number of the element

- *cfrontiersolid*(1,*nsol*) [A10]

  Function describing the fire curve on frontier 1 of element *nsol*.

  …

- *cfrontiersolid*(numberoffrontier,*nsol*) [A10]

  Function describing the fire curve on the last frontier of element *nsol*.

If *ndim* = 2, *numberoffrontier* = 4. For triangular elements, *cfrontiersolid*(4,*nsol*) = NO

If *ndim* = 3, *numberoffrontier* = 6. For elements which have a triangular base, *cfrontiersolid*(7,*nsol*) = NO, *cfrontiersolid*(8,*nsol*) = NO.

The order of the surfaces on the elements is linked to the node numbers.

In a triangular element, see Figure 18,

frontier 1 is from node 1 to node 2
frontier 2 is from node 2 to node 3
frontier 3 is from node 3 to node 1

In a quadrangular element, see Figure 18,

frontier 1 is from node 1 to node 2,
frontier 2 is from node 2 to node 3,
frontier 3 is from node 3 to node 4,
frontier 4 is from node 4 to node 1.

In a 3D element with 6 nodes, see Figure 19,

frontier 1 comprises is defined by the nodes 1, 4, 6, 3,
frontier 2 comprises is defined by the nodes 2, 5, 6, 3,
frontier 3 comprises is defined by the nodes 1, 2, 5, 4,
frontier 4 comprises is defined by the nodes 1, 2, 3,

frontier 5 comprises is defined by the nodes 4, 5, 6.

In a 3D element with 8 nodes, see Figure 20,

frontier 1 comprises is defined by the nodes 1, 5, 8, 4,
frontier 2 comprises is defined by the nodes 4, 3, 7, 8,
frontier 3 comprises is defined by the nodes 2, 6, 7, 3,
frontier 4 comprises is defined by the nodes 1, 2, 6, 5,
frontier 5 comprises is defined by the nodes 1, 2, 3, 4,
frontier 6 comprises is defined by the nodes 5, 6, 7, 8.

The function describing the fire curve can be one of the SAFIR defined function (e.g. `FISO` or `F20`), or a user defined function located in a txt file (e.g. `myfire.txt`), or the command `NO` if no fire curve is applied on a particular surface. The meaning of `NO` is not "remove any previously fire curve"; it is more to be understood as "disregard this information". For example, the two following examples are equivalent:

```
    F     5        FISO       FISO         NO         NO
```

is equivalent to

```
    F     5        FISO         NO         NO         NO
    F     5          NO       FISO         NO         NO
```

c) 1 card for automatic generation of frontiers from the previously defined element to this one.

GF, *nsol, cfrontiersolid*(1,*nsol*), … ,*cfrontiersolid*(*numberoffrontier*,*nsol*), *kgene*

- GF [A2]
  Command
- *nsol* [integer]
  Number of the element
- *cfrontiersolid*(1,*nsol*) [A10]
  Function describing the fire curve on frontier 1 of element *nsol*.

  …
- *cfrontiersolid*(numberoffrontier,*nsol*) [A10]
  Function describing the fire curve on the last frontier of element *nsol*.
- *kgene* [integer]
  increment on the element numbers

```
    F     5        FISO       FISO         NO         NO
   GF    20        FISO       FISO         NO         NO     5
```

is equivalent to

```
    F     5        FISO         NO         NO         NO
    F    10        FISO         NO         NO         NO
    F    15        FISO         NO         NO         NO
    F    20        FISO         NO         NO         NO
```

Figure 23 shows the boundary conditions created by these cards applied on the section that was shown on Figure 21.

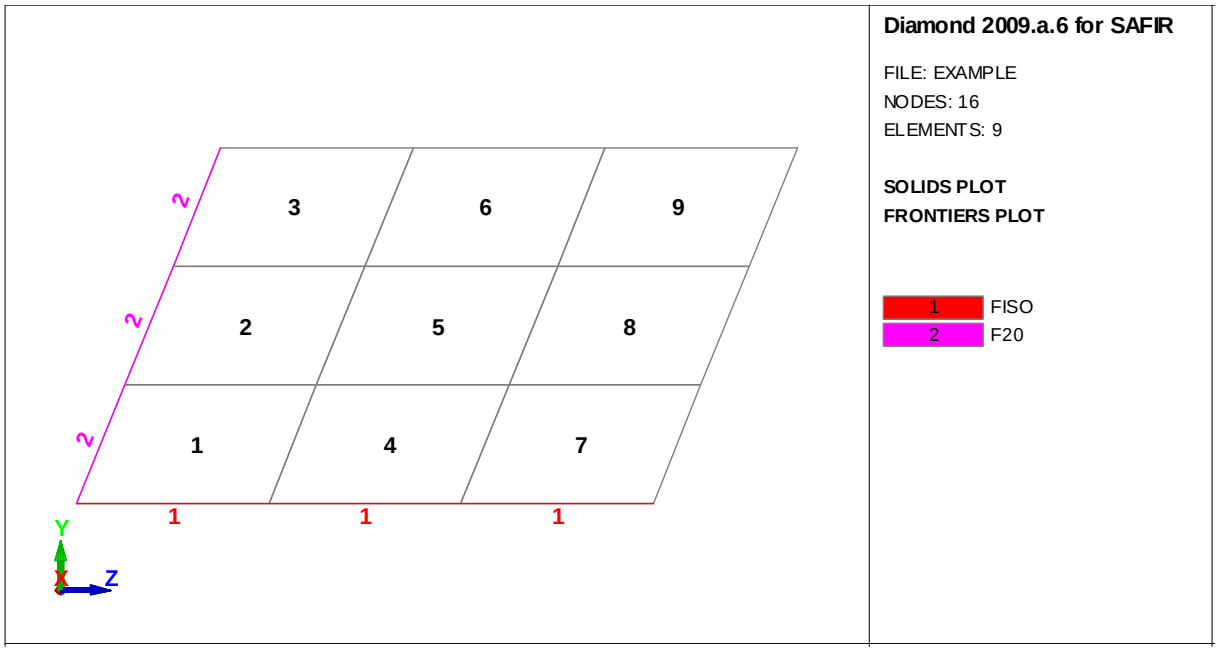


Figure 23: Boundary conditions

d) 1 card for each element that has at least one surface exposed to a defined heat flux.

FLUX, *nsol, cfrontiersolid*(1,*nsol*), … ,*cfrontiersolid*(*numberoffrontier*,*nsol*)

- FLUX [A4]
  Command
- *nsol* [integer]
  Number of the element
- *cfrontiersolid*(1,*nsol*) [A10]
  Function describing the flux (in W/m²) on frontier 1 of element *nsol*.

  …
- *cfrontiersolid*(numberoffrontier,*nsol*) [A10]
  Function describing the flux (in W/m²) on the last frontier of element *nsol*.

The function describing the flux can be one of the SAFIR defined function (e.g. `F1000`) or, more likely, a user defined function located in a txt file (e.g. `myflux.txt`), or the command NO if no flux is applied on a particular surface.

A positive flux means energy introduced in the section.

It is possible to have a flux condition and a fire curve on the same surface.

Each surface of the model heated by the HASEMI fire must have a FLUX command and the function HASEMI where appropriate. For example:

```
FLUX    5          NO          NO      HASEMI          NO
```

There is no need to add a frontier `F20` on a surface heated by a HASEMI flux (for reemitting energy); it will be automatically added by SAFIR.

Note: if the function HASEMI_FR is used in the FLUX command, the flux as given by the Equations of EN 1991-1-2 is multiplied by 0.85, as indicated in the French National Annex.

e) 1 card for automatic generation from the previously defined element.

GFLUX, *nsol, cfrontiersolid*(1,*nsol*), … ,*cfrontiersolid*(*numberoffrontier*,*nsol*), *kgene*

- GFLUX [A5]
  Command
- *nsol* [integer]
  Number of the element
- *cfrontiersolid*(1,*nsol*) [A10]
  Function describing the flux (in W/m²) on frontier 1 of element *nsol*.

  …
- *cfrontiersolid*(numberoffrontier,*nsol*) [A10]
  Function describing the flux (in W/m²) on the last frontier of element *nsol*.
- *kgene* [integer]
  increment on the element numbers

g) 1 card

- END_FRONT [A9]
  Command indicating the end of the series.

SERIES 15: VOIDS

The following cards a), b), c) and d) are repeated *nvoid* times, see series 9. If *nvoid* = 0, this series is skipped.

a) 1 card

- VOID [A4]

  Command

b) 1 card for each surface of an element that is adjacent to an internal cavity.

ELEM, *nsol, nfrontier*

- ELEM [A4]

  Command

- *nsol* [integer]

  Number of the element

- *frontiersolid* [integer]

  Number of the frontier of this element that is adjacent to a cavity

If one element has two surfaces that define a void, two cards must be entered. For example:

```
ELEM    5    2
ELEM    5    3
```

c) 1 card for automatic generation from the previously defined element.

GELEM, *nsol, nfrontier, kgene*

- GELEM [A4]

  Command

- *nsol* [integer]

  Number of the element

- *frontiersolid* [integer]

  Number of the frontier of this element that is adjacent to a cavity

- *kgene* [integer]

  Increment on the element numbers. Can be positive r negative.

d) 1 card

- END_VOID [A9]

  Command indicating the end of the cards defining this cavity.

The elements surrounding a cavity must be described in the same order as they are encountered by someone walking counter clockwise on the frontier of this cavity.

Any element from the frontier of the cavity can be chosen as the first one in the list. The only restriction is that the first 2 elements must touch each other. This is because, if they are separated by an axis of symmetry, SAFIR cannot check whether the cavity is really described counter clockwise.

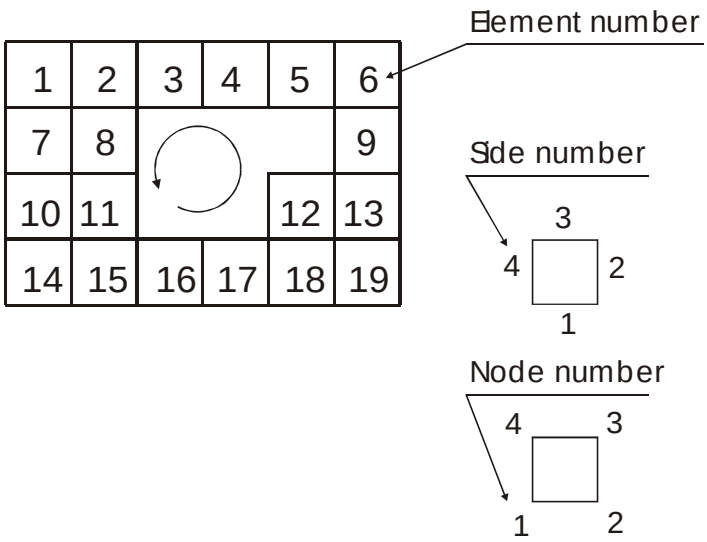


Figure 24: description of a cavity

For example, Figure 24 shows a model with 19 rectangular finite elements and an internal cavity. The cavity can be described by the following cards[11]:

```
VOID
      ELEM    16     3
      ELEM    17     3
      ELEM    12     4
      ELEM    12     3
      ELEM     9     4
      ELEM     5     1
     GELEM     3     1    -1
      ELEM     8     2
      ELEM    11     2
END_VOID
```

[11] In this case, *nfrontiervoid* = 10, see series 9.

SERIES 16: SYMMETRIES

a) 1 card

- SYMMETRY [A4]

  Command

The following cards b), c) and d) are optional. If no symmetry is used when creating the model, only cards a) and e) are present. Symmetries can only be used in 2D models.

b) 1 card for each real axis of symmetry in the section.

REALSYM, *nno1, nno2*

- REALSYM [A7]

  Command

- *nno1* [integer]

  Number of the first node defining the axis of symmetry

- *nno2* [integer]

  Number of the second node defining the axis of symmetry

The line passing through nodes *nno1* and *nno2* is a real axis of symmetry for the model. This possibility is used when the 2D model represents part of a beam section. When creating the .TEM file, the fibres located on the other side of the axis of symmetry are created, with the same sectional area, same material number and same temperature. This option is used when there is a thermal axis of symmetry in the section, which will not be a structural axis of symmetry in the structural calculation.

c) 1 card if the Y axis is an axis of symmetry in the section.

YSYM

- YSYM [A4]

  Command

This card is used if the first axis of coordinate of the model, i.e. the local y axis for the beam element, is a thermal and structural axis of symmetry. When creating the .TEM file, the area of the fibres is simply multiplied by 2. This card can only be used when the structural model is a 2D model. Another possible option is not to use the YSYM command and divide the loads applied on the structure by 2[12].

The YSYM command can be used simultaneously with one or several REALSYM commands.

[12] It is even simpler to model the complete section and not to bother about symmetries. This will nevertheless slightly increase the run times and could be an issue, for example, with HASEMI fires.

d) 1 card for each axis of symmetry in a cavity.

SYMVOID, *nno1, nno2, nvoid*

- SYMVOID [A7]
  Command
- *nno1* [integer]
  Number of the first node defining the axis of symmetry
- *nno2* [integer]
  Number of the second node defining the axis of symmetry
- *nvoid* [integer]
  Number of the cavity in where this axis of symmetry must be considered.

This card has only an effect on the thermal calculation (through the calculation of the view factors in the cavity). If this axis is also a real axis of symmetry, a REALSYM command must also be defined for this axis. It is theoretically possible to have as much as 4 different axes of symmetry in a cavity, but the validity of the procedure has been verified only with 2 axis of symmetry in a cavity.

e) 1 card

- END_SYM [A7]
  Command to indicate the end of this series

SERIES 17: PRECISION

a) 1 card

PRECISION, *precision*

- PRECISION [A9]
  Command
- *precision* [real]
  Small value which must be reached to have convergence.

The value of *precision* is used as a small number at different locations in SAFIR. A good value depends on the type of model that is being analyzed and the experience of the user is useful to choose a good value. 2×10^{-3} is normally acceptable for thermal calculations.

SERIES 18: MATERIALS

a) 1 card

- MATERIAL [A8]

  Command

All *nmat* materials (see Series 8) have to be described here. The description of a material comprises:

- A number, from 1 to *nmat*. the number is defined automatically by the order of appearance, from 1 for the first material to *nmat* for the last one.
- A name, given in cards b). Different materials in a model can have the same name (with the exception of USERx, see below). This will be the case if similar materials have different properties.
- Materials properties. They are given in cards c).

The descriptions given hereafter give the cards b) and c) for the materials that can be used in a thermal analysis.

INSULATION: material with constant thermal properties

b) 1 card

- INSULATION [A10]

  Name of the material

c) 1 card

k, c, rho, w, h_h, h_c, ε

- *k* [real]

  Thermal conductivity, in W/mK
- *c* [real]

  Specific heat, in J/m³K
- *rho* [real]

  Specific mass of the dry material, in kg/m³
- *w* [real]

  Water content, in kg/m³
- h_h [real]

  Coefficient of convection on heated surfaces, in W/m²K
- h_c [real]

  Coefficient of convection on unheated surfaces, in W/m²K
- ε [real]

  Emissivity (no dimension)

USERx: material with some thermal properties that vary with temperature.

b) 1 card

USERx*, ntemperature*

- USERx [A5]

  Name of the material. Five different materials are possible, namely USER1, USER2, USER3, USER4 and USER5. As an exception to all other material types, only one material number can have the name USER1, only one material can have the name USER2, etc.

- *ntemperature* [real]

  Number of temperatures at which the thermal properties are given. Properties are given at a certain number of temperatures, given in increasing order. Linear interpolation is made for intermediate temperatures.

c) *ntemperature* cards

Card 1.

T, k, c, rho, w, h_h, h_c, ε, r

- *T* [real]

  First[13] temperature (in degree Celsius) at which thermal properties are given

- *k* [real]

  Thermal conductivity at *T*, in W/mK

- *c* [real]

  Specific heat at *T*, in J/m³K

- *rho* [real]

  Specific mass of the dry material at *T*, in kg/m³

- *w* [real]

  Water content, in kg/m³

- h_h [real]

  Coefficient of convection on heated surfaces, in W/m²K

- h_c [real]

  Coefficient of convection on unheated surfaces, in W/m²K

- ε [real]

  Emissivity (no dimension)

- *r* [real]

  Any positive value (and 0) will force *k, c, rho* to be non reversible. This means that, during cooling from a maximum temperature T_{max}, these properties will keep the value that was valid for T_{max}. Any negative value will force these properties to be reversible, which means that the value of the property only depends on its current temperature, be it during heating or during cooling.

[13] and, therefore, lowest

Cards 2 to *ntemperature*.

T, k, c, rho

- *T* [real]

  Temperature (in degree Celsius) at which thermal properties are given

- *k* [real]

  Thermal conductivity at *T*, in W/mK

- *c* [real]

  Specific heat at *T*, in J/m³K

- *rho* [real]

  Specific mass of the dry material at *T*, in kg/m³

As an example, the following 12 cards have been used to represent concrete of EN 1992-1-2 according to the French National Annex.

```
   USER1    12
    0.   2.0000    900.   2300.    46.   25.   4.   0.7   1.
   50.   1.8801    900.   2300.
  100.   1.7656    900.   2300.
  115.   1.7323    915.   2300.
  140.   1.6778    940.   2286.
  160.   1.1570    960.   2276.
  200.   1.1108   1000.   2254.
  400.   0.9072   1100.   2185.
  600.   0.7492   1100.   2145.
  800.   0.6368   1100.   2105.
 1000.   0.5700   1100.   2064.
 1200.   0.5488   1100.   2024.
```

GYPSUM: Gypsum plaster boards

b) 1 card

- `X_GYPSUM or C_GYPSUM` [A8]

  Name of the material

c) 1 card

h_h, h_c, ε

- h_h [real]

  Coefficient of convection on heated surfaces, in W/m²K

- h_c [real]

  Coefficient of convection on unheated surfaces, in W/m²K

- ε [real]

  Emissivity (no dimension)

C_GYPSUM has a 20℃ density of 732 kg/m³, whereas X _GYPSUM has a 20℃ density of 648kg/m³.

Gypsum type materials lead to a slow convergence of the iterations during the time integration process because of the various peaks in the curve of equivalent specific heat. A time step as small as 1 second may be required.

CONCRETE of EN 1992-1-2:

b) 1 card

- `CALCONC_EN or SILCONC_EN` [A10]

  Name of the material. `CALCONC_PR` , `SILCONC_PR` , `CALCON_ETC` , `SILCON_ETC` can also be used; they have the same thermal properties.

c) 1 card

rho, w, h_h, h_c, ε, r

- *rho* [real]

  Specific mass of concrete (including moisture content), in kg/m³

- *w* [real]

  Water content, in kg/m³

- h_h [real]

  Coefficient of convection on heated surfaces, in W/m²K

- h_c [real]

  Coefficient of convection on unheated surfaces, in W/m²K

- ε [real]

  Emissivity (no dimension)

- *r* [real]

  Parameters that allows tuning the thermal conductivity between the lower limit and the upper limit, see clause 3.3.3 of EN 1992-1-2. At any temperature *T*, the thermal conductivity will be interpolated between the lower limit k_L and the upper limit k_U by the following equation:

  $$k = k_L + r\left(k_U - k_L\right)$$

CARBON STEEL of EN 1993-1-2 or EN 1992-1-2:

b) 1 card

- `STEELEC3EN or STEELEC2EN` [A10]

  Name of the material

c) 1 card

h_h, h_c, ε

- h_h [real]

  Coefficient of convection on heated surfaces, in W/m²K

- h_c [real]

  Coefficient of convection on unheated surfaces, in W/m²K

- ε [real]

  Emissivity (no dimension)

STAINLESS STEEL of EN 1993-1-2:

b) 1 card

- `SLS1.4301, SLS1.4401, SLS1.4404, SLS1.4571, SLS1.4003, SLS1.4462, SLS1.4311` [A9]

  Name of the material

c) 1 card

h_h, h_c, ε

- *h_h* [real]

  Coefficient of convection on heated surfaces, in W/m²K

- *h_c* [real]

  Coefficient of convection on unheated surfaces, in W/m²K

- ε [real]

  Emissivity (no dimension)

ALUMINIUM of EN 1999-1-2:

b) 1 card

- `AL6061_T6, AL6063_T6, AL5083_O, AL5083_H12` [A9]

  Name of the material

c) 1 card

h_h, h_c, ε

- *h_h* [real]

  Coefficient of convection on heated surfaces, in W/m²K

- *h_c* [real]

  Coefficient of convection on unheated surfaces, in W/m²K

- ε [real]

  Emissivity (no dimension)

WOOD of EN 1995-1-2:

b) 1 card

- WOODEC5 [A10]

  Name of the material

c) 1 card

rho, w, h_h, h_c, ε, r, l, m, n

- *rho* [real]

  Specific mass of wood (including moisture content), in kg/m³

- *w* [real]

  Water content, in % of the dry mass

  If, for example, *rho* = 450 kg/m³ and *w* = 12, the density of dry wood is 450/(1+12/100) = 401.7 kg/m³

- h_h [real]

  Coefficient of convection on heated surfaces, in W/m²K

- h_c [real]

  Coefficient of convection on unheated surfaces, in W/m²K

- ε [real]

  Emissivity (no dimension)

- *r* [real]

  Ratio of conductivity along the grain by conductivity perpendicular to the grain (no dimension), usually greater than 1.

- *l, m, n* [3 real]

  The vector <*l*;*m*;*n*> gives the direction of the grain

  Direction of the grain is essential for 3D analyses.

  Vector components for 2D analysis are given in the order <y;z;x>. Most 2D analyses are performed on the section of a beam or a column and the grain is perpendicular to the section, with a vector <0;0;1>. If grain direction is parallel to z-axis, for example, then the vector should be <0;1;0>.

  In 2D problems, two different possibilities exist to define an isotropic material:

  1) specifying a vector <0;0;1>
  2) specify *r* = 1

SERIES 19: TIME DISCRETISATION

a) 1 card

- `TIME` [A4]

  Command

b) at least 1 card, maximum 100 cards

dt, t

- *dt* [real]

  Time step (in seconds) used for the time integration

- *t* [real]

  Time (in seconds) until which *dt* is used

c) 1 card

- `END_TIME` [A8]

  Command

The following example leads to a time step of 12 seconds being used during the first hour, then 20 seconds up 2 hours, time when the simulation will be stopped:

```
     TIME
                12.     3600.
                20.     7200.
END_TIME
```

SERIES 20: OUTPUT RESULTS

a) 1 card

- `OUTPUT` [A6]

  Command

b) 1 card

- `TIMEPRINT` [A9]

  Command

c) at least 1 card, maximum 16 cards

dt, t

- *dt* [real]

  Time step (in seconds) used for writing the calculated temperatures in the .OUT file and in the eventual .TEM or .TSH files.

- *t* [real]

  Time (in seconds) until which *dt* is used

d) 1 card

- `END_TIMEPR` [A10]

  Command to indicate the end of the `TIMEPRINT` cards.

The following example leads to a time step of 1 minutes being used for writing the temperatures during the first hour, then 15 minutes up 2 hours:

```
OUTPUT
  TIMEPRINT
                60.     3600.
               900.     7200.
END_TIMEPR
```

e) 1 card (optional)

- PRINTDEPL [A9]

  Writes the increment of temperatures at each iteration of each time step of the time integration, see series 19. This command is used only for debugging convergence problems or to decide of an appropriate time step for the time integration. It is not used for usual calculations because it slightly increases the run time but, first of all, it increases significantly the size of the .OUT file.

f) 1 card (optional)

- PRINTFHE [A9]

  Writes the out of balance thermal loads at the nodes at each iteration of each time step of the time integration, see series 19. This command is used only for debugging convergence problems or to decide of an appropriate time step for the time integration. It is not used for usual calculations because it slightly increases the run time but, first of all, it increases significantly the size of the .OUT file.

g) 1 card

1 blank line to indicate that the input file is finished. [-]

Finally, the example input file given hereafter yields the temperatures calculated after 2 hours shown on Figure 25.

```
This file has been created as an example for the users manual of SAFIR
Date: 16/12/2010
Author: Jean-Marc Franssen

     NNODE   16
      NDIM    2
   NDOFMAX    1
EVERY_NODE    1
  END_NDOF

TEMPERAT
      TETA       0.9
  TINITIAL      20.0

      NMAT    2

ELEMENTS
     SOLID    9
        NG    2
     NVOID    0
  END_ELEM
```

```
NODES
      NODE    1     0.000     0.000
     GNODE    4     0.000     0.120
    REPEAT    4     0.025     0.010    3

FIXATIONS
   END_FIX

NODOFSOLID
      ELEM    1    1    2    6    5    1   0.
     GELEM    3    9   10   14   13    1   0.    4
    REPEAT    3    1                           2
   NEW_MAT    5    2

FRONTIER
         F    1      FISO        NO        NO        NO
        GF    7      FISO        NO        NO        NO    3
         F    1        NO        NO        NO       F20
        GF    3        NO        NO        NO       F20    1
 END_FRONT

  SYMMETRY
   END_SYM

 PRECISION     0.002

 MATERIALS
STEELEC3EN
               25.        4.       0.7
CALCONC_EN
              2300.       46.       25.        4.       0.8        0.

      TIME
               12.     3600.
               20.     7200.
  END_TIME

OUTPUT
 TIMEPRINT
               60.     3600.
              900.     7200.
END_TIMEPR
```

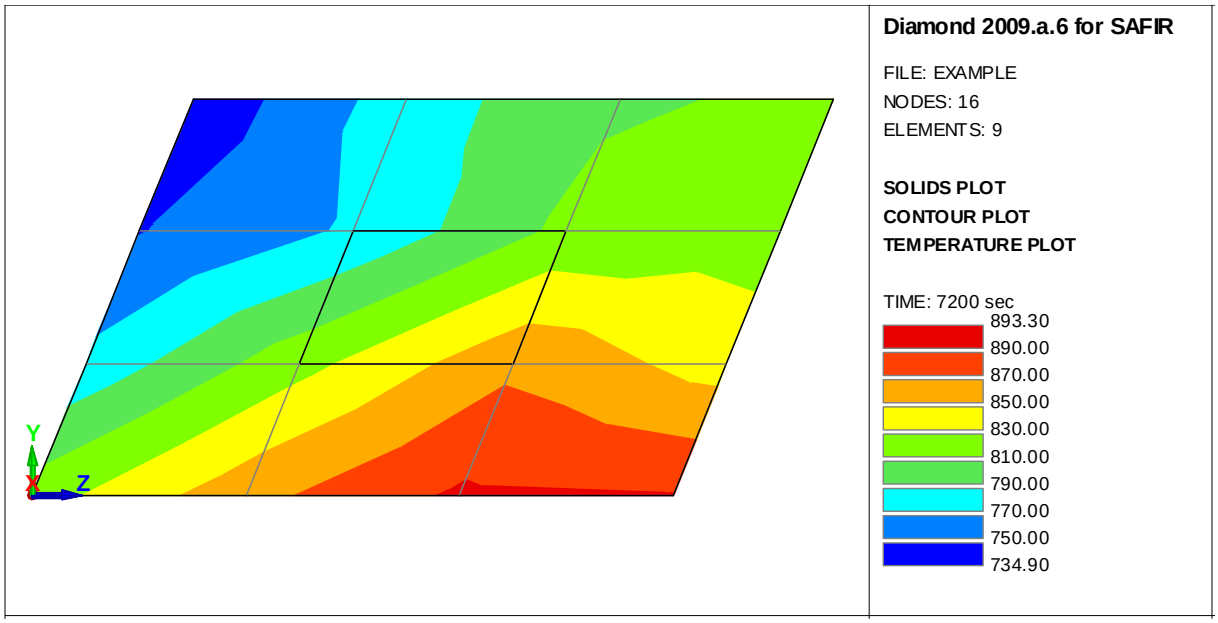


Figure 25: calculated temperatures

D.1.3 Detailed description of the file describing the HASEMI fire(s)

If a HASEMI function is used in a FLUX command of the input file, a file called *hasemi.txt* must be present in the same folder as the input file. This file describes the local Hasemi fire(s). The format of this file is described in this section.

SERIES 1: COMMENTS

a) Multiple cards (can be 0)
   As many lines as required may be introduced by the user at the beginning of the file in order to comment the content of the file. These comments will not influence in any manner the results of the calculation. They will help understand what the file is about, which may be particularly useful if the file is found and opened several years after it has been created.

b) 1 card
   1 blank line to indicate that the comments are finished. [-]

SERIES 2: NUMBER OF FIRE(S)

a) 1 card
   NFIRE*, nfire*
   - NFIRE [A5]
     Command
   - *nfire* [integer]
     Number of localised fires that will influence the model in the thermal analysis.

The following series 3 to 6 have to be written *nfire* times, once for each localised fire.

SERIES 3: POSITION OF THE FIRE

a) 1 card
   FIRE_POS*, posi1, posi2, [posi3]*
   - FIRE_POS [A8]
     Command
   - *posi1* [real]
     Position of the fire (in meters) along the first axis of coordinates of the structural analysis.
   - *posi2* [real]
     Position of the fire (in meters) along the second axis of coordinates of the structural analysis.

- *[posi3] (optional)* [real]

  Position of the fire (in meters) along the third axis of coordinates of the structural analysis. Only required for 3D structural models.

SERIES 4: CEILING HEIGHT

a) 1 card

`HEIGHT`, *hc*

- `HEIGHT` [A6]

  Command

- *hc* [real]

  Vertical distance (in meters) from the fire source to the ceiling.

  Only this value is introduced in the local fire model of EN 1991-1-2 and will influence the severity of the attack of the fire on the structure. The vertical position of the fire given in Series 3 (*posi2* for 2D structural models and *posi3* for 3D structural models) is in fact not used in the model.

SERIES 5: DIAMETER

a) 1 card

- `DIAMETER` [A8]

  Command

b) at least 2 cards, maximum 100 cards

t, diameter

- *t* [real]

  Time (in seconds) when the diameter is given. The evolution of the diameter as a function of time will be linearly interpolated between the values given in this series.

- *diameter* [real]

  Value of the diameter (in meters) of the fire at time *t*

c) 1 card

- `END_DIAM` [A8]

  Command

SERIES 6: RATE OF HEAT RELEASE

a) 1 card

- RHR [A8]

  Command

b) at least 2 cards, maximum 100 cards

t, rhr

- *t* [real]

  Time (in seconds) when the rate of heat release is given. The evolution of the rate of heat release as a function of time will be linearly interpolated between the values given in this series.

- *rhr* [real]

  Value of the rate of heat release (in W) of the fire at time *t*

c) 1 card

- END_RHR [A7]

  Command

Hereafter is given an example of a *hasemi.txt* file.

```
Description of the local fire by the Hasemi method.
There are 2 local fires.
This file has been used with the structure described in the file TEST.IN

     NFIRE    2                   Number of localised fires

  FIRE_POS   0.        0.  0.5
    HEIGHT                 2.5    Distance between the fire and the ceiling
  DIAMETER
            0.        0.          Time (s) - Fire source diameter (m)
          300.        3.
          600.        4.
  END_DIAM
       RHR
            0.        0.          Time (s) - Rate of heat release (W)
          600.   900000.
   END_RHR

  FIRE_POS   1.5       0.  0.5
    HEIGHT                 2.5
  DIAMETER
            0.        0.
          100.        0.
          400.        3.
          600.        4.
  END_DIAM
       RHR
            0.        0.
          100.        0.
          500.   900000.
          600.   900000.
   END_RHR
```

D.2 Input techniques

There are four known techniques for creating input files.

D.2.1 Text editor

Any text editor that allows creating, opening, modifying and saving text files can be used to create input files or *hasemi.txt* files.
This technique is the best choice if one or a few input data have to be modified in an existing input file. New input files can easily be created by the *SaveAs* option of the editor if the model to be analysed is very close to an existing model.
When an input file has to be created from scratch, or if extensive modifications have to be made in an existing file, it is recommended to use an editor that has sufficient capabilities such as, e.g., working in column mode or comparing files. The developers have been using ultraedit (http://www.ultraedit.com/) with great satisfaction.

D.2.2 Personal programming

Users with programming skills can write their own software for creating input files. This requires significant efforts for a program that, typically, will allow creating a very specific type of model to be created, with very little, if any, versatility. This technique can be the best choice if a very large number of similar models have to be created, for example in order to perform a parametric analysis. This technique could also be considered for a user that has only one type of structure to analyse with varying dimensions.

Any programming technique can be used, as far as it can create a text file. Spreadsheets such as Excel have been used with success.

D.2.3 Wizard

A particular tool belonging to the category described in section D.2.2 has been written by the developers. Given the name of Wizard, it allows to create in a few clicks an input file for a thermal analysis of a 2D section based on a hot rolled H section. The section can have a concrete slab on the top flange and be protected by applied thermal insulation. The catalogue of H sections of ArcelorMittal is embedded in the software.

The last version of Wizard, Wizard2004, version 4.0.0 is still distributed on the CD provided with SAFIR and can be downloaded for free from the SAFIR website (http://www.argenco.ulg.ac.be/logiciels/SAFIR/downloads-3.html). It is still compatible with

SAFIR2011 but will probably not be maintained in the future, because a similar tool has been incorporated in the tool described in Section D.2.4.

This tool is sometimes used as a quick way to create a file that has the correct format for SAFIR, to be modified subsequently with a text editor to produce a file describing the model to analyse.

D.2.4 GID

GID (http://gid.cimne.upc.es/) is a commercial pre-processor that can help creating a finite element model with a graphic user interface, see Figure 26.

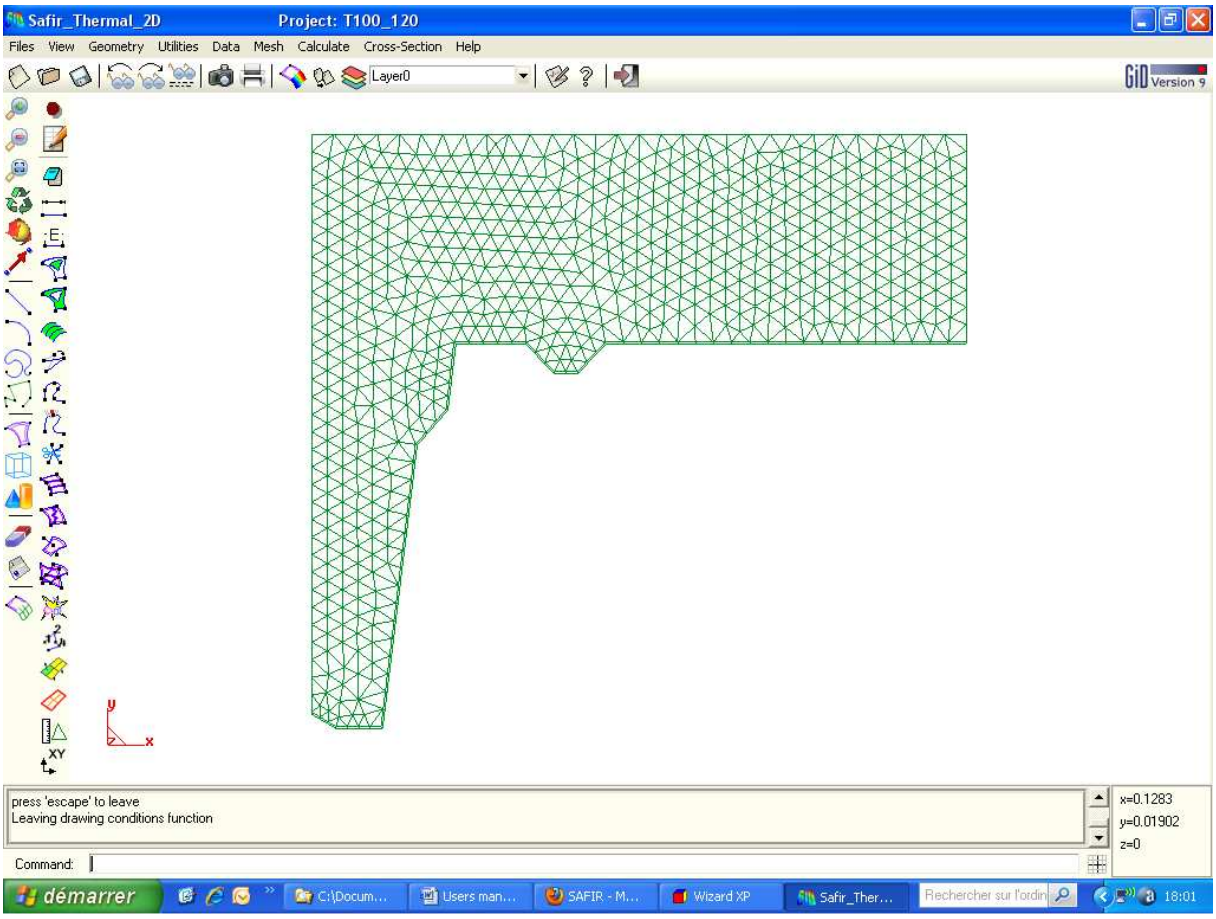


Figure 26: Graphic interface of GID

A reasonably powerful version can be downloaded and run for free, while the full professional version can be purchased with a price that is different for educational organisations or for corporate organisations.

The modules that can make GID writing the input files in the format of SAFIR (for thermal as well as for structural calculations), as well as some short tutorials, can be downloaded on the SAFIR web site (http://www.argenco.ulg.ac.be/logiciels/SAFIR/downloads-3b.html).

GID is clearly the best choice when various ands complex models have to be created. Experience nevertheless shows that users very quickly learn how to create their models in GID, but have a tendency to let the number of nodes and elements increase significantly compared to models made with a text editor. This is not detrimental for single 2D thermal calculations. More care should be exercised and more skills in using GID should be gained in order to improve the quality of the model when multiple 2D calculations have to be performed because of exposure to a local HASEMI fire or when really big 3D models have to be analysed.

D.3 Limits on input based on the stability, accuracy and practicality of the data

The main limits based on stability and accuracy are based on the link that must exist between the time step used during time integration, see Series 19 in Section D.1.2, and the size of the finite elements in the direction of the heat flow. They have been studied in detail in [1], section xxx.

Accuracy increases when the time step decreases.

Suitable values of the time step depend on the types of material that are present in the model and on the geometry of the model. With usual finite element sizes, in the order of 10 mm perpendicular to the boundary of the model in the zones near the surfaces of the models, times steps from 12 to 20 seconds have been used without any problem in concrete or steel models, protected or not. Higher values can eventually be used after the first 30 minutes of an ISO fire. When gypsum material is all or part of the model, time steps as short as 1 second may have to be used from the very first minutes of the simulation. When the section comprises one or several internal cavities, time steps in the order of 1 second must also be used from the moment when the heat wave reaches the cavities and heat exchanges by radiation in the cavities start playing a role.

D.4 Default variables - Process for setting those variables to user-defined values

There is no default variable in the model other than physical constants.

Default thermal properties are embedded for the thermal models of materials from the Eurocodes (steels, concretes, aluminums and wood) and for gypsum materials. For using other material properties, the user has to use either `INSULATION` material or `USERx` materials.

D.5 Consecutive cases

Consecutive cases can be executed one by one, if the user starts *safir2011.exe* directly from the Windows Explorer, or automatically by the SafirShell.

In any case, to the knowledge of the developers, all variable are completely re-initialized; there is no data retention from case to case. Any observation of the contrary should be reported as a bug to the developers.

E. External data files

E.1 Content and organization of external data files

There is no external data file other than those required for the input, see sections D.1.2 and D.1.3, and those created by execution of SAFIR, namely the .OUT file and eventual .TEM or .TSH files.

Terms and definitions

Card

A series of information of the input file that is read by in single READ statement by SAFIR. The rules for reading a card are the rules of the FORTRAN language. The input data of a single card can be split in the input file on several consecutive lines.

Discretisation

The process by which the user represents to structure (that, by nature, is a continuous object) into a model made of a finite number of nodes and elements.

Iteration

Because the thermal problem to be solved is non linear, SAFIR cannot calculate the temperature directly at a given time step as a function of the temperatures that were calculated at the previous time step. In order to calculate the temperature at a given time step, SAFIR has to perform several trial and error calculations during which the temperatures are progressively refined. Each of these trials and errors is called an iteration.

Line

A physical record of the input file.

Time step

A particular time during the course of the fire fat which SAFIR will calculate the temperatures (times steps used for the calculation). The time steps are chosen by the user.

User

The person who collects the input data for the problem to be solved, who creates a model representing the problem (the fire and the structure) that can be analyzed by SAFIR and who interprets the results provided by SAFIR.

List of symbols

t	Thickness of a shell finite element [m]