EDF R&D

Fluid Dynamics, Power Generation and Environment Department
Single Phase Thermal-Hydraulics Group

6, quai Watier
F-78401 Chatou Cedex

Tel: 33 1 30 87 75 40
Fax: 33 1 30 87 79 16

JUNE 2013

Code_Saturne documentation

Code_Saturne version 3.1.0 tutorial

contact: saturne-support@edf.fr

TABLE OF CONTENTS

I Introduction 5

1 Introduction . . . 6

II Simple junction testcase 7

1 General description . . . 8
1.1 Objective . . . 8
1.2 Description of the configuration . . . 8
1.3 Characteristics . . . 9
1.4 Mesh characteristics . . . 9
2 CASE 1: Basic calculation . . . 10
2.1 Calculation options . . . 10
2.2 Initial and boundary conditions . . . 10
2.3 Parameters and User routines . . . 11
2.4 Results . . . 11

III Full domain 13

1 General description . . . 14
1.1 Objective . . . 14
1.2 Description of the configuration . . . 14
1.3 Characteristics . . . 14
1.4 Mesh characteristics . . . 15
1.5 Summary of the different calculations . . . 15
2 CASE 2: Passive scalar with various boundary conditions and output management . . . 16
2.1 Calculation options . . . 16
2.2 Initial and boundary conditions . . . 17
2.3 Parameters and User routines . . . 17
2.4 Output management . . . 18

2.5 Results . . . 18
3 CASE 3: Time dependent boundary conditions and variable fluid density . . . 21
3.1 Calculation options . . . 21
3.2 Initial and boundary conditions . . . 21
3.3 Variable Density . . . 22
3.4 Parameters . . . 22
3.5 User routine . . . 23
3.6 Output management . . . 23
3.7 Calculation restart . . . 24
3.8 Results . . . 24
4 CASE 4: Head losses, parallelism and spatial average . . . 27
4.1 Calculation options . . . 27
4.2 Initial and boundary conditions . . . 27
4.3 Variable Density . . . 28
4.4 Head losses . . . 28
4.5 Parameters . . . 28
4.6 User routines . . . 29
4.7 Output management . . . 30
4.8 Results . . . 31

IV Stratified junction 33

1 General description . . . 34
1.1 Objective . . . 34
1.2 Description of the configuration . . . 34
1.3 Characteristics . . . 34
1.4 Mesh characteristics . . . 35
2 CASE 5: Stratified junction . . . 35
2.1 Calculation options . . . 35
2.2 Initial and boundary conditions . . . 36
2.3 Variable density and dynamic viscosity . . . 36
2.4 Parameters . . . 36
2.5 Output management . . . 37
2.6 User routines . . . 37
2.7 Results . . . 38

V Three 2D disks **41**

1 General description . . . 42
1.1 Objective . . . 42
1.2 Remarks . . . 42
1.3 Description of the configuration . . . 42
1.4 Characteristics . . . 43
1.5 Mesh characteristics . . . 44
2 CASE 6: 3 2D disks . . . 45
2.1 Parameters . . . 45
2.2 Output management . . . 45
2.3 Coupling computation . . . 46
2.4 Results . . . 46

VI Step by step solution **48**

1 Solution for case1 . . . 49
2 Solution for case2 . . . 87
3 Solution for case3 . . . 117
4 Solution for case4 . . . 125
5 Solution for case5 . . . 128
6 Solution for case6 . . . 139
6.1 Launching the SYRTHES computation alone . . . 140
6.2 Launching the *Code_Saturne* computation alone . . . 149
6.3 Launching the *Code_Saturne*-SYRTHES coupling computation . . . 156

Part I

Introduction

1 Introduction

Code_Saturne is a system designed to solve the Navier-Stokes equations in the cases of 2D, 2D axisymmetric or 3D flows. Its main module is designed for the simulation of flows which may be steady or unsteady, laminar or turbulent, incompressible or potentially dilatable, isothermal or not. Scalars and turbulent fluctuations of scalars can be taken into account. The code includes specific modules, referred to as "specific physics", for the treatment of lagrangian particle tracking, semi-transparent radiative transfer, gas, pulverized coal and heavy fuel oil combustion, electricity effects (Joule effect and electric arcs) and compressible flows. *Code_Saturne* relies on a finite volume discretization and allows the use of various mesh types which may be hybrid (containing several kinds of elements) and may have structural non-conformities (hanging nodes).

The present document is a tutorial for *Code_Saturne* version 3.1.0. It presents five simple test cases and guides the future *Code_Saturne* user step by step into the preparation and the computation of the cases.

The test case directories, containing the necessary meshes and data are available in the `examples` directory.

This tutorial focuses on the procedure and the preparation of the *Code_Saturne* computations. For more elements on the structure of the code and the definition of the different variables, it is higly recommended to refer to the user manual.

Code_Saturne is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. *Code_Saturne* is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

Part II

Simple junction testcase

1 General description

The first thing to do before running *Code_Saturne* is to prepare the computation directories. In this first example, the study directory `T_junction/` will be created, containing a single calculation sub-directory `case1`. This is done by typing the command:

```
$ code_saturne create -s T_junction -c case1
```

The mesh files should be copied in the directory `MESH/` as below:

```
$ cd T_junction/MESH/
$ cp ITECH_CS_training_2012/meshes/1-simple_junction/downcomer.des .
```

The *Code_Saturne* Graphic User Interface (GUI) is launched by typing the following command lines:

```
$ cd T_junction/case1/DATA
$ ./SaturneGUI &
```

1.1 Objective

The aim of this case is to train the user of *Code_Saturne* on an oversimplified 2D junction including an inlet, an outlet, walls and symmetries.

1.2 Description of the configuration

The configuration is two-dimensional.
It consists of a simple junction as shown on figure II.1. The flow enters through a hot inlet into a cold environment and exits as indicated on the same figure. This geometry can be considered as a very rough approximation of the cold branch and the downcomer of the vessel in a nuclear pressurized water reactor. The effect of temperature on the fluid density is not taken into account in this first example.

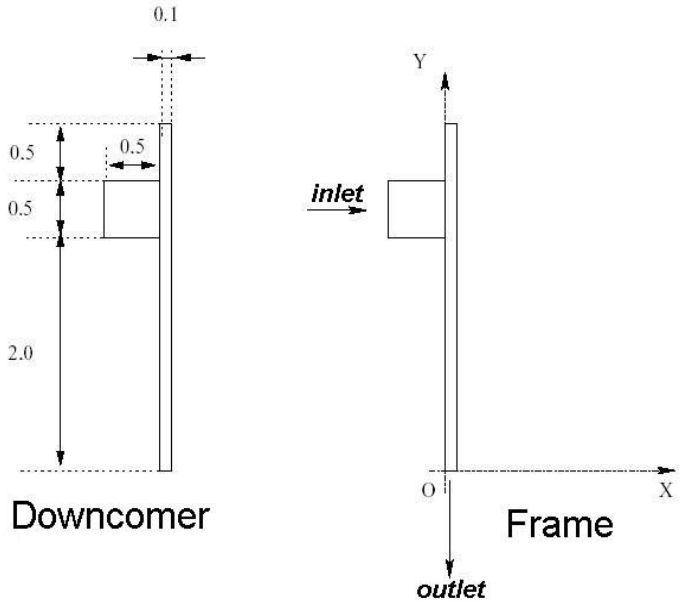


Figure II.1: Geometry of the downcomer

1.3 Characteristics

Characteristics of the geometry and the flow:

Height of downcomer	$H = 3.00\ m$
Thickness of downcomer	$E_d = 0.10\ m$
Diameter of the cold branch	$D_b = 0.50\ m$
Inlet velocity of fluid	$V = 1\ m.s^{-1}$

Physical characteristics of fluid:
The initial water temperature in the domain is equal to 20°C. The inlet temperature of water in the cold branch is 300°C. Water characteristics are considered constant and their values taken at 300°C and $150 \times 10^5\ Pa$:

- density: $\rho = 725.735\ kg.m^{-3}$
- dynamic viscosity: $\mu = 0.895 \times 10^{-4}\ kg.m^{-1}.s^{-1} = 8.951 \times 10^{-5}\ Pa.s$
- specific heat: $C_p = 5\,483\ J.kg^{-1}.°C^{-1}$
- Thermal Conductivity $= 0.02495\ W.m^{-1}.K^{-1}$

1.4 Mesh characteristics

Figure II.2 shows a global view of the downcomer mesh. This two-dimensional mesh is composed of 700 cells, which is very small compared to those used in real studies. This is a deliberate choice so that tutorial calculations run fast.

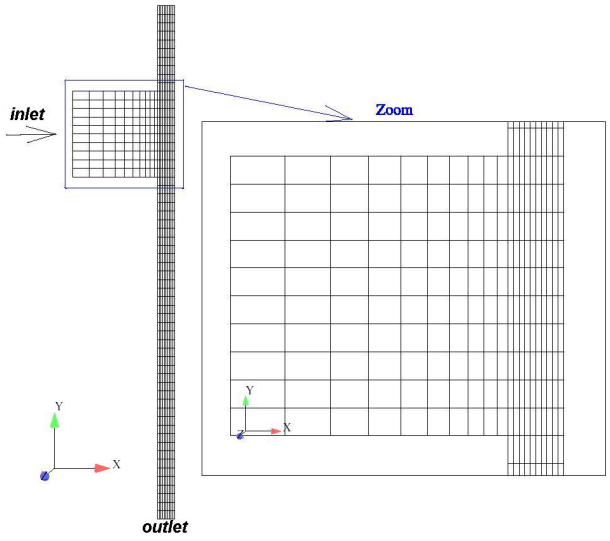


Figure II.2: Geometry of the downcomer

Note that here the case is two-dimensional but *Code_Saturne* always operates on three-dimensional mesh elements (cells). The present mesh is composed of a layer of hexahedrons created from the 2D mesh shown on figure II.2 by extrusion (elevation) in the z direction. The virtual planes parallel to Oxy will have "sliding" ("symmetry") conditions to account for the two-dimensional character of the configuration.

Type: structured mesh

Coordinates system: cartesian, origin on the edge of the main pipe at the outlet level, on the nozzle side (figure II.1)

Mesh generator used: SIMAIL

Color definition: see figure II.3. To specifiy boundary conditions on the boundary faces of the mesh, the latter have to be identified. It is commonly done by assigning an integer to each of them, characteristic of the boundary region they belong to. This integer is refered to as "color" or "reference".

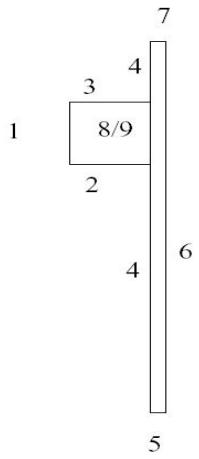


Figure II.3: Colors of the boundary faces

2 CASE 1: Basic calculation

2.1 Calculation options

Most of the options used in this calculation are default options of *Code_Saturne*.

→ Flow type: steady flow

→ Turbulence model: $k - \epsilon$

→ Scalar(s): 1 - temperature

→ Physical properties: uniform and constant

2.2 Initial and boundary conditions

→ Initialization: none (default values)

The boundary conditions are defined as follows:

- **Flow inlet**: Dirichlet condition, an inlet velocity of 1 $m.s^{-1}$ and an inlet temperature of 300°C are imposed
- **Outlet**: default values
- **Walls**: default values

Figure II.3 shows the colors used for boundary conditions and table II.1 defines the correspondance between the colors and the type of boundary condition to use.

Do not forget to enter the value of the hydraulic diameter, adapted to the current inlet (used for turbulence entry conditions).

Colors	Conditions
1	Inlet
5	Outlet
2 3 4 6 7	Wall
8 9	Symmetry

Table II.1: Boundary conditions and associated references

2.3 Parameters and User routines

All parameters necessary to this study can be defined through the Graphical Interface without using any user Fortran files.

Calculation control parameters	
Pressure-Velocity coupling	SIMPLE algorithm
Number of iterations	30
Relaxation coefficient	0.9
Output period for post-processing files	1

2.4 Results

Figure II.4 presents the results obtained at different iterations in the calculation. They were plotted from the post-processing files, with EnSight.

Note: since the "steady flow" option has been chosen, the evolution of the flow iteration after iteration has no physical meaning. It is merely an indication of the rapidity of convergence towards the (physical) steady state.

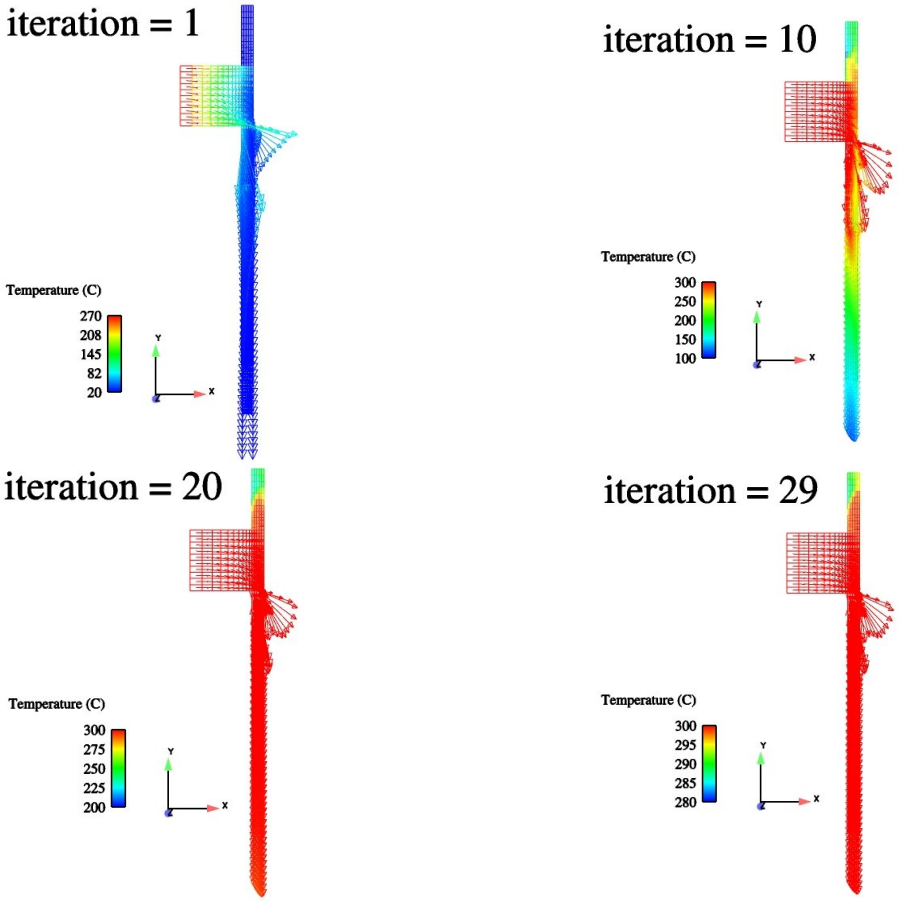


Figure II.4: Water velocity field colored by temperature at different time steps

Part III

Full domain

1 General description

1.1 Objective

This aim of this case is to tackle the merging of initially separate meshes into a single fluid domain. The questions of mesh joining and hanging nodes will be addressed. The test case will then be used to present more complex calculations, with time dependent variables and Fortran user routines.

1.2 Description of the configuration

The fluid domain is composed of three separate meshes, very roughly representing elements of a nuclear pressurized water reactor vessel:

- the downcomer
- the vessel's bottom
- the lower core plate and core

Figure III.1 represents the complete domain. The flow circulates from the top left horizontal junction to the right vertical outlet.

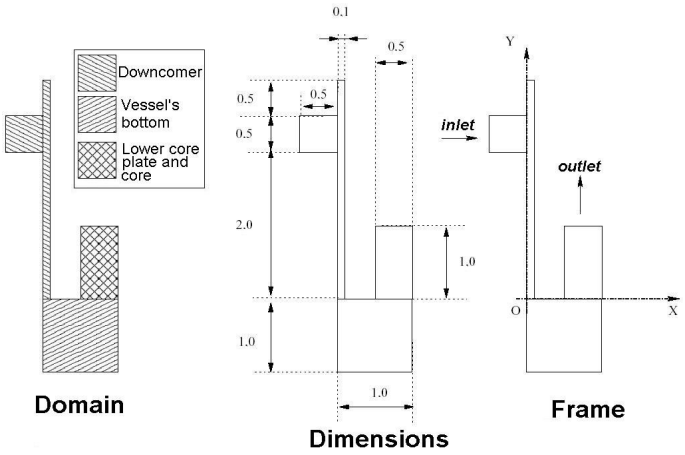


Figure III.1: Geometry of the complete domain

1.3 Characteristics

The characteristics of the geometry and the flow are:

Height of downcomer	$H = 3.00\ m$
Thickness of downcomer	$E_d = 0.10\ m$
Diameter of the inlet cold branch	$D_b = 0.50\ m$
Height of vessel's bottom	$H_{fc} = 1.00\ m$
Width of vessel's bottom	$l_{fc} = 1.00\ m$
Height of core above the lower core plate	$H_{pic} = 1.00\ m$
Width of core above the lower core plate	$l_{pic} = 0.50\ m$
Inlet velocity of fluid	$V = 1\ m.s^{-1}$

Physical characteristics of fluid:
The initial water temperature in the domain is equal to 20°C. The inlet temperature of water in the cold branch is 300°C. Water characteristics are considered constant[1] and their values taken at 300°C and 150×10^5 Pa, except density which is considered variable in cases 3 and 4:

- density: $\rho = 725.735\ kg.m^{-3}$
- dynamic viscosity: $\mu = 0.895 \times 10^{-4}\ kg.m^{-1}.s^{-1} = 8.951 \times 10^{-5}\ Pa.s$
- heat capacity: $C_p = 5\,483\ J.kg^{-1}.°C^{-1}$
- Thermal Conductivity $= 0.02495\ W.m^{-1}.K^{-1}$

1.4 Mesh characteristics

Figure III.2 shows a global view of the mesh and some details of the joining zones, to show that *Code_Saturne* can deal with hanging nodes. This mesh is composed of 1 650 cells, which is very small compared to those used in real studies. This is a deliberate choice so that tutorial calculations run fast.

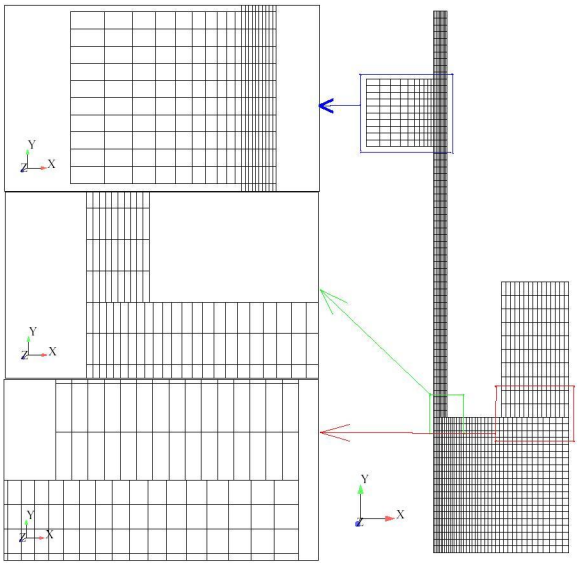


Figure III.2: View of the full domain mesh with zoom on the joining regions

Type: block structured mesh

Coordinates system: cartesian, origin on the edge of the main pipe at the outlet level, on the nozzle side (figure III.2)

Mesh generator used: SIMAIL and mesh joining with the Preprocessor of *Code_Saturne* (in order to deal with hanging nodes)

Color definition: see figure III.3

1.5 Summary of the different calculations

Three cases will be studied with this geometry. The following table gives a summary of their different characteristics.

[1] which makes temperature a passive scalar ... but it is only for simplification purposes

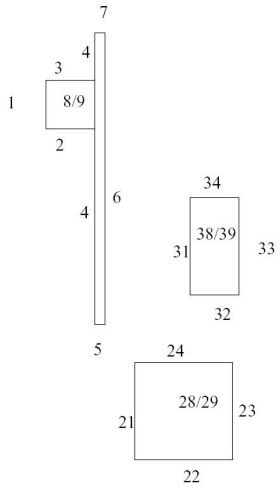


Figure III.3: Colors of the boundary faces

CASE	characteristics
CASE 2	unsteady flow, additionnal passive scalar, output management
CASE 3	same as case 2 with time dependent boundary conditions, fluid density depending on the temperature and calculation restart
CASE 4	same as case 3 with head losses, parallelism and spatial average

Remark: In this case, you must add three meshes which have to be joined. In order to join the three meshes, you must add a selection criteria in the box *Selection criteria*. In this case, only faces of colors 5, 24 and 32 are liable to be joined (different colors can be entered on a single line, separated by comma).
You can verify the quality of your mesh in *Mesh quality criteria*.

2 CASE 2: Passive scalar with various boundary conditions and output management

2.1 Calculation options

Some options are similar to case 1:

→ Turbulence model: $k - \epsilon$

→ Scalar(s): 1 - temperature

→ Physical properties: uniform and constant

The new options are:

→ Flow type: unsteady flow

→ Time step: uniform and constant

→ Scalar(s): 2 - passive scalar[2]

→ Management of monitoring points

[2]could correspond to a tracer concentration for instance

2.2 Initial and boundary conditions

→ Initialization: 20°C for temperature (default value)
10 for the passive scalar

The boundary conditions are defined in the user interface and depend on the boundary zone.

- **Flow inlet**: Dirichlet condition, an inlet velocity of 1 $m.s^{-1}$, an inlet temperature of 300°C and an inlet value of 200 for the passive scalar are imposed
- **Outlet**: default value
- **Walls**: velocity, pressure and thermal scalar: default value
passive scalar: different conditions depending on the color and geometric parameters

In order to test the ability to specify boundary condition regions in the Graphical Interface, various conditions will be imposed for the passive scalar, as specified in the following table:

Wall	Nature	Value
wall_1	Dirichlet	0
wall_2	Dirichlet	5
wall_3	Dirichlet	0
wall_4	Dirichlet	25
wall_5	Dirichlet	320
wall_6	Dirichlet	40

The "wall_1" to "wall_6" regions are defined as follows, through color references and geometric localization:

Label	Color and geometric parameters
wall_1	24 and $0.1 \leqslant x$ and $x \leqslant 0.5$
wall_2	2 or 3
wall_3	4 or 7 or 21 or 22 or 23
wall_4	6 and $y > 1$
wall_5	6 and $y \leqslant 1$
wall_6	31 or 33

Figure III.3 shows the colors used for boundary conditions and table III.1 defines the correspondance between the colors and the type of boundary condition to use.

Colors	Conditions
1	Inlet
34	Outlet
2 3 4 6 7 21 22 23 24 31 33	Wall
8 9 28 29 38 39	Symmetry

Table III.1: Boundary faces colors and associated references

2.3 Parameters and User routines

All parameters necessary to this study can be defined through the Graphical Interface without using any user Fortran files.

Calculation control parameters	
Pressure-Velocity coupling	SIMPLEC algorithm
Number of iterations	300
Reference time step	0.05
Output period for post-processing files	2

2.4 Output management

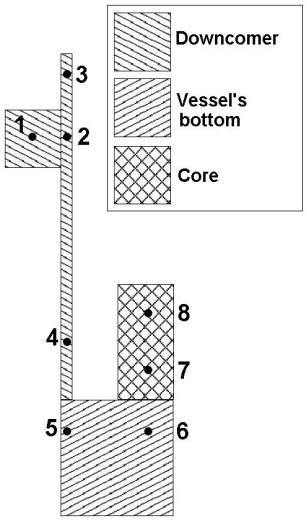
In this case, different aspects of output management will be addressed.

By default in the Graphical Interface, all variables are set to appear in the listing, the post-processing and the chronological records. This default choice can be modified by the user.

In this case, the *Pressure*, the *Tubulent energy* and the *Dissipation* will be removed from the listing file.

The *Courant number* (CFL) and *Fourier number* will be removed from the post-processing results[3].

Eventually, probes will be defined for chronological records, following the data given in figure III.4. Then the *total pressure* will be deactivated for all probes and the *Velocity U* will only be activated on probes 1, 2, 6, 7 and 8.



Probe n^o.	x (m)	y (m)	z (m)
1	-0.25	2.25	0
2	0.05	2.25	0
3	0.05	2.75	0
4	0.05	0.5	0
5	0.05	-0.25	0
6	0.75	-0.25	0
7	0.75	0.25	0
8	0.75	0.75	0

Figure III.4: Position and coordinates of probes in the full domain

In addition the domain boundary will be post-processed. This allows to check the boundary conditions, and especially that of the passive scalar.

2.5 Results

Figure III.5 shows the boundary domain colored by the passive scalar boundary conditions. The different regions of boundary conditions defined earlier can be checked.

Figure III.6 presents results obtained at different times of the calculation. They were plotted from the post-processing files, with EnSight.

[3]this can be very useful to save some disk space if some variables are of no interest, as post-processing files can be large

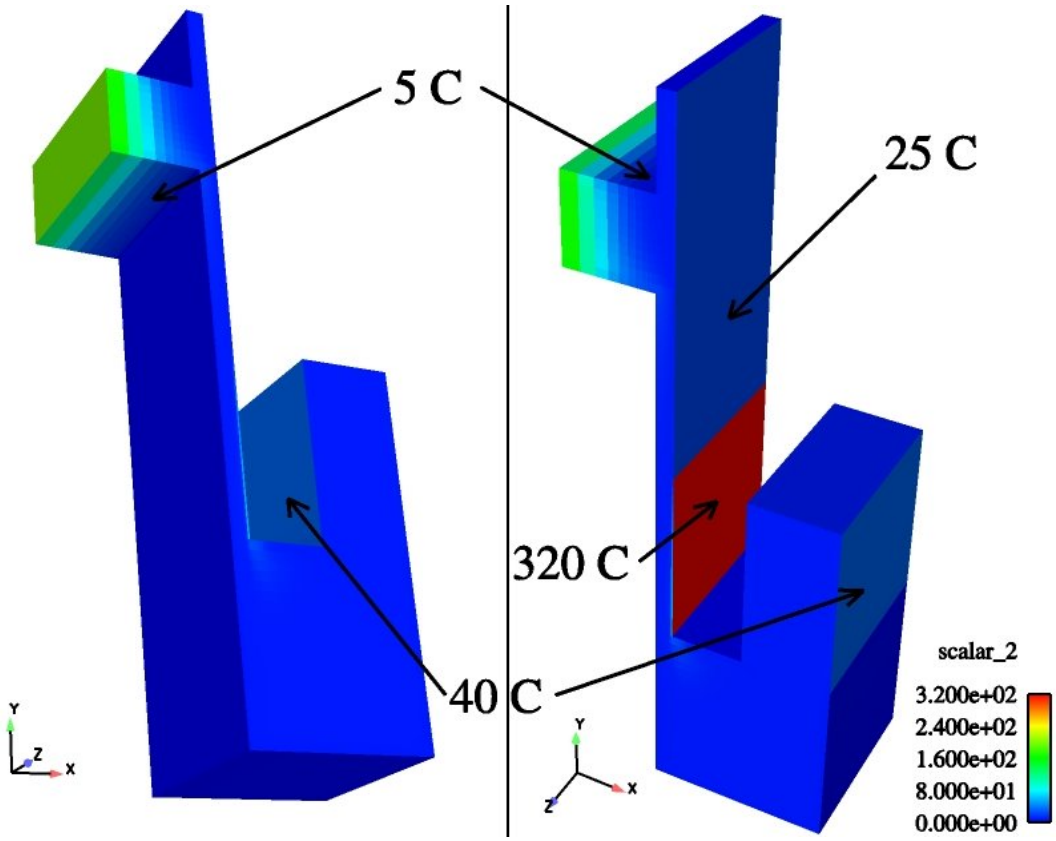


Figure III.5: View of the boundary domain colored by the scalar_2 variable - Case 2

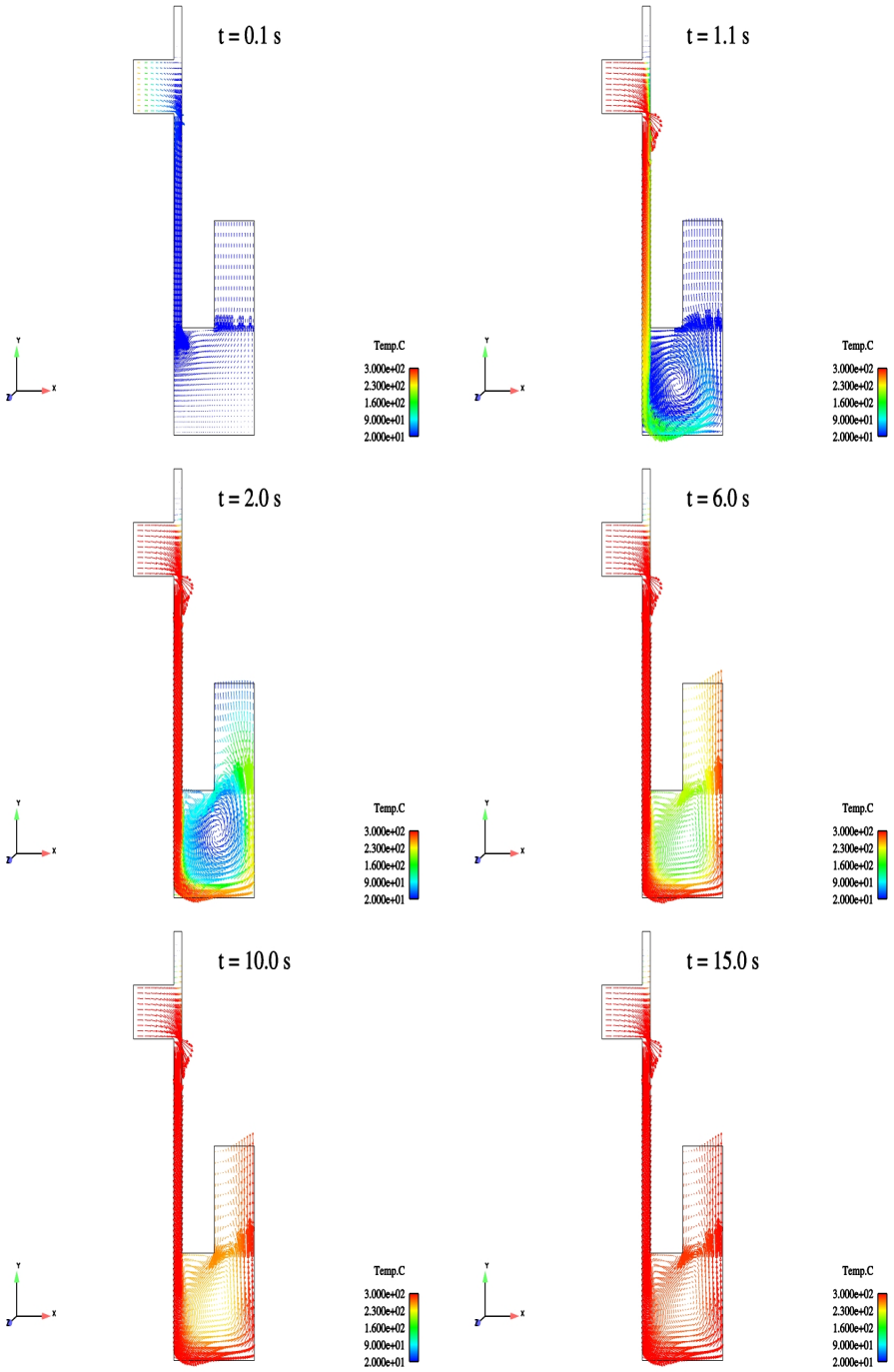


Figure III.6: Water velocity field colored by temperature at different time steps - Case 2

3 CASE 3: Time dependent boundary conditions and variable fluid density

In this case some boundary conditions will be time dependent and some physical characteristics of the fluid will be dependent on the temperature.

Remark: You can copy your `case2` in order to make the `case3`:

```
$ code_saturne create --copy-from case2 case3
```

3.1 Calculation options

The options for this case are the same as in `case2`, except for the variable fluid density:

→ Flow type: unsteady flow

→ Time step: uniform and constant

→ Turbulence model: $k - \epsilon$

→ Scalar(s): 1 - temperature
2 - passive scalar

→ Physical properties: uniform and constant (except density)

→ Management of monitoring points

3.2 Initial and boundary conditions

→ Initialization: 20°C for temperature (default value)
10 for the passive scalar

The boundary conditions are defined in the user interface and depend on the boundary zone. The time dependence of the temperature boundary condition implies the use of a Fortran user routine (see below).

- **Flow inlet**: Dirichlet condition, an inlet velocity of 1 $m.s^{-1}$, a time dependent inlet temperature and a value of 200 for the passive scalar are imposed
- **Outlet**: default value
- **Walls**: velocity, pressure and thermal scalar: default value
passive scalar: different conditions depending on the color and geometric parameters

The boundary conditions for the passive scalar are identical as in case 2, as specified in the following table:

Wall	Nature	Value
wall_1	Dirichlet	0
wall_2	Dirichlet	5
wall_3	Dirichlet	0
wall_4	Dirichlet	25
wall_5	Dirichlet	320
wall_6	Dirichlet	40

The "wall_1" to "wall_6" regions are defined as follows, through color references and geometric localization:

Label	Color and geometric parameters
wall_1	24 and $0.1 \leqslant x$ and $x \leqslant 0.5$
wall_2	2 or 3
wall_3	4 or 7 or 21 or 22 or 23
wall_4	6 and $y > 1$
wall_5	6 and $y \leqslant 1$
wall_6	31 or 33

Figure III.3 shows the colors used for boundary conditions and table III.2 defines the correspondance between the colors and the type of boundary condition to use.

Colors	Conditions
1	Inlet
34	Outlet
2 3 4 6 7 21 22 23 31 33	Wall
24 for $0.1 \leq x \leq 0.5$	Wall
8 9 28 29 38 39	Symmetry

Table III.2: Boundary faces colors and associated references

3.3 Variable Density

In this case the density is a function of temperature, the variation law is defined in the Graphical User Interface although it can also be defined in a Fortran user routine. The expression is:

$$\rho = T.(A.T + B) + C \tag{III.1}$$

where ρ is the density, T is the temperature, $A = -4.0668 \times 10^{-3}$, $B = -5.0754 \times 10^{-2}$ and $C = 1\,000.9$

In order for the variable density to have an effect on the flow, gravity must be set to a non-zero value. $\underline{g} = -9.81\underline{e}_y$ will be specified in the Graphical Interface.

Remark:
The temperature is "TempC" is the user expression if the temperature is in Celsius. Don't forget **";"** at the end of the expression.

3.4 Parameters

All the parameters necessary to this study can be defined through the Graphical Interface, except the time dependent boundary conditions that have to be specified in user routines.

Parameters of calculation control	
Number of iterations	300
Reference time step	0.05
Output period for post-processing files	2

In order to join the separate meshes into a single domain, colors 5, 24 and 32 will have to be joined through the Graphical Interface.

3.5 User routine

The following routine has to be copied from the folder SRC/REFERENCE/base into the folder SRC[4]: `cs_user_boundary_conditions.f90`.

cs_user_boundary_conditions.f90
This routine allows to define advanced boundary conditions on the boundary faces.
Even if `cs_user_boundary_conditions.f90` is used, all boundary conditions have to be defined in the Graphical User Interface. Only the conditions that differ from this first definition need to appear in `cs_user_boundary_conditions.f90`. The boundary conditions defined in `cs_user_boundary_conditions.f90` will replace those specified in the Graphical Interface.

In this case, the temperature at entry is supposed variable in time, following the law:

$$\begin{cases} T = 20 + 100t & \text{for } 0 \leqslant t \leqslant 3.8 \\ T = 400 & \text{for } t > 3.8 \end{cases} \tag{III.2}$$

where T is the temperature in °C and t is the time in second (s).

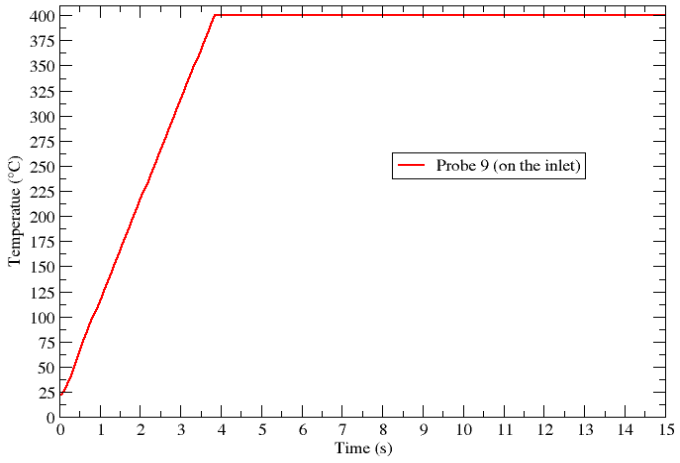


Figure III.7: Time evolution of the temperature at inlet.

Remark:
`ttcabs` is the current physical time. See the example file in the subdirectory `SRC/EXAMPLES` for the complete `cs_user_boundary_conditions.f90` file.

3.6 Output management

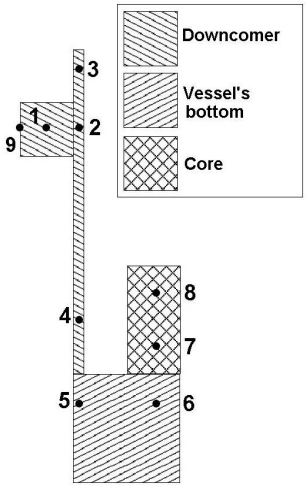
The output management is the same as in `case2`, except that a nineth monitoring point will be added, just at the entry, to monitor the temperature evolution at inlet.

In this case, the *Pressure*, the *Tubulent Energy* and the *Dissipation* will be removed from the listing file.

The *Courant number* (CFL) and *Fourier number* will be removed from the post-processing results[5].

Eventually, probes will be defined for chronological records, following the data given in figure III.4. Then the *total_pressure* will be deactivated from all probes and the *Velocity U* will be only activated on probes 1, 2, 6, 7 and 8.

[4] only when it appears in the SRC directory will it be taken into account by the code
[5] this can be very useful to save some disk space if some variables are of no interest, as post-processing files can be large



Probe n^o.	x (m)	y (m)	z (m)
1	-0.25	2.25	0
2	0.05	2.25	0
3	0.05	2.75	0
4	0.05	0.50	0
5	0.05	-0.25	0
6	0.75	-0.25	0
7	0.75	0.25	0
8	0.75	0.75	0
9	-0.5	2.25	0

Figure III.8: Position and coordinates of probes in the full domain

In addition the domain boundary will be post-processed. This allows to check the boundary conditions, and especially that of the temperature and passive scalar.

3.7 Calculation restart

After the first run, the calculation will be continued for another 400 time steps. The calculation restart is managed through the Graphical Interface.

3.8 Results

Figure III.9 shows the time evolution of temperature recorded on each monitoring probe. Figure

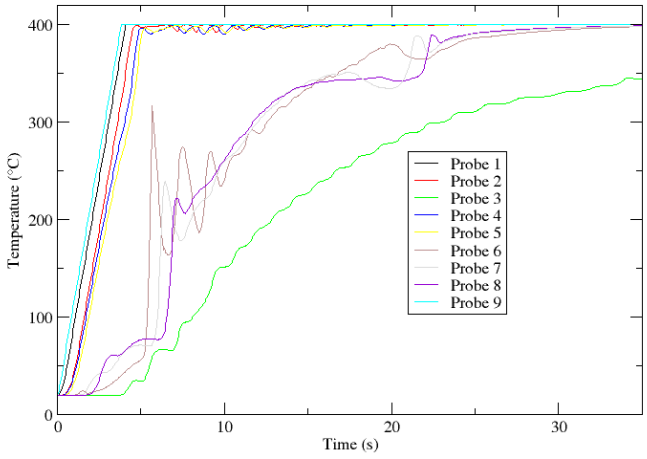


Figure III.9: Time evolution of temperature at monitoring probes for case 3

III.10 shows the velocity fields colored by temperature in the first run of calculation. Figure III.11 shows the velocity fields in the second calculation (restart of the first one).

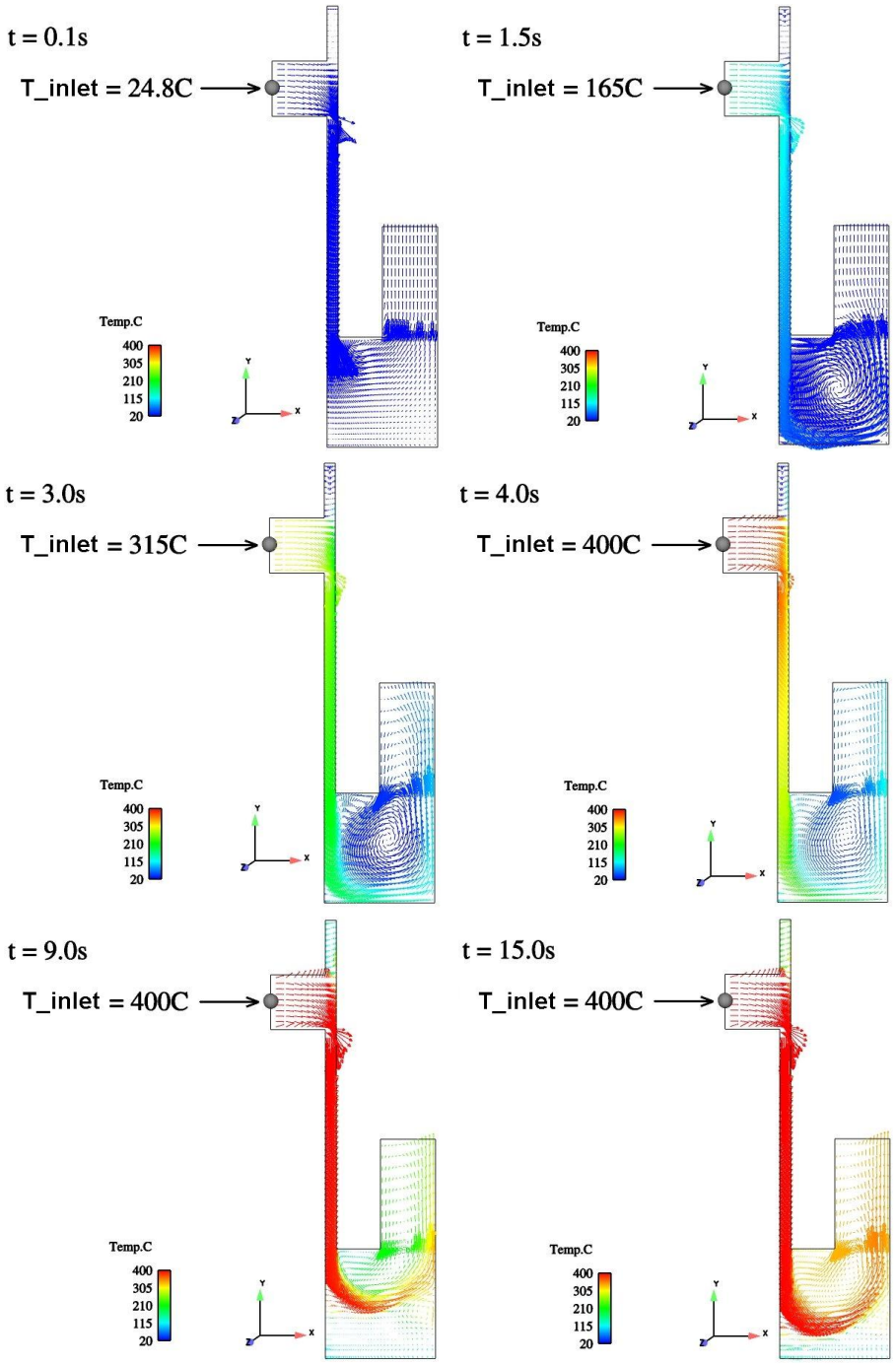


Figure III.10: Water velocity field colored by temperature and inlet temperature value at different time steps (first calculation)

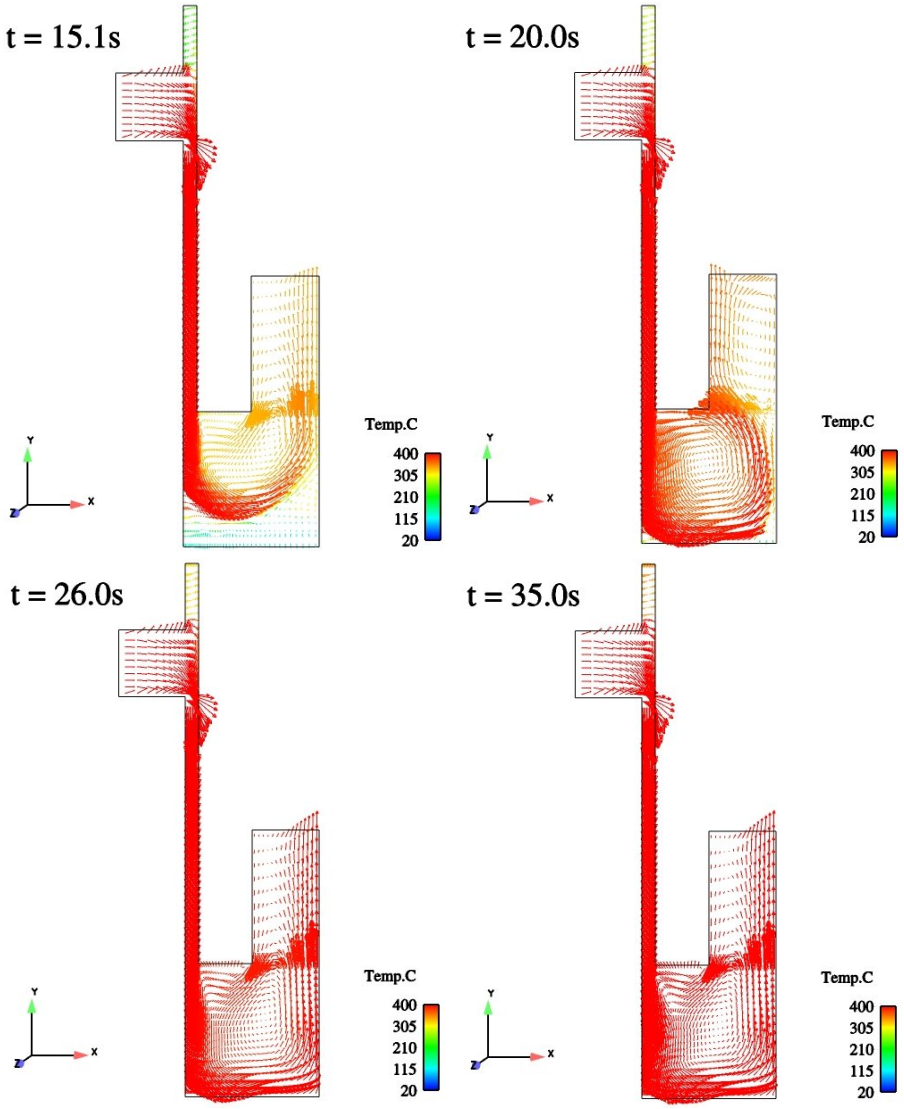


Figure III.11: Water velocity field colored by temperature and inlet temperature value at different time steps (second calculation)

4 CASE 4: Head losses, parallelism and spatial average

This case will be run in parallel on two processors. Head losses will be used to simulate the presence of an obstacle in the flow and the spatial average of the temperature will be calculated at each time step.

4.1 Calculation options

The options for this case are the same as in case 3:

→ Flow type: unsteady flow

→ Time step: uniform and constant

→ Turbulence model: $k - \epsilon$

→ Scalar(s): 1 - temperature
2 - passive scalar

→ Physical properties: uniform and constant (except density)

→ Management of monitoring points

4.2 Initial and boundary conditions

→ Initialization: 20°C for temperature (default value)
10 for the passive scalar

The boundary conditions are defined in the user interface and depend on the boundary zone.

- **Flow inlet**: Dirichlet condition, an inlet velocity of 1 $m.s^{-1}$ and a time dependent inlet temperature are imposed
- **Outlet**: default value
- **Walls**: velocity, pressure and thermal scalar: default value
passive scalar: different conditions depending on the color and geometric parameters

The boundary conditions for the passive scalar are identical as in case 2, as specified in the following table:

Wall	Nature	Value
wall_1	Dirichlet	0
wall_2	Dirichlet	5
wall_3	Dirichlet	0
wall_4	Dirichlet	25
wall_5	Dirichlet	320
wall_6	Dirichlet	40

The "wall_1" to "wall_6" regions are defined as follows, through color references and geometric localization:

Label	Color and geometric parameters
wall_1	24 and $0.1 \leqslant x$ and $x \leqslant 0.5$
wall_2	2 or 3
wall_3	4 or 7 or 21 or 22 or 23
wall_4	6 and $y > 1$
wall_5	6 and $y \leqslant 1$
wall_6	31 or 33

Figure III.3 shows the colors used for boundary conditions and table III.3 defines the correspondance between the colors and the type of boundary condition to use.

Colors	Conditions
1	Inlet
34	Outlet
2 3 4 6 7 21 22 23 31 33	Wall
24 for $0.1 \leq x \leq 0.5$	Wall
8 9 28 29 38 39	Symmetry

Table III.3: Boundary faces colors and associated references

4.3 Variable Density

In this case the density is a function of temperature, the variation law is defined in the Graphical User Interface although it can also be defined in a Fortran user routine. The expression is:

$$\rho = T.(A.T + B) + C \tag{III.3}$$

where ρ is the density, T is the temperature, $A = -4.0668 \times 10^{-3}$, $B = -5.0754 \times 10^{-2}$ and $C = 1\,000.9$

In order for the variable density to have an effect on the flow, gravity must be set to a non-zero value. $\underline{g} = -9.81\underline{e}_y$ will be specified in the Graphical Interface.

4.4 Head losses

To simulate the presence of an obstacle 0.20 (m) large and 0.5 (m) high in the vessel, a zone of head losses will be created in the domain (fig III.12).

The head losses zone is located between the coordinates $x = 0.2$ (m) and $x = 0.4$ (m), and $y = -0.75$ (m) and $y = -0.25$ (m).

The head losses coefficient to apply is $K_{ii} = 10^4 = \frac{1}{2}\ \alpha_{ii}$ and is isotropic.

4.5 Parameters

All the parameters necessary to this study can be defined through the Graphical Interface. However, the calculation of the spatial average is defined by a user routine.

Parameters of calculation control	
Number of iterations	900
Reference time step	0.01
Output period for post-processing files	2
The calculation will be run in parallel	2 procs.

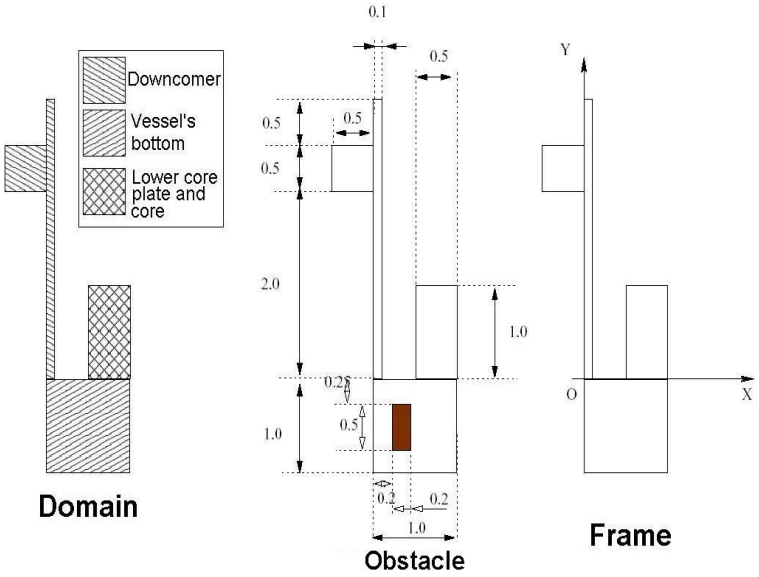


Figure III.12: Full domain geometry with the obstacle

In order to join the separate meshes into a single domain, colors 5, 24 and 32 will have to be joined through the Graphical Interface.

Note that the time step has been reduced because of the head losses: the pressure step is more difficult to be solved in presence of head losses.

4.6 User routines

The following routines have to be copied from the folder **SRC/REFERENCE/** into the folder **SRC/**[6]: `cs_user_boundary_conditions.f90` and `cs_user_extra_operations.f90`. We can find and copy some basic and specific boundary conditions examples in the folder **SRC/EXAMPLES/** to correctly impose the *Code_Saturne* boundary conditions.

• **cs_user_boundary_conditions.f90**
This routine allows to define advanced boundary conditions on the boundary faces.

Even if cs_user_boundary_conditions.f90 is used, all boundary conditions have to be defined in the Graphical User Interface (GUI). Only the conditions that differ from this first definition need to appear in cs_user_boundary_conditions.f90. The boundary conditions defined in cs_user_boundary_conditions.f90 will replace those specified in the Graphical Interface.

In this case, the temperature at entry is supposed variable in time, following the law:

$$\begin{cases} T = 20 + 100t & \text{for } 0 \leqslant t \leqslant 3.8 \\ T = 400 & \text{for } t > 3.8 \end{cases} \tag{III.4}$$

where T is the temperature in °C and t is the time in s.

• **cs_user_extra_operations.f90**
This routine is called at the end of each time step and has access to the whole set of variables of the code. It is therefore useful for many user-specific post-processing, including the calculation of a spatial average in the present case.
The spatial average of the temperature will be calculated at each time step and the result wrote in

[6] only when they appear in the SRC directory will they be taken into account by the code.

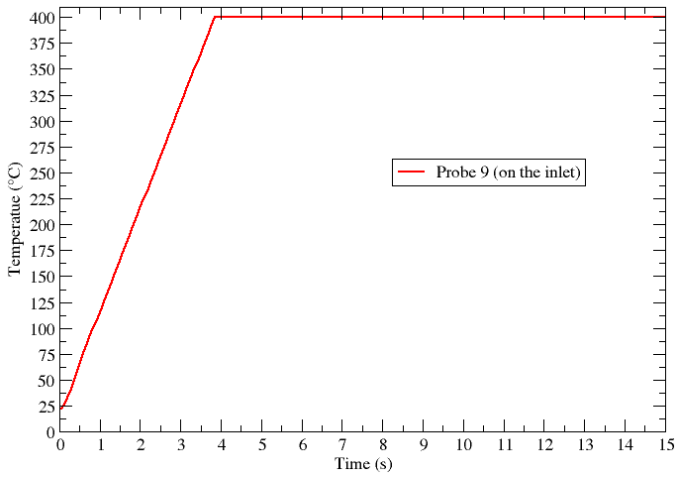


Figure III.13: Time evolution of the temperature at inlet

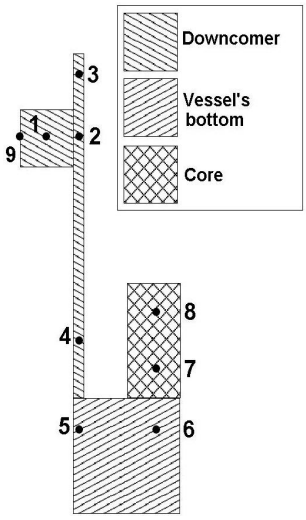
a file named ‘‘`moy.dat`’’. The values are saved in order to draw the time evolution of the average temperature.

Beware when calculating the average. Since the calculation is running in parallel, computing the sum of the temperatures on “all the cells” will only yield for each processor the sum on the cells managed by this processor. In order to obtain the full sum, the parallelism routine `parsom` must be used (see example in the cs_user_extra_operations-parallel_operations.f90 routine).

Remark: `cs_user_extra_operations-xxx.f90` are different example routines present in the subdirectory `SRC/EXAMPLES`. They should be removed from the `SRC/` before running the case.

4.7 Output management

The output management is the same as in case 3.



Points	X(m)	Y(m)	Z(m)
1	-0.25	2.25	0
2	0.05	2.25	0
3	0.05	2.75	0
4	0.05	0.5	0
5	0.05	-0.25	0
6	0.75	-0.25	0
7	0.75	0.25	0
8	0.75	0.75	0
9	-0.5	2.25	0

Figure III.14: Position and coordinates of probes in the full domain

In this case, the *Pressure*, the *Tubulent Energy* and the *Dissipation* will be removed from the listing file.

The *Courant number* (CFL) and *Fourier number* will be removed from the post-processing results[7].

Eventually, probes will be defined for chronological records, following the data given in figure III.4. Then the *total pressure* will be deactivated from all probes and the *Velocity U* will be only activated on probes 1, 2, 6, 7 and 8.

In addition the domain boundary will be post-processed. This allows to check the boundary conditions, and especially that of the temperature and passive scalar.

4.8 Results

Figure III.15 shows the evolution of the spatial average of the temperature.

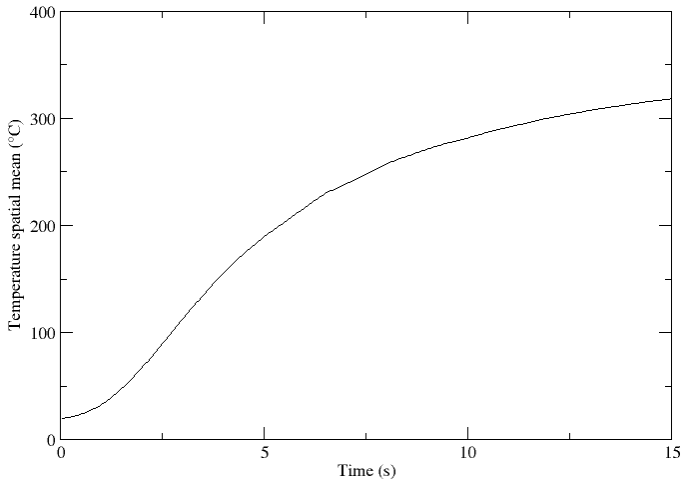


Figure III.15: Evolution of spatial average of the temperature as a function of time

Figure III.16 shows velocity fields colored by temperature. The effect of the head loss modeling the obstacle is clearly visible.

[7]this can be very useful to save some disk space if some variables are of no interest, as post-processing files can be large

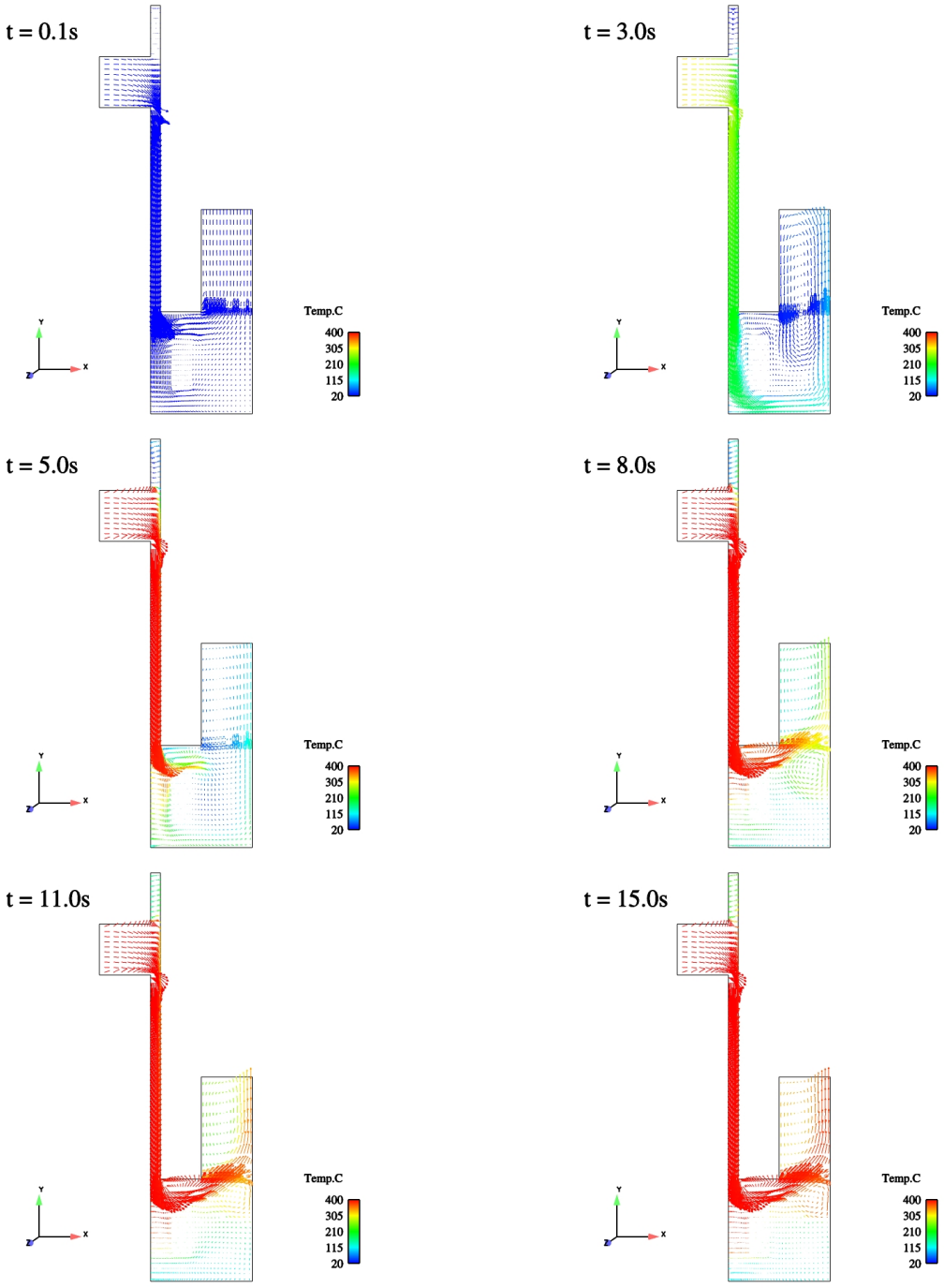


Figure III.16: Water velocity field colored by temperature

Part IV

Stratified junction

1 General description

1.1 Objective

The aim of this case is to train the *Code_Saturne* user on a simplified but real 3D computation. It corresponds to a stratified flow in a T-junction. The test case will be used to present some advanced post-processing techniques.

1.2 Description of the configuration

The configuration is based on a real mock-up designed to characterize thermal stratification phenomena and associated fluctuations. The geometry is shown on figure V.1.

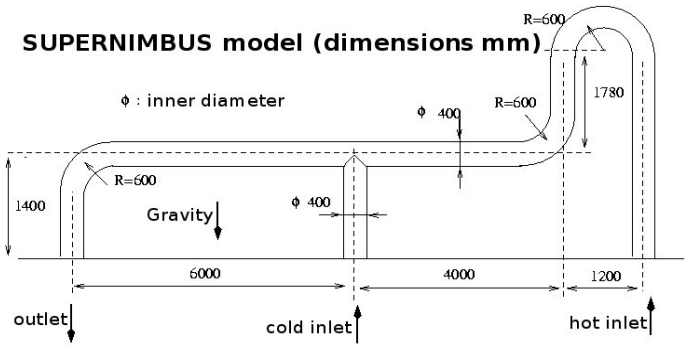


Figure IV.1: Geometry of the case

There are two inlets, a hot one in the main pipe and a cold one in the vertical nozzle. The volumic flow rate is identical in both inlets. It is chosen small enough so that gravity effects are important with respect to inertia forces. Therefore cold water creeps backwards from the nozzle towards the elbow until the flow reaches a stable stratified state.

1.3 Characteristics

Characteristics of the geometry:

Diameter of the pipe	$D_b = 0.40\ m$

Characteristics of flow:

Cold branch volume flow rate	$Dv_{cb} = 4\ l.s^{-1}$
Hot branch volume flow rate	$Dv_{hb} = 4\ l.s^{-1}$
Cold branch temperature	$T_{cb} = 18.26°\mathrm{C}$
Hot branch temperature	$T_{hb} = 38.5°\mathrm{C}$

The initial water temperature in the domain is equal to 38.5°C.

Water specific heat and thermal conductivity are considered constant and calculated at 18.26°C and $10^5\ Pa$:

- heat capacity: $C_p = 4{,}182.88\ J.kg^{-1}.°\mathrm{C}^{-1}$

- thermal conductivity: $\lambda = 0.601498\ W.m^{-1}.°C^{-1}$

The water density and dynamic viscosity are variable with the temperature. The functions are given below.

1.4 Mesh characteristics

The mesh used in the actual study had 125 000 elements. It has been coarsened for this example in order for calculations to run faster. The mesh used here contains 16 320 elements.

Type: unstructured mesh

Coordinates system: cartesian, origin on the middle of the horizontal pipe at the intersection with the nozzle.

Mesh generator used: SIMAIL

Color definition: see figure V.3.

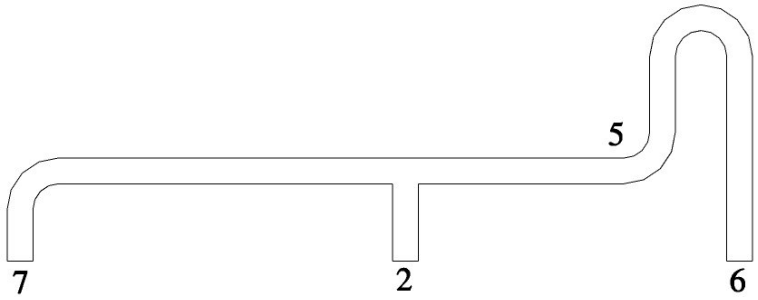


Figure IV.2: Colors of the boundary faces

2 CASE 5: Stratified junction

In this case, advanced post-processing features will be used. A specific pos-processing sub-mesh will be created, containing all the cells with a temperature lower than 21°C, so that it can be visualized (with EnSight for instance). The variable "temperature" will be post-processed on this sub-mesh. A 2D clip plane will also be extracted along the symmetry plane of the domain and temperature will be written on it.

2.1 Calculation options

The following options are considered for the case:

→ Flow type: unsteady flow

→ Time step: variable in time and uniform in space

→ Turbulence model: $k - \epsilon$

→ Scalar(s): temperature

→ Physical properties: uniform and constant for specific heat and thermal conductivity and variable for density and dynamic viscosity

→ Pressure interpolation in stratified flows : improved

Colors	Conditions
2	Cold inlet
6	Hot inlet
7	Outlet
5	Wall

Table IV.1: Boundary faces colors and associated references

2.2 Initial and boundary conditions

→ Initialization: temperature initialization at 38.5°C

The boundary conditions are defined as follows:

- **Flow inlet**: Dirichlet condition
  - velocity of 0.03183 $m.s^{-1}$ for both inlets
  - temperature of 38.5°C for the hot inlet
  - temperature of 18.6°C for the cold inlet
- **Outlet**: default value
- **Walls**: default value

Figure V.3 shows the colors used for boundary conditions and table IV.1 defines the correspondance between the colors and the type of boundary condition to use.

2.3 Variable density and dynamic viscosity

In this case the density and the dynamic are functions of the temperature, the following variation laws are specified in the Graphical User Interface:

$$\rho = T.(A.T + B) + C \tag{IV.1}$$

where ρ is the density, T is the temperature, $A = -4.0668 \times 10^{-3}$, $B = -5.0754 \times 10^{-2}$ and $C = 1\,000.9$

For the dynamic viscosity, the variation law is:

$$\mu = T.(T.(AM.T + BM) + CM) + DM \tag{IV.2}$$

where μ is the dynamic viscosity, T is the temperature, $AM = -3.4016 \times 10^{-9}$, $BM = 6.2332 \times 10^{-7}$, $CM = -4.5577 \times 10^{-5}$ and $DM = 1.6935 \times 10^{-3}$

In order for the variable density to have an effect on the flow, gravity must be set to a non-zero value. $\underline{g} = -9.81\underline{e}_z$ will be specified in the Graphical Interface.

2.4 Parameters

All the parameters necessary to this study can be defined through the Graphical Interface, except the variable fluid characteristics and the advanced post-processing features that have to be specified in user routines.

Parameters of calculation control	
Reference time step	1 s
Number of iterations	100
Maximal CFL number	20
Maximal Fourier number	60
Minimal time step factor	0.01 s
Maximal time step factor	70 s
Time step maximal variation	0.1
Period of output chronological files	10

The time step limitation by gravity effects will also be activated.

2.5 Output management

The standard options for output management will be used. Four monitoring points will be created at the following coordinates:

Points	X(m)	Y(m)	Z(m)
1	0.010025	0.01534	-0.011765
2	1.625	0.01534	-0.031652
3	3.225	0.01534	-0.031652
4	3.8726	0.047481	7.25

Two vertical temperature profiles will be extracted, at the following locations:
profil16: $x = 1.6$ $y = 0$ $-0.2 \leqslant z \leqslant 0.2$ (m)
profil32: $x = 3.2$ $y = 0$ $-0.2 \leqslant z \leqslant 0.2$ (m)

2.6 User routines

The following filess must to be copied from the folder `SRC/REFERENCE/base` into the folder `SRC`[1]: **cs_user_postprocess.c** and **cs_user_postprocess_var.f90**.

In this test case, advanced post-processing features will be used. A clip plane will be created, along the symmetry plane of the domain, on which the temperature will be written. This plane will be added to the standard "writer" (*i.e.* it will be an extra part in the standard `RESULTS.case` output). The periodicity of output on the standard writer will be 10 iterations.
An additional writer will also be created, with a periodicity of 5 iterations. It will only contain one part (*i.e.* one sub-mesh): the set cells where the temperature is lower than 21°C. The temperature will be written on this part. The interest of this part is that it is time dependent as for the cells it contains.

The following user functions and subroutines will be used:

• **cs_user_postprocess_meshes** (in **cs_user_postprocess.c**)
This function is called only once, at the beginning of the calculation. It allows to define the different writers and parts. In this function, adapt the example using the **cs_post_define_volume_mesh_by_func**, replacing `He_fraction_05` with `T_lt_21` (do not forget to set the enclosing test to `true`). If argument matching *the automatic variables output* is set to *true*, all variables (including temperature) postprocessed on the main output will be added to this one. For finer control, we set it to *false* here, and we will use a user-defined output with **cs_user_postprocess_var**. The associated writer list should contain writer 1, which may be created either using the GUI, or the **cs_user_postprocess_writers** (in the same file). Make sure this writers allows for *transient connectivity*. The **_he_fraction_05_select** near the beginning of the file must also be adapted, renaming it to **_t_lt_21_select**, and adapting its

[1] only when they appear in the `SRC` directory will they be taken into account by the code

contents (mainly calling **cs_field_by_name** on *temperature* instead of *He_fraction*, and replacing $>$ *5.e-2* with $<$ *21*). This selection function is called automatically at each output time step so as to update the selected sub-mesh.

- **cs_user_postprocess_var.f90**

This routine is called at each time step. It allows to specify which variable will be written on which part (in this case, temperature).

2.7 Results

Figure V.6 shows the evolution of the temperature in the domain at different time steps. The evolution of the stratification is clearly visible.

Figure V.5 shows the cells where the temperature is lower than 21°C. It is not an isosurface created from the full domain, but a visualization of the full sub-domain created through the post-processing routines.

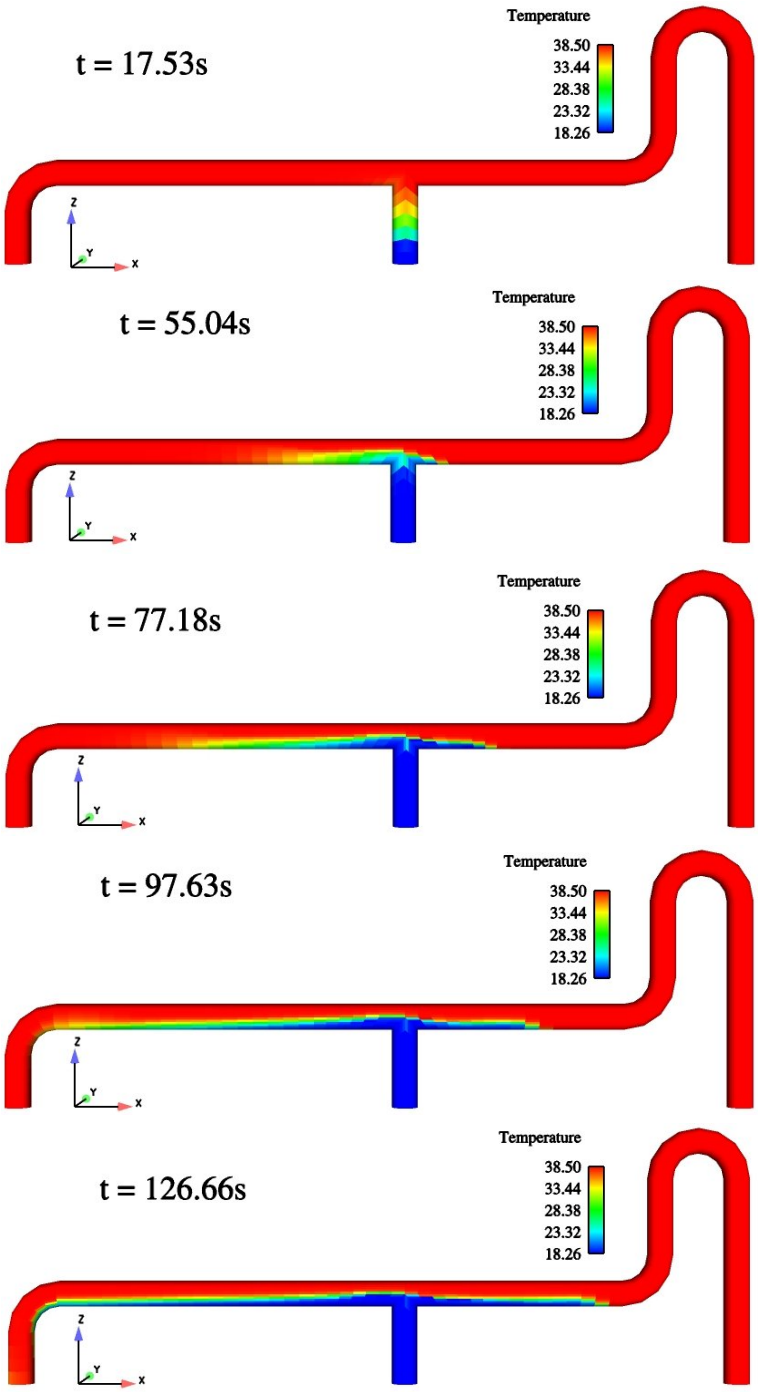


Figure IV.3: Evolution of temperature

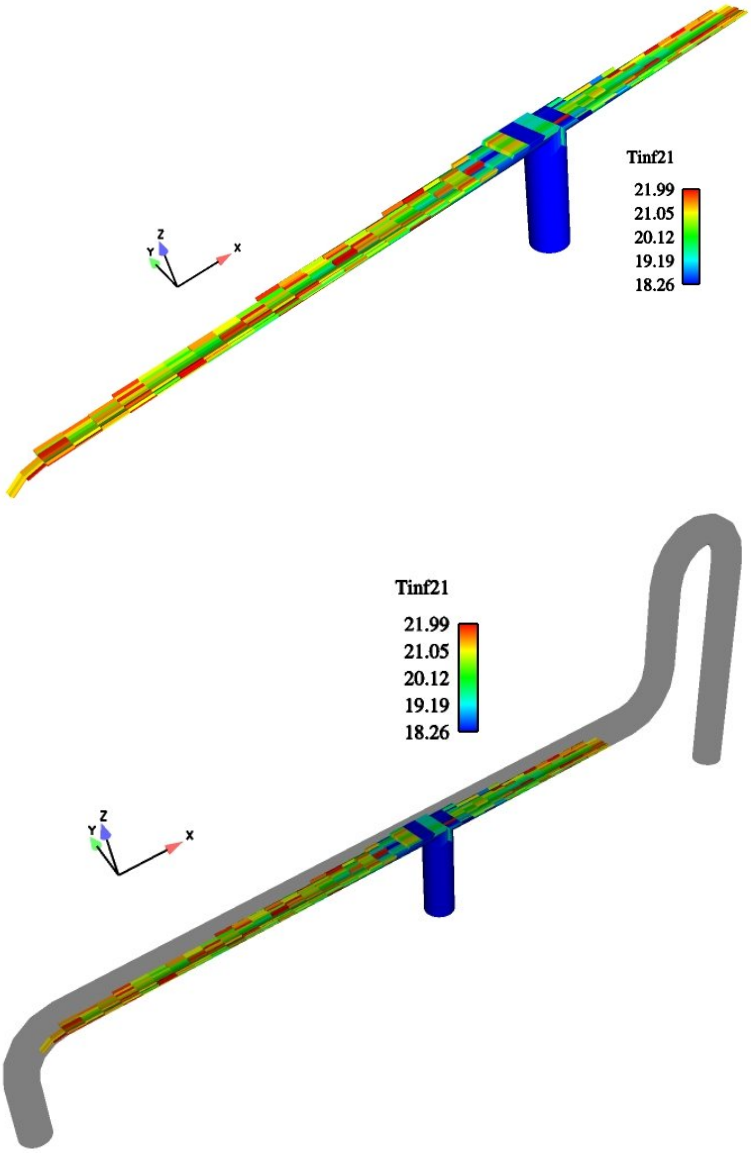


Figure IV.4: Sub-domain where the temperature is lower than 21°C (upper figure) and localization in the full domain (lower figure)

Part V

Three 2D disks

1 General description

1.1 Objective

The aim of this case is to train the *Code_Saturne* coupling with a thermal conduction and radiation code SYRTHES on a simplified 2D problem. It corresponds to a natural convection inside a sheath with different electric wires.

We can see with this test-case the conjugate heat transfer phenomenon between the solid and fluid domains.

1.2 Remarks

• **Remark - 1**: create the `3disks2D` study directory, two subdirectories `fluid` and `solid` as below:

```
$ code_saturne create -s 3disks2D -c fluid --syrthes solid
```

• **Remark - 2**: The fluid mesh must be copied in the directory `MESH`. The solid mesh must be copied in the subdirectory `solid`.

• **Remark - 3**: launch the SYRTHES Graphical User Interface (Gui) (`$ syrthes.gui &`) inside the subdirectory `solid` for the first solid computation alone.

• **Remark - 4**: launch the *Code_Saturne* Graphic User Interface (GUI) inside the subdirectory `fluid` for the fluid computation alone.

• **Remark - 5**: launch the *Code_Saturne*-SYRTHES coupling computation with the `runcase_coupling` script.

1.3 Description of the configuration

The 2D configuration represents a simplification of the real 3D geometry of the wires inside an electric sheath. As we can see, we have 3 different wires represented as 3 different disks inside a bigger disk for the sheath. We assume that the 3 disks are in contact with an air flow inside the electric sheath.

The geometry is shown on figure V.1.

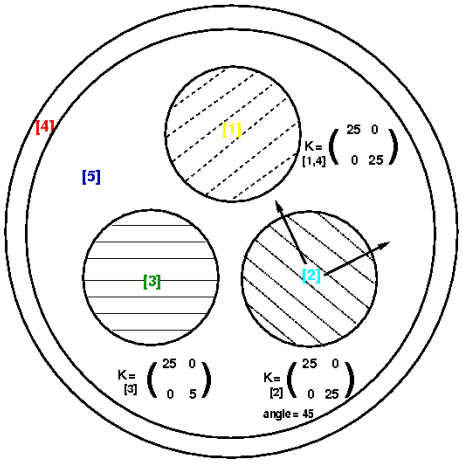


Figure V.1: Geometry of the test-case with [1,2,3,4] the solid domain and [5] the fluid domain. The 4 disk physical properties are specified for the solid domain.

For the fluid domain, there are two symmetry conditions and walls conditions imposed to the faces

coupling with the solid domain. We have no velocity imposed to create movement inside the fluid area and gravity force is taken into account.

Nevertheless, we define a density which is variable in function of the temperature for the air flow. The 3 disks, which are warmer than the air flow, generate a temperature difference creating a fluid movement. The warmer air flow is moving to the top and the colder air flow to the bottom of the fluid domain.

With this test-case, we can easily observe the effect of the solid disks on the air flow contained in the electric sheath.

1.4 Characteristics

- **Solid domain:**

The initial and boundary conditions to choose without conjugate heat transfer for the solid domain are defined hereafter:

Initial conditions	
Temperature condition	$T_{ini,s} = 20°C$

Boundary conditions	value	surface reference
Heat exchange conditions ($q_{w,ext}$)	$T_{ext} = 90°C.$; $h_{ext} = 1000(W/m^2.K)$	color 2 or 5 or 8

Characteristics of the solid domain with the 4 different disks (1 to 3 for the electric wires and 4 for the disk for the electric sheath):

	Conductivity type	values (W/m/°C)	volume reference
disk 1	isotropic	$k_{11} = 25$	color 1
disk 2	orthotropic	$k_{11} = 25$; $k_{22} = 5$	color 2
disk 3	orthotropic	$k_{11} = 25$; $k_{22} = 5$ $\alpha = 45^o$	color 3
disk 4	anisotropic	$k_{11} = 25$	color 4

Physical properties	values
Density [ρ]	7700 (kg/m^2)
Specific heat [C_p]	460 $(J/kg/m^3)$

- **Fluid domain:**

The characteristics of the air flow inside the fluid domain are defined as following:

Thermophysical models	choosen type
Time step	constant in time and uniform in space
Turbulence model	ε -k
Scalar	Temperature (°C)

The initial and boundary conditions to choose without conjugate heat transfer for the solid domain are defined below:

Initial conditions	
Temperature condition	$T_{ini,f} = 20°C.$

Boundary conditions	values	surface reference
Walls (Heat exchange $q_{w,ext}$)	$T_{ext} = 30°C$; $h_{ext} = 10(W/m^2.K)$	color 1
Symmetry		color 2 or 3

In this case, the fluid density is function of the temperature, the following ideal gas law is specified in the Graphical User Interface (GUI):

$$\rho = \frac{p_0}{R_g\ (T + 273.15)} \tag{V.1}$$

where ρ is the density, T is the temperature (°C), ideal gas constant $R_g = 287\ (m^2.s^{-2}.K^{-1})$, $p_0 = 101325\ (Pa)$ the reference pressure choosen as $p \approxeq p_{atmos}$.

1.5 Mesh characteristics

- **Description of the solid mesh:**

The solid mesh used in the conduction problem contains 11688 nodes (P_1 discretization) and 5688 elements. We have to take care of the references allowing to identify materials properties and boundary conditions which are specified in this solid mesh by reference colors.

Type: unstructured mesh **Mesh generator used**: SIMAIL **Color definition**: see figure V.3.

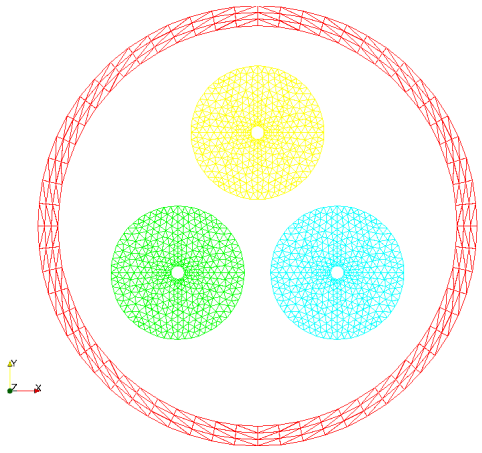


Figure V.2: Colors of the boundary faces

- **Description of the fluid mesh:**

The fluid mesh contains 3866 nodes. We have to apply the **check mesh** available in the *Code_Saturne* Graphical User Interface to check the quality criteria and identify the reference colors associated to the boundary conditions.

Type: unstructured mesh **Mesh generator used**: SIMAIL **Color definition**: see figure V.3.

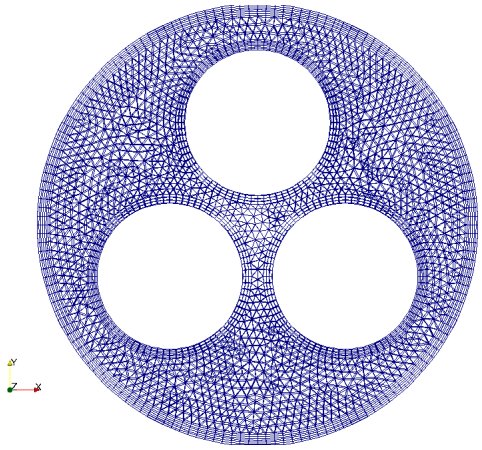


Figure V.3: Colors of the boundary faces

2 CASE 6: 3 2D disks

The post-processing containing the "temperature" field will be post-processed on a sub-mesh with ParaView. A 2D clip plane will also be extracted along the symmetry plane of the fluid domain and temperature will be written on it.

2.1 Parameters

All the parameters necessary to this study can be defined through the *Code_Saturne* (GUI) and SYRTHES (Gui) respectively, as below:

Numerical parameters of **solid computation**	
Reference time step	0.1 (s)
Number of iterations	100
Numerical parameters of **fluid computation**	
Reference time step	0.1 (s)
Number of iterations	100

These numerical time steps and iterations number have been defined to run the fluid and solid computations independently one from each other. Thus, we can test the setting data for the fluid computation with *Code_Saturne* and the solid conduction computation with SYRTHES. After that we will be able to run the coupling computation with the computation option **conjugate heat transfer** activated on both data setting.

2.2 Output management

The standard options for output management will be used. Only one monitoring point will be created for the solid conduction computation at the following coordinates:

Probe	x (m)	y(m)
1	0.003	-1.2

For this probing we choose to save the temperature value every 10 time steps and the temperature field every 25 time steps.

2.3 Coupling computation

The numerical parameters used for the coupling computation must be modified to be sure to see the conjugate heat transfer phenomenon between the solid and fluid domains. For this reason, we increase the iterations number and the time step for the fluid and solid data setting.

By default, the smaller iterations number will be used to drive the coupling computation. If we choose an iterations number of 10000 for the fluid domain and 5000 for the solid domain, the coupling computation will be stopped after 5000 instead of 10000.

Numerical parameters of **solid computation**	
Reference time step	0.5 (s)
Number of iterations	50000

Numerical parameters of **fluid computation**	
Reference time step	0.5 (s)
Number of iterations	50000

2.4 Results

Figure V.6 shows the evolution of the temperature in the solid domain without **conjugate heat transfer** with the fluid domain. We have represented below the evolution of the temperature in the fluid domain without coupling with SYRTHES.

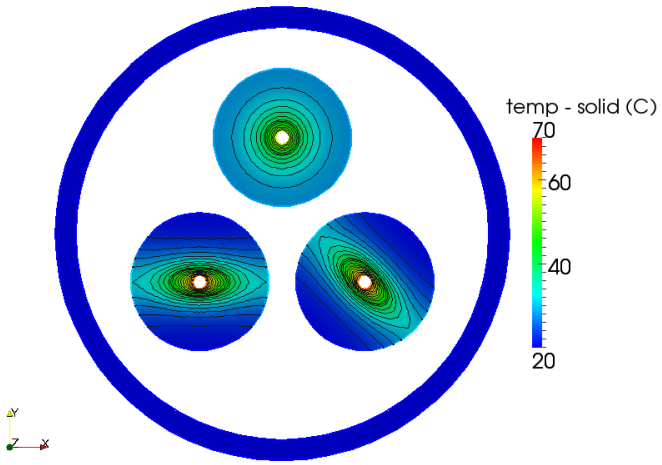


Figure V.4: The temperature evolution in the **solid domain without coupling method**

Figure V.6 shows the evolution of the temperature in the solid and fluid area with the **conjugate heat transfer activated**. The natural convection in the fluid domain due to the temperature difference imposed by the solid disks is clearly visible with the velocity field and vector.

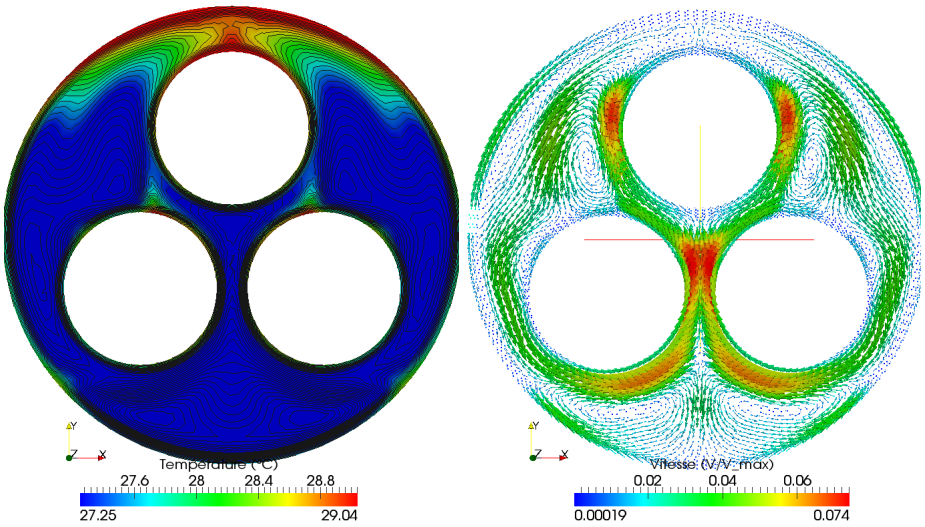


Figure V.5: The temperature evolution in the **fluid domain without coupling method**

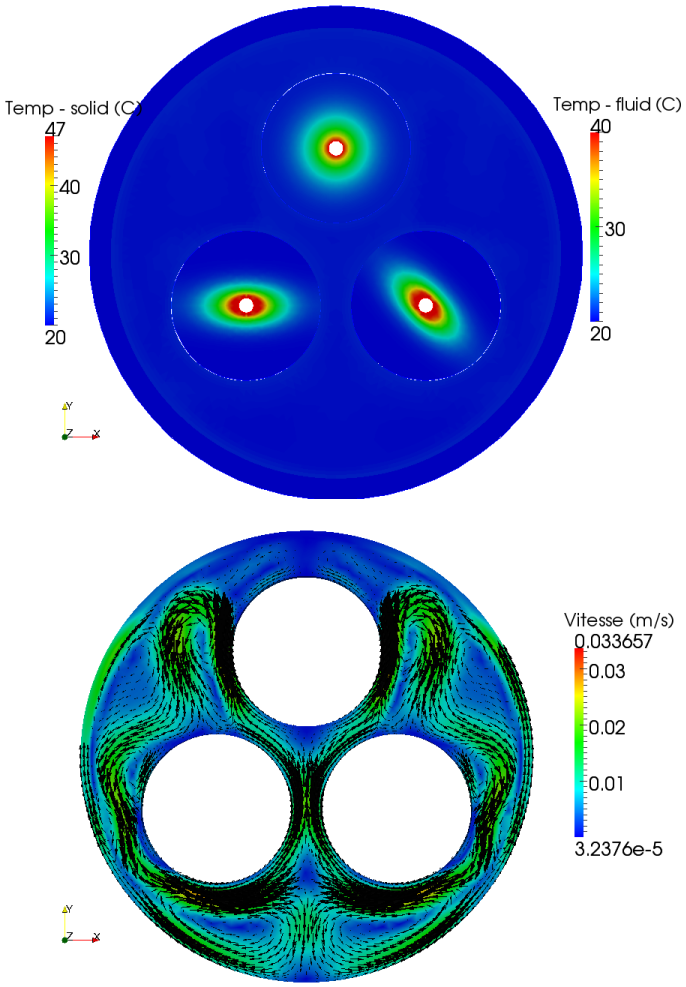


Figure V.6: Evolution of temperature

Part VI

Step by step solution

1 Solution for case1

The first thing to do before running *Code_Saturne* is to prepare the computation directories. In this first example, the study directory "simple_junction" will be created, containing a single calculation directory `case1`. This is done by typing the command:

```
$ code_saturne create -s simple_junction -c case1
```

The mesh files should be copied in the directory `MESH/`, as follow:

```
$ cd simple_junction/MESH/
$ cp ITECH_CS_training_2012/meshes/1-simple_junction/downcomer.des .
```

The *Code_Saturne* Graphical User Interface (GUI) is launched by typing the command lines as below:

```
$ cd simple_junction/case1/DATA
$ ./SaturneGUI &
```

And the following graphic window opens (fig VI.1).

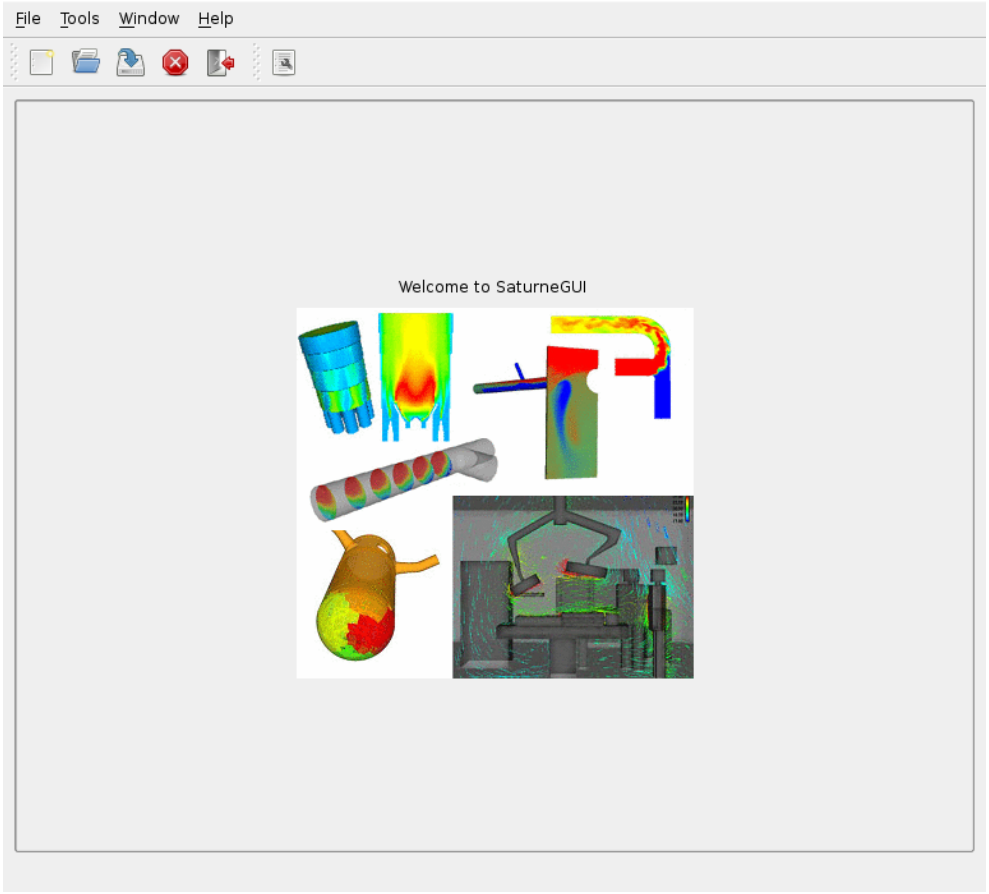


Figure VI.1: *Code_Saturne* (GUI) graphic window

Go to the *File* menu and click on *New file* to open a new calculation data file. The interface automatically updates the following information:

- Study name
- Case name
- Directory of the case
- Associated sub-directories of the case

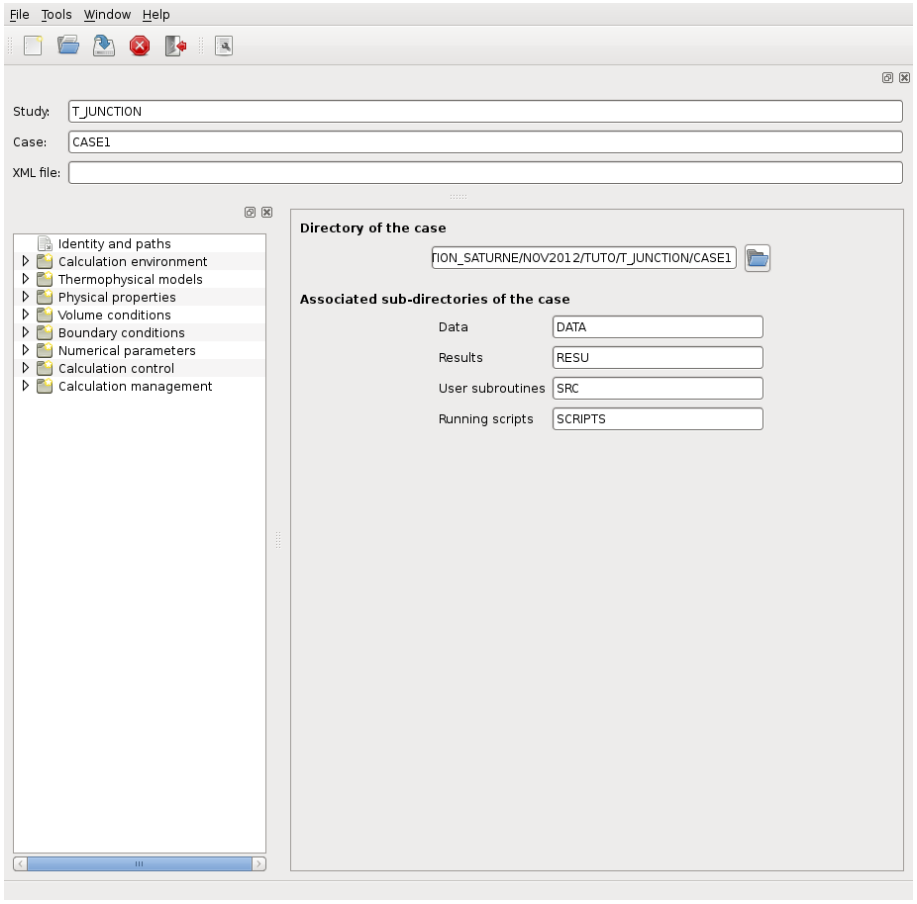


Figure VI.2: Identity and paths

Save the case to give a name to the new `xml` file (such as `case1.xml`) by opening the *File* menu and clicking on *Save as....*

A new window will appear, enter the name of the case in *File Name* then click on *Save*.

Remember to save the case regularly throughout the preparation of the calculation.

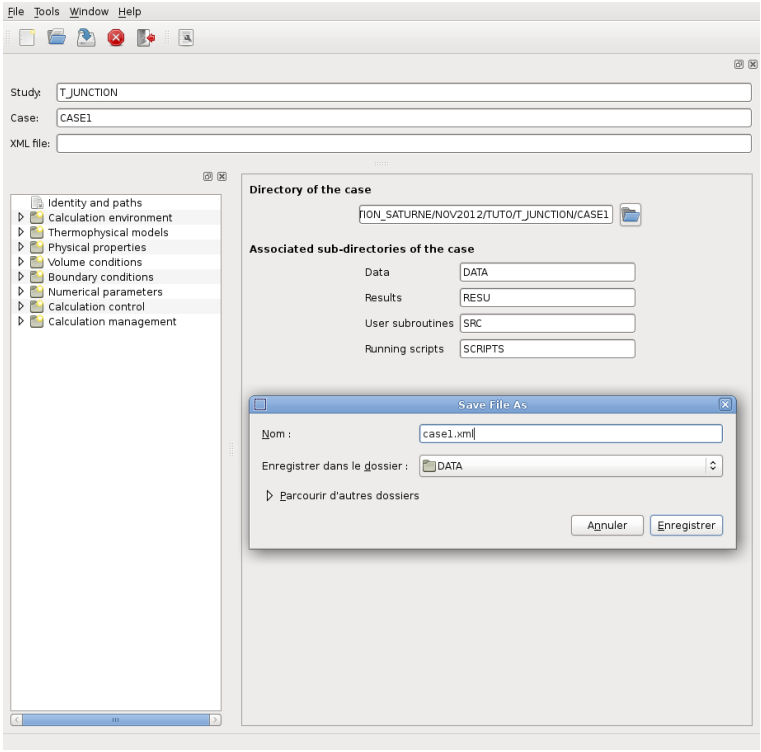


Figure VI.3: Saving the `xml` file

The next step is to specify the mesh(es) to be used for the calculation. Click on the *Mesh selection* item under the heading *Calculation environment*. Click to "+" to add meshes.

The list of meshes in the folder *Meshes* appears in the window *List of meshes*. In this case only the mesh `downcomer.des` is needed.

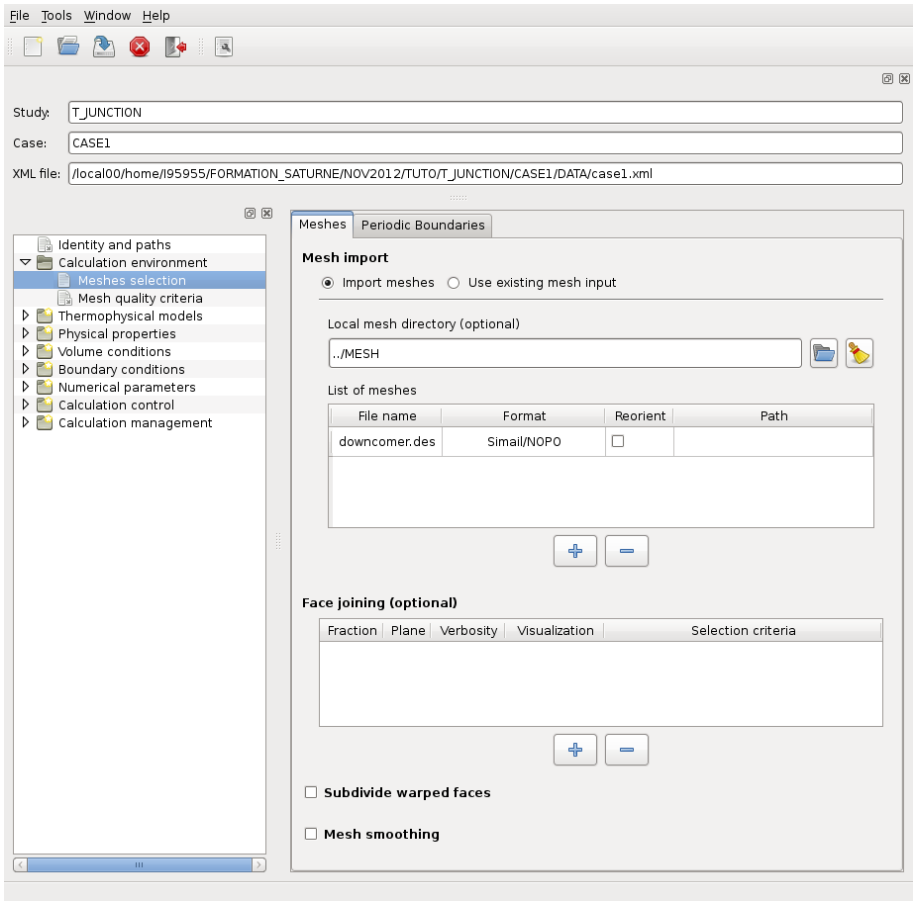


Figure VI.4: Meshes: list of meshes

The *Periodic Boundaries* is not used in this case. Keep the default values.

The *Calculation features* item under the heading *Thermophysical models* allows to define the type of flow to be simulated. In this case, a steady flow will be chosen.

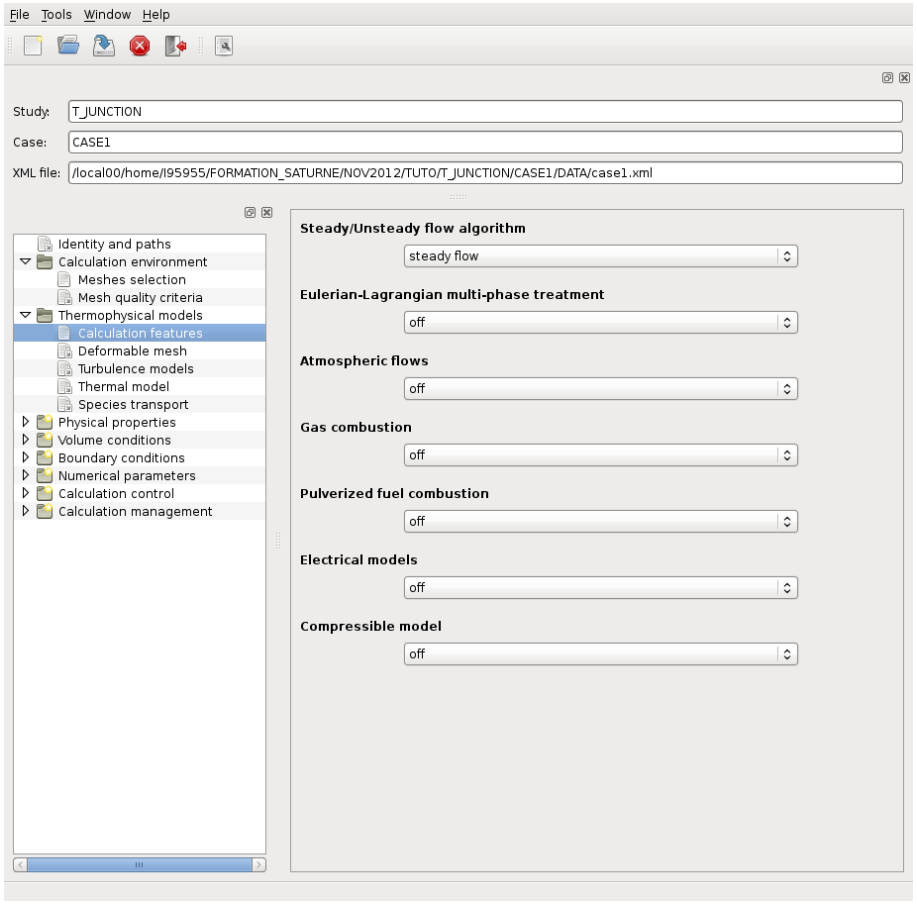


Figure VI.5: Flow type

The turbulence model is selected in the following list:

- laminar flow (no model)
- mixing length
- k-ε
- k-ε Linear Production
- Rij-ε LLR
- Rij-ε SSG
- v2f (φ model)
- k-ω SST
- Spalart-Allmaras
- LES (Smagorinsky)
- LES (classical dynamic model)
- LES (WALE)

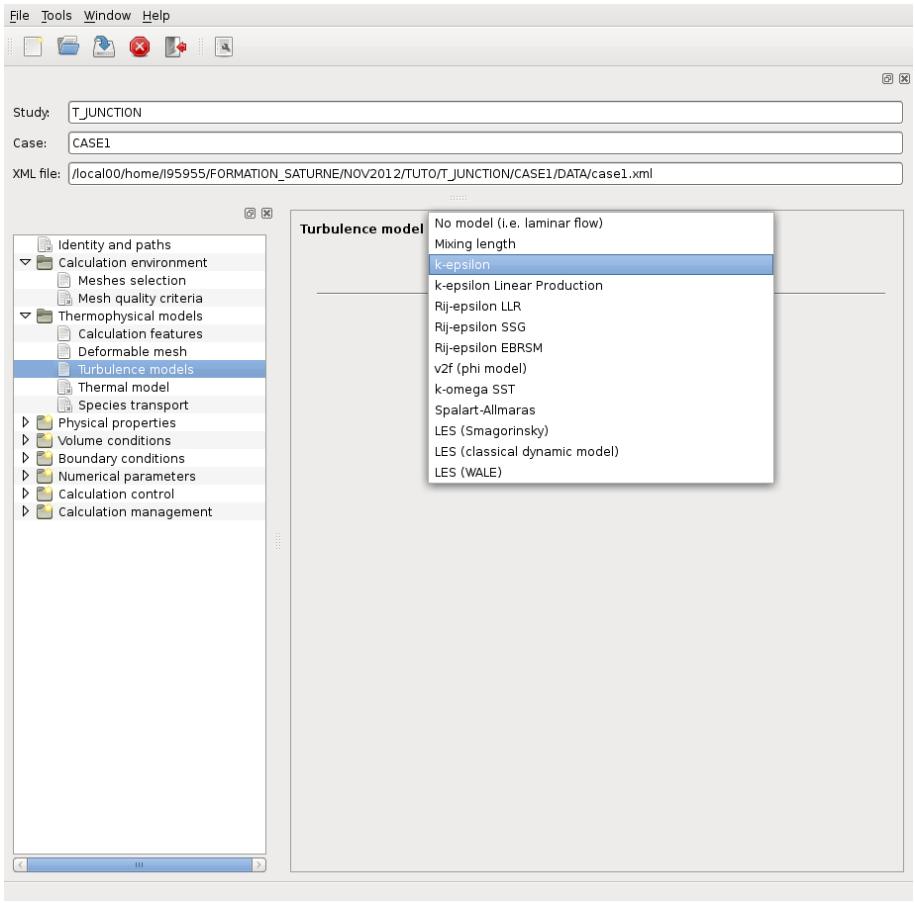


Figure VI.6: Turbulence model: list of models

In this case, the k-ε model is used.

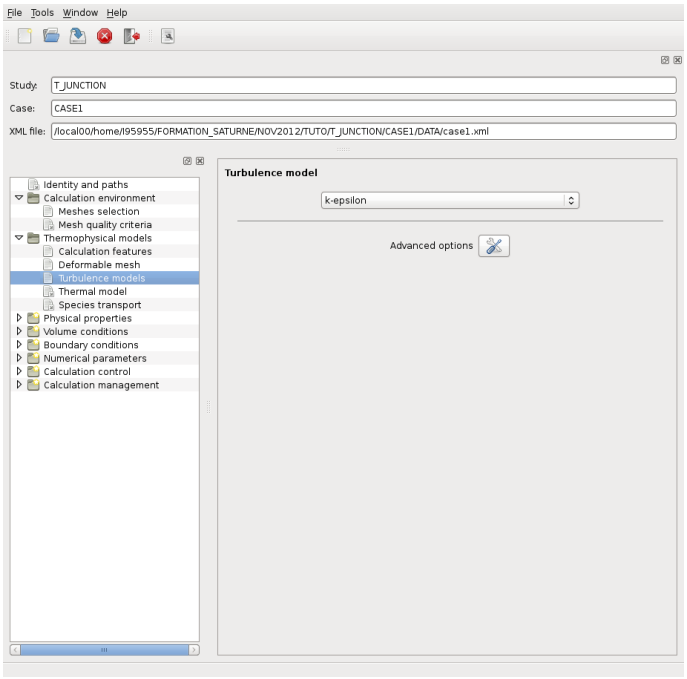


Figure VI.7: Turbulence model: choice of a model

For this study the equation for temperature must be solved. Click on the *Thermal model* item to choose between:

- No thermal scalar
- Temperature (Celsius degrees)
- Temperature (Kelvin)
- Enthalpy

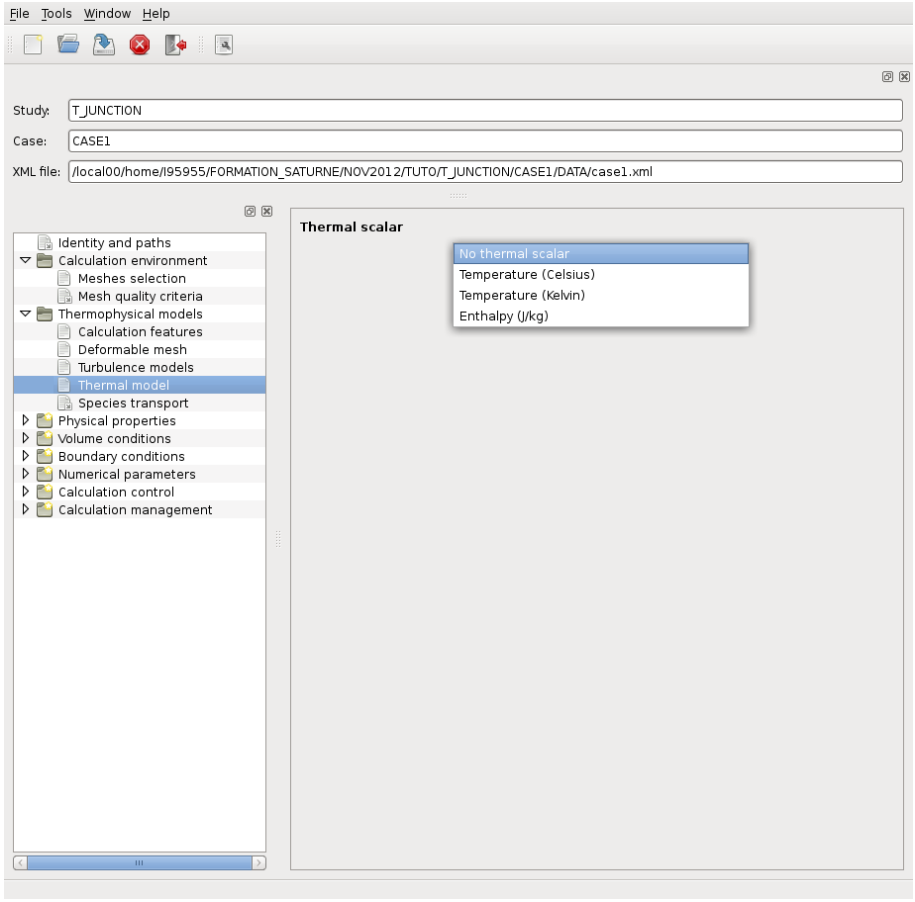


Figure VI.8: Thermal scalar conservation: list of models

In the present case, select *Temperature (Celsius)*.

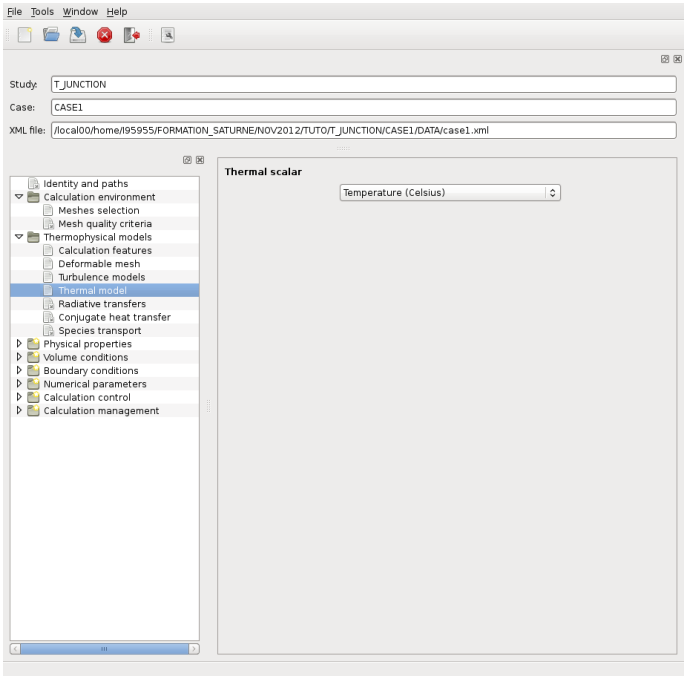


Figure VI.9: Thermal scalar conservation: choice of a model

Once the thermal scalar selected, additional items appear. There are no radiative transfers in our case, so this item can be ignored.

Initialization:

To initialize variables at the instant $t = 0$ (s), go to the *Initialization* item under the heading *Volume conditions*. Here the velocity, the thermal scalar and the turbulence can be initialized.

In this case, the default values can be kept: zero velocity, an initial temperature of **20**°C and a turbulence level based on a reference velocity of **1** $(m.s^{-1})$. Specific zones can be defined with different initializations. In this case, only the default "all cells" is used.

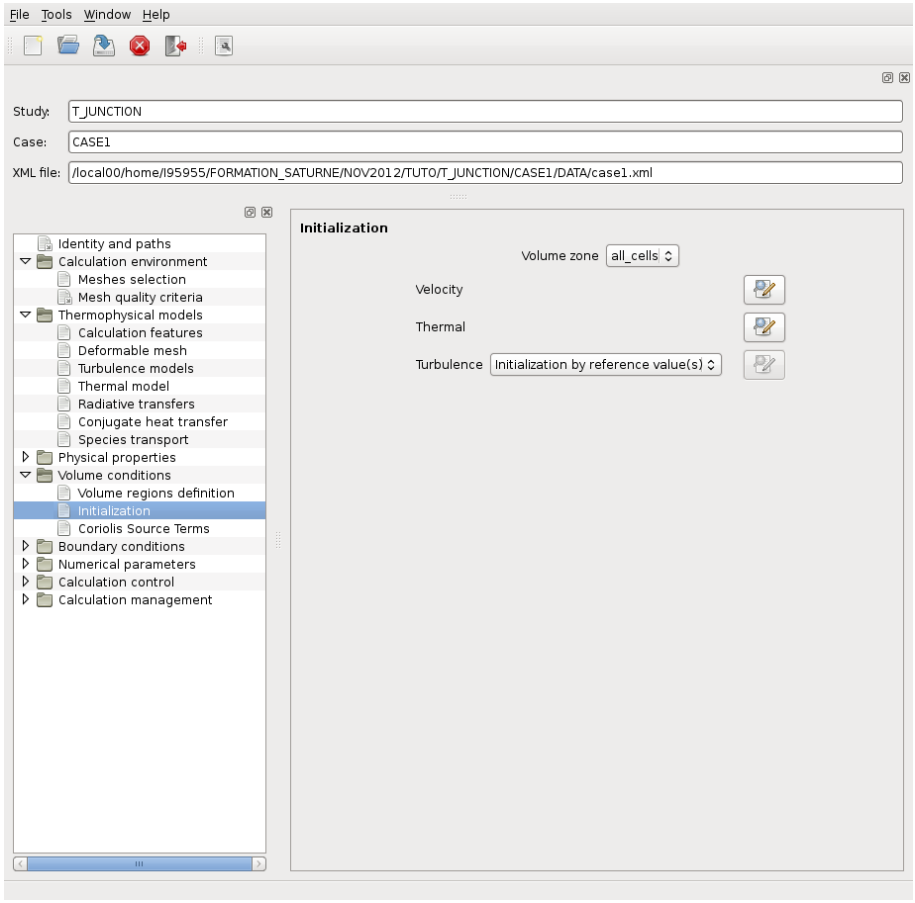


Figure VI.10: Initialization of the scalar, velocity and turbulence

- Click on the icon near *"Thermal"* in order to specify the initial value of the thermal scalar. It can be a value or a user expression.

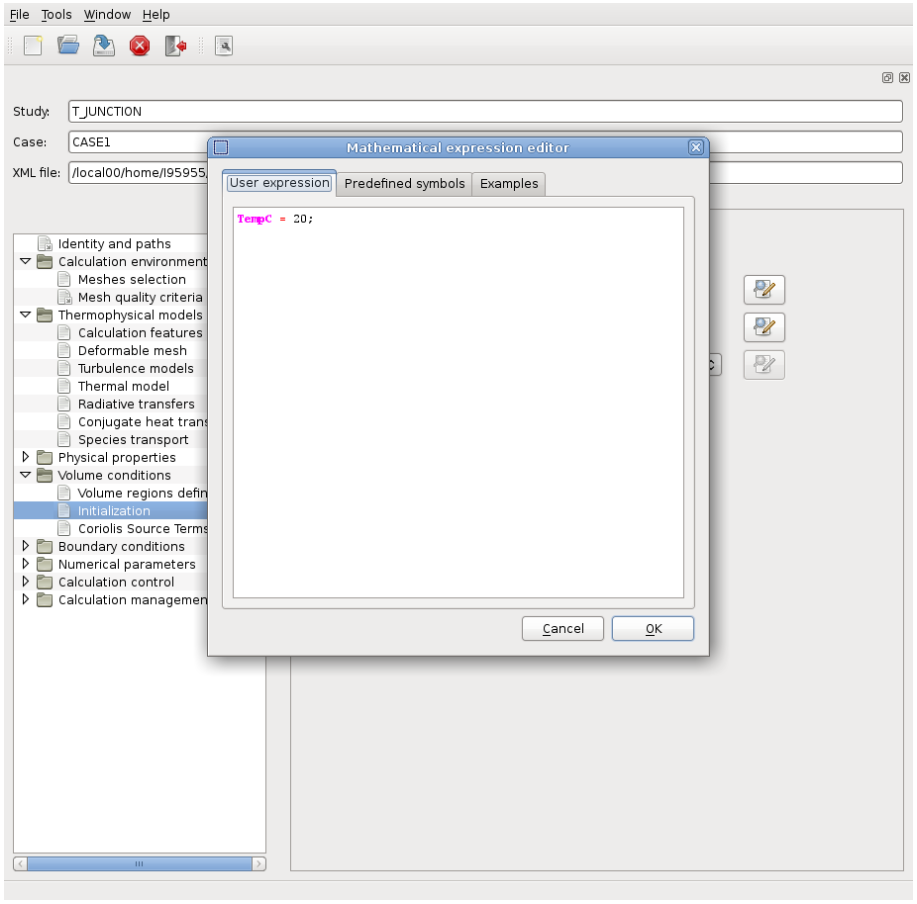


Figure VI.11: Initialization of the scalar

- To initialize the velocity, click also on the icon near *"Velocity"*.

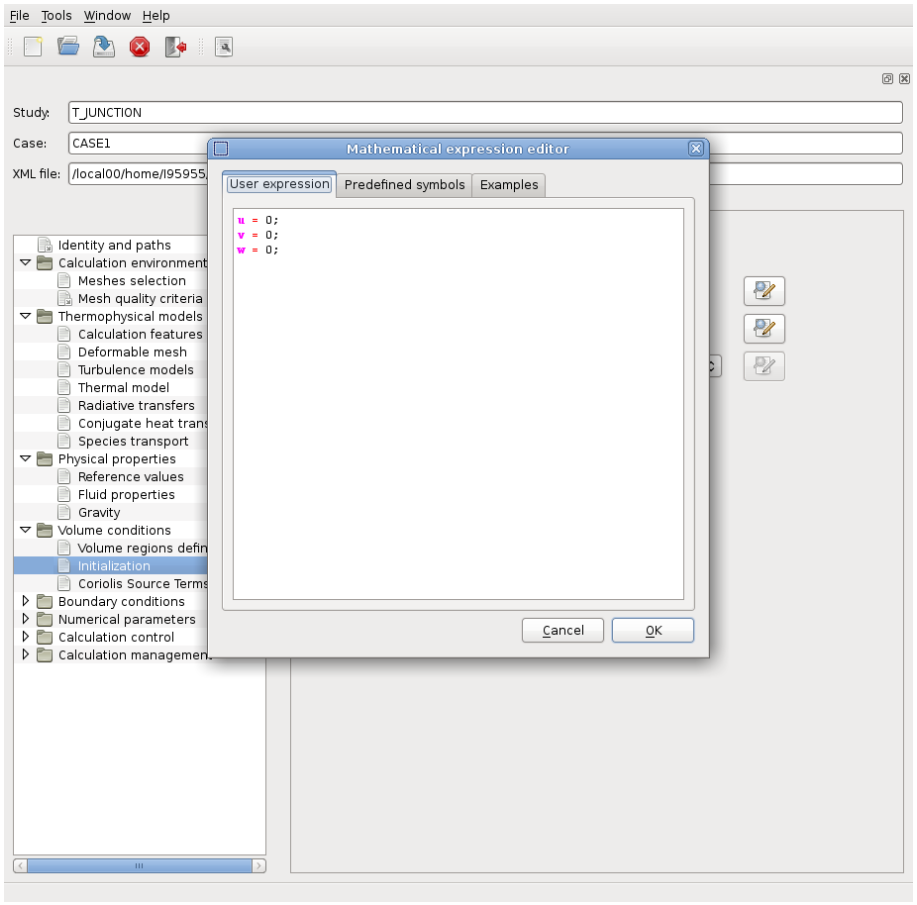


Figure VI.12: Initialization of the velocity

Under the heading *Physical properties* in the main list, the *Reference values* item allows to set the reference pressure, the reference velocity and the reference length.

Use the default value of **101 325** (Pa) for the pressure and **1** (m/s) for the velocity.

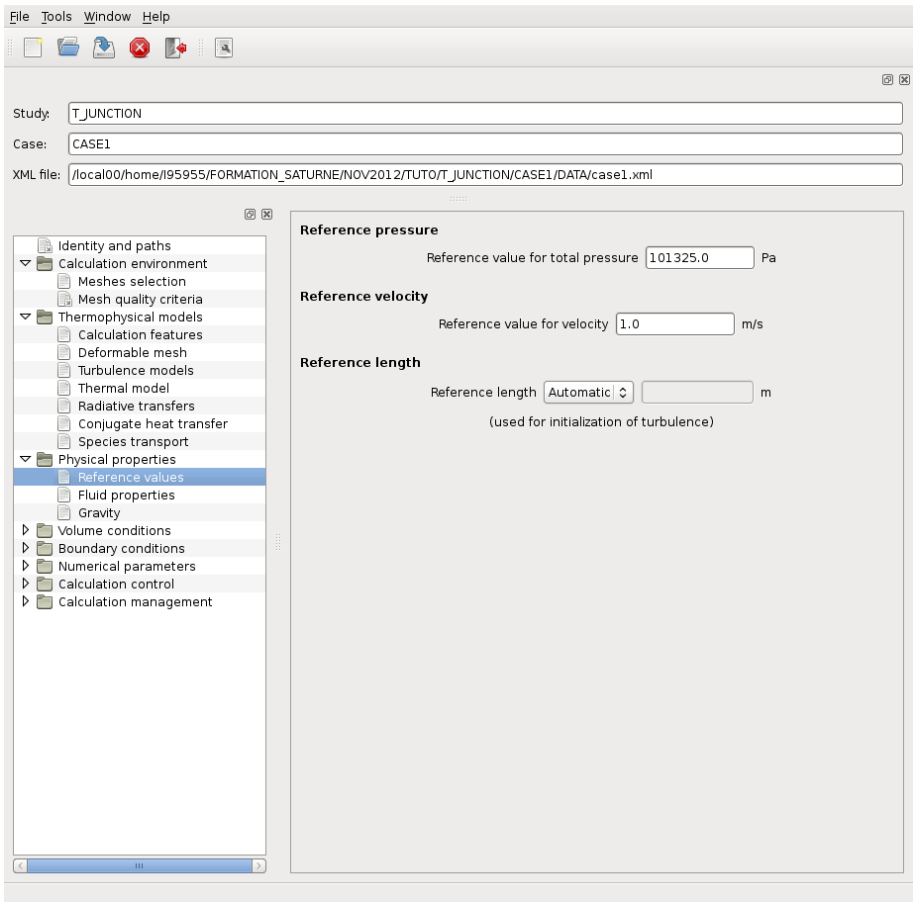


Figure VI.13: Physical properties: reference pressure

Specify the fluid physical characteristics in the *Fluid properties* item:

- Density
- Viscosity
- Specific Heat
- Thermal Conductivity

In this case they are all constant.

- $\rho \quad = 725.735\ kg.m^{-3}$
- $\mu \quad = 0.895 \times 10^{-4}\ kg.m^{-1}.s^{-1}$
- $C_p \quad = 5\,483\ J.kg^{-1}.°\mathrm{C}^{-1}$
- $(\lambda/C_p) = 0.02495\ W.m^{-1}.K^{-1}$

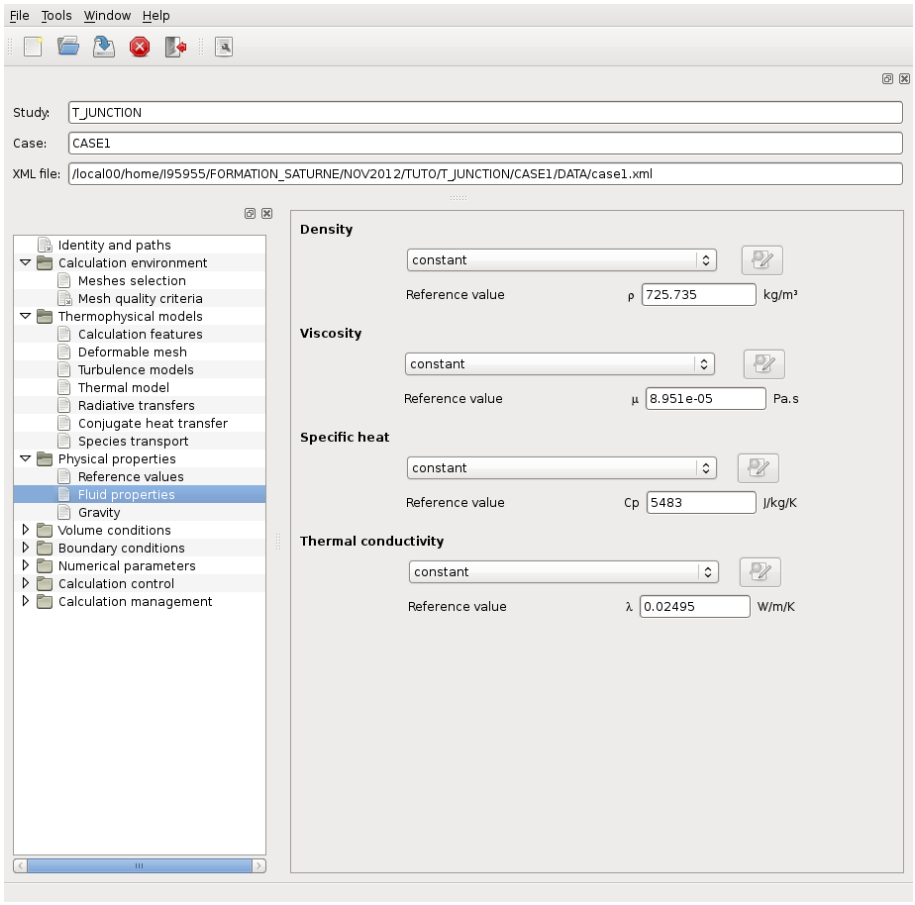


Figure VI.14: Physical properties: fluid properties

Set the three components of gravity in the *Gravity* item. In this case, since the gravity doesn't have any influence on the flow, gravity can be set to **0**.

As for the pressure interpolation, the interpolation method keeps the standard default value.

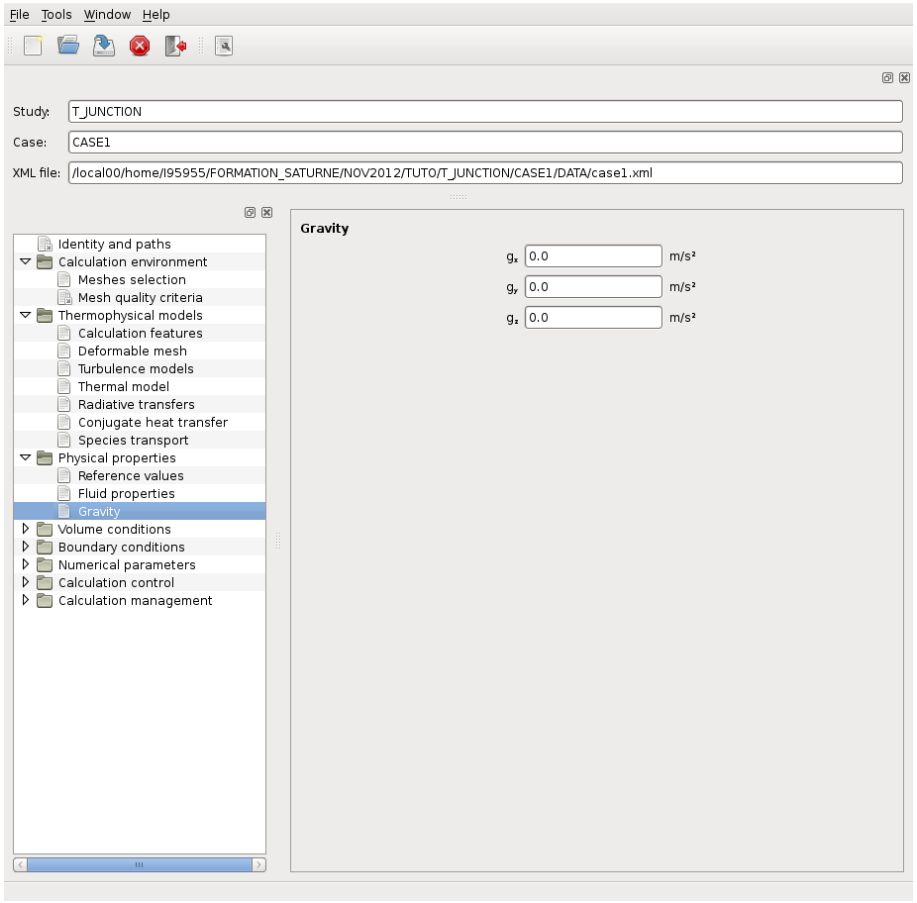


Figure VI.15: Physical properties: gravity and hydrostatic pressure

Boundary conditions now need to be defined. Go to the *Define boundary regions* item under the heading *Boundary conditions*. The following window opens (fig VI.18).

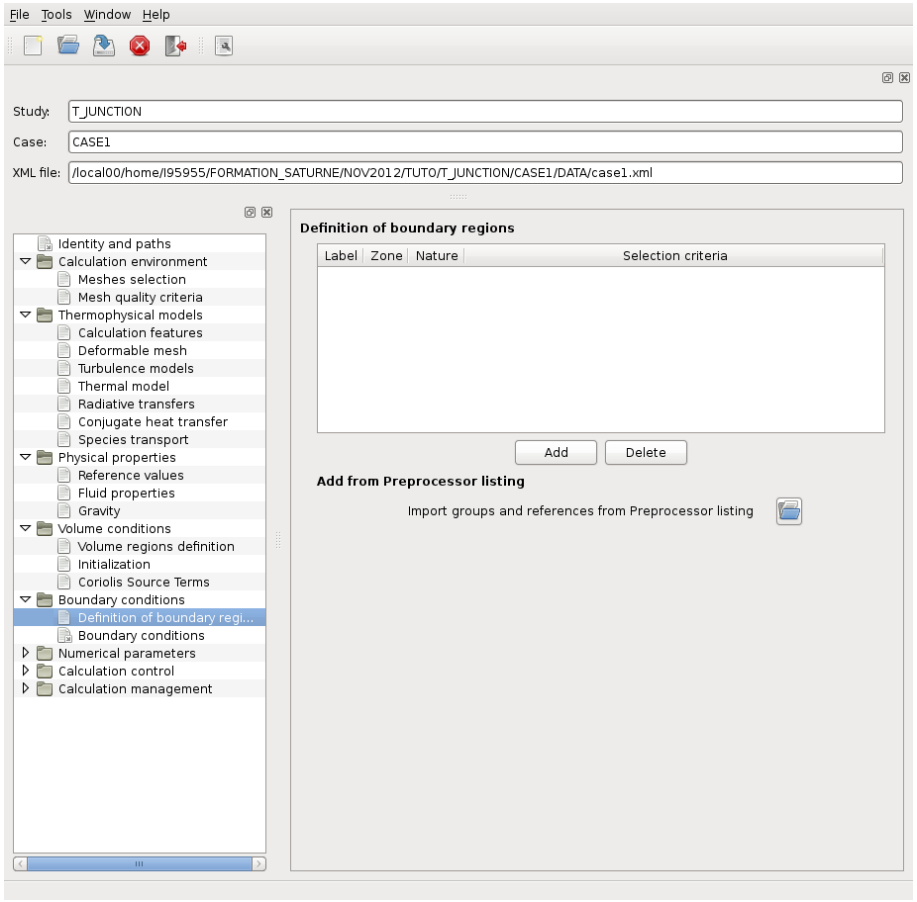


Figure VI.16: Creation of a boundary region

Each boundary must be defined. Click on *Add* to edit a new boundary. The boundary faces will be grouped in user-defined zones, based on their color or on geometrical conditions. For each zone, a reference number, a label, a nature and a selection criteria must be assigned. The different natures that can be assigned are:

```
- wall
- inlet
- symmetry
- outlet
```

The *Label* can be any character string. It is used to identify the zone more easily. It usually corresponds to the nature of the zone.

The *Zone* number can be any integer. It will be used by the code to identify the zone. No specific order or continuity in the numbering is needed.

The *Selection criteria* is used to define the faces that belong to the zone. It can be a color number, a group reference, geometrical conditions, on a combination of them, related by ‘‘or’’ or ‘‘and’’ keywords.

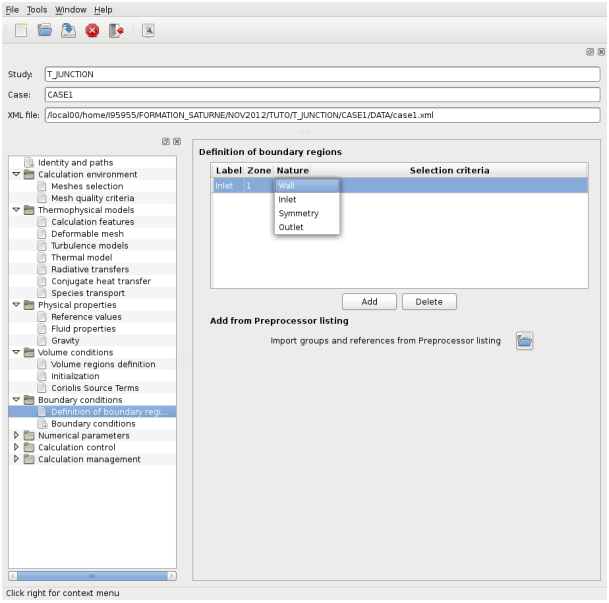


Figure VI.17: Creation of a boundary region

The specification of the inlet condition is detailled in the following pages. The settings will be as follows:

```
Label:  inlet,
Zone:  1,
Nature:  inlet,
Selection criteria:  1
```

Type all the information in the fields, the result diplays as figure VI.18

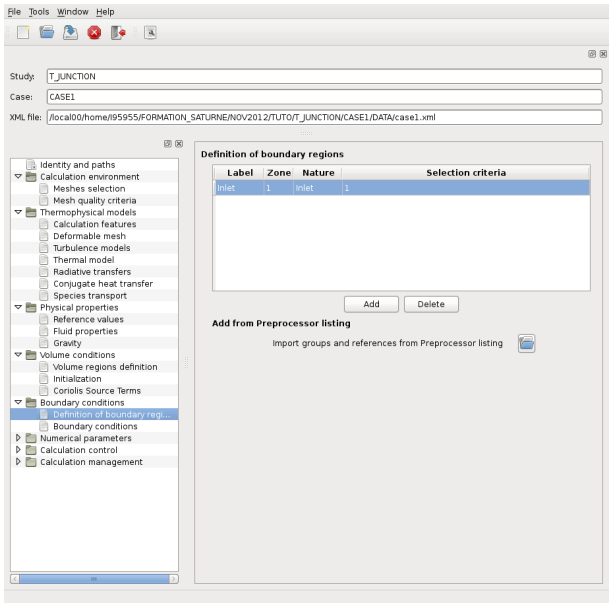


Figure VI.18: Creation of a boundary region

Remember to save the `xml` file regularly!

Do the same thing for the other boundaries.

In our case, colors 8 and 9 are symmetry boundaries. One option can be to define a separate zone for each color, as follows:

Label	symmetry_1	symmetry_2
Zone	3	4
Nature	symmetry	symmetry
Localization	8	9

But it is usually faster to regroup the different colors in one single zone, as shown on figure VI.19. In our case, the localization for this zone is the string ‘‘`8 or 9`’’.

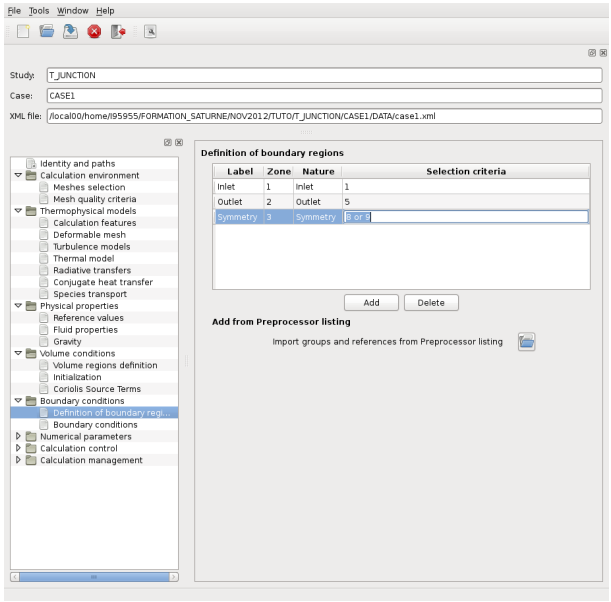


Figure VI.19: Creation of boundary regions: symmetry region

The same treatment must be done for the wall conditions. All colors 2, 3, 4, 6 and 7 can be grouped in a single boundary zone.

After defining all the boundary zones, the Interface window will look as in figure VI.20.

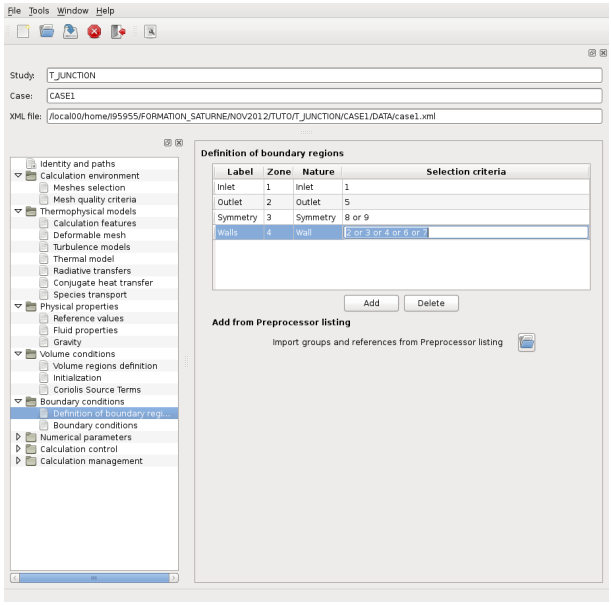


Figure VI.20: Creation of boundary regions

Now that the boundary zones are defined, the boundary conditions assigned to them will be specified. Click on the *Boundary conditions* item to set the inlet boundary conditions for velocity, turbulence and themal scalar.

As shown on figure VI.23, outlet and wall boundary zones also appear in the window.

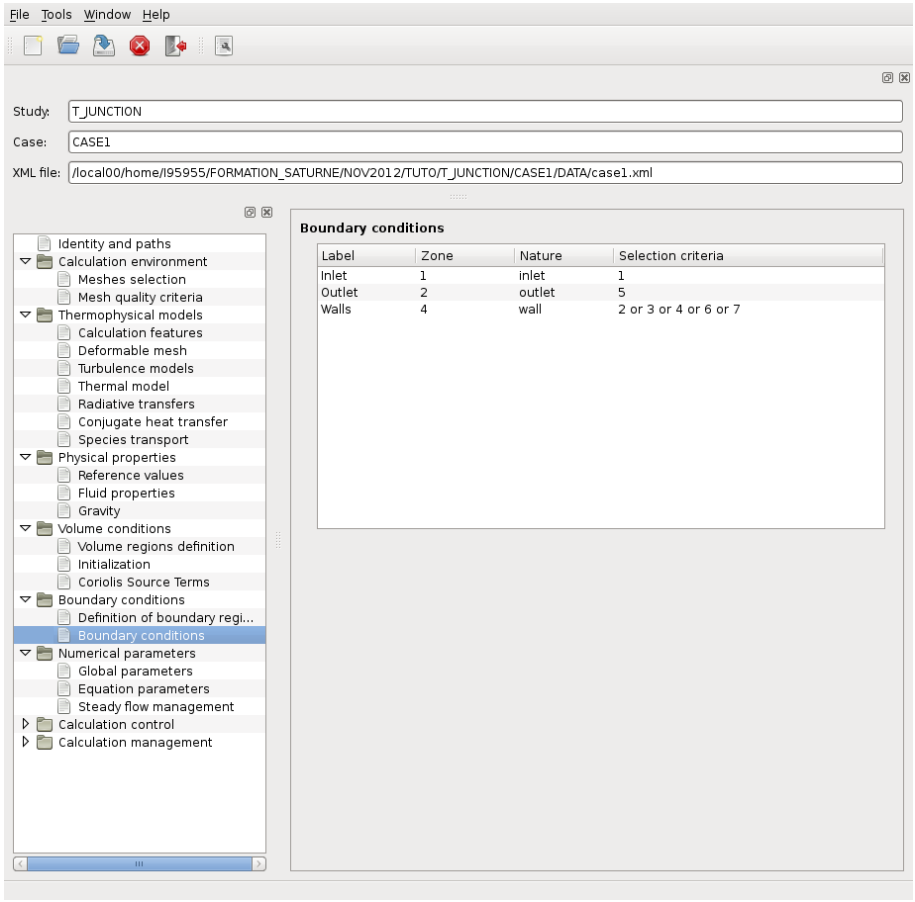


Figure VI.21: Dynamic variables boundary conditions

Click on the label *inlet*. In the section *Velocity*, select *norm*, then in the sub-section *Direction* choose *specified coordinates* and enter the normal vector components of the inlet velocity.

For the turbulence, choose the inlet condition based on a hydraulic diameter and specify it as below:

```
x = 1.0 (m) ; y = 0.0 (m) ; z = 0.0 (m)
hydraulic diameter = 0.5 (m)
```

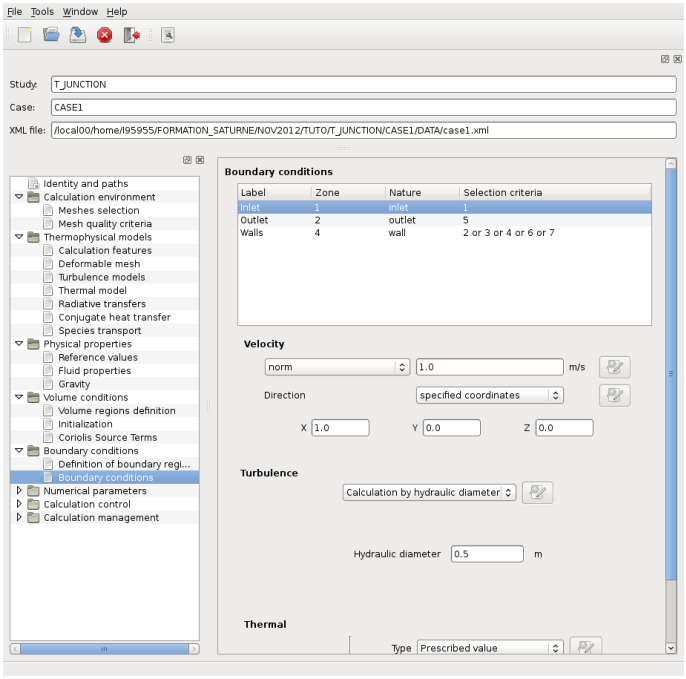


Figure VI.22: Dynamic variables boundary conditions: inlet

Click on *inlet* to choose the temperature inlet value. Here this value is **300**°C.

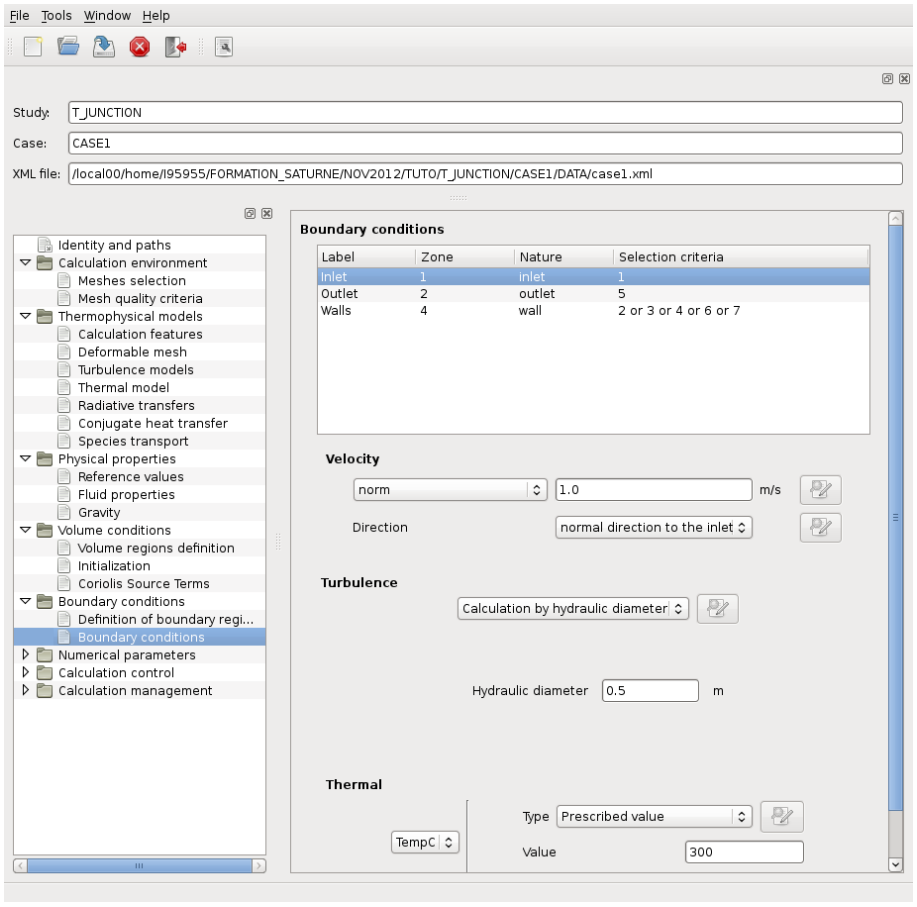


Figure VI.23: Dynamic variables boundary conditions: inlet

As for the wall boundary zone, the specifications the user might have to give is when the wall is sliding, and if the wall is `"smooth"` or `"rough"`. In this case, the walls are fixed so the option is not selected, and the wall is considered as `"smooth"`.

Note that if one of the walls had been sliding, it would have been necessary to isolate the corresponding boundary faces in a specific boundary region.

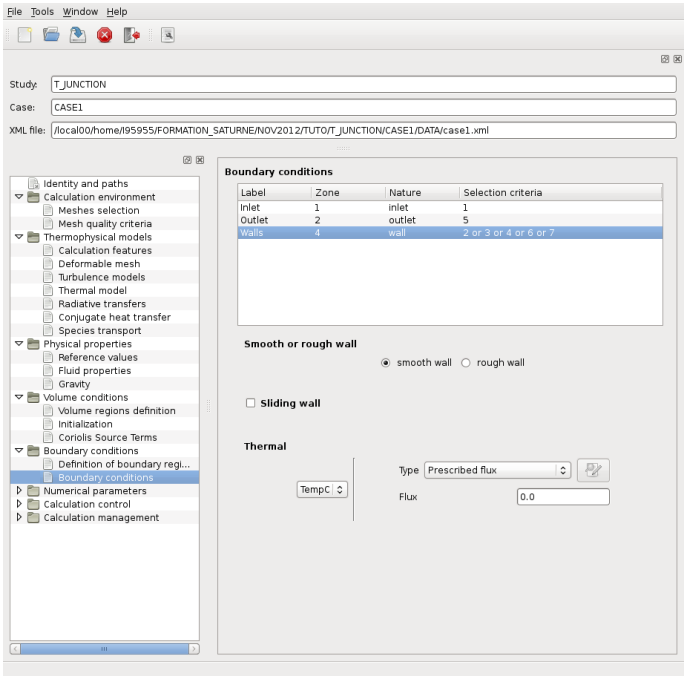


Figure VI.24: Dynamic variables boundary: walls

The boundary conditions on the temperature are only applied on inlets, outlets and walls.

For the walls, three conditions are available:

```
- Prescribed value
- Prescribed flux
- Exchange Coefficient
```

For the outlet, only *Prescribed value* and *Prescribed flux* are available, but they are taken into account only when the flow re-enters from the outlet. Otherwise, homogeneous *Prescribed flux* is considered by *Code_Saturne*.

For the inlets, only *Prescribed value* is available.

In this case all walls are adiabatic. So the boundary condition for the temperature will be a *Prescribed flux* set to **0**.

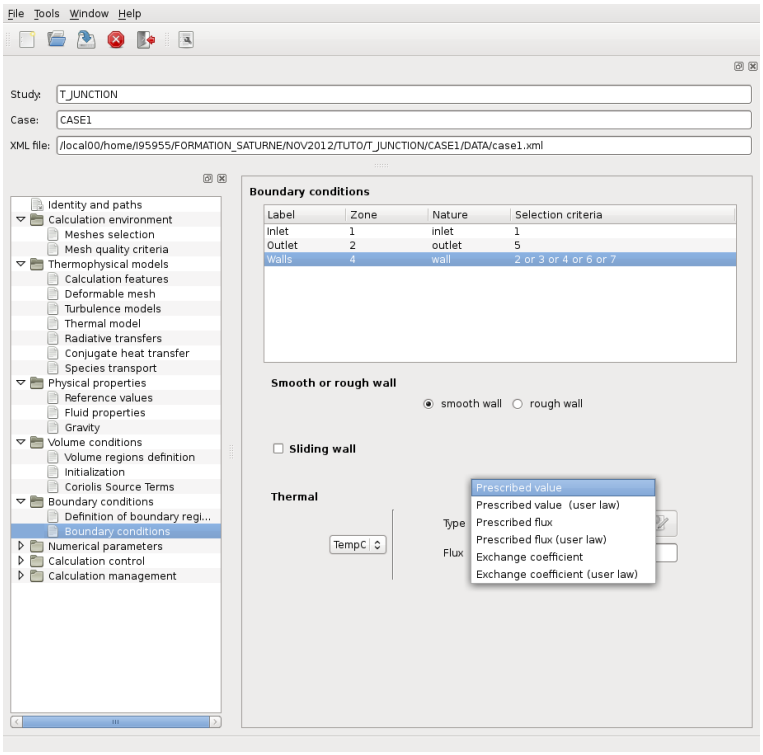


Figure VI.25: Scalars boundaries: walls

The Global parameters need then to be specified, under the header *Numerical parameters*. In this case, the SIMPLE algorithm must be chosen

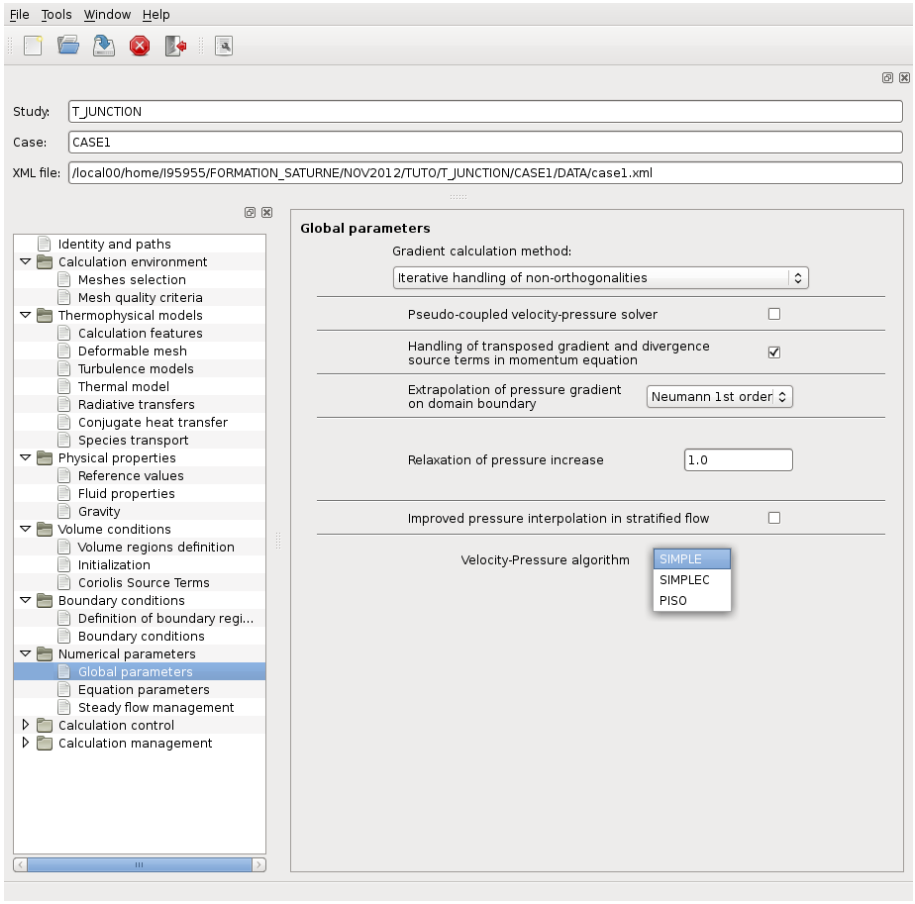


Figure VI.26: Steady flow management

After selecting the *Equation parameters* item, the tab *Scheme* allows to change different more advanced numerical parameters.

In this case none of them should be changed from their default value.

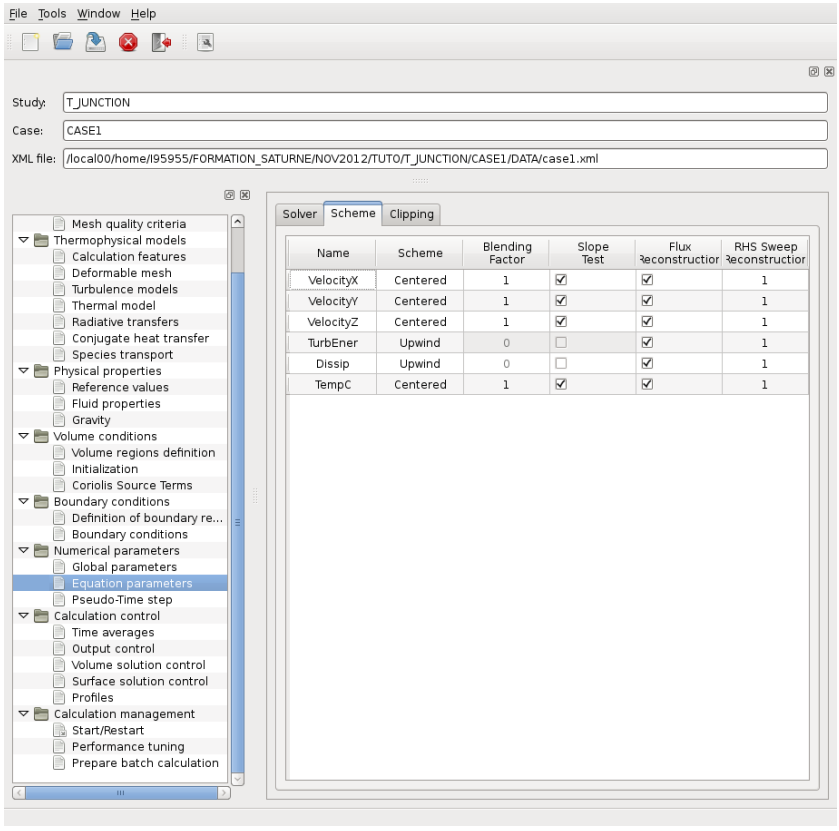


Figure VI.27: Numerical parameters

The tab *Clipping* in the *Equation parameters* item permits to vanish the too small or too big value.

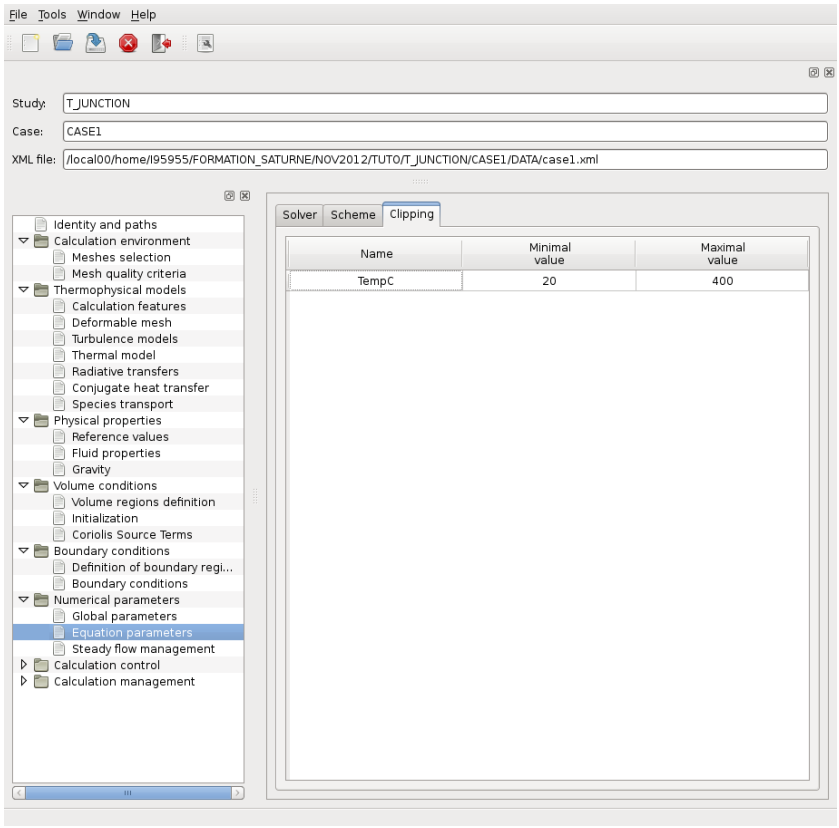


Figure VI.28: Clipping

Go to the *Steady flow management* item to specify the number of iterations, **300** in this case. The relaxation coefficient is equal to **0.9** and the *Zero iteration option* will not be activated.

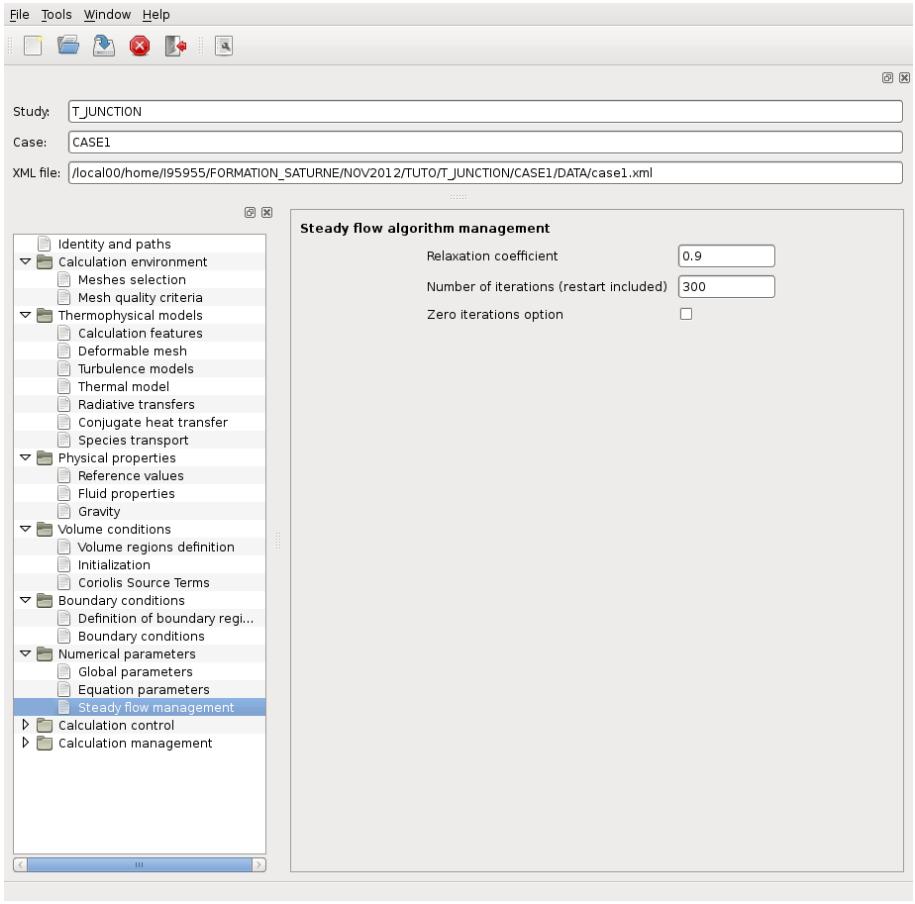


Figure VI.29: Steady flow management

Under the heading *Calculation control*, click on the *Output control* item to change the frequency for the printing of information in the output listing.

The options are:

```
- No output
- Output listing at each time step
- Output at each 'n' time step (the value of 'n' must then be specified)
```

Here and in most cases, the second option should be chosen.

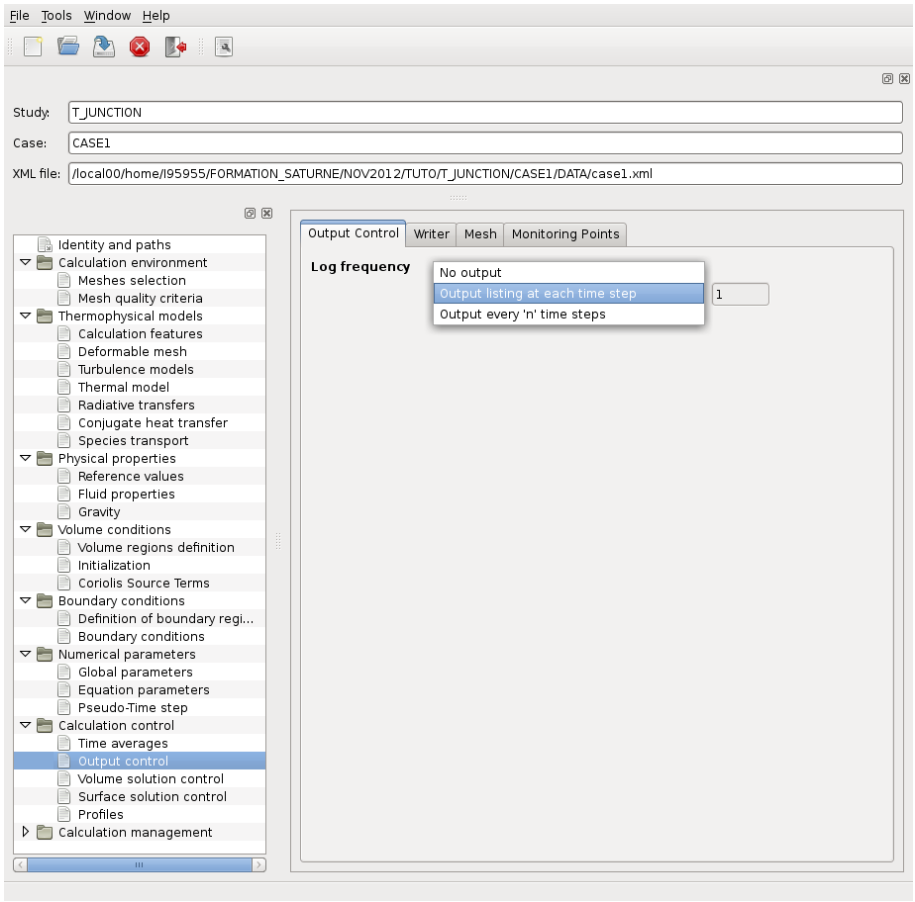


Figure VI.30: Output control: output listing

For the post-processing (by default EnSight format files), there are three options:

```
- Only at the end of calculation
- At each time step
- Post-processing every 'n' time steps
```

In this case, we are interested in the evolution of the variables during the calculation, so the second option is chosen.

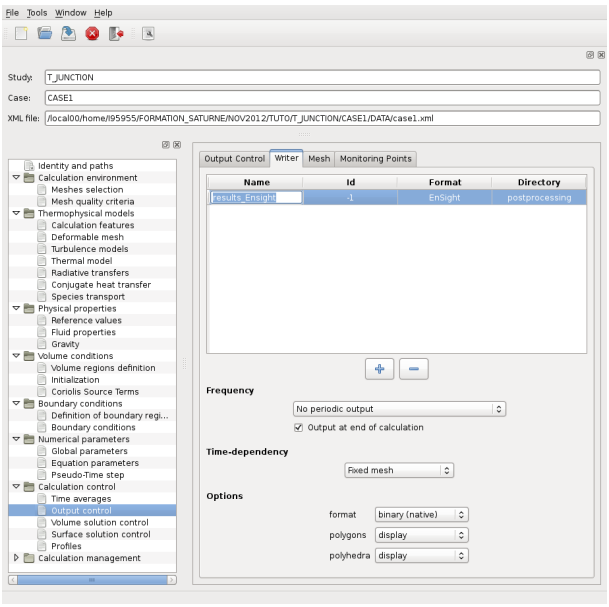


Figure VI.31: Output control: post-processing

The other options are kept to their default value.

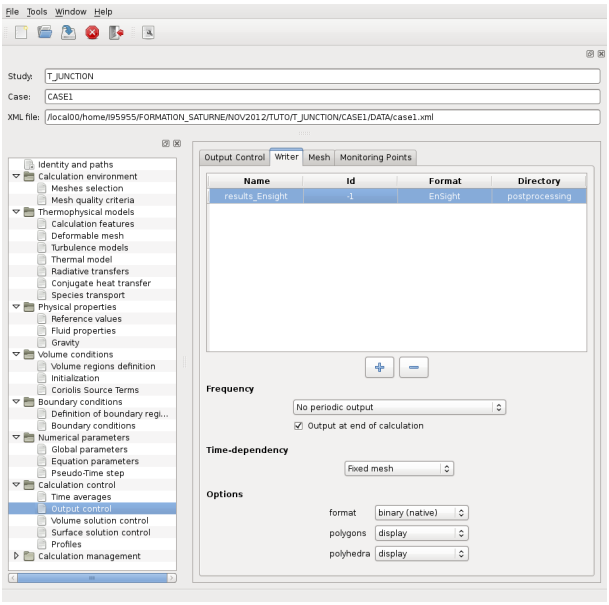


Figure VI.32: Output control

The *Monitoring Points* tab allows to define specific points in the domain (monitoring probes) where the time evolution of the different variables will be stored in historic files. In this case no monitoring points are defined.

The *Volume solution control* item allows to specify which variable will appear in the output listing, in the post-processing files or on the monitoring probes. In this case, the default value is kept, where every variable is activated.

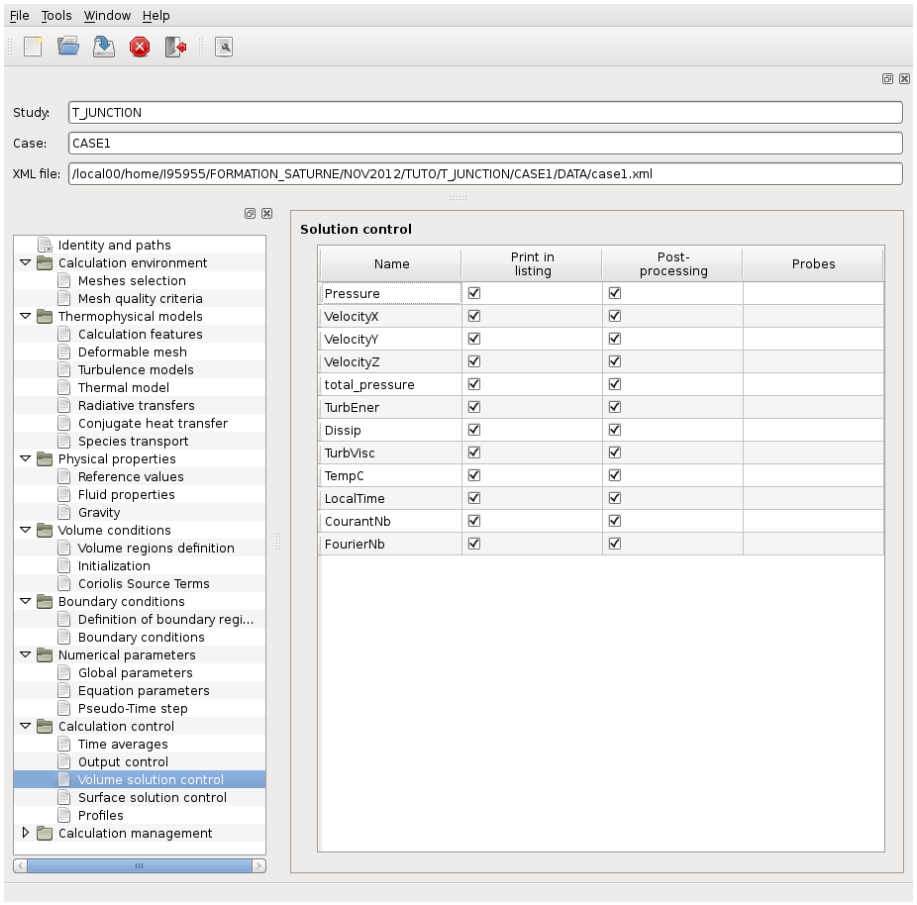


Figure VI.33: Solution control

The *Start/Restart* item allows to start a new calculation from the results of a former one. It is not the case in the present calculation so nothing has to be modified.

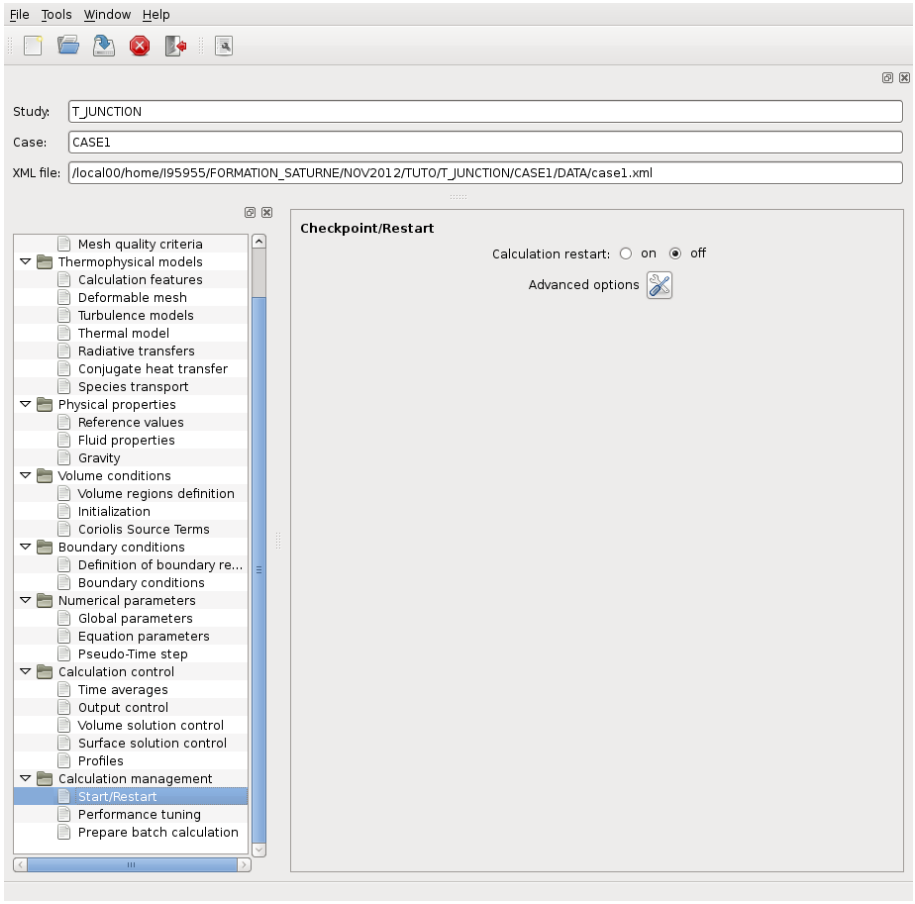


Figure VI.34: Start/Restart

The final item, *Prepare batch calculation*, is used to prepare the launch script and, on certain architectures, launch the calculation.

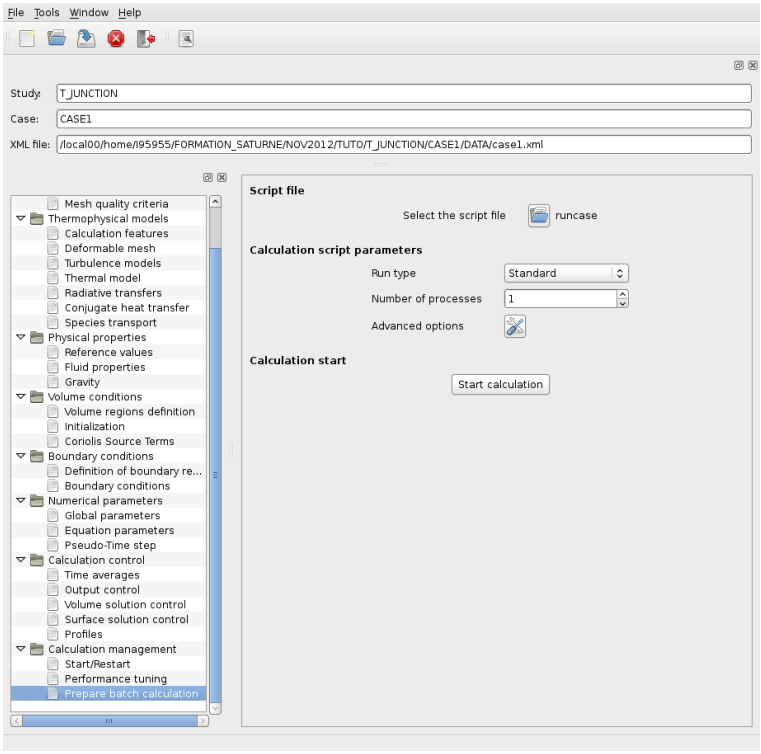


Figure VI.35: Prepare batch analysis: Computer selection

Click on the icon to *Select the batch script file* to select the launch script. The default launch script is named `runcase` and is located in the `SCRIPTS/` directory. Select it and click on *Open*.

Remember to save the `xml` file before opening the launch script.

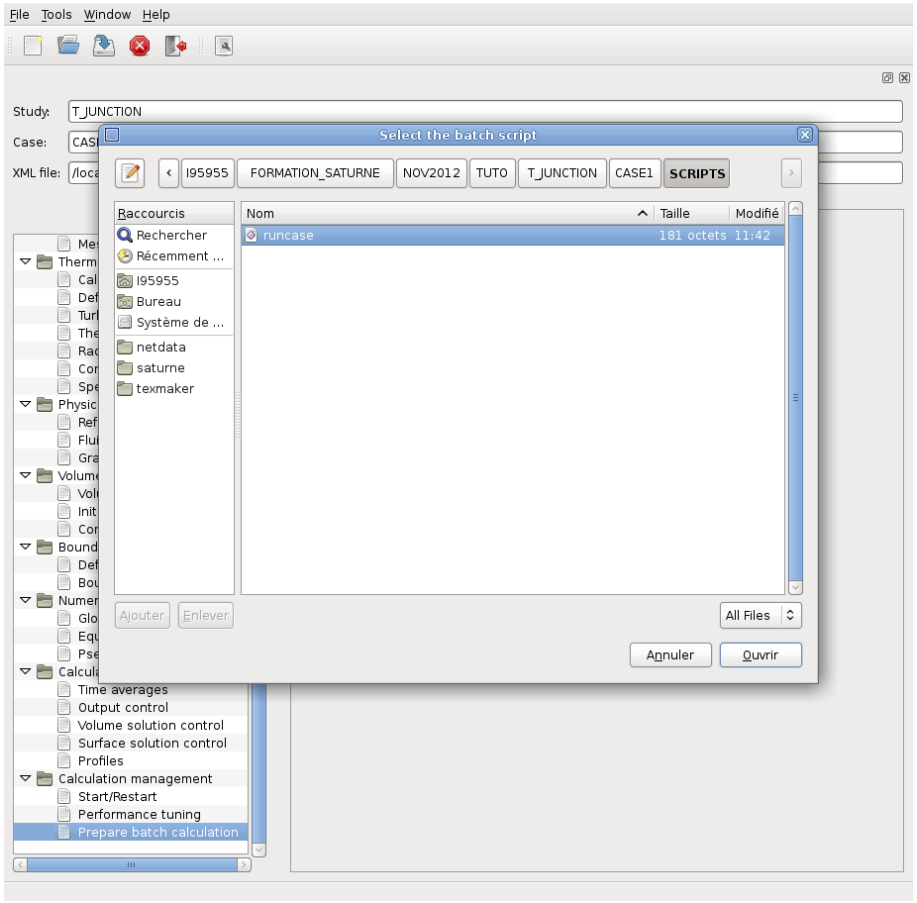


Figure VI.36: Prepare batch analysis: batch script file selection

When the script is selected, new options will appear. On this calculation, the number of processors used will be left to 1.

Finally, the *Advanced options* icon allows to change some more advanced parameters that will not be needed in this simple case.

Eventually, save the `xml` file and execute it by clicking on *start calculation*. The results will be copied in the `RESU/` directory.

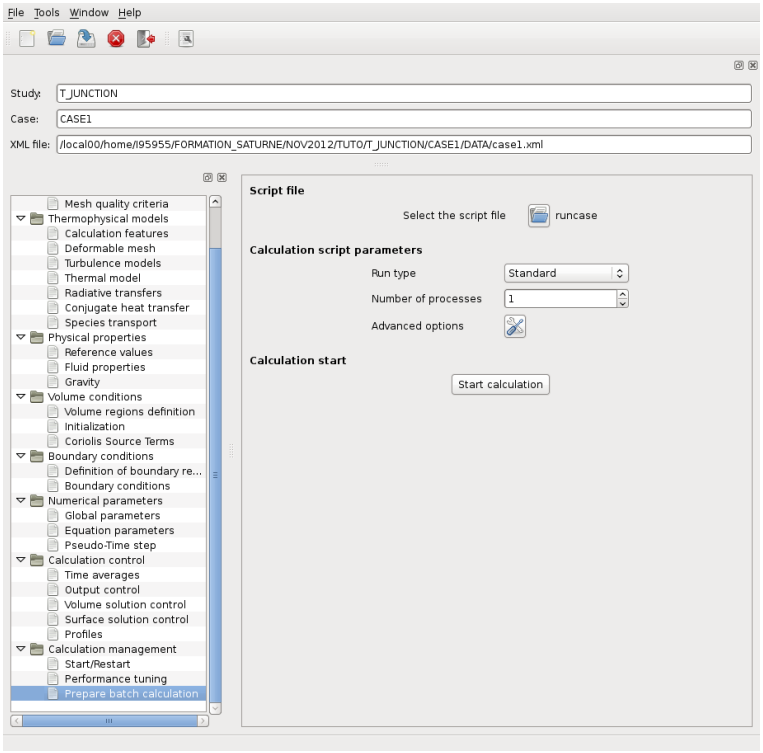


Figure VI.37: Prepare batch analysis: Execution

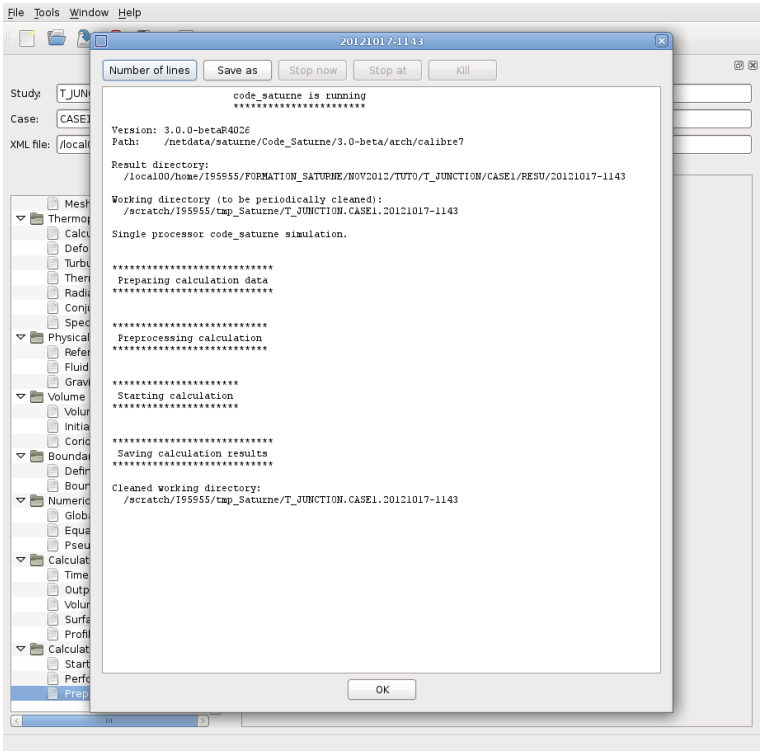


Figure VI.38: Run

2 Solution for case2

This case corresponds to a new study, in which there will be three calculation cases (cases 2, 3 and 4). We can create one case in a single `code_saturne create` command and additional cases can be added later. To test this functionality, first create the study directory, with case subdirectory `case2`, as below:

```
$ code_saturne create -s full_domain -c case2
$ cd full_domain
```

Go to the `DATA` directory in `case2`, open a new case and select the meshes to use.

Click on the heading *Calculation environment* then on the *Meshes selection* item. In this case, you must add three meshes which have to be joined.

In order to join the three meshes, you must add a selection criteria in the box *Selection criteria*. In this case, only faces of colors 5, 24 and 32 are liable to be joined (different colors can be entered on a single line, separated by comma).

Click on the + icon to enter the list of colors to be joined in the *Face joining (optional)* item.

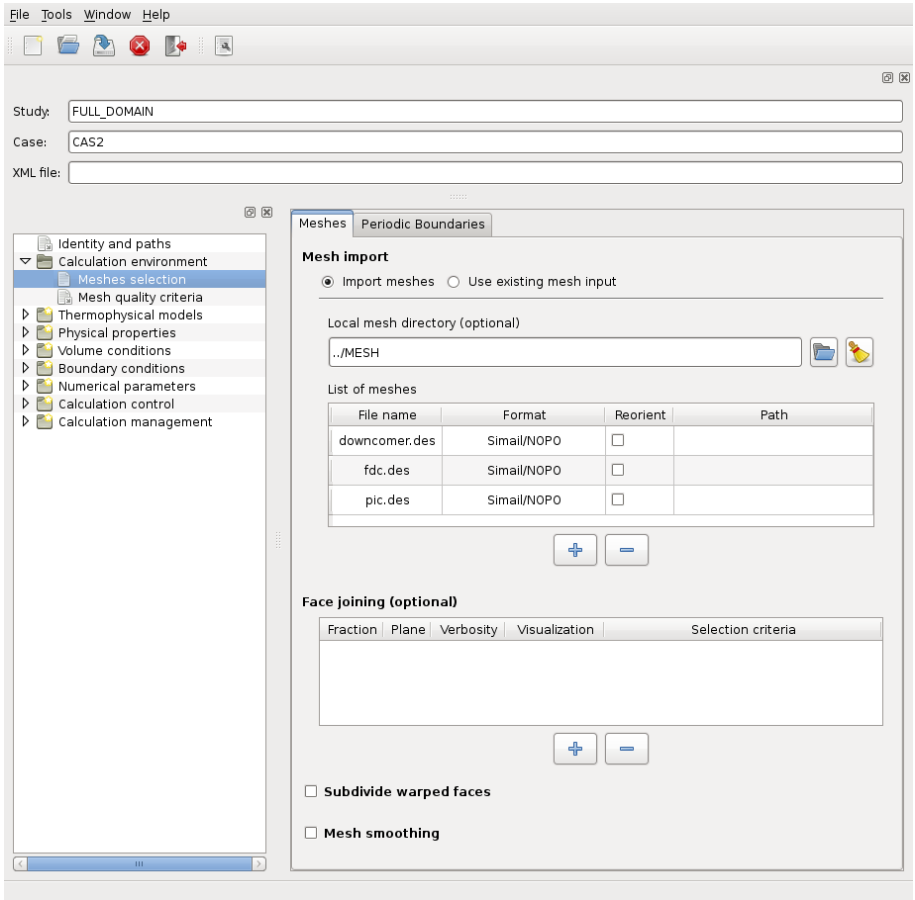


Figure VI.39: Meshes: list of meshes

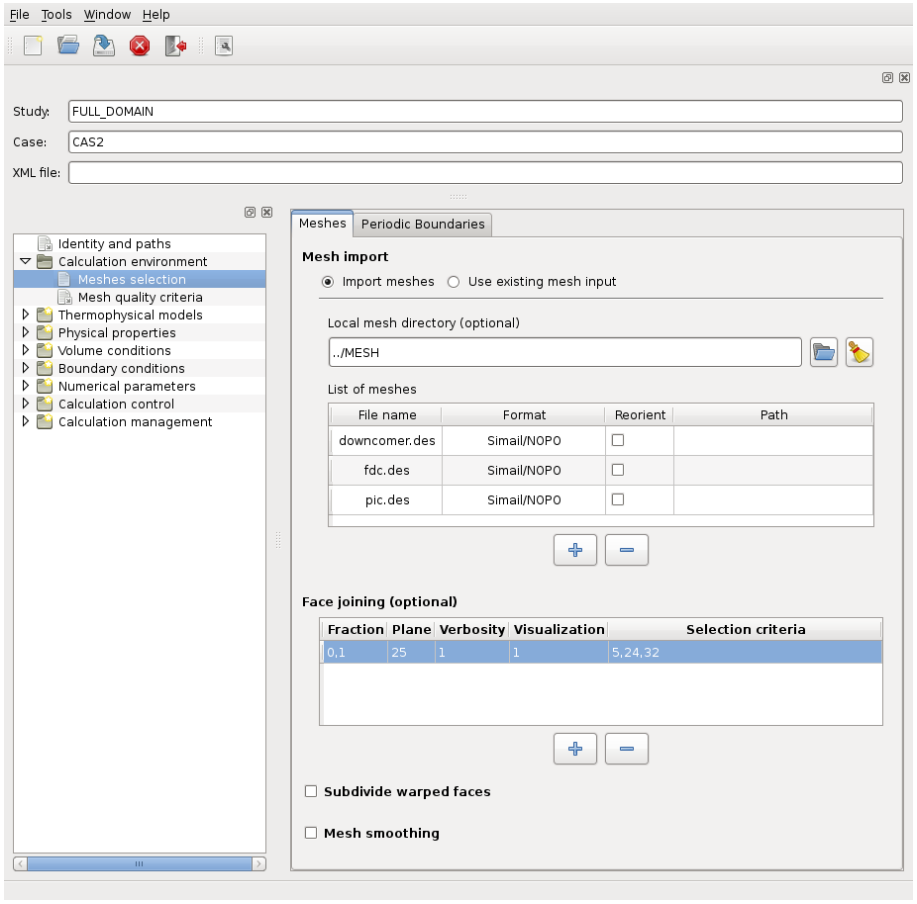


Figure VI.40: Join a Mesh

You can now verify the quality of your mesh.

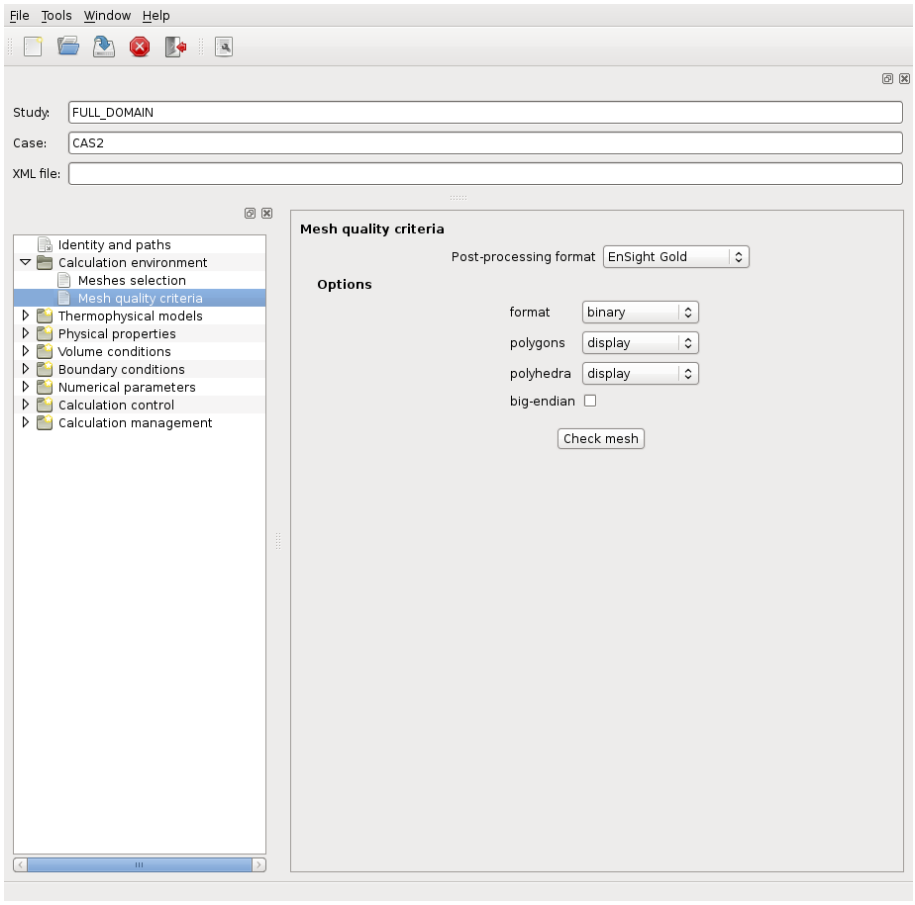


Figure VI.41: Mesh quality criteria

In this case "Unsteady flow" must be selected in the *Calculation features* item.

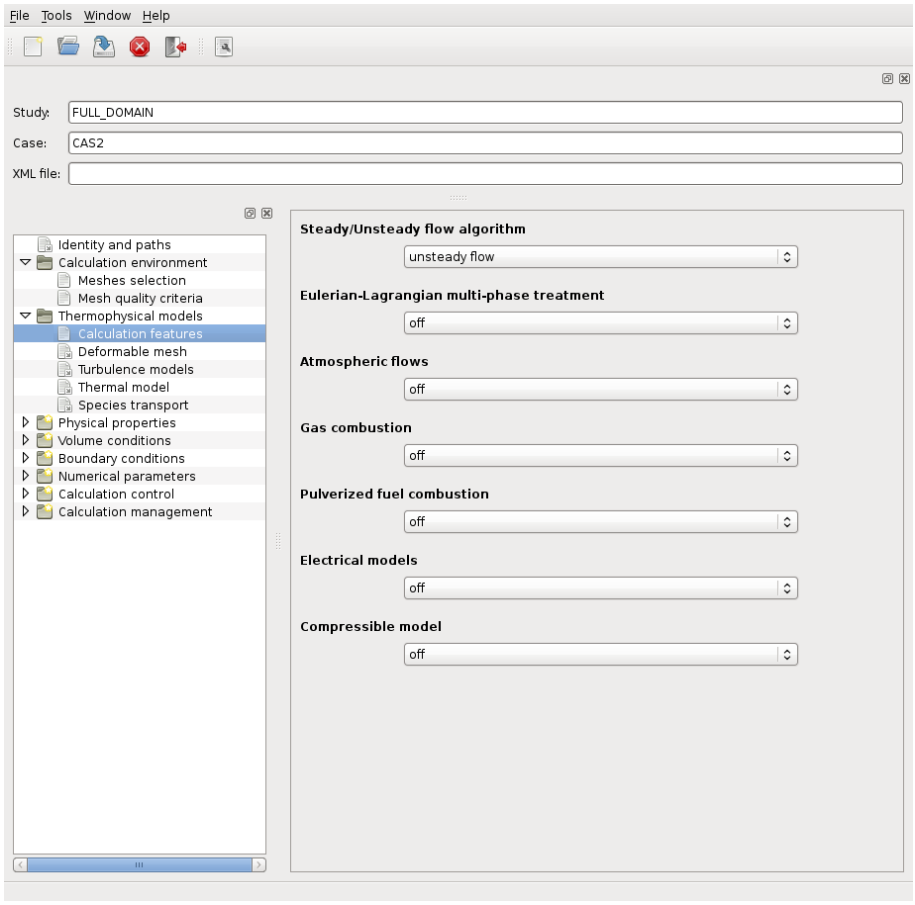


Figure VI.42: Thermophysical models - Analysis features - Unsteady flow

The rest of the heading *Thermophysical models* is identical to `case1`.

To add an additional scalar, click on the *Species transport* item under the *Thermophysical models* heading.

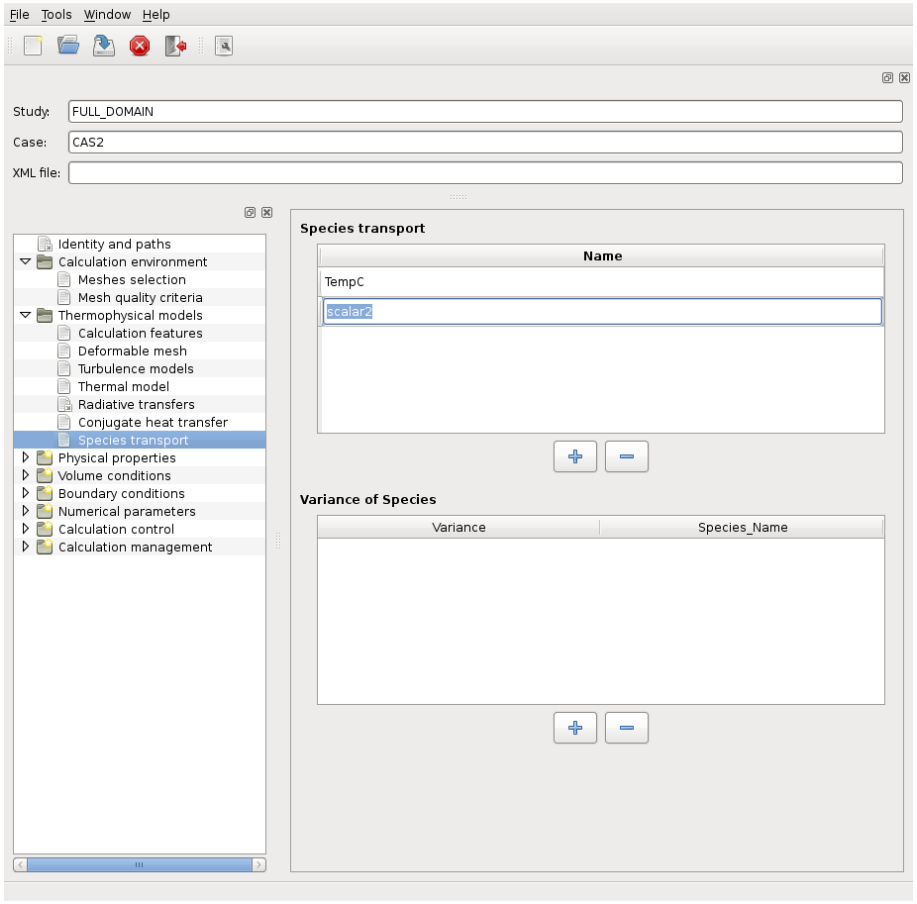


Figure VI.43: Additional scalar

The heading of *Physical properties* is identical to `case1`.
In the *Fluid properties* item, still under the heading *Physical properties*, specify the diffusion coefficient of this new scalar.

Click on the scalar name to highlight it, then enter the value in the box. In this case, the species diffusion coefficient value is **0.855** ($\times 10^{-5}$ $m^2.s^{-1}$) for the `scalar2` scalar to solve.

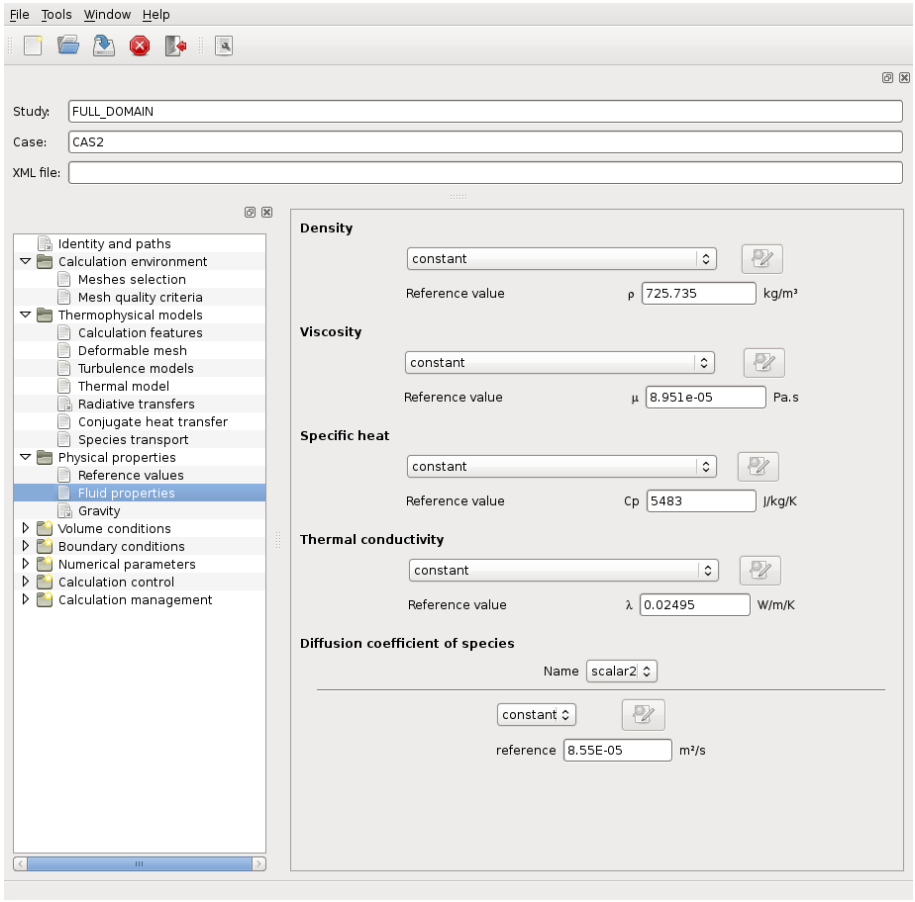


Figure VI.44: Fluid properties

Initialization:
To initialize variables at the instant $t = 0\ s$, go to the *Initialization* item under the heading *Volume conditions*.

Here the velocity, the thermal scalar and the turbulence can be initialized. In this case, the default values can be kept: zero velocity, an initial temperature of **20**°C and a turbulence level based on a reference velocity of **1** $(m.s^{-1})$. You must also initialize the `scalar2` species at **10**°C.

Specific zones can be defined with different initializations. In this case, only the default "all cells" is used.

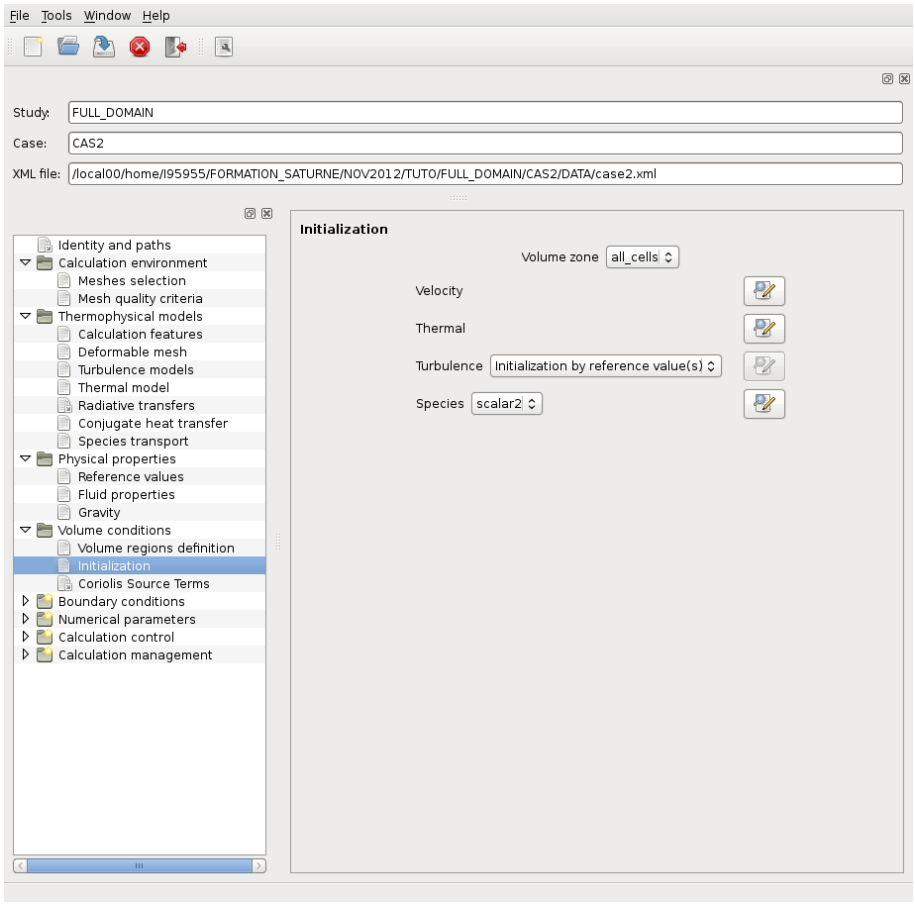


Figure VI.45: Initialization

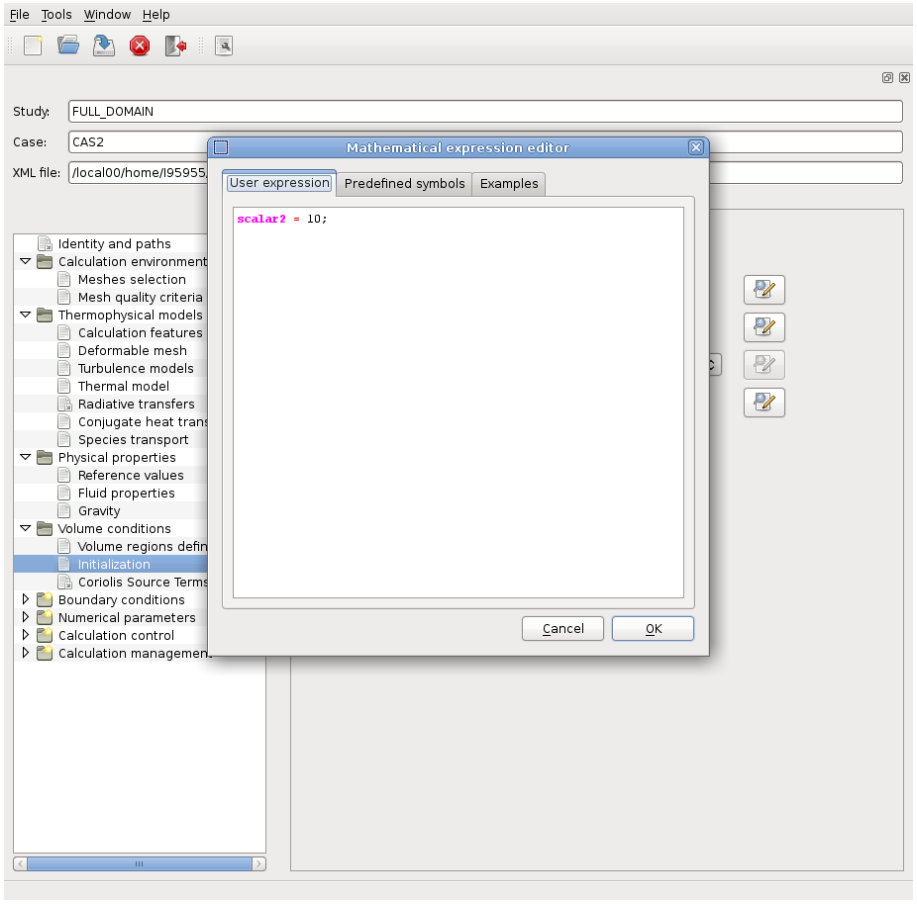


Figure VI.46: Initialization- Species

•**Create the boundary zones**:
The procedure is the same as in case 1, but the colors are different. Note that colors 5 and 32 have completely disappeared in the joining process (they are now internal faces and are not considered as boundaries), while some boundary faces of color 24 remain.

Create the inlet, outlet and symmetry boundary zones with the following colors:

```
- inlet   :  ''1''
- outlet  :  ''34''
- symmetry:  ''8 or 9 or 28 or 29 or 38 or 39''
```

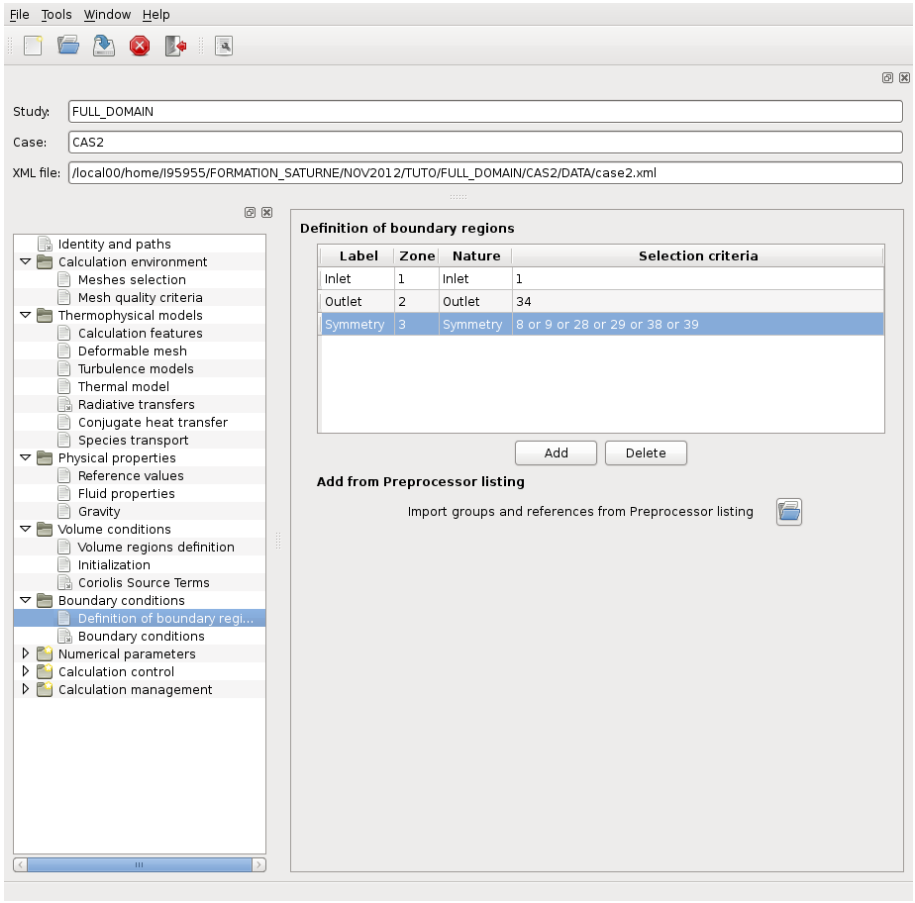


Figure VI.47: Creation of the boundary zones

In this case, different conditions are applied for the walls. Separate corresponding wall boundary regions must therefore be created, following the data in the following table.

Label	Zone	Nature	Selection criteria
wall_2	5	wall	2 or 3
wall_3	6	wall	4 or 7 or 21 or 22 or 23
wall_4	7	wall	6 and y>1
wall_5	8	wall	6 and y⩽1
wall_6	9	wall	31 or 33

The “wall_1” region combines color and geometrical criteria. The associated character string to enter in the “Selection criteria” box[1] is as follows:

```
''24 and 0.1 <= x and 0.5 >= x''
```

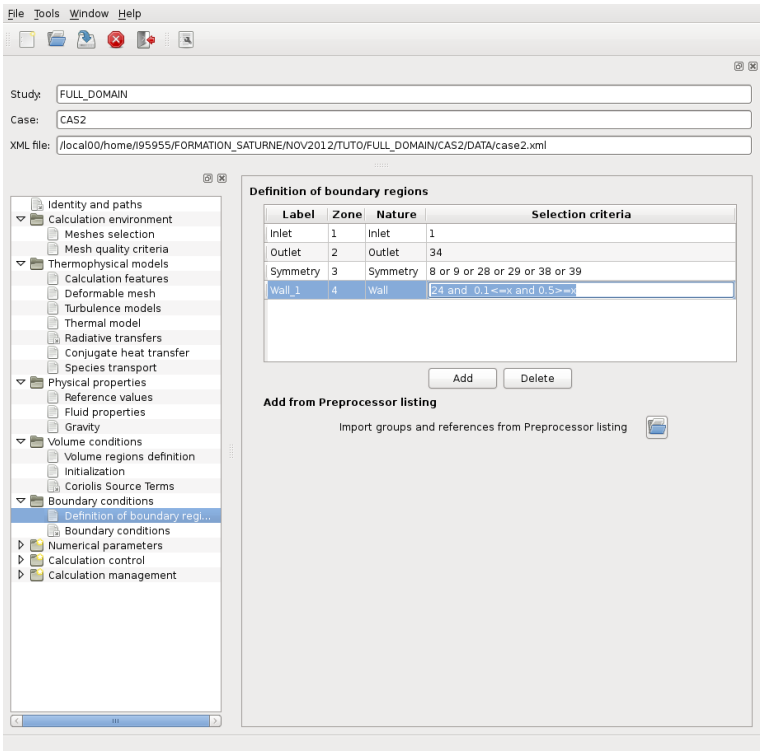


Figure VI.48: Creation of a wall boundary region

[1]Note that, due to the joining process, there are in fact no boundary faces of color 24 with x coordinate outside the [0.1;0.5] interval. The geometrical criterium is therefore not necessary. It is presented here to show the capacity of the face selection module.

Define the other wall boundary zones. The faces of color 6 have to be divided in two separate zones, based on a geometrical criterium on y.

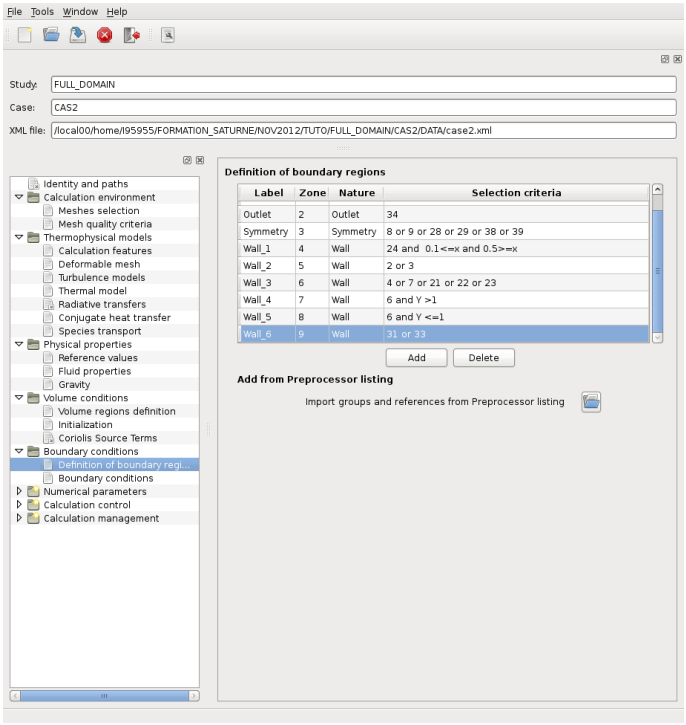


Figure VI.49: Creation of wall boundary regions

The dynamic boundary conditions are the same as in case 1 for the inlet, and there are still no sliding walls.

- **Inlet:**

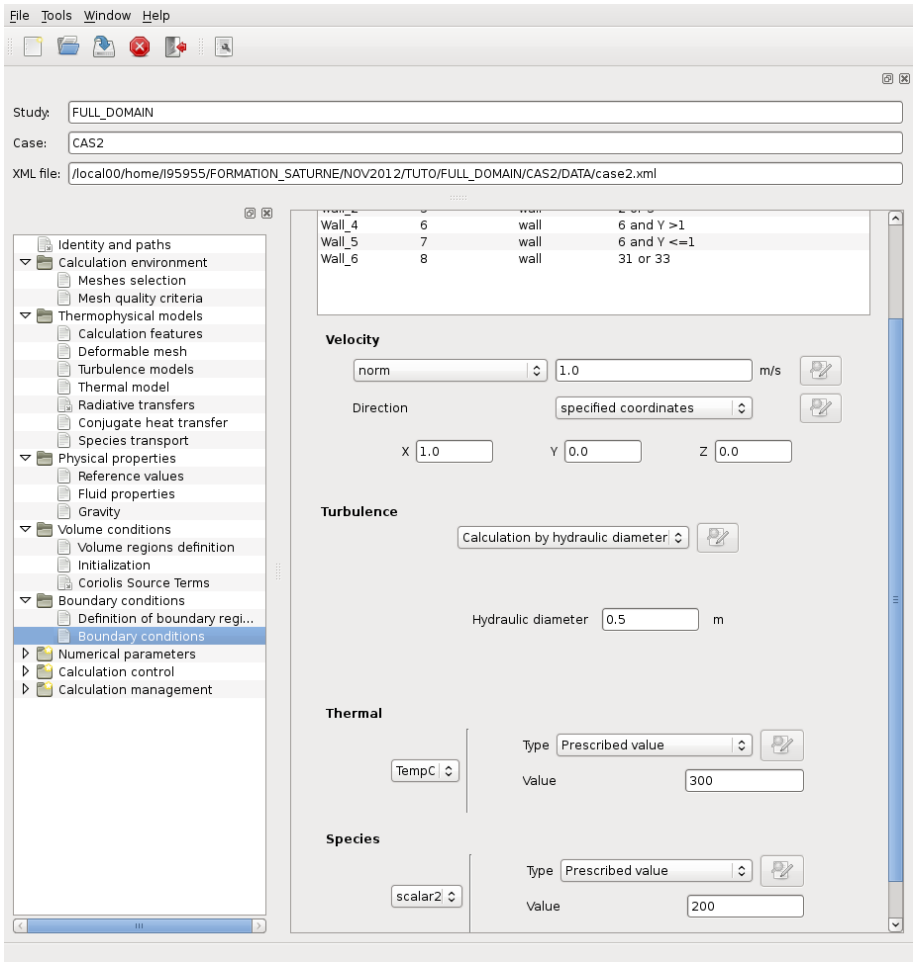


Figure VI.50: Dynamic variables boundary: Inlet

- **Outlet:**

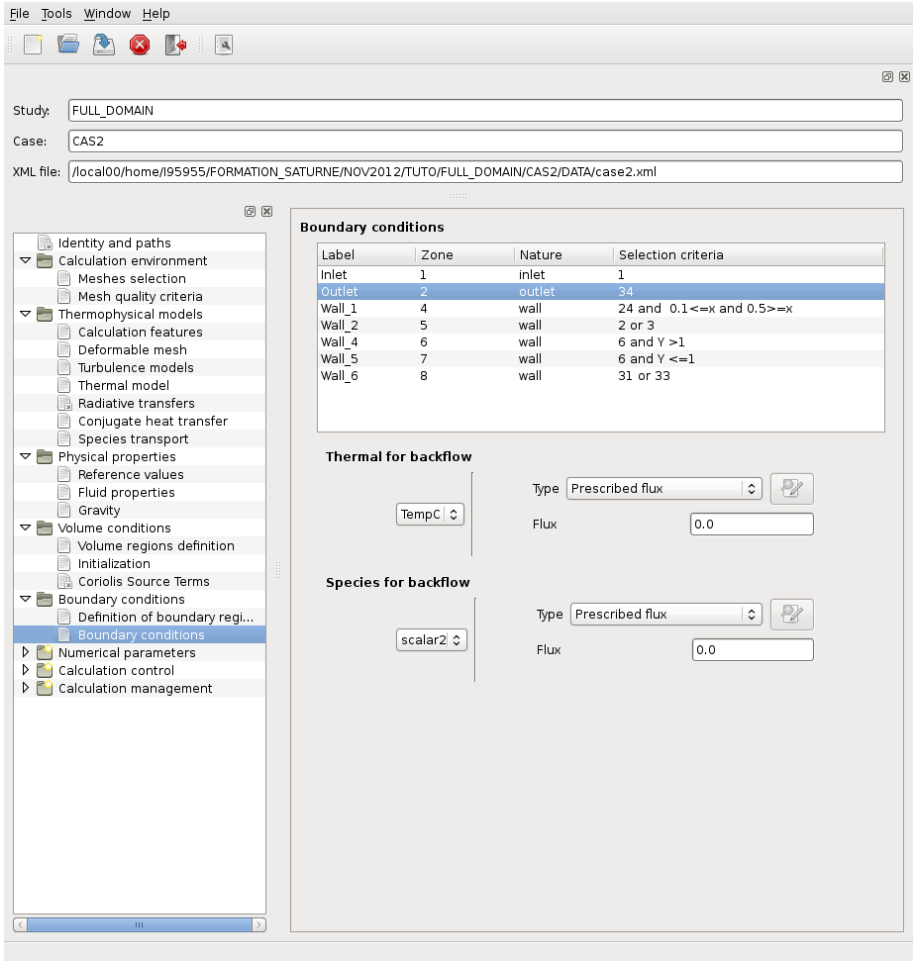


Figure VI.51: Dynamic variables boundary: Outlet

To configure the scalar boundary conditions on the walls, select individually each wall in the *Boundary conditions* item.

On all the walls, a default homogeneous `prescribed flux` is set for temperature, and `prescribed values` are specified for the passive scalar, named `scalar2`, according to the following table:

Wall	Nature	`Scalar2` value
wall_1	Prescribed value	0
wall_2	Prescribed value	5
wall_3	Prescribed value	0
wall_4	Prescribed value	25
wall_5	Prescribed value	320
wall_6	Prescribed value	40

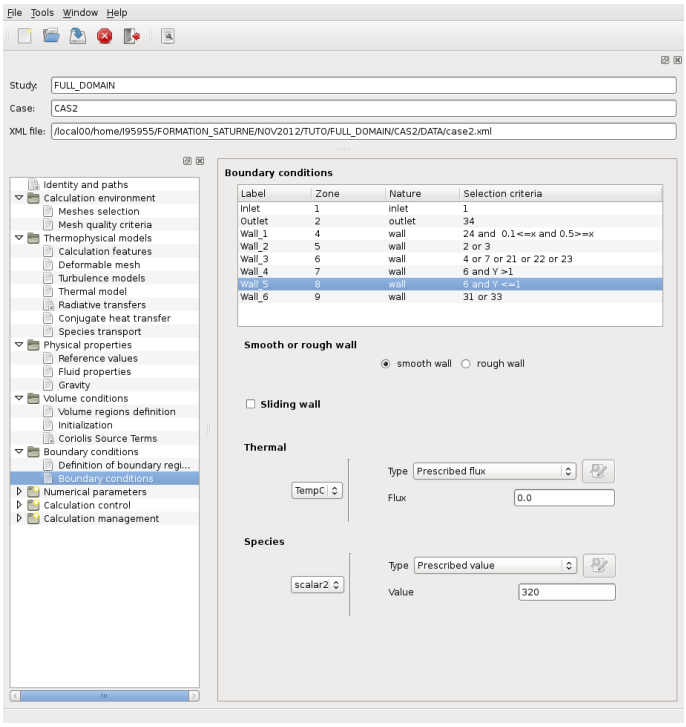


Figure VI.52: Scalars boundaries: wall_5

Some calculation parameters now need to be defined. Go to the *Global parameters* item under the heading *Numerical parameters*. In our case the Pressure-Velocity algorithm is *SIMPLEC*.

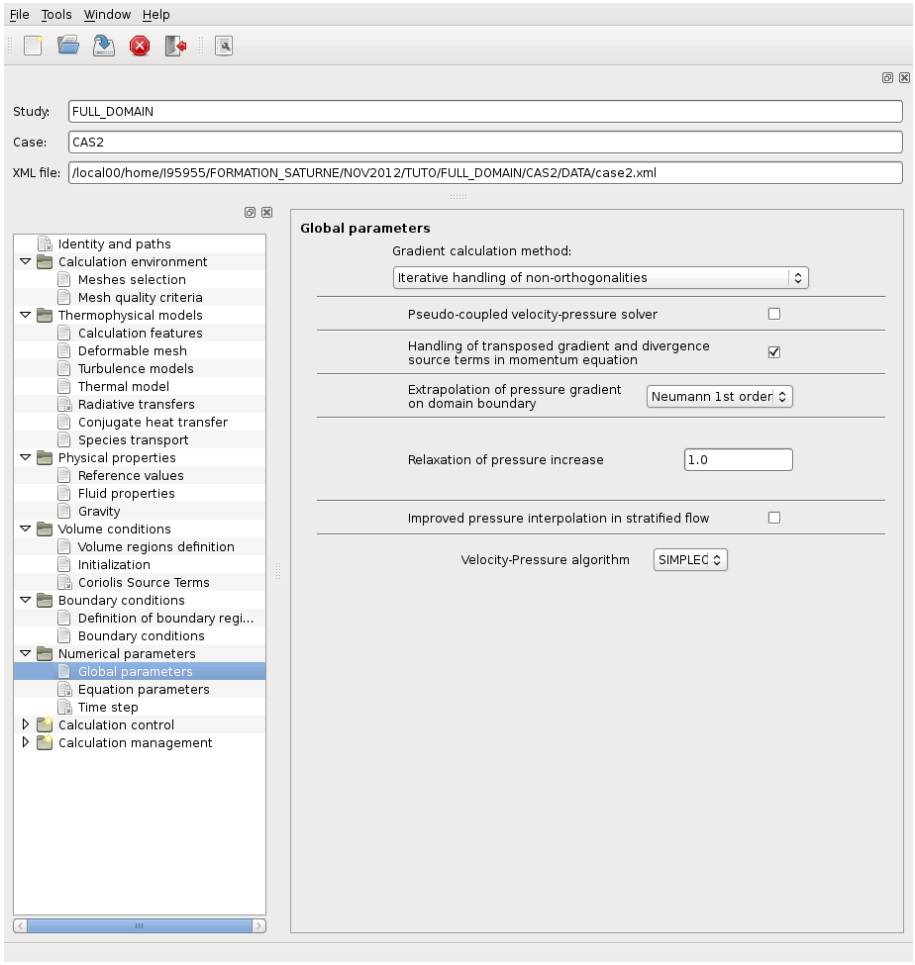


Figure VI.53: Time step setting

Go to the *Equations parameters* item under the heading *Numerical parameters*. You can define the maximum and minimun value for the `TempC` and for the `scalar2` scalars.

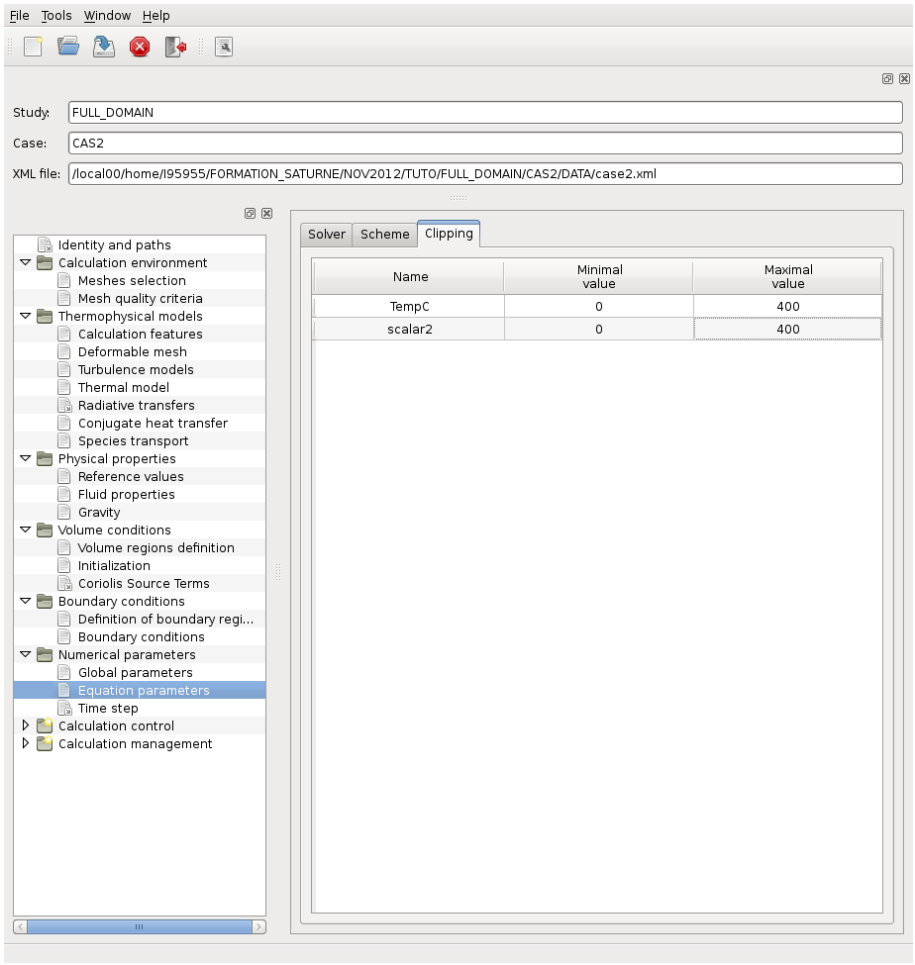


Figure VI.54: Clipping

Go to the *Time step* item under the heading *Numerical parameters*. In our case the time step is *Constant*. Set the number of iterations to **300** and the reference time step to **0.05** (s).

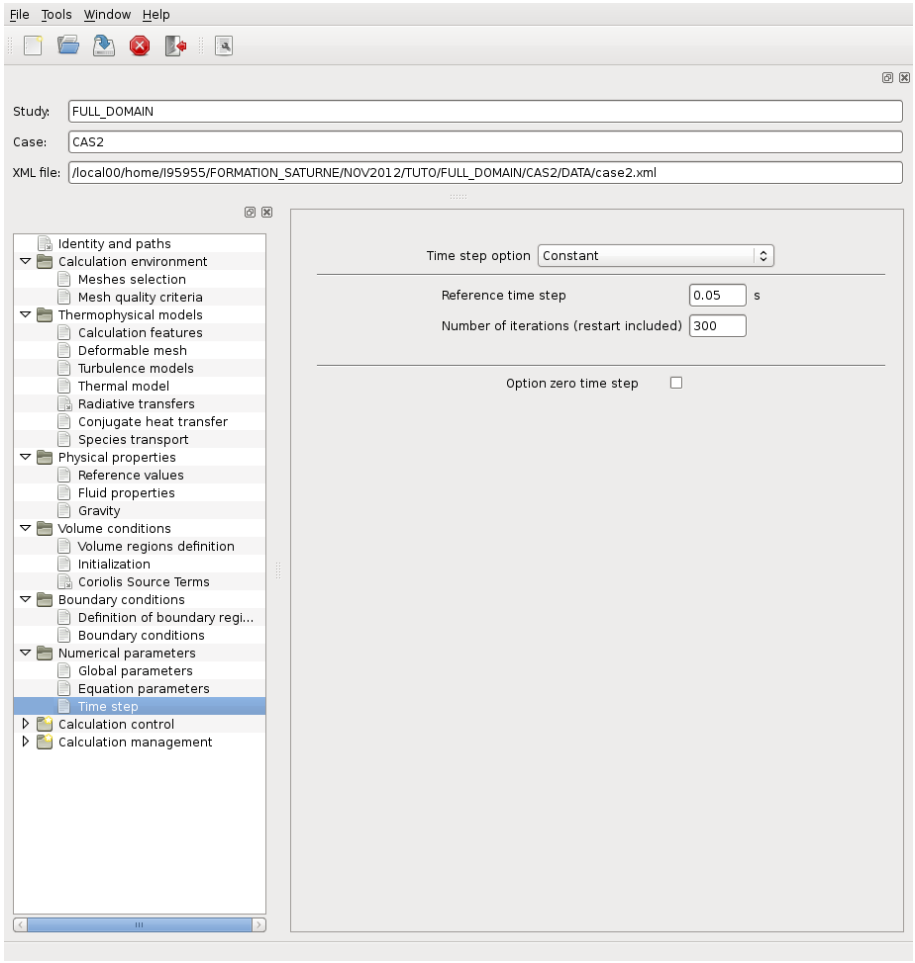


Figure VI.55: Time step setting

Go to the *Output control* item under the heading *Calculation control* to set the output parameters. In the *Output control* item, keep the default value for the output listing frequency.

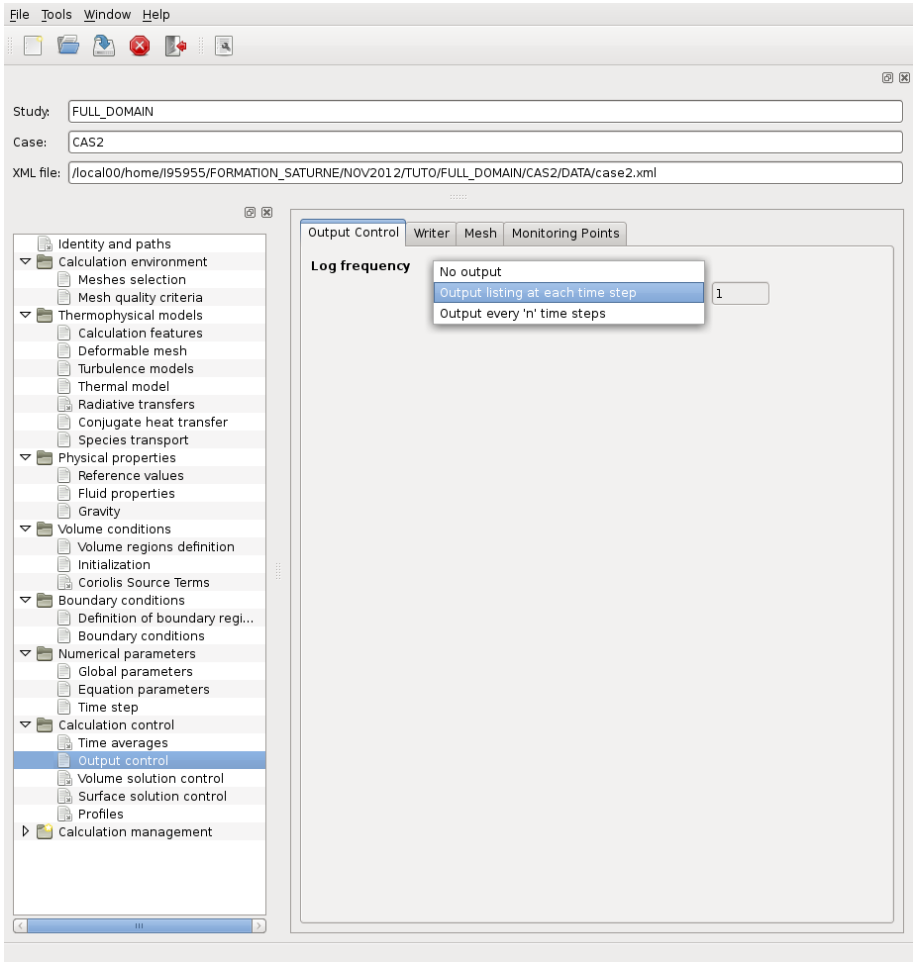


Figure VI.56: Output control: log frequency

For the Post-processing, go to the *Writer* item and click on "results".

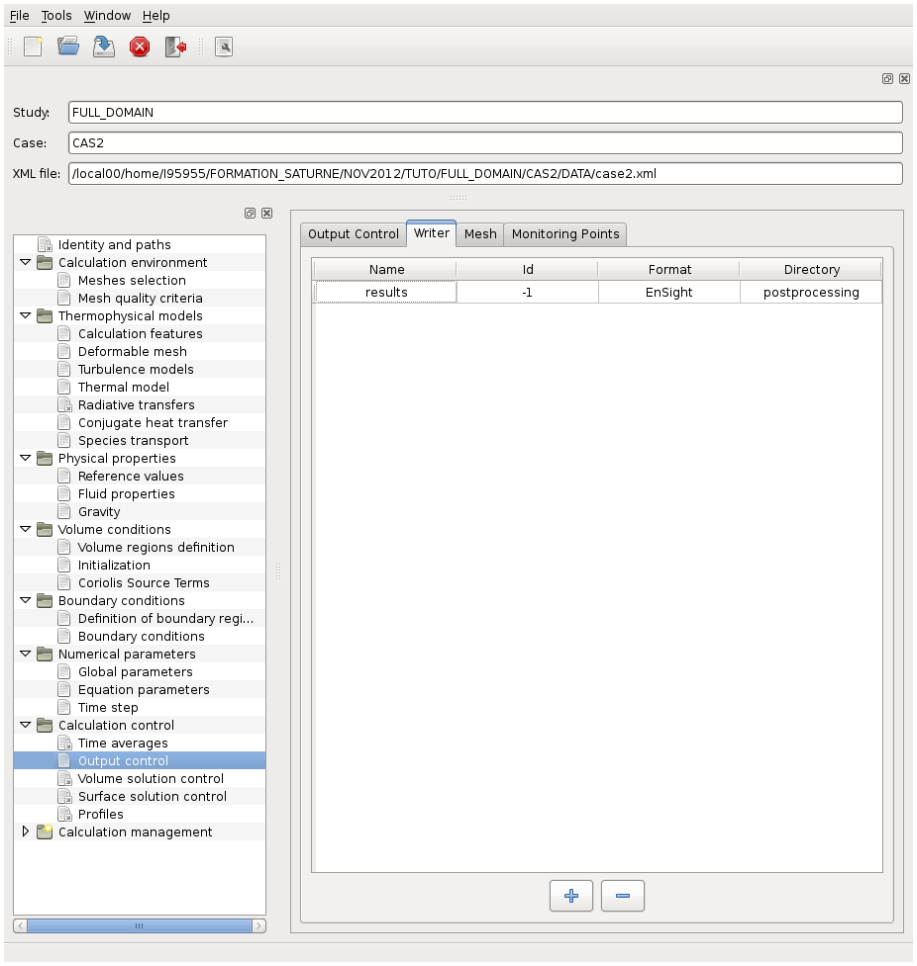


Figure VI.57: Output control: Writer

Now you can select the third option in the *Frequency (output every 'n' time steps)* item and set the value of 'n' to **2**. By default, the boundary faces are selected.

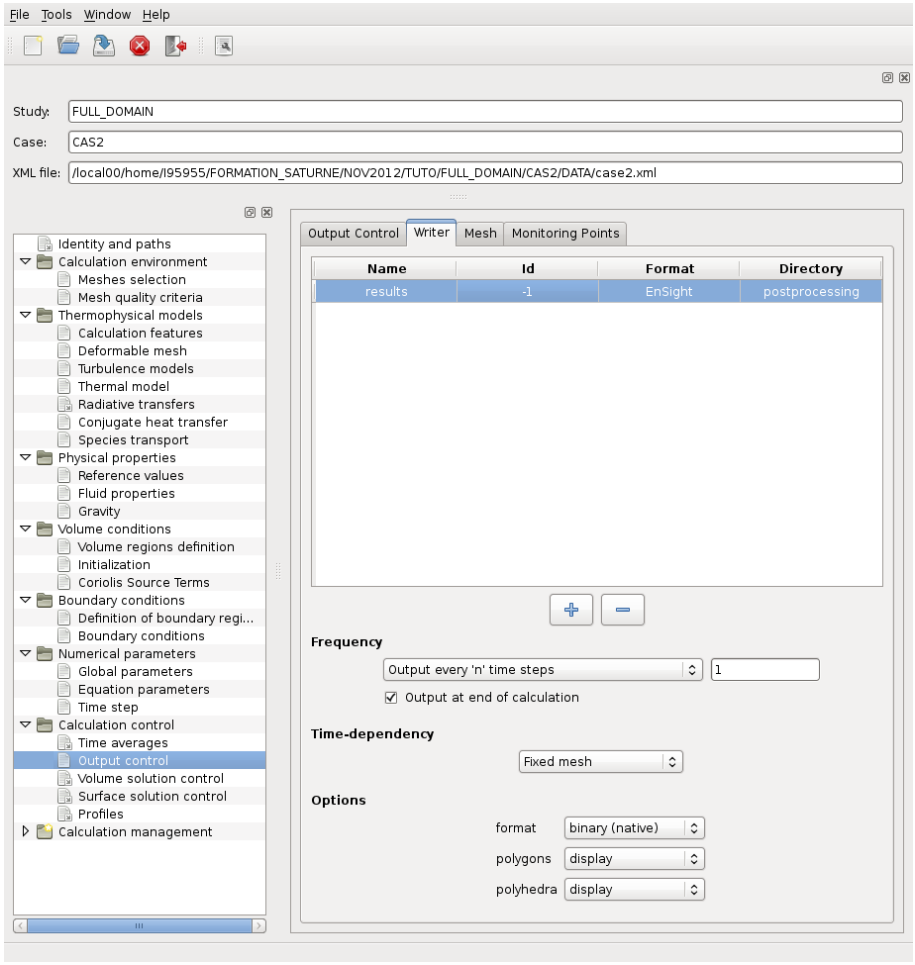


Figure VI.58: Output control: results

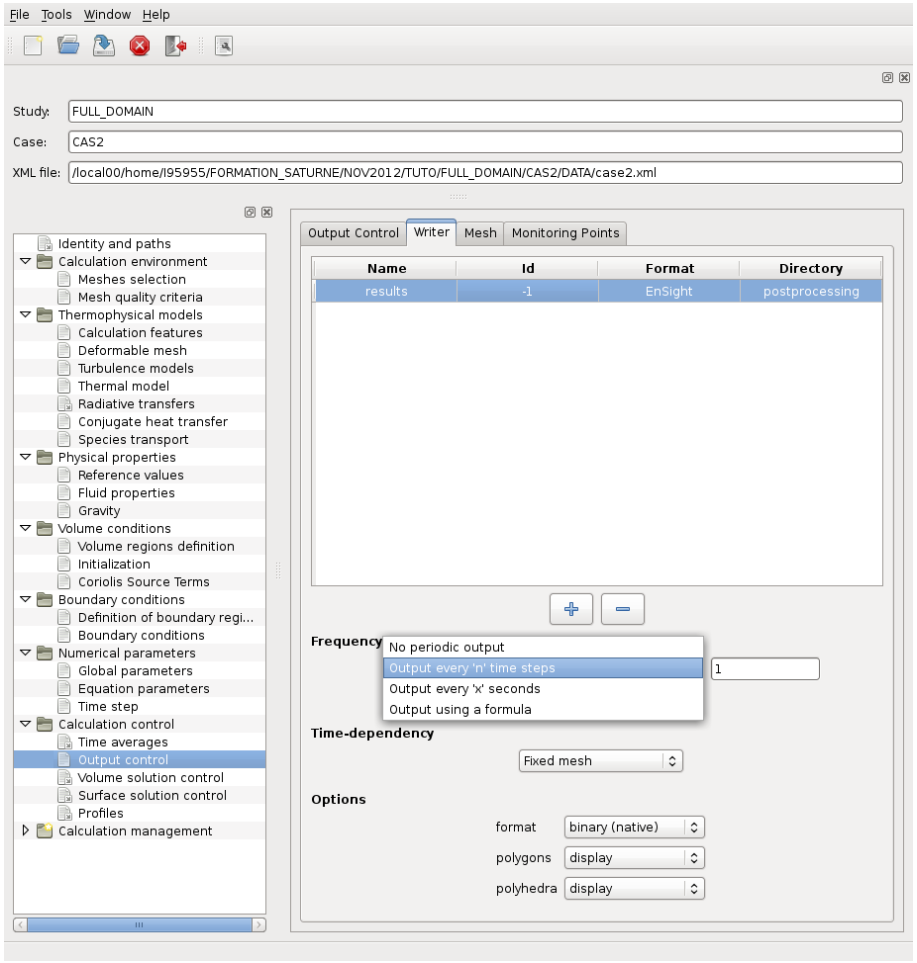


Figure VI.59: Output control: frequency

You can also choose the format. In this case, you will choose the EnSight format.

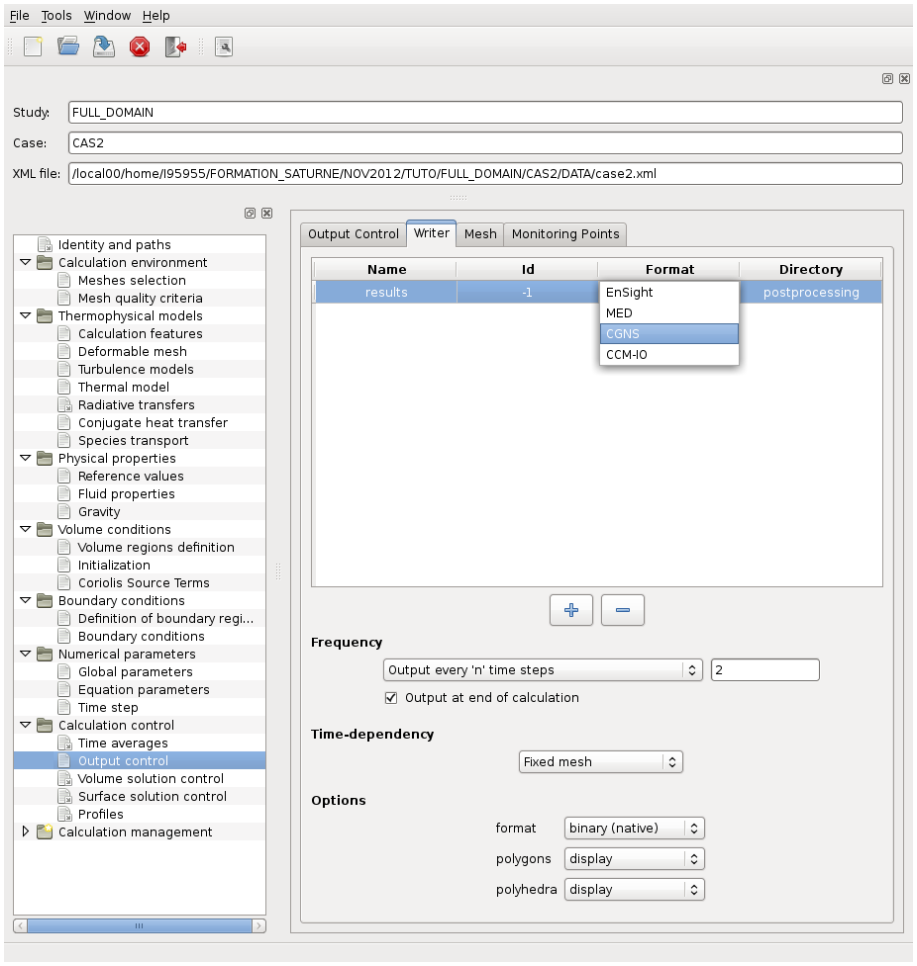


Figure VI.60: Output control: format

Go to the *Mesh* item.

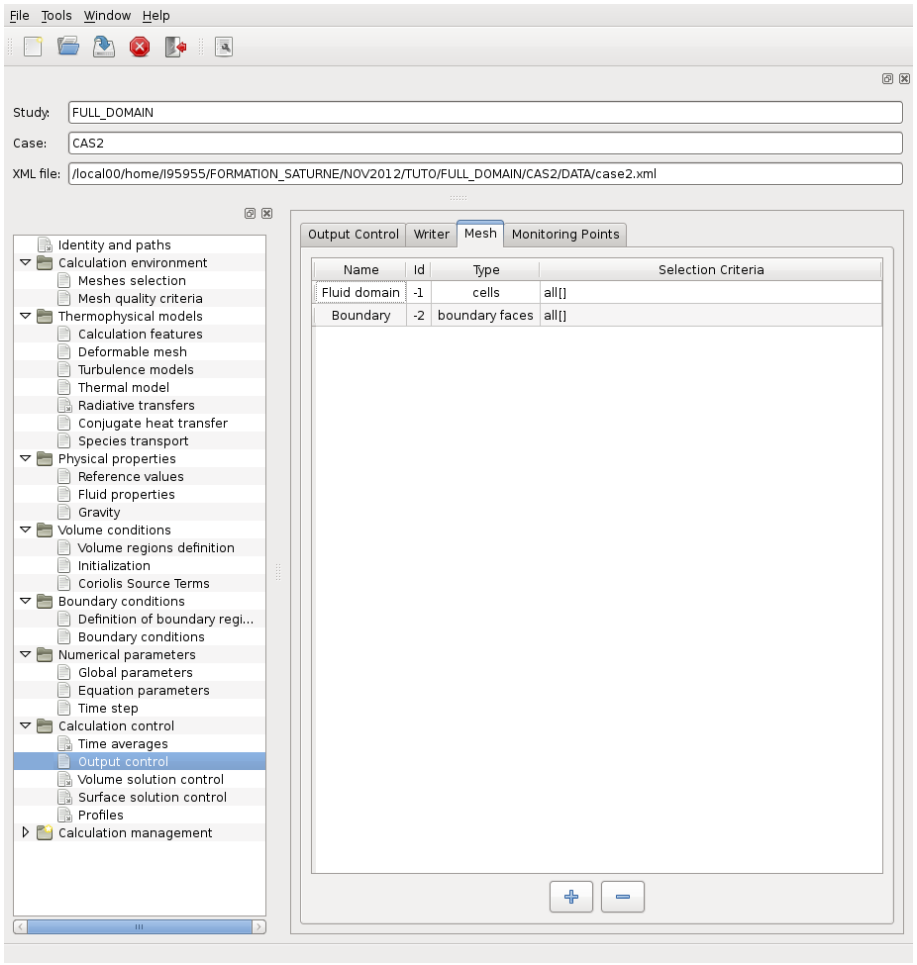


Figure VI.61: Output control: mesh

You can click on the *Fluid domain Mesh Name* item and new options will appear.

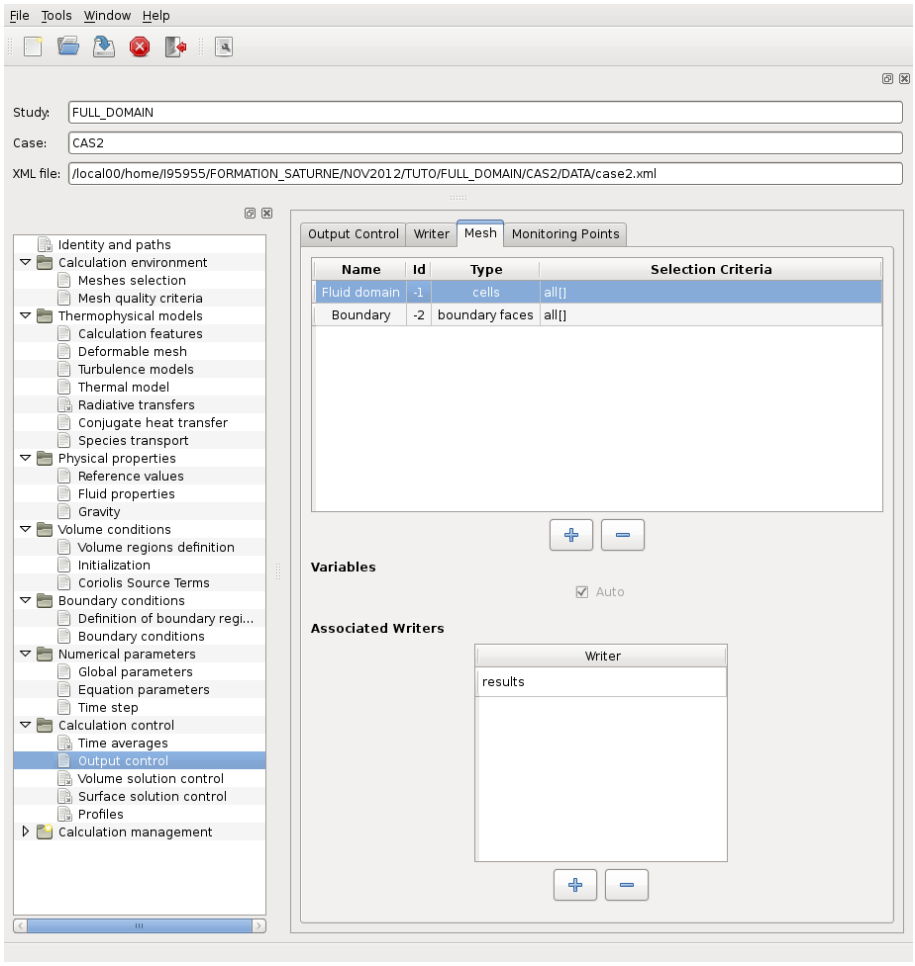


Figure VI.62: Output control: post-processing

You can associated a mesh with several writers.

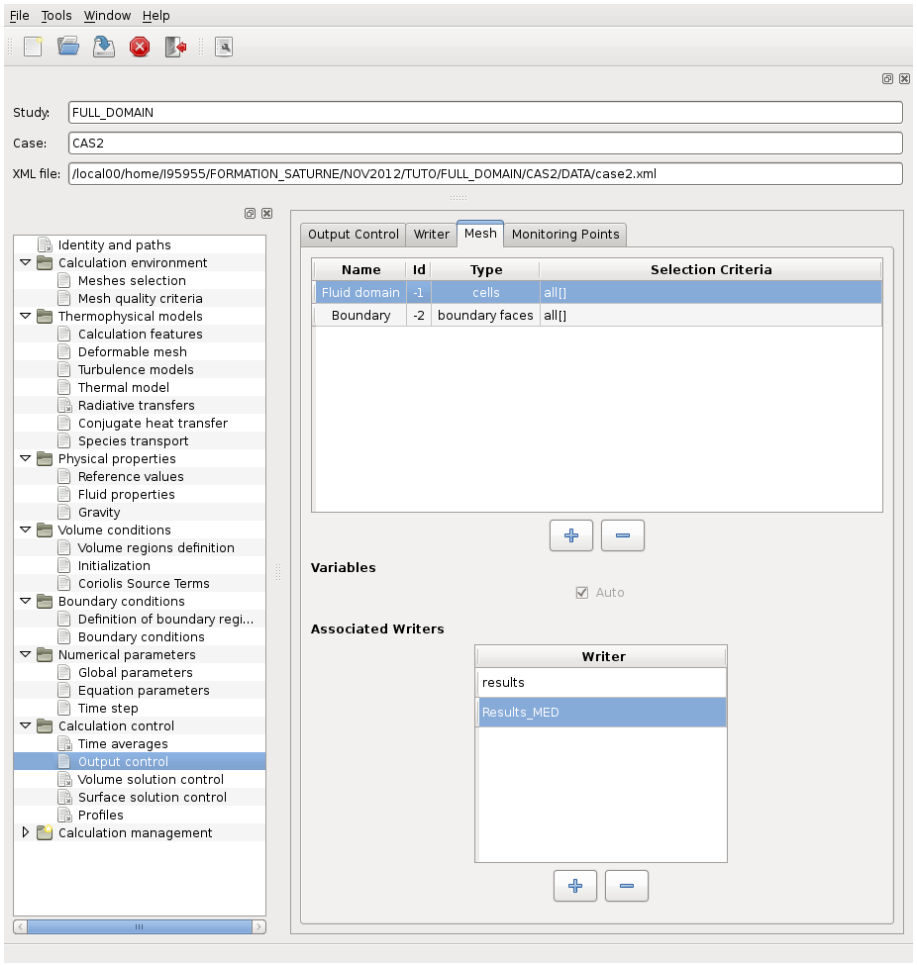


Figure VI.63: Output control:associated writers

In this case, chronological records on specified monitoring probes are needed. To define the probes, click on the *Monitoring points Coordinates* tab.

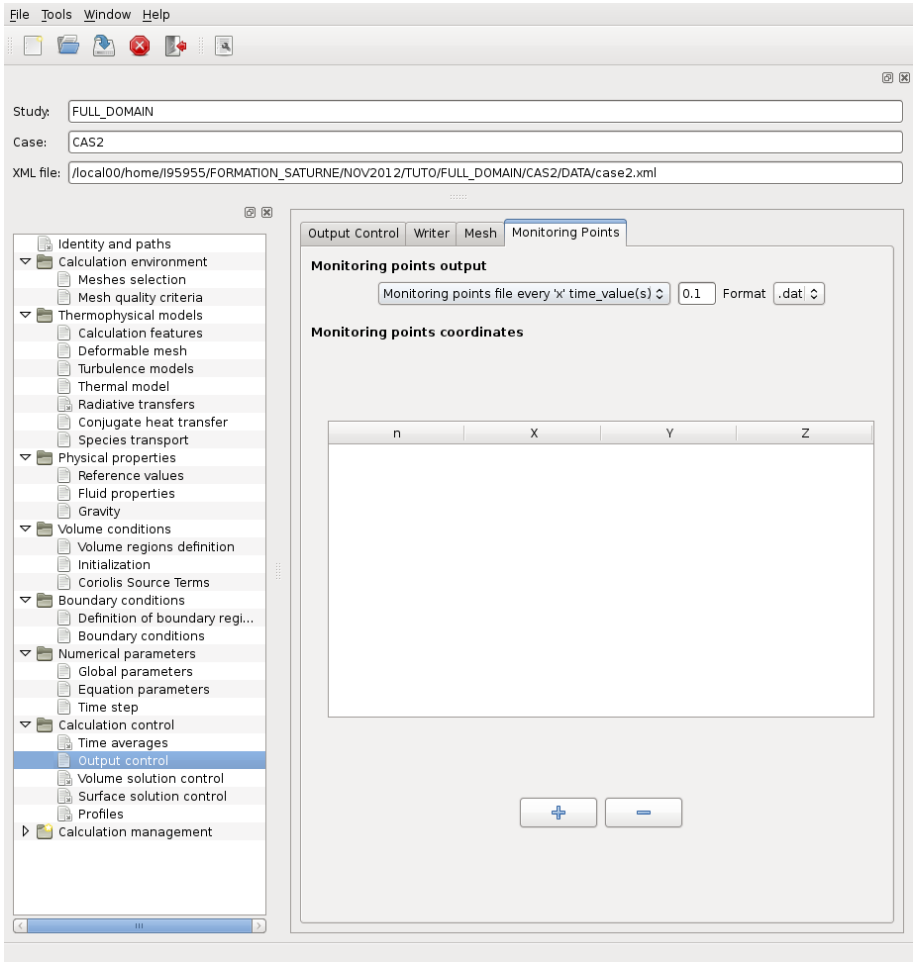


Figure VI.64: Output control: monitoring points

Click on "+" and enter the coordinates of the monitoring points you want to define.

For the first probe:

```
Probe (1) :  x = -0.25 (m) ; y = 2.25 (m) ; z = 0.0 (m)
```

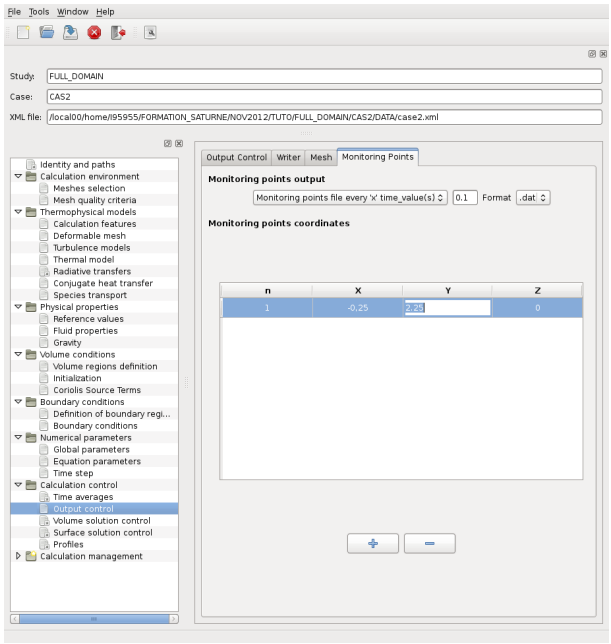


Figure VI.65: Output controls: monitoring points - 1^{st} point

Repeat the procedure for the other probes. Their coordinates are indicated in the following table (the z coordinate is always 0).

Probe n^o.	x (m)	y (m)
2	0.05	2.25
3	0.05	2.75
4	0.05	0.50
5	0.05	-0.25
6	0.75	-0.25
7	0.75	0.25
8	0.75	0.75

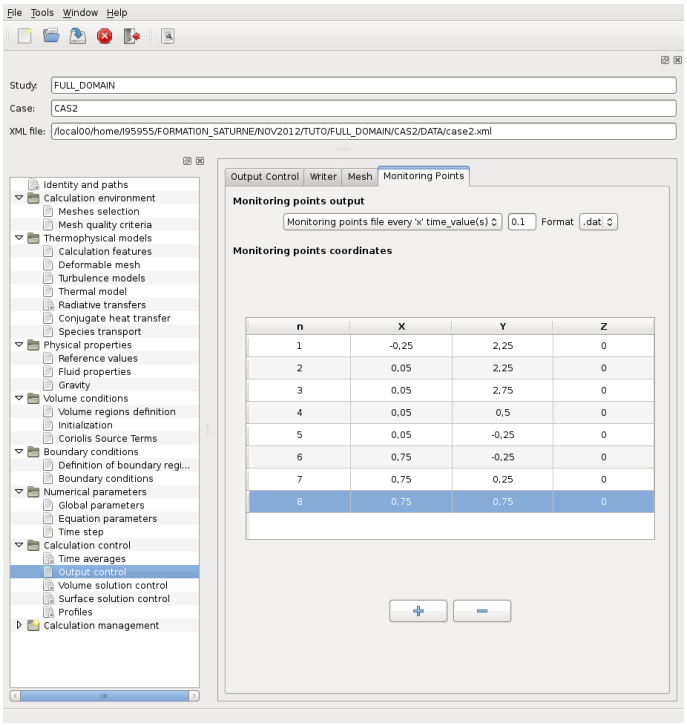


Figure VI.66: Output control: monitoring points

Remember to save the `xml` file regularly.

Go to the *Volume solution control* item to define which variables will appear in the listing, the post-processing and the chronological records.

Uncheck the boxes in front of the *Pressure*, *Tubulent Energy* and *Dissipation* variables, in the *Print in listing* column. Information on these three variables will not appear in the output listing anymore.

Uncheck the boxes in front of the *Courant number* and *Fourier number* variables in the *Post-processing* column. These variables will be removed from the post-processing results.

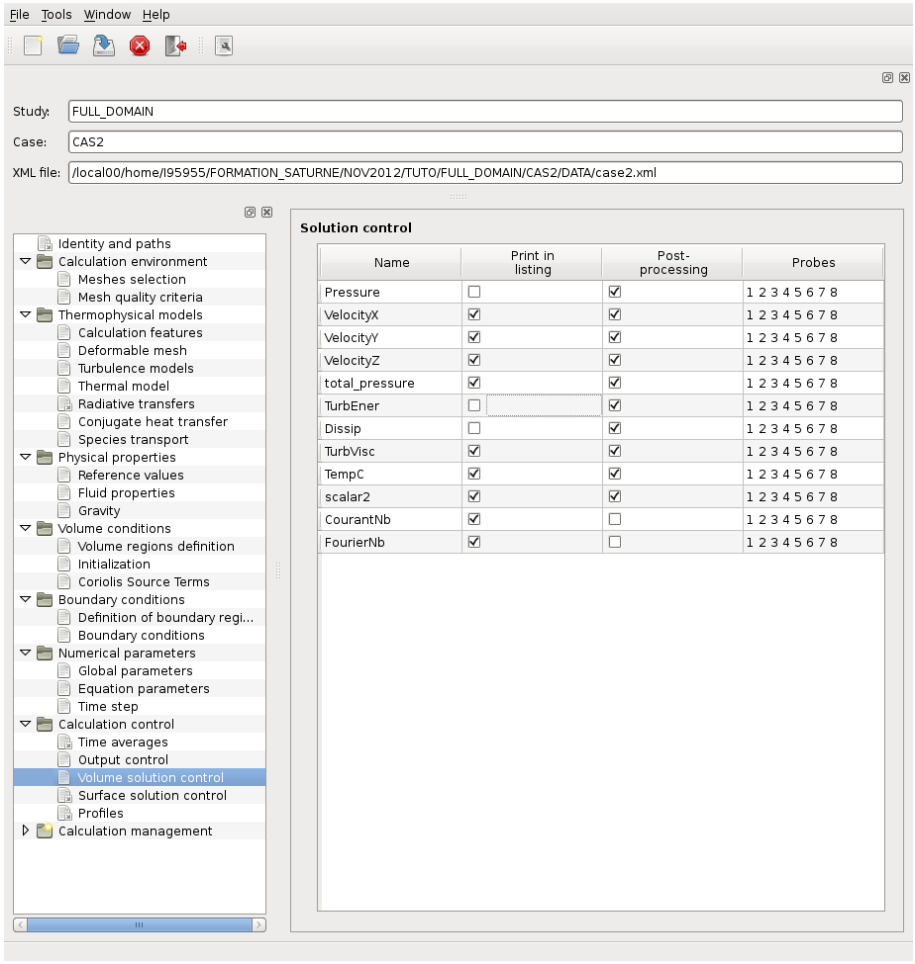


Figure VI.67: Solution control - Output configuration

Delete all the probe numbers for the *total_pressure* variable. No chronological record will be created for this variable.

As for the *VelocityX* variable, only select probes 1, 2, 6, 7 and 8. Time evolution on the other probes will not be recorded.

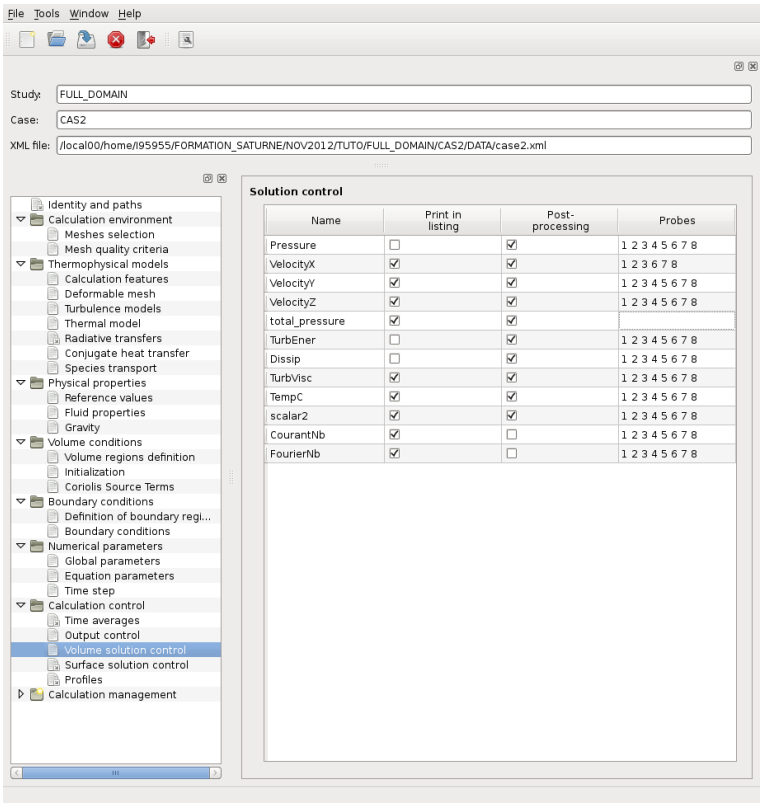


Figure VI.68: Solution control - Probes

Switch to the *Calculation management* heading to prepare the launch script and run the calculation.

3 Solution for case3

Only a few elements are different from `case2`.

In this case the density becomes variable. Go to the *Fluid properties* item under the heading, *Physical properties* and change the nature of the density from `constant` to `user law`.

The user law of the density is defined as following in the *Code_Saturne* (GUI):

```
rho = TempC * ( -4.668E-03*TempC - 5.0754E-02 ) + 1000.9 ;
```

Click on the highlighted icon and define the user law in the window that pops up. Follow the format used in the *Examples* tab.

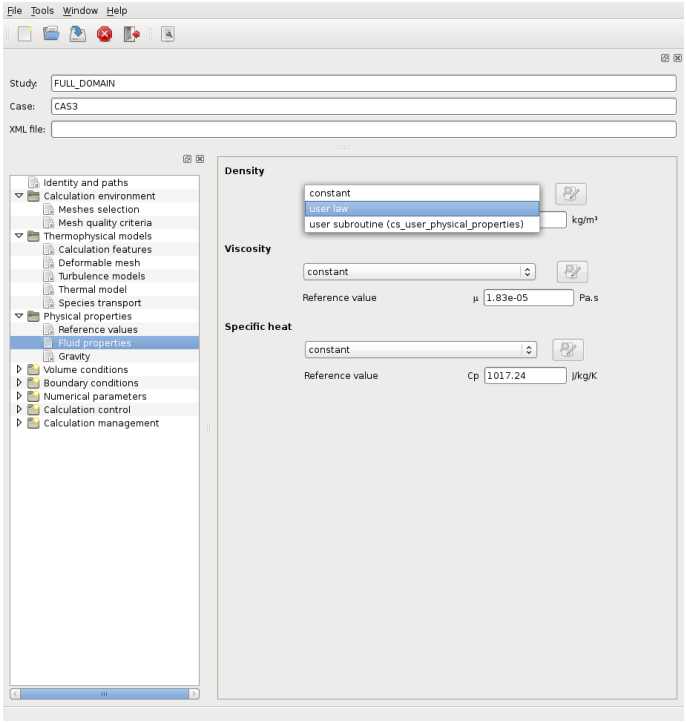


Figure VI.69: Fluid properties - Variable density

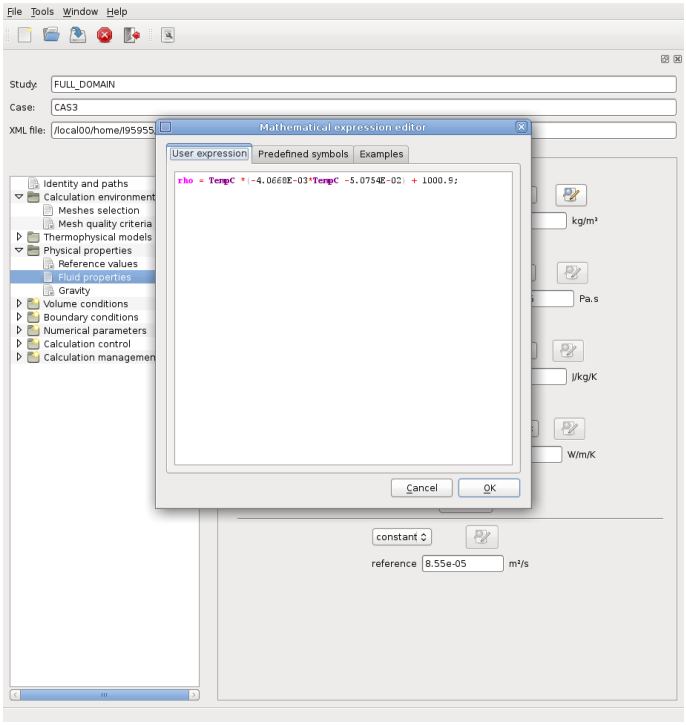


Figure VI.70: Fluid properties - Variable density - User expression

As the density is variable, the influence of gravity has to be considered. In the heading *Physical properties* go to *Gravity* and set the value of each component of the gravity vector.

```
g_x = 0.0 ; g_y = -9.81 ; g_z = 0.0
```

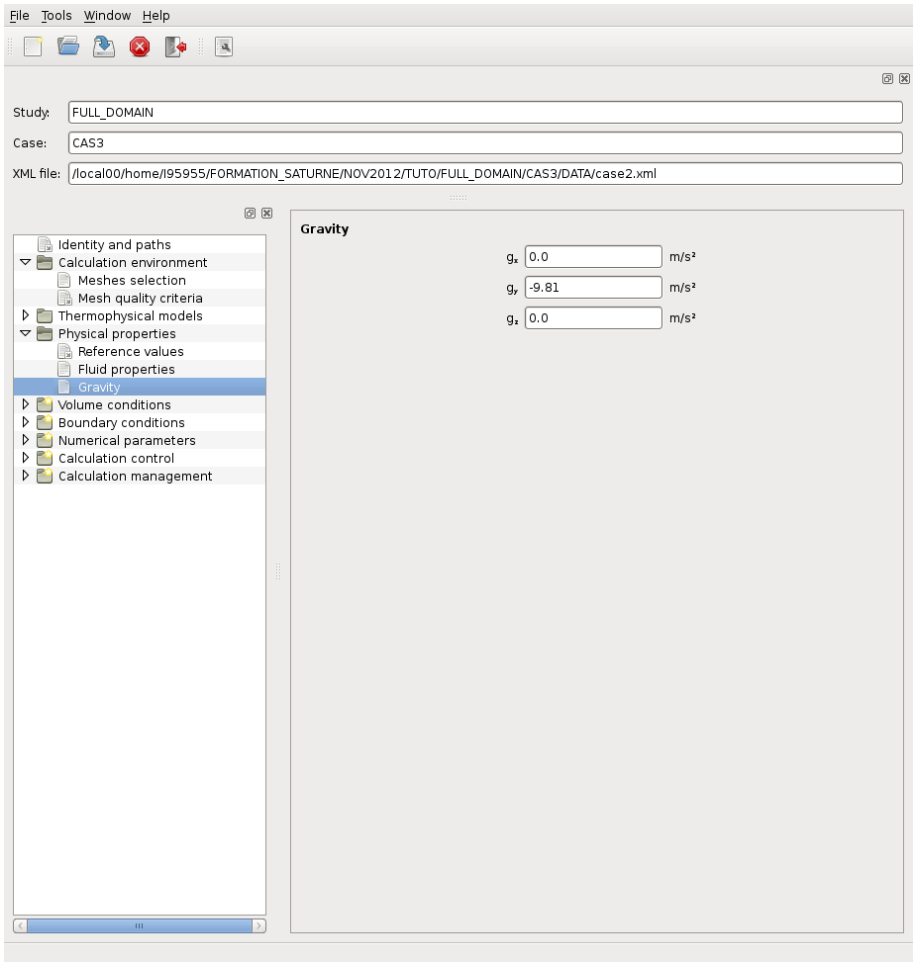


Figure VI.71: Fluid properties - Gravity

Add a monitoring point close to the entry boundary condition in the *Output control* item.

Probe	x (m)	y (m)	z (m)
9	-0.5	2.25	0.0

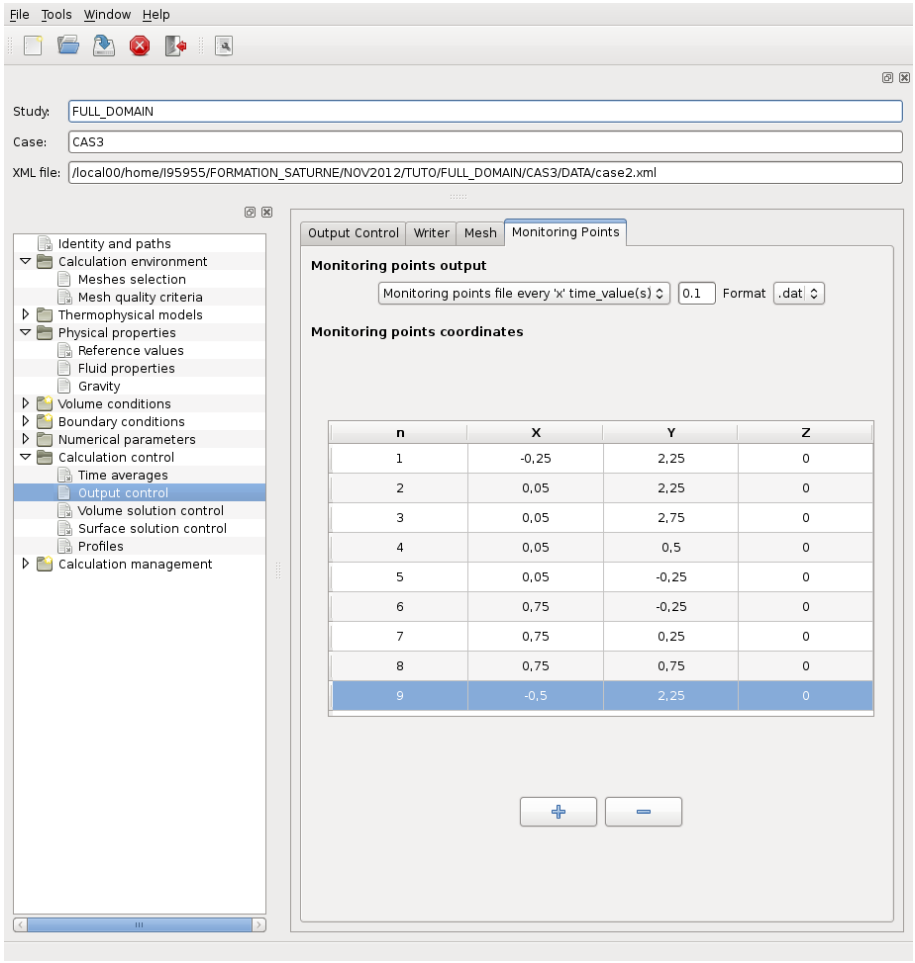


Figure VI.72: New monitoring probe

After completing the interface, before running the calculation, some Fortran user routines need to be modified.

Go to the folder SRC/REFERENCE/base and copy cs_user_boundary_conditions.f90 in the SRC directory.

• `cs_user_boundary_conditions.f90`:
In this case, cs_user_boundary_conditions.f90 is used to specify the time dependent boundary condition for the temperature. Refer to the comments in the routine or to the *Code_Saturne* user manual for more information on this routine.

In our case, you need to identify the boundary faces of color **'1'**.
The command `call getfbr('1',nlelt,lstelt)` will return an integer `nlelt`, corresponding to the number of boundary faces of color 1, and an integer array `lstelt` containing the list of the `nlelt` boundary faces of color 1.

• **Remark**: Note that the string '1' **can be more complex and combine different colors, group references or geometrical criteria**, with the same syntax as in the Graphical Interface.

For each boundary face `ifac` in the list, the Dirichlet value is given in the multi-dimension array `rcodcl` as follows:

```
if (ttcabs.lt.3.8d0) then
do ielt = 1, nlelt
ifac = lstelt(ielt)
rcodcl(ifac,isca(1),1) = 20.d0 + 100.d0*ttcabs
enddo
else
do ielt = 1, nlelt
ifac = lstelt(ielt)
rcodcl(ifac,isca(1),1) = 400.d0
enddo
endif
```

`isca(1)` refers to the first scalar and ttcabs is the current physical time.

See the example `cs_user_boundary_conditions-base.f90` file in the subdirectory `SRC/EXAMPLES` to complet correctly your boundary conditions for this `case3`.

• Remark: Note that, although the inlet boundary conditions for temperature are specified in the `cs_user_boundary_conditions.f90` file, it is necessary to specify them also in the Graphical Interface.

The value given in the Interface can be anything, it will be overwritten by the Fortran routine.

After updating the Fortran file, run the calculation as explained in `case2`.

When a calculation is finished, *Code_Saturne* stores all the necessary elements to continue the computation in another execution, with total continuity. These elements are stored in several files, grouped in a `yyyymmdd-hhmm/checkpoint` subdirectory, in the `RESU` directory.

In this case, after the first calculation is finished, a second calculation will be run, starting from the results of the first one.

Go directly on the *Start/Restart* item under the heading *Calculation management*. Activate the *Calculation restart* by clicking the **"on"** box.

Then click on the folder icon next to it to specify the restart files to use.

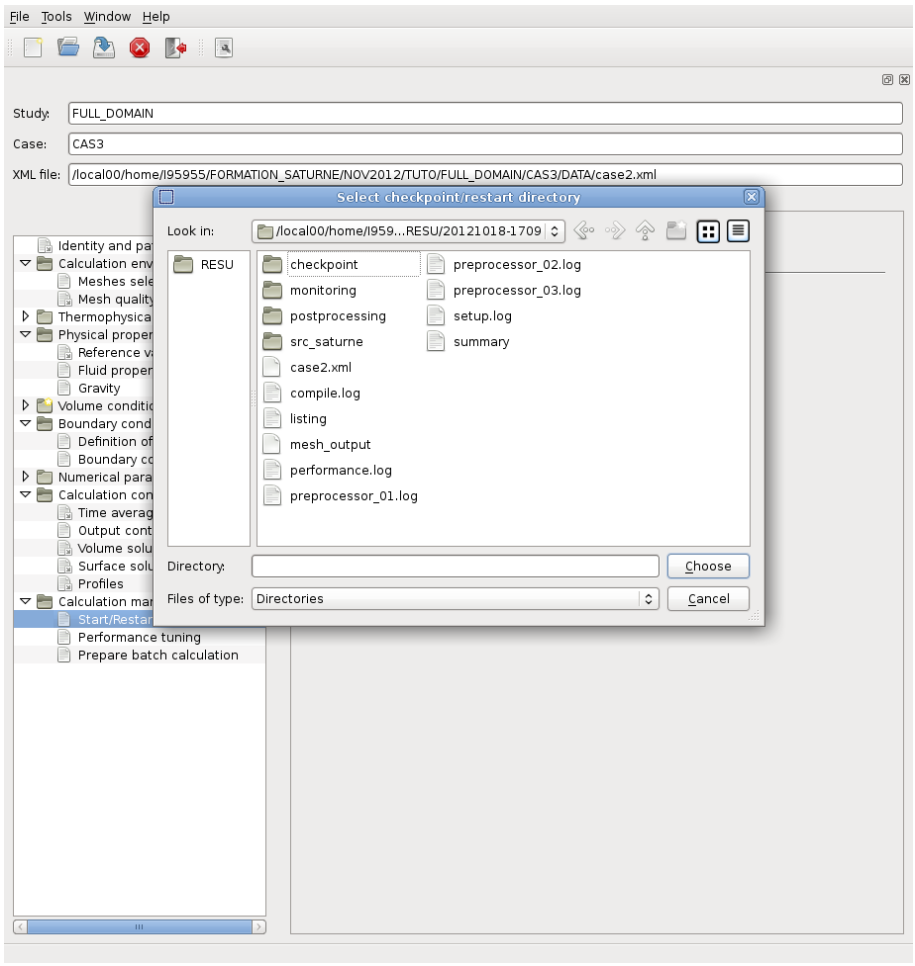


Figure VI.73: Start / Restart

A window opens, with the architecture of the study sub-directories. Open the `RESU` folder and click on the folder `yyyymmdd-hhmm/checkpoint` (where `yyyymmdd-hhmm` corresponds to the reference of the first calculation results). Then click on *Validate*.

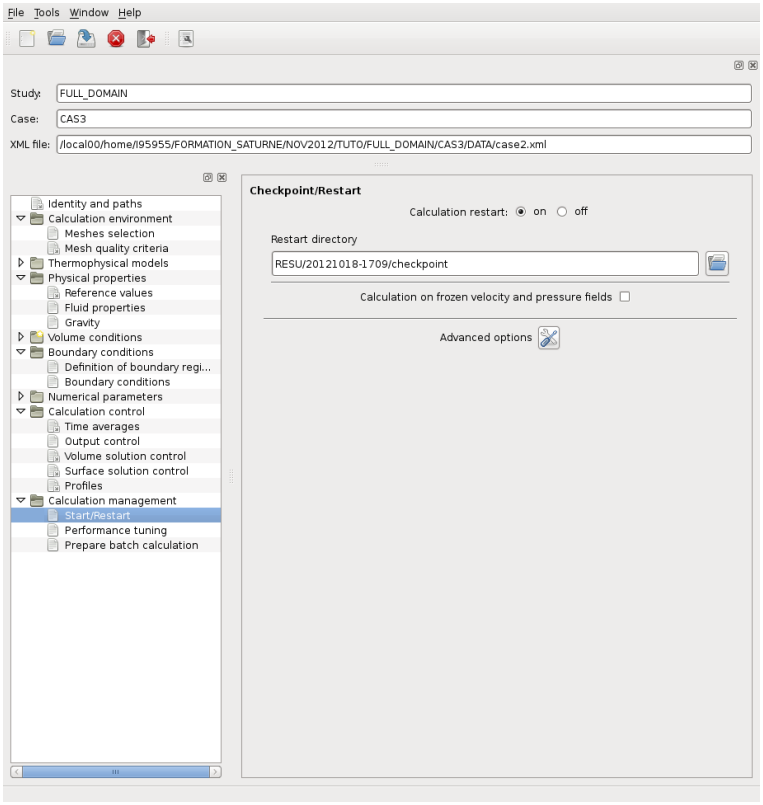


Figure VI.74: Start / Restart - Selection of the restart directory

Go to the *Time step* item under the heading *Numerical parameters* and change the number of iterations. It must be the total number of iterations, from the beginning of the first calculation.

The first calculation was done with 300 iterations and another 400 iterations are needed for the present case. Therefore the value 700 must be entered.

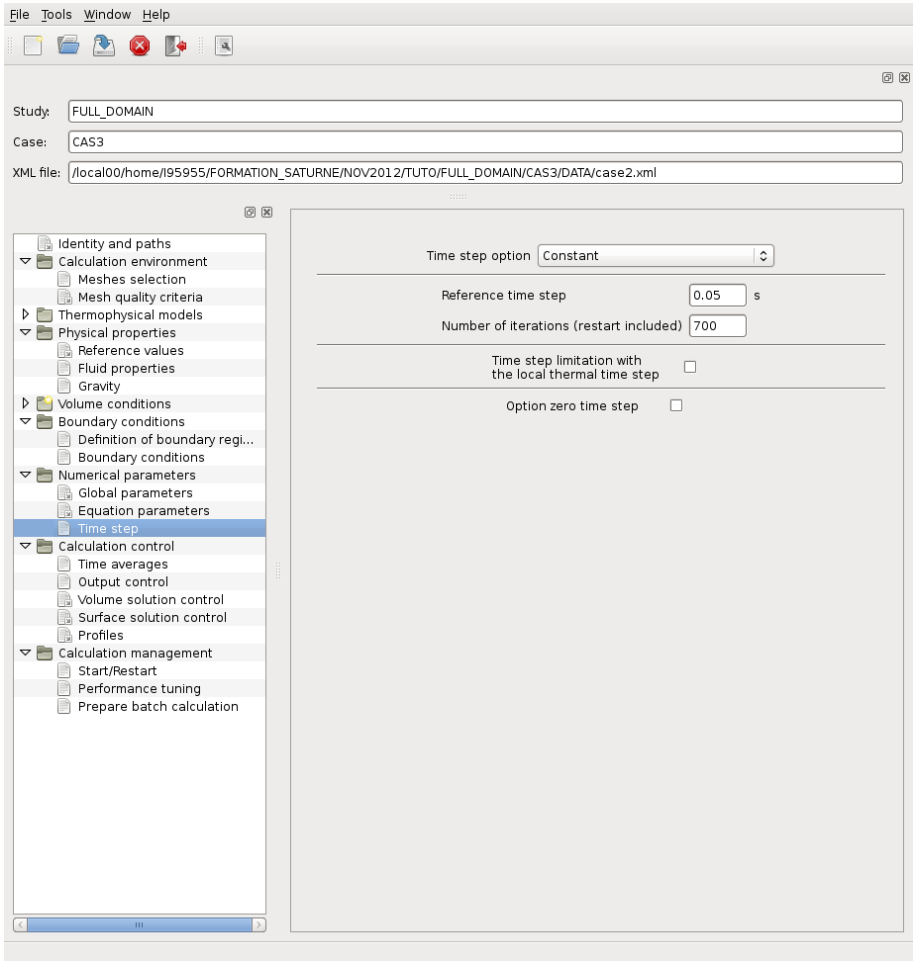


Figure VI.75: Time step

Eventually, run the calculation.

4 Solution for case4

This case is similar to `case3`, with the following differences:

- **Step 1**: define head losses in the fluid domain,
- **Step 2**: compute the spatial average of temperature scalar,
- **Step 3**: parallel computation on 2 processors,
- **Step 4**: dealing with a user results file.

• **Step 1-1**: Define the head losses in the Graphical User Interface (GUI).
Go to *Volume regions definition* under the heading *Volume conditions*. Click on **"Add"**, unselect "Initialization" and select **"Head losses"** in the box named *Nature*. In the box named *Label*, name the head losses region.
Define the limits of the head losses region in *Selection criteria*. The associated character string to enter is as below:

```
''0.2 <= x and 0.4 >= x and -0.75 <= y and -0.25 >= y''
```

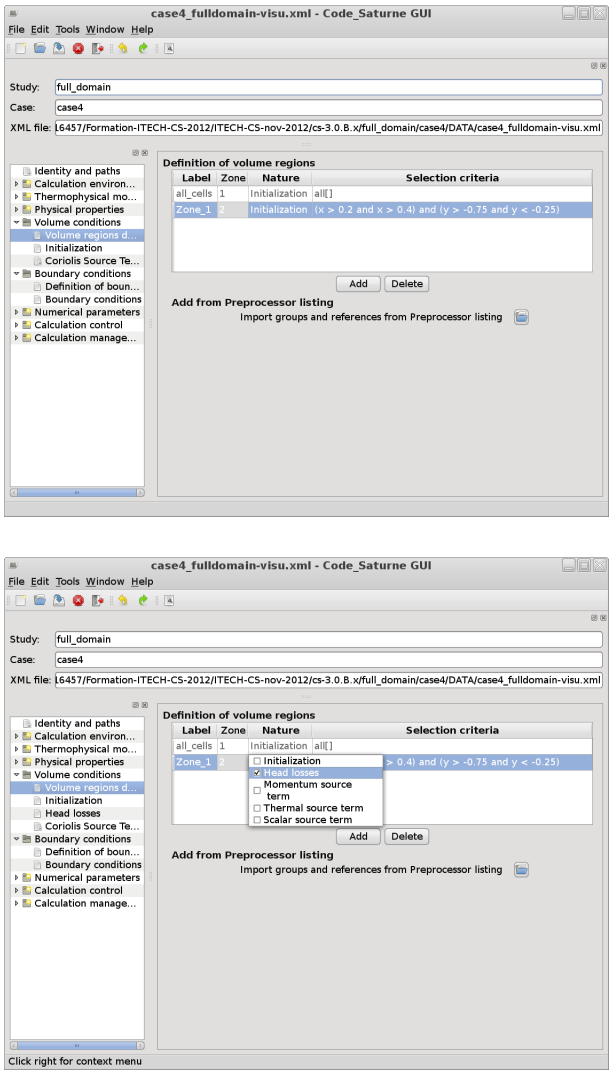


Figure VI.76: Creation of head losses region

• **Step 1-2**: Specify the head losses coefficients α_{ii}.
To specify the head losses coefficients go to the *Head losses* item and select the name of the head losses volume region. In this example, the coefficient is isotropic so that we use the same value for each α_{ii}. Please note that $\alpha_{ii} = 2 \times K_{ii}$, therefore if $K_{ii} = 10^4$, $\alpha_{ii} = 2\ 10^4$.

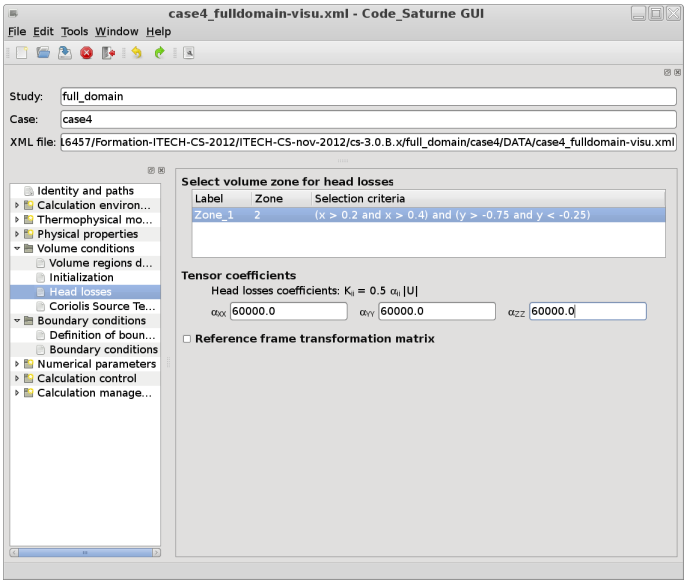


Figure VI.77: Head losses coefficients

• **Step 2**: compute the spatial average of temperature **TempC**
The computation of the spatial average is done in the `cs_user_extra_operations.f90` routine.

• Remark: Refer to the example files in the subdirectory `SRC/EXAMPLES` which names are:

```
cs_user_extra_operations-energy_balance.f90
cs_user_extra_operations-extract_1d_profile.f90
cs_user_extra_operations-force_temperature.f90
cs_user_extra_operations-global_efforts.f90
cs_user_extra_operations-parallel_operations.f90
cs_user_extra_operations-print_statistical_moment.f90
```

To correctly complet your `cs_user_extra_operations.f90` routine (copied in the `SRC` directory), you can use mainly these examples files: `cs_user_extra_operations-print_statistical_moment.f90` and `cs_user_extra_operations-extract_1d_profile.f90`.

• **Step 3**: choose a computation with 2 processors
This modification will be done in the *Prepare batch analysis* item.
To run the calculation on two processors, simply change the number of processors indicator to 2. The launch script will automatically deal with the rest.

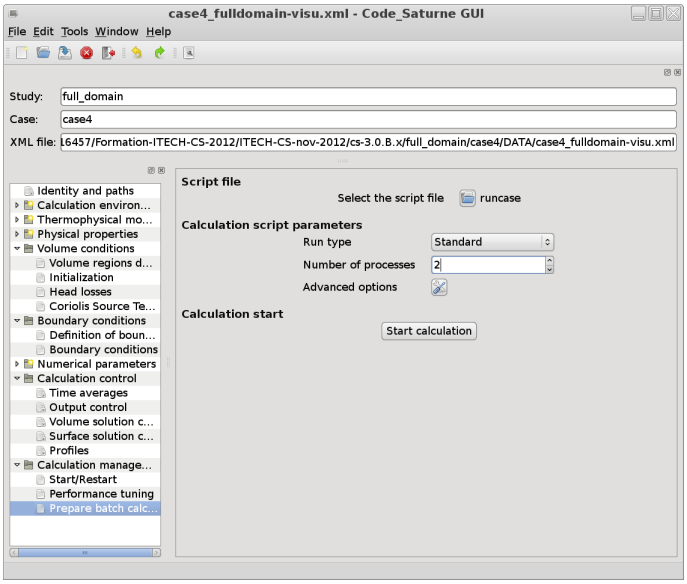


Figure VI.78: Number of processors

• **Step 4**: dealing with a user results file **"moy.dat"**
The new user file "moy.dat" created by `cs_user_extra_operations.f90` will be written directly in the `yyyymmdd-hhmm` results subdirectory created at the end of the computation in the `RESU` directory.

• Remark : We do not have to specify the name of the new user file in the Graphical User Interface (GUI), like in previous *Code_Saturne* versions.
The name of the new user file had to be identified in the launch script in order to be automatically copied in the `RESU` directory, this is not requested.

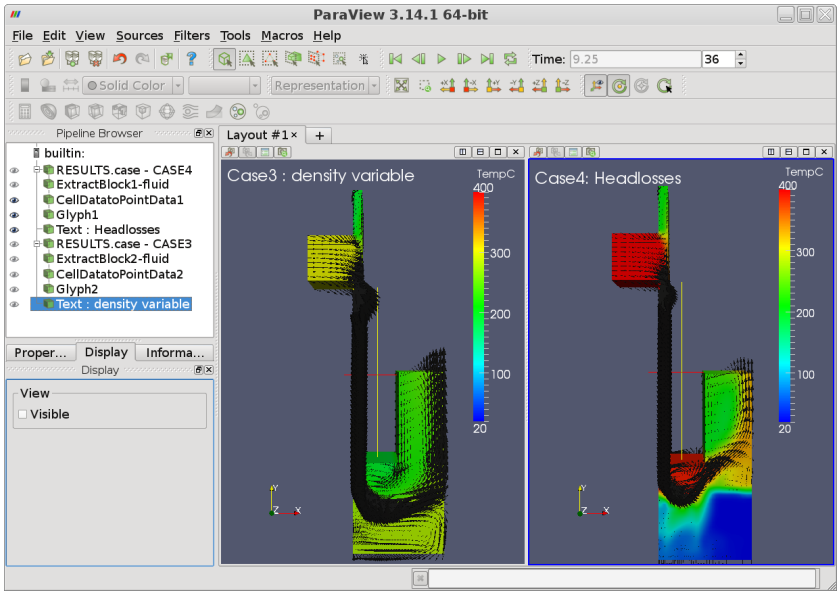


Figure VI.79: User results files

5 Solution for case5

The first thing to do before running *Code_Saturne* is to prepare the computation directories. In this first example, the study directory "T_junction" will be created, containing a single calculation directory `case5`. This is done by typing the command:

```
$ code_saturne create -s T_junction -c case5
```

- Open the *Code_Saturne* interface
- Open a new case
- Check the name of the mesh
- Select a k-ε model
- Use a thermal scalar in Celsius degrees

In the item *Initialization* under the heading *Volume conditions*, set the initial value of the temperature in the domain to 38.5°C. Initialize the turbulence with the reference velocity 0.03183 $m.s^{-1}$.

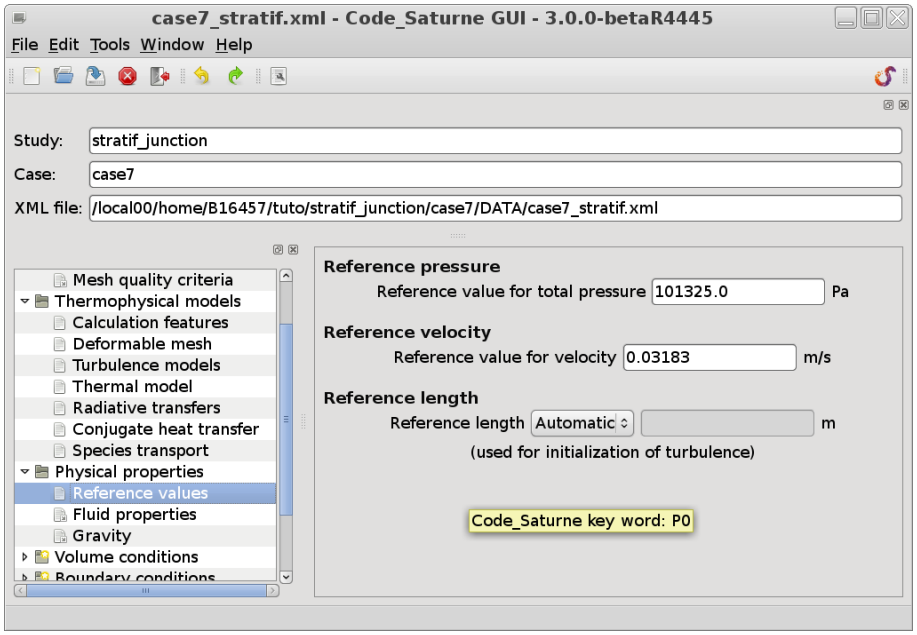


Figure VI.80: Thermophysical models - Initialization

In the item *Fluid properties*, under the heading *Physical properties*, enter the following information:

Variable	Type	Value
Density	user law	998.671 $kg.m^{-3}$
Viscosity	user law	0.445×10^{-4} $kg.m^{-1}.s^{-1}$
Specific Heat	Constant	4 182.88 $J.kg^{-1}.°C^{-1}$
Thermal Conductivity	Constant	0.601498 $W.m^{-1}.K^{-1}$

For density and viscosity, the value given here will serve as a reference value (see user manual for details).

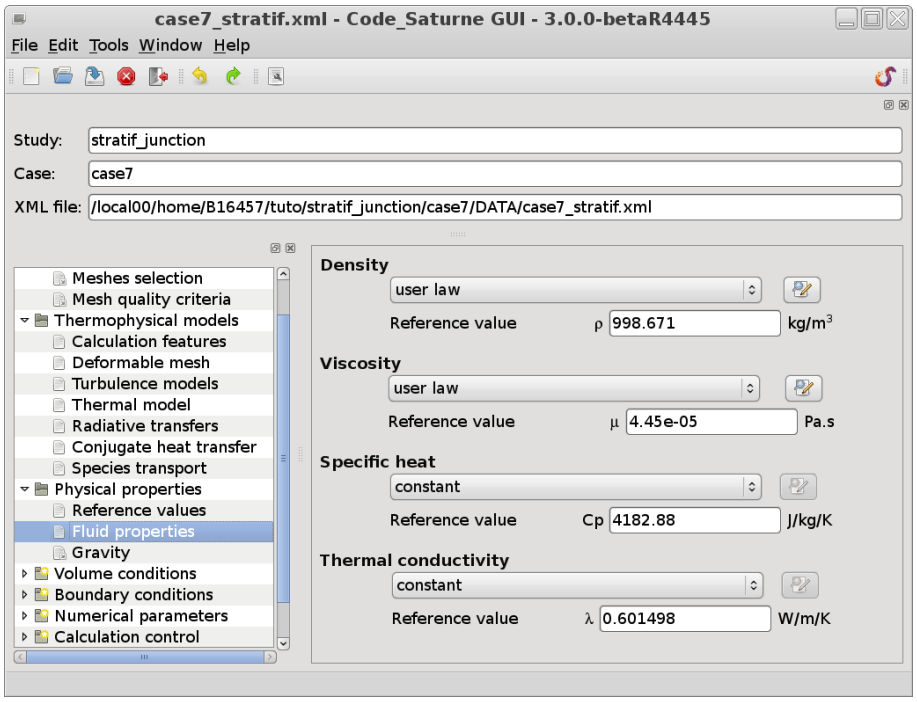


Figure VI.81: Physical properties: fluid properties

For the density and viscosity, enter the expressions of the user laws as showed in figures VI.82 and VI.83, in the windows poping while clicking on the highlighted boxes.

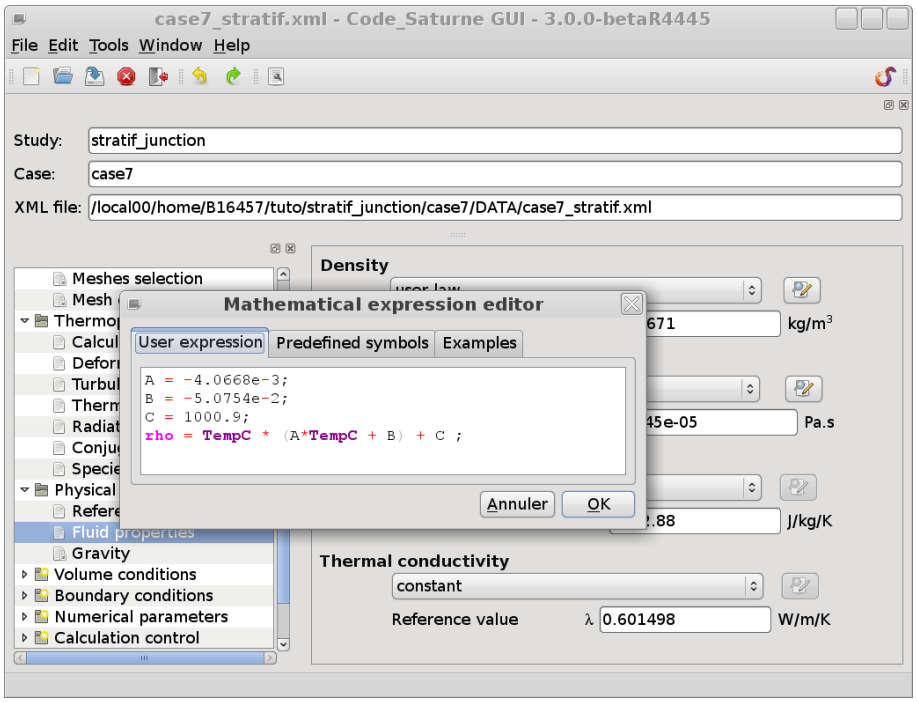


Figure VI.82: Variable density

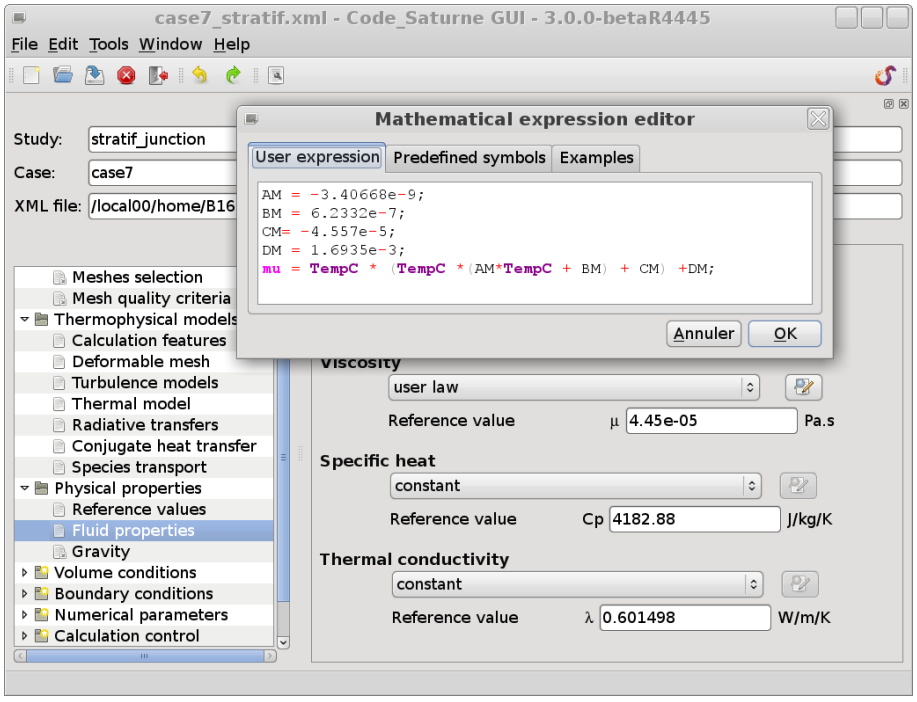


Figure VI.83: Variable viscosity

The aim of the calculation is to simulate a stratified flow. It is therefore necessary to have gravity. Set it to the right value in the item *Gravity, hydrostatic pressure*. In order to have a sharper stratification, the pressure interpolation method will be set to *improved*.

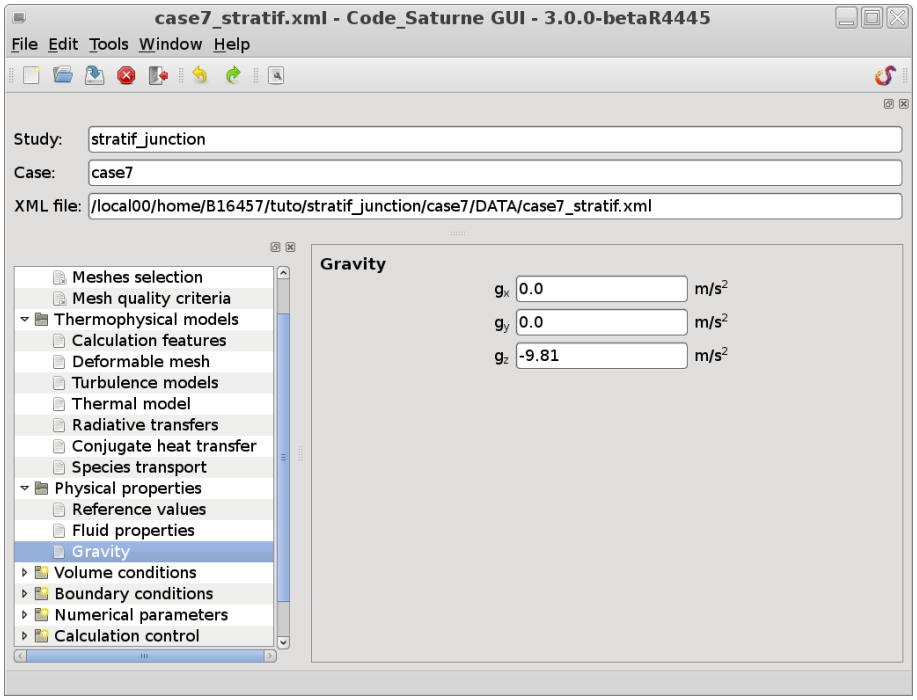


Figure VI.84: Fluid properties - Gravity

Go to the item *Definition and initialization* under the heading *Additional scalars* to specify the minimal and maximal values for the temperature: 18.26°C and 38.5°C. Note that the initial value of 38.5°C set earlier is properly taken into account.

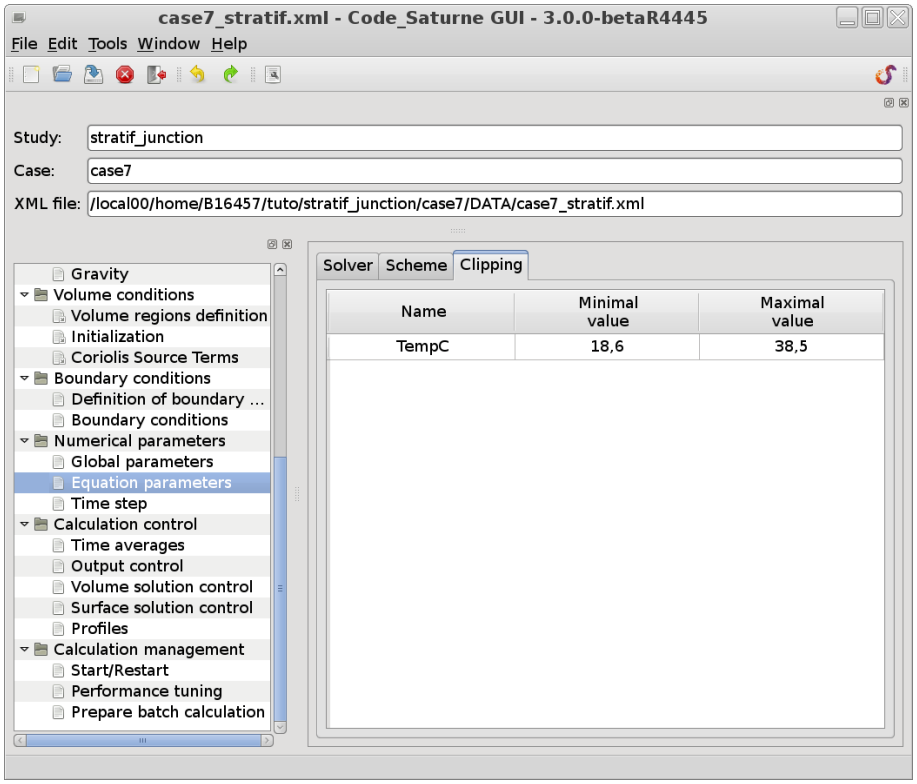


Figure VI.85: Scalar initialization

Create the boundary regions.

Colors	Conditions
2	inlet
6	inlet
7	outlet
5	wall

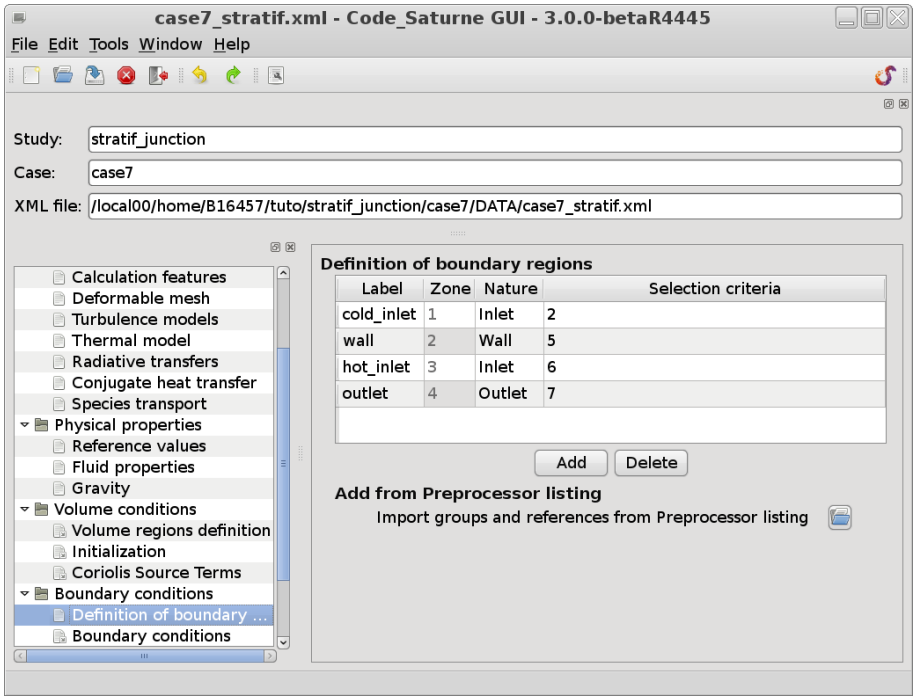


Figure VI.86: Boundary regions

For the dynamic boundary conditions, the velocity is 0.03183 $m.s^{-1}$ in the z direction and the hydraulic diameter 0.4 m for both inlets.

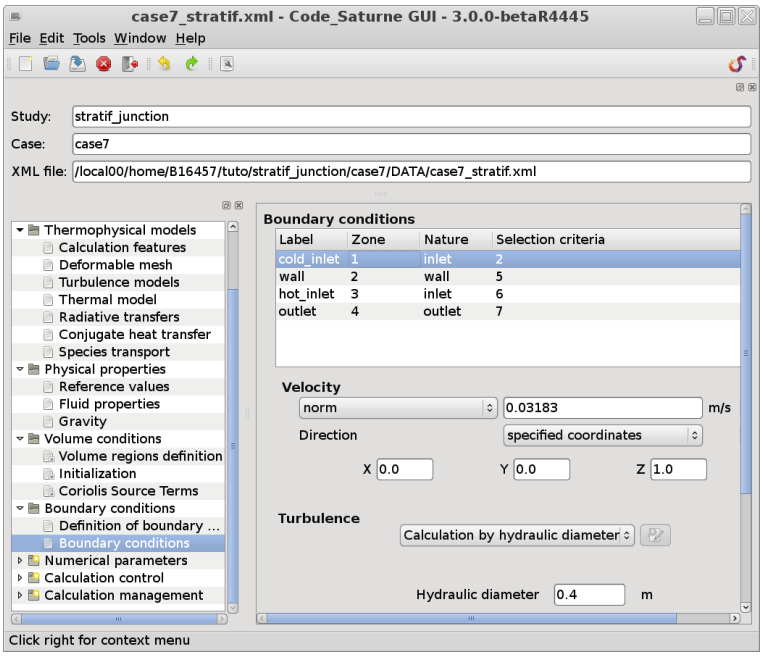


Figure VI.87: Dynamic boundary conditions

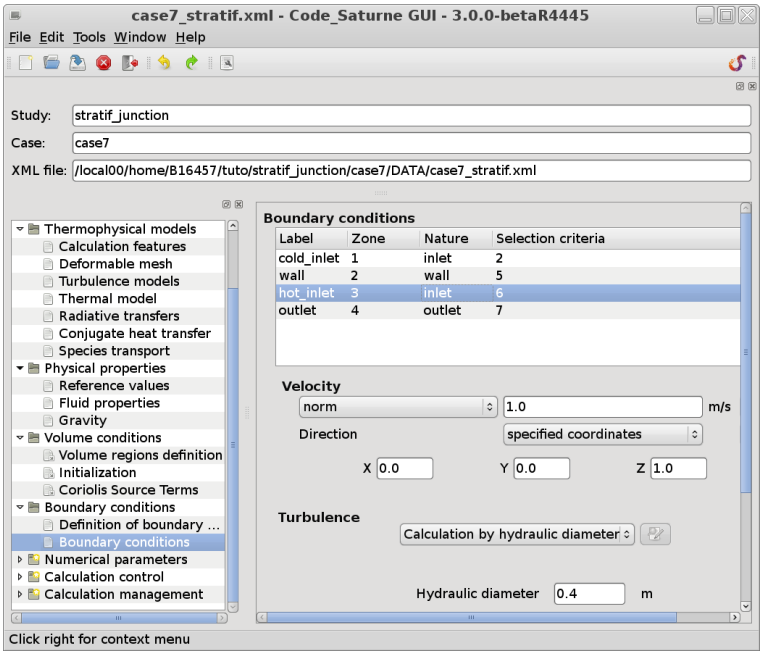


Figure VI.88: Dynamic boundary conditions

For the scalar boundary conditions, the temperature of the cold inlet is 18.6°C and that of the hot inlet is 38.5°C.

- **Cold inlet:**

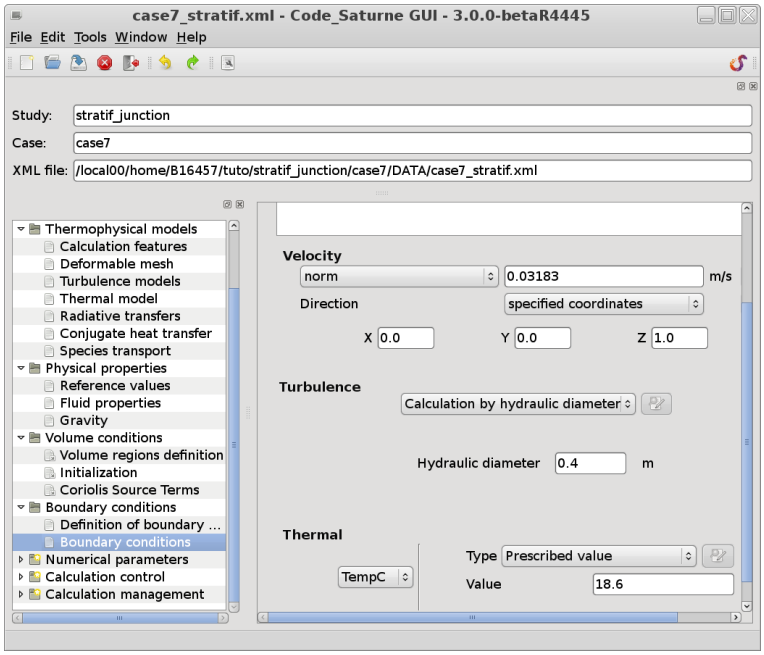


Figure VI.89: Temperature boundary conditions

- **Hot inlet:**

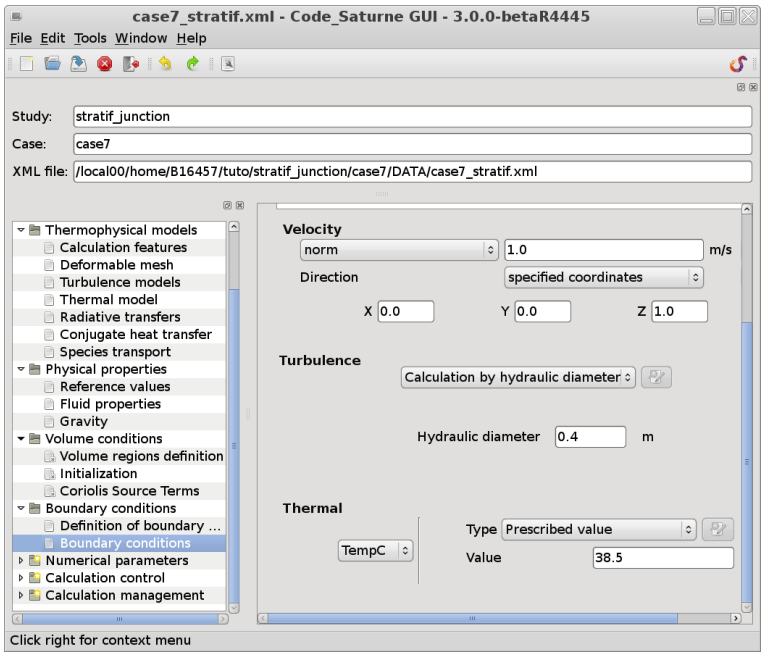


Figure VI.90: Temperature boundary conditions

Tick the appropriate box for the time step to be variable in time and uniform in space. In the boxes below, enter the following parameters:

Parameters of calculation control	
Number of iterations	100
Reference time step	1 s
Maximal CFL number	20
Maximal Fourier number	60
Minimal time step	0.01 s
Maximal time step	70 s
Time step maximal variation	0.1

And activate the option *Time step limitation with the local thermal time step*

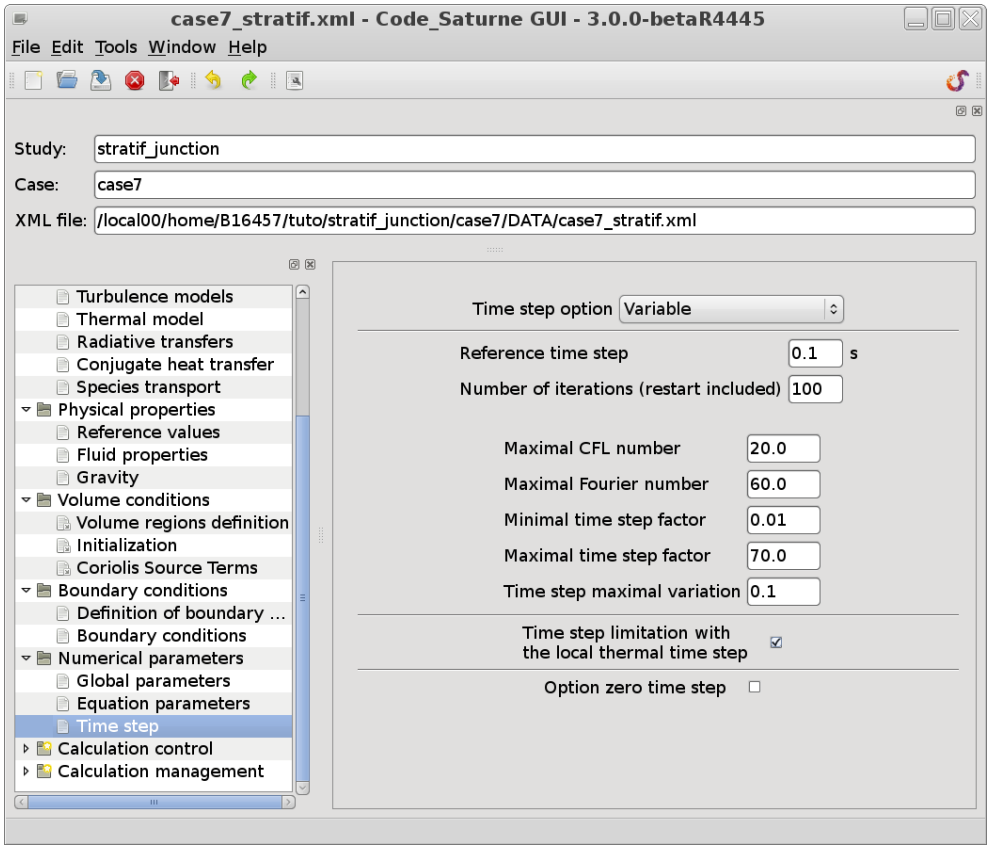


Figure VI.91: Time step

Set the frequency of post-processing for the main writer **r**esults to 10.

Create four monitoring probes at the following coordinates:

Points	X(m)	Y(m)	Z(m)
1	0.010025	0.01534	-0.011765
2	1.625	0.01534	-0.031652
3	3.225	0.01534	-0.031652
4	3.8726	0.047481	7.25

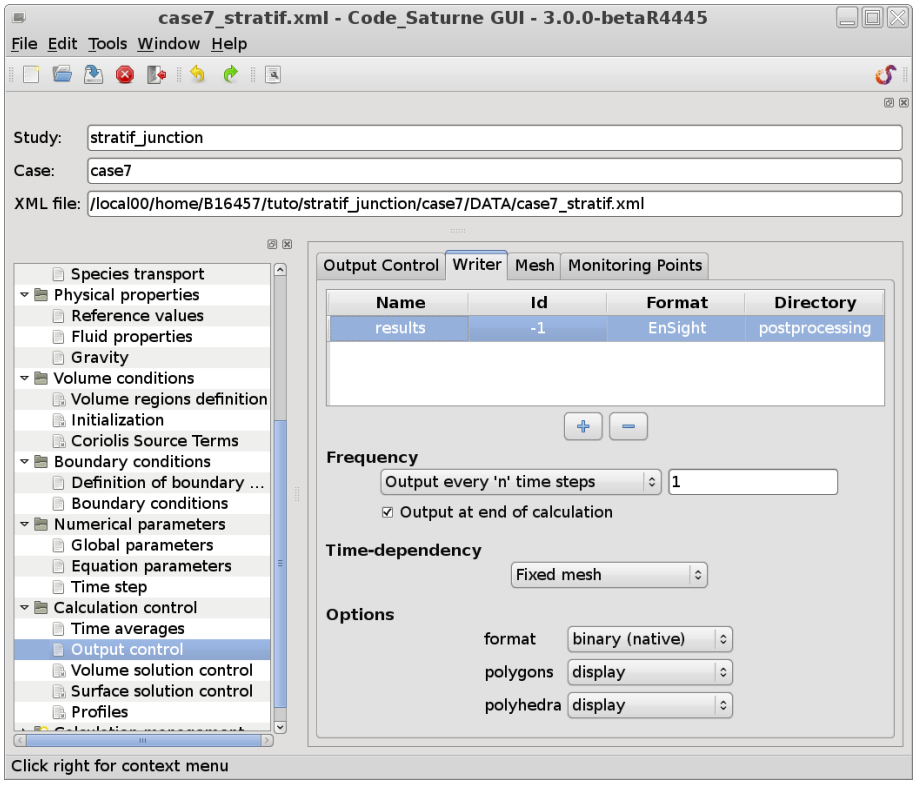


Figure VI.92: Output management

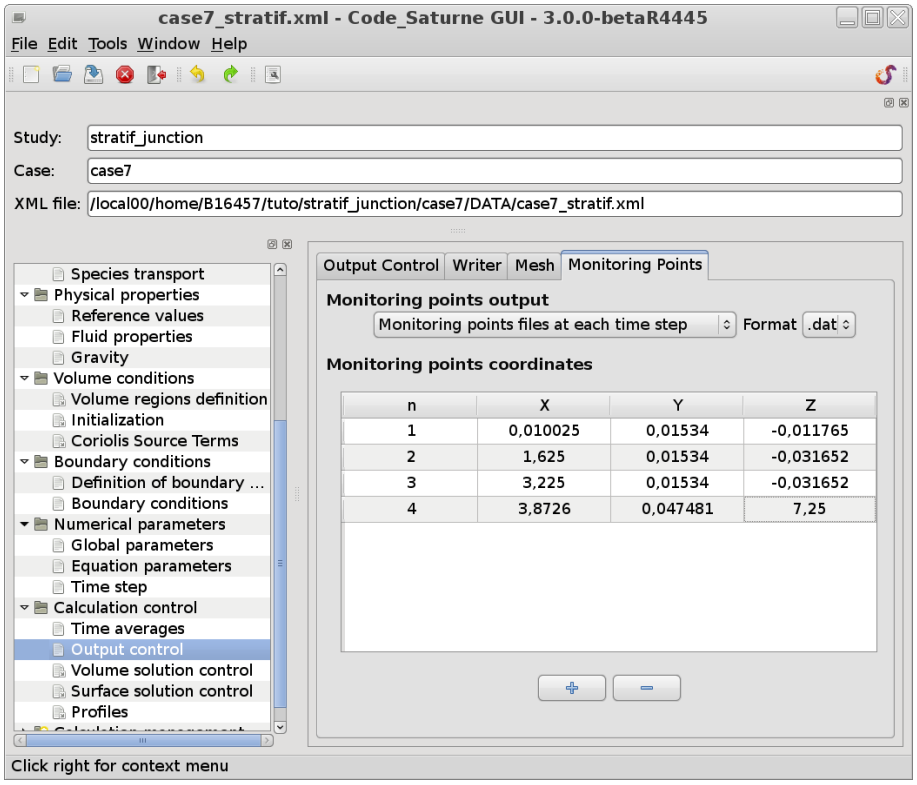


Figure VI.93: Monitoring points

For the advanced post-processing features, copy the files *cs_user_postprocess.c* and *cs_user_postprocess_var.f90* to the SRC directory. The general content of these routines is described in the user manual or in the examples available in the directory `SRC/REFERENCE/base`. The modified routines adapted to this test case are available in the `examples` directory. Only the main elements are mentioned here.

• **cs_user_postprocess_meshes** (in **cs_user_postprocess.c**)
This is called only once, at the beginning of the calculation. It allows to define the different writers and parts.

• **cs_user_postprocess_var.f90**
This routine is called at each time step. It allows to specify which variable will be written on which part.

6 Solution for case6

• **Step 1**: check the post-install required for coupling *Code_Saturne* with SYRTHES.
The first step is to check the post-install required for coupling with SYRTHES and verify if the SYRTHES `PATH` is correctly known in the system environment. We just need to edit the batch file[2] name `code_saturne.cfg` as below:

```
$ vim <install-prefix>/etc/code_saturne.cfg
>### Set the location to the SYRTHES installation directory.
> syrthes = <install-prefix-syrthes>
```

• **Step 2**: source the `syrthes.profile` file in your user environment.
Before using SYRTHES alone, you have to copy and source this file to define SYRTHES environment variables (like `$SYRTHES4_HOME`) in your terminal, as follows:

```
$ cp <install-prefix-syrthes>/bin/syrthes.profile .
$ source syrthes.profile
$ echo $SYRTHES4_HOME (to check the SYRTHES PATH in your environment)
```

After having defined correctly your environment, to be able to launch a coupling computation *Code_Saturne*-SYRTHES or a SYRTHES computation alone, you just have to create the coupling study directory.

• **Step 3**: create the `3disks2D` study directory, two subdirectories `fluid` and `solid`.

This is done using the standard command:

```
$ code_saturne create -s 3disks2D -c fluid --syrthes solid
> code_saturne 3.0 study/case generation
> o Creating study '3disk2D' ...
> o Creating case 'fluid' ...
> SYRTHES4 home directory:  <install-prefix-syrthes>
> MPI home directory:  /usr
>
>************************************************
> solid :  creating SYRTHES case ...
> <install-prefix-syrthes>
> OK !
>************************************************
```

• **Remark**: The fluid mesh must be copied in the directory `MESH`. The solid mesh must be copied in the subdirectory `solid`.

[2] see the installation guide, name `install.pdf`, in `<install-prefix>/share/doc/code_saturne/` directory.

6.1 Launching the SYRTHES computation alone

The preparation of the computation for `case5` is defined below:

- **Step 1**: launch the SYRTHES Graphical USer Interface (syrthes.gui),
- **Step 2**: open a **New Data File**,
- **Step 3**: check the name of the mesh and convert this one in .syr format,
- **Step 4**: define the initial and boundary conditions for the conduction problem,
- **Step 5**: define the physical properties of each disk {1, 2, 3 and 4},
- **Step 6**: running the SYRTHES computation alone.

• **Step 1**: launch the SYRTHES Graphical User Interface (Gui).
The SYRTHES Graphical User Interface is launched by the following command lines in the solid sub-directory:

```
$ cd 3disks2D/solid/
$ syrthes.gui &
```

• **Step 2**: choose a New Data File inside the (Gui).

Figure VI.94: Running the SYRTHES's IHM with syrthes.gui

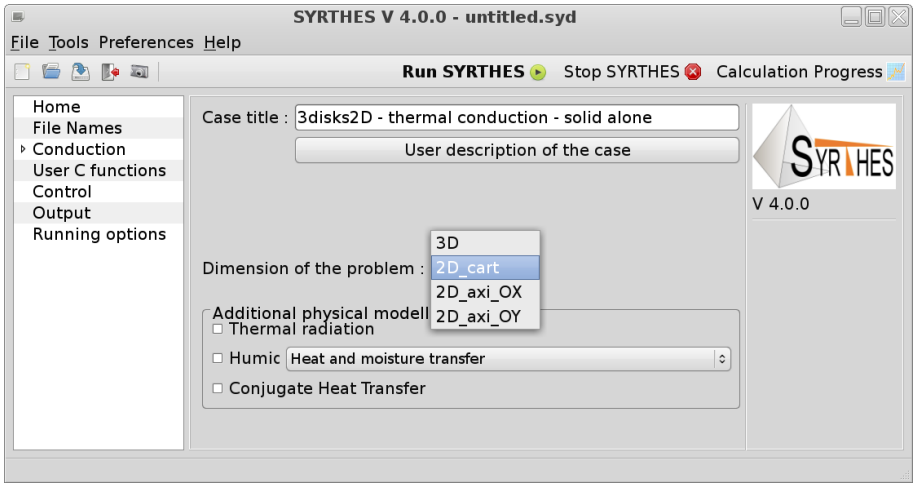


Figure VI.95: Define the dimension and physical modelling of the problem treated

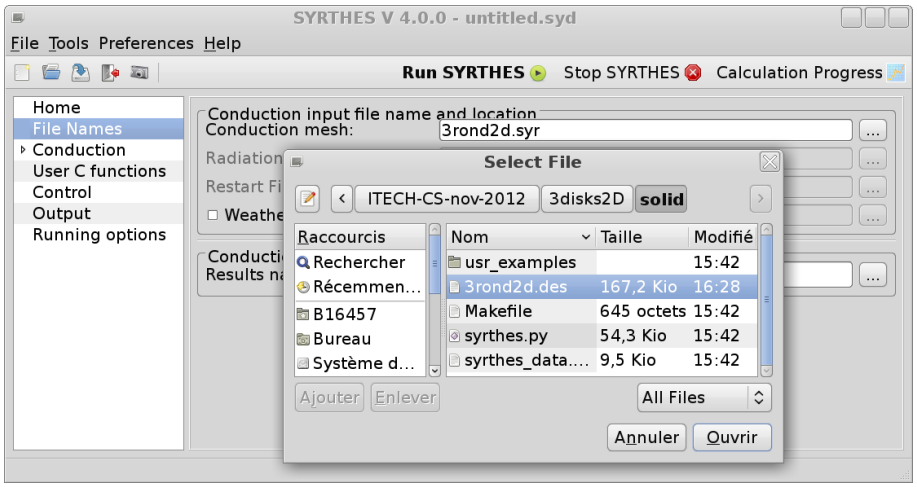


Figure VI.96: Choose the 2D solid mesh file with the format `.des`.

Figure VI.97: The SYRTHES (Gui) directly converts the `.des` to the `.syr` format.

• **Remark**: Inside the SYRTHES Graphical USer Interface (Gui), we can load the SIMAIL format `*.des` for the solid mesh. This one will be automatically transformed to the `*.syr` format.
It can also be done with the following command line:

```
$ convert2syrthes4 -m 3rond2d.des
```

• **Remark**: You can convert the `*.syr` format into a `*.med` format. Like that, you can load the `*.med` file inside SALOME, after having used this command line below:

```
$ syrthes4med30 -m 3rond2d.syr -o 3rond2d.med
```

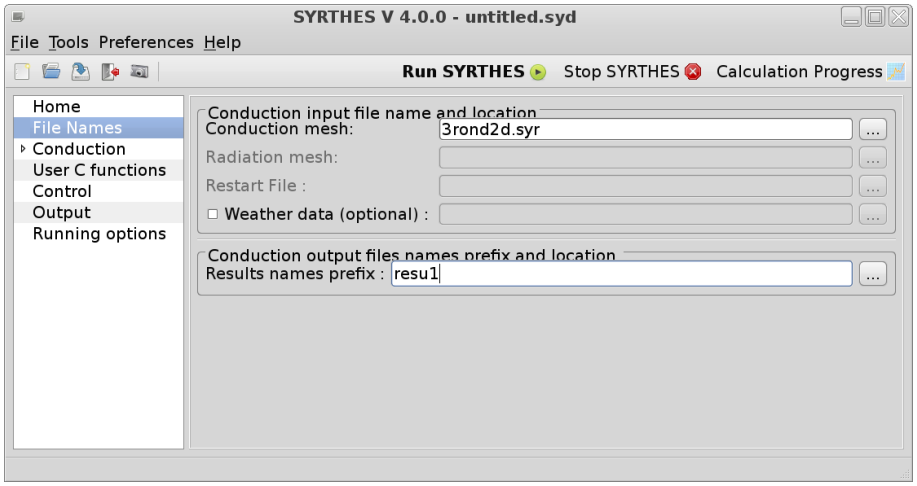


Figure VI.98: Choose a name for the results files `.res`, `.his` and `.rdt`

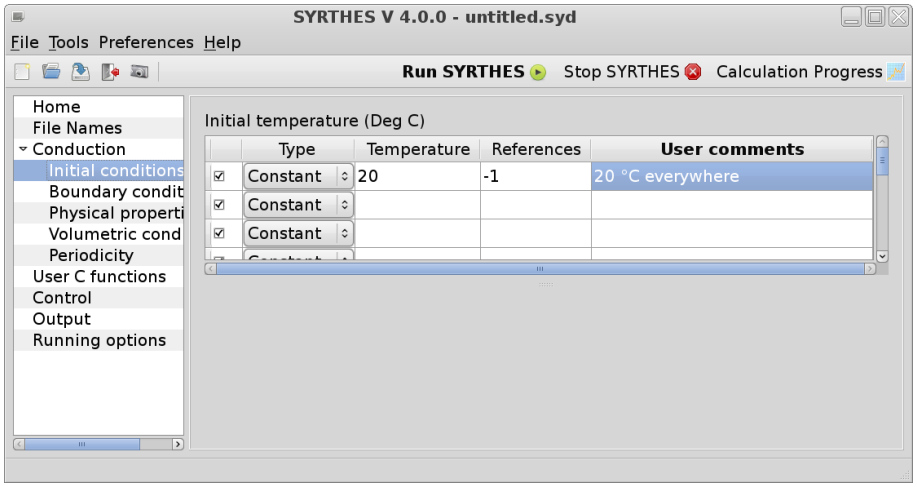


Figure VI.99: Define the initial temperature conditions inside the different disks.

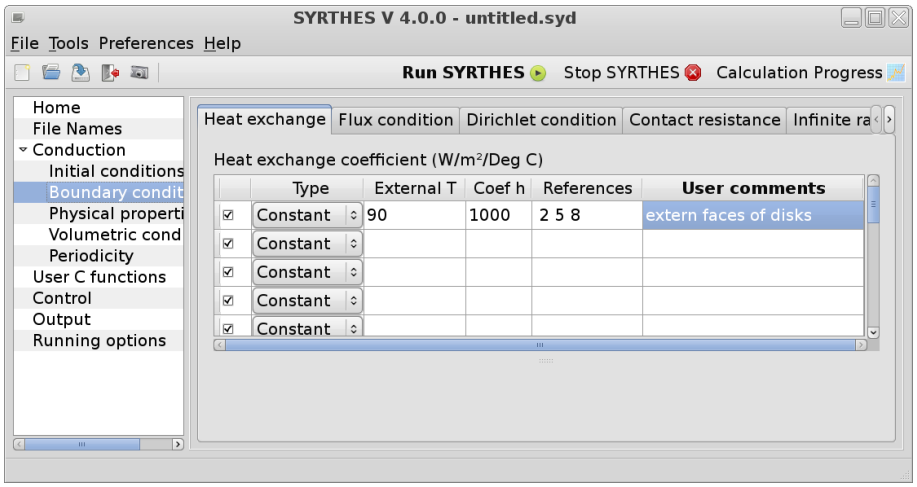


Figure VI.100: Define the temperature boundary conditions for the extern face of the three disks.

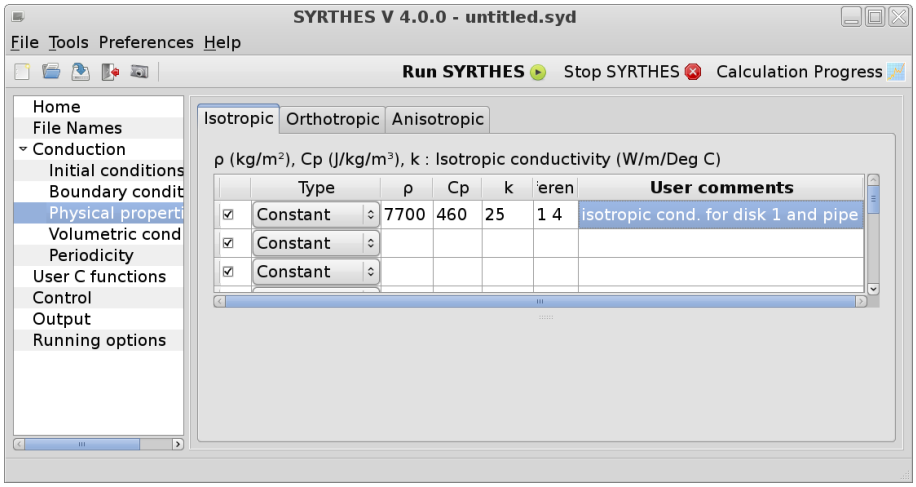


Figure VI.101: Define the physical properties for the disk 1 and 4 with isotropic conductivity.

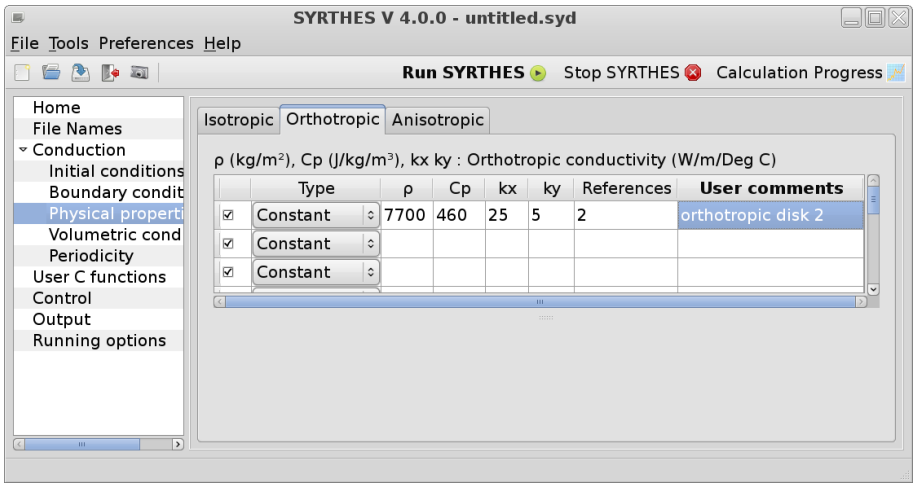


Figure VI.102: Define the physical properties for the disk 2 with isotropic conductivity.

• **Remark**: To correctly identify the volume references associated to a specific physical property, we can check the mesh regions directly inside ParaView after having used following command line:

```
$ syrthes4ensight -m 3rond2d.syr -o mesh_3rond2d
*****************************************************
--> geometry file name :  mesh_3rond2d.ensight.geom
--> case file name :  mesh_3rond2d.ensight.case
```

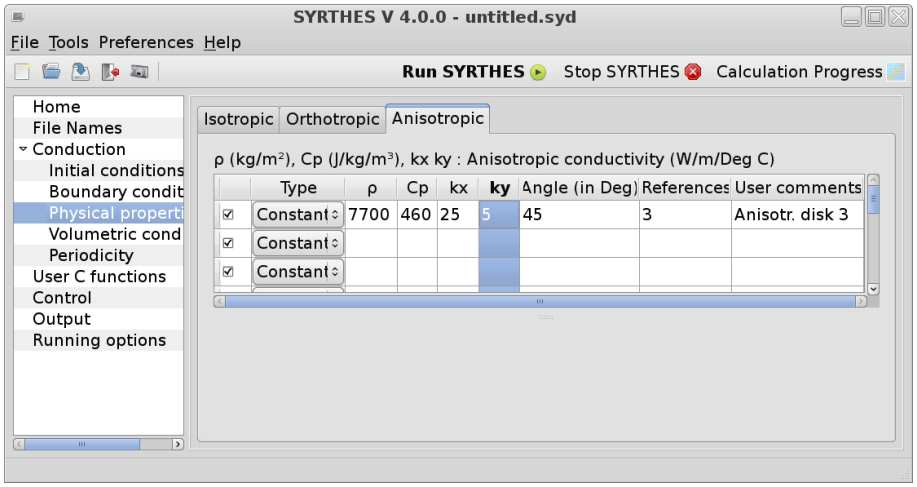


Figure VI.103: Define the Physical properties for the disk 3 with anisotropic conductivity.

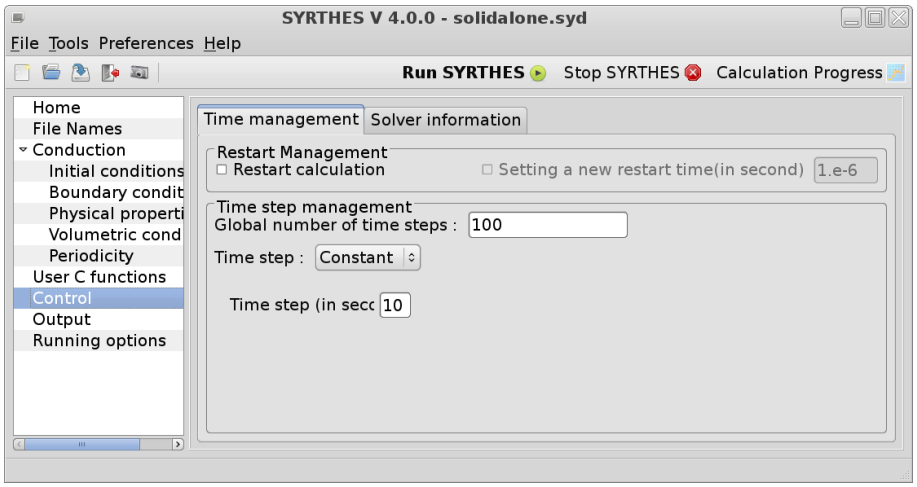


Figure VI.104: Define the global number of time steps and the time step for the 2D solid conduction computation.

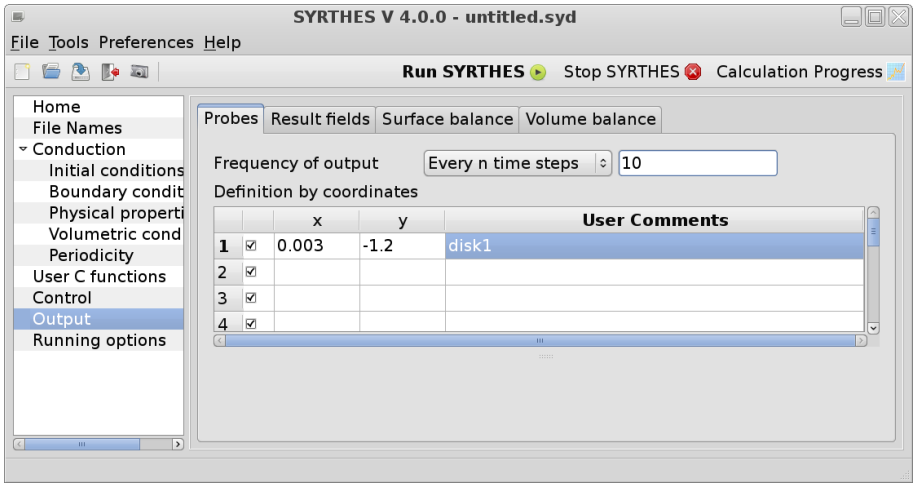


Figure VI.105: Define the probe coordinates for output management.

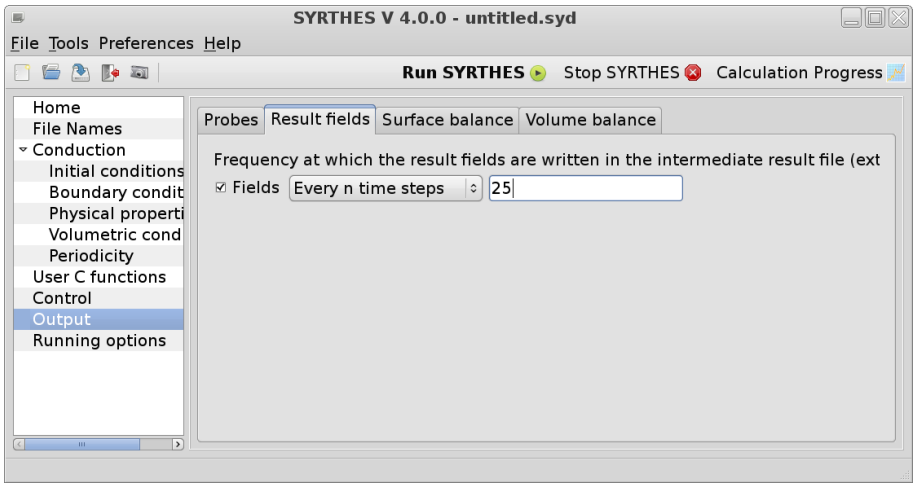


Figure VI.106: Define the frequency at which the results fields are written

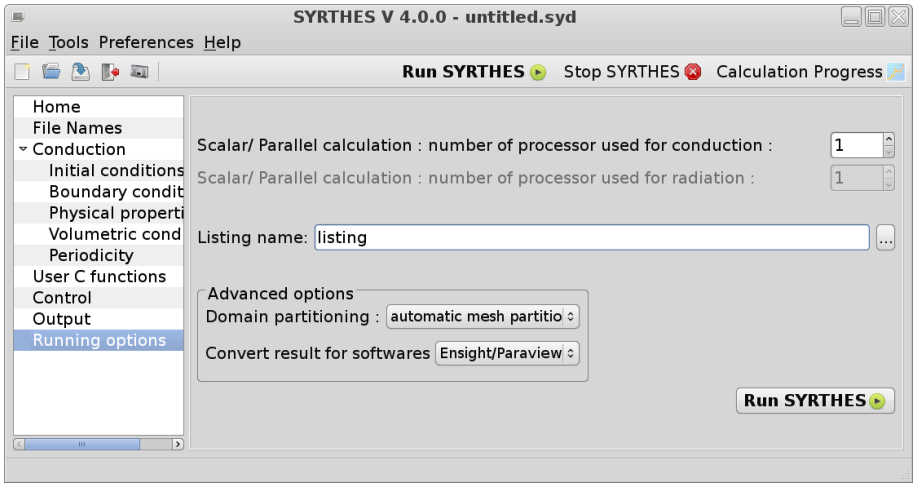


Figure VI.107: Define the file name of the SYRTHES listing and the number of processors used.

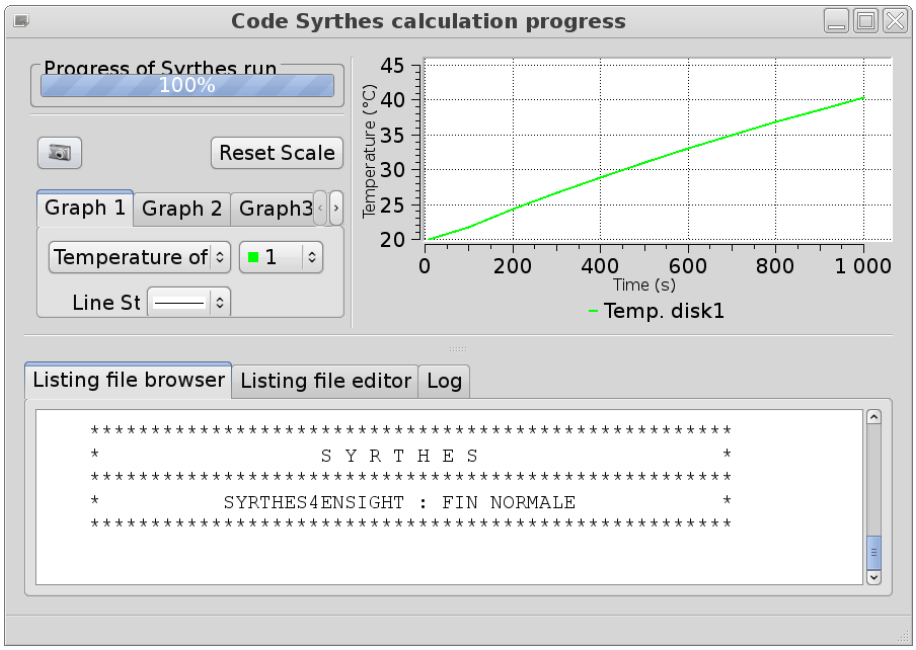


Figure VI.108: Screenshot of the computation progress window.

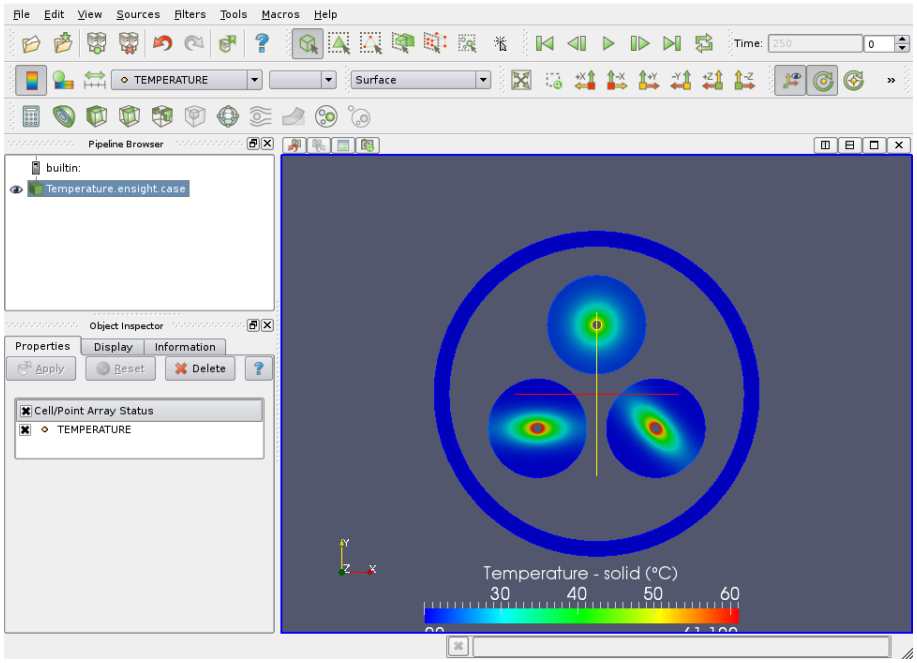


Figure VI.109: Screenshot of the 2D solid temperature Field.

• **Remark**: We can visualize the temperature results fields by applying the following command line to the results file `resu1.res` or `resu1.rdt` (for the results saved at the last time step or the results saved at each time step):

```
$ syrthes4ensight -m 3rond2d.syr -r resu1.res -o Results_Temp
$ syrthes4ensight -m 3rond2d.syr -r resu1.rdt -o Chrono_Temp
```

6.2 Launching the *Code_Saturne* computation alone

The preparation of the fluid computation alone for `case5` is defined below:

- **Step 1**: launch the *Code_Saturne* Graphical User Interface (`./SaturneGUI`),
- **Step 2**: open a **New case**,
- **Step 3**: check the quality of the fluid mesh with the `check_mesh`,
- **Step 4**: define the initial and boundary conditions for the air flow problem,
- **Step 5**: define the physical properties of the disk for the air flow,
- **Step 6**: running the *Code_Saturne* computation alone.

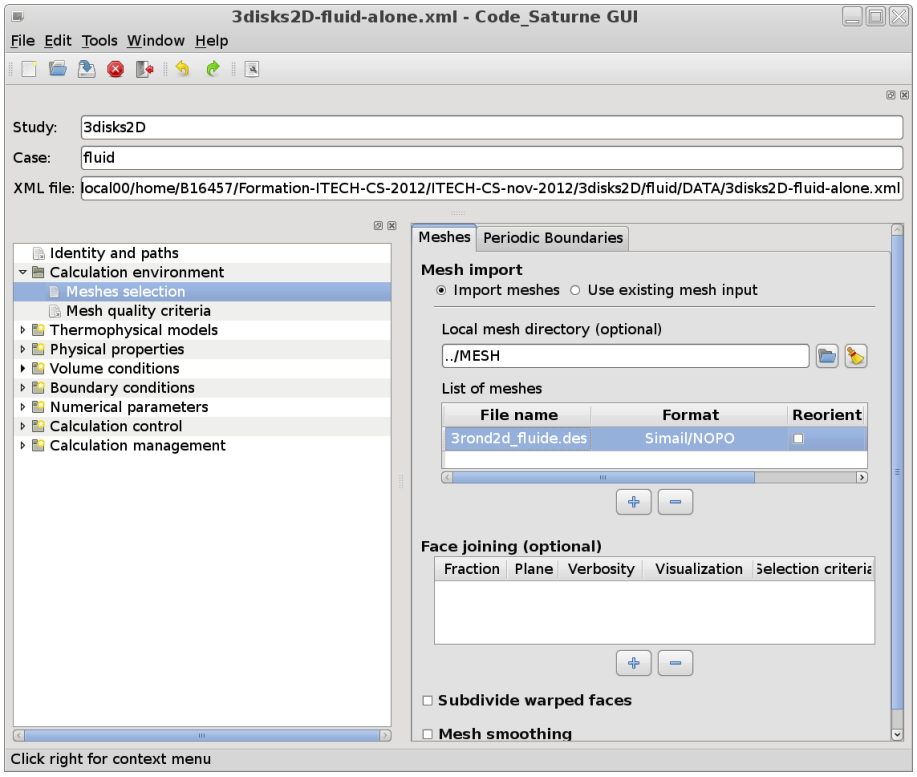


Figure VI.110: Choose the fluid mesh with *Code_Saturne* (GUI)

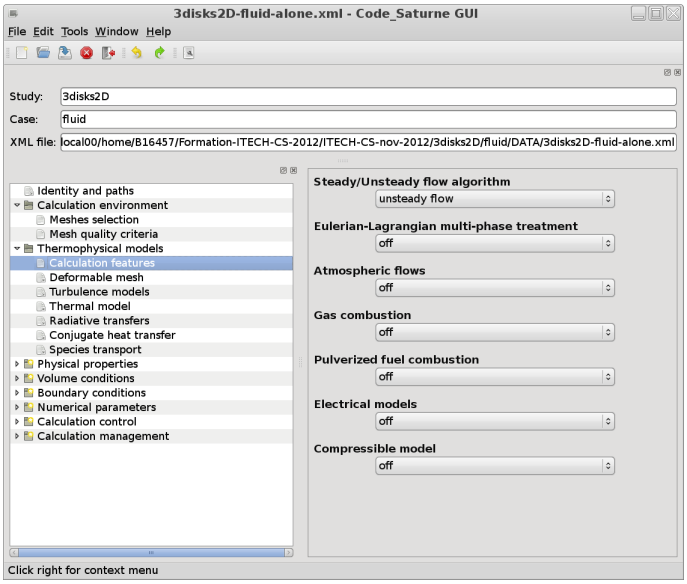


Figure VI.111: Define the physical modelling associated to the air flow inside the fluid domain.

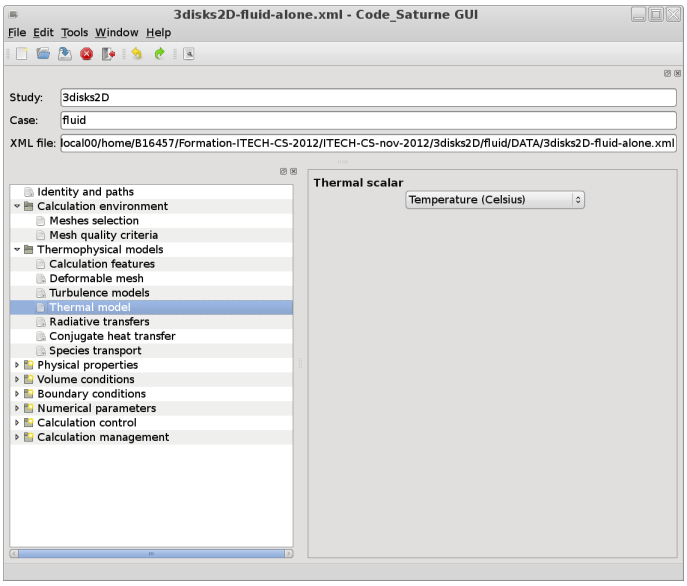


Figure VI.112: Choose the Temperature scalar.

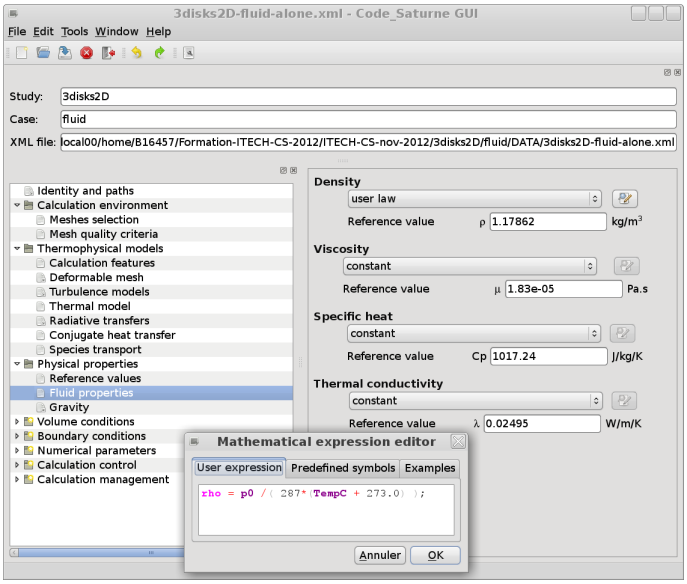


Figure VI.113: Define the variable density with a ideal gas law inside the *Code_Saturne* (GUI).

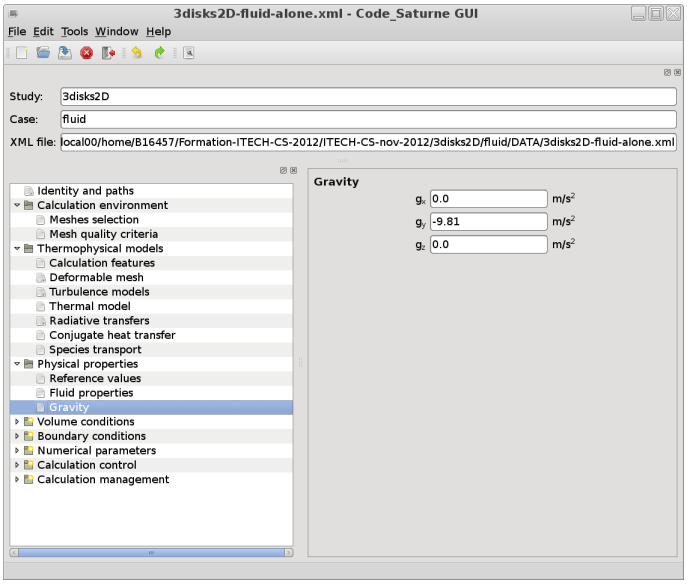


Figure VI.114: Define the gravity

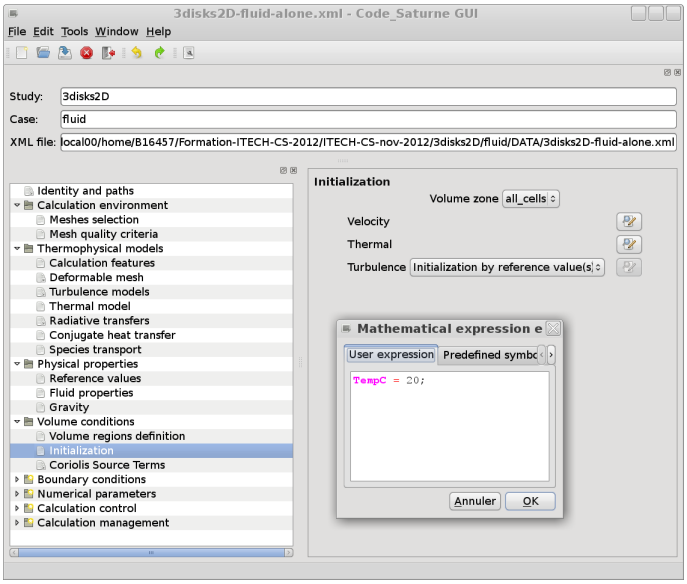


Figure VI.115: Initalization of the velocity components and temperature variables.

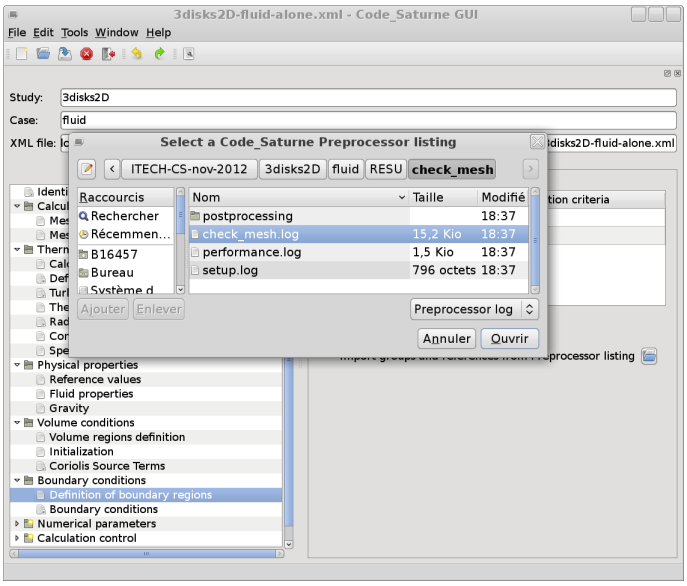


Figure VI.116: Load the check_mesh.log file inside the *Code_Saturne* (GUI).

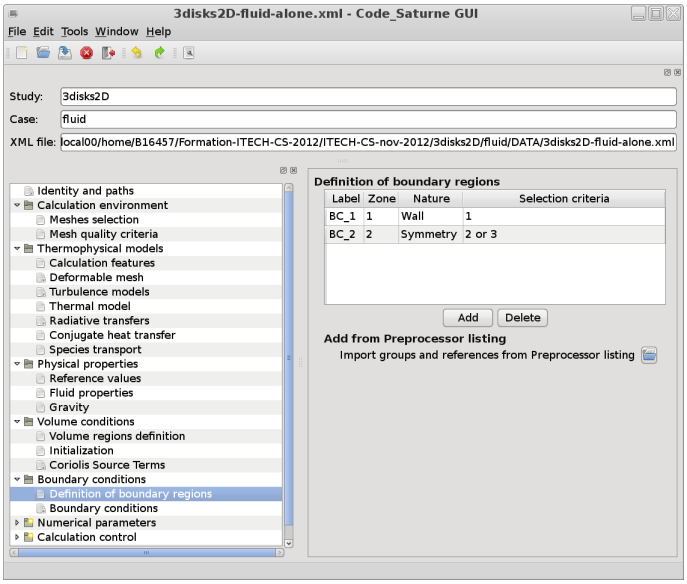


Figure VI.117: Loading the check_mesh.log file automatically defines the boundary regions.

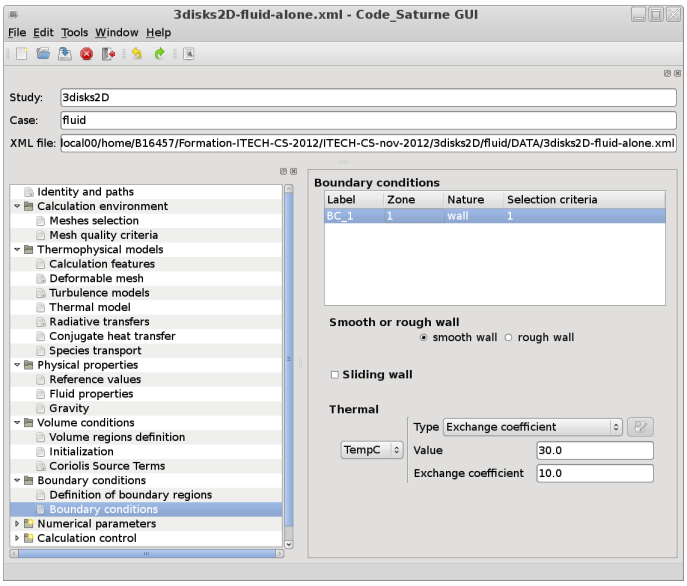


Figure VI.118: Define a thermal transfer condition as wall boundary condition with a extern wall temperature $T_{ext} = 30°C$ and a exchange coefficient $q_{ext} = 10$ $(W/m^2.K)$.

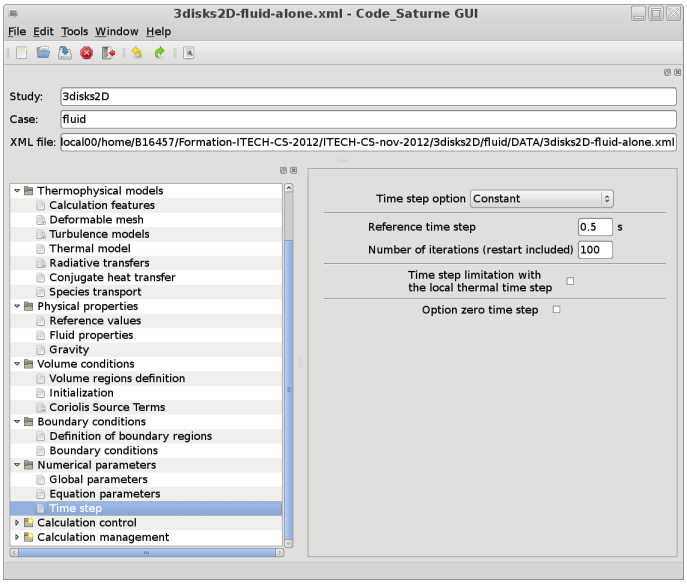


Figure VI.119: Define the iterations number and time step.

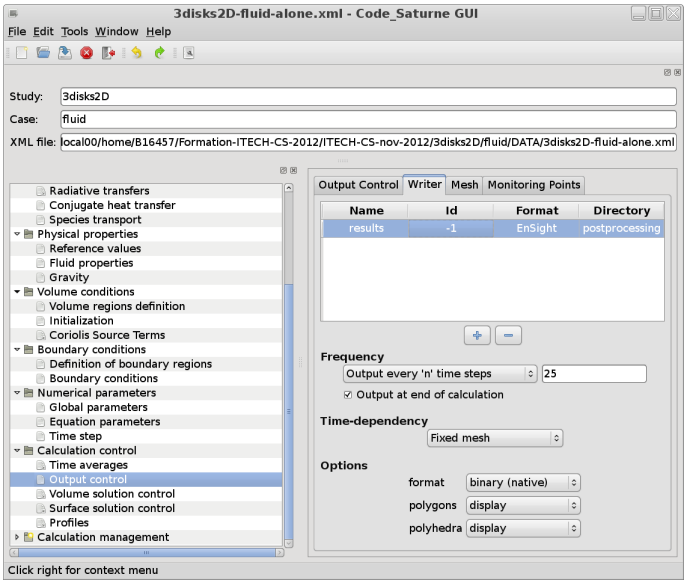


Figure VI.120: Define the writer and frequency output inside the *Code_Saturne* (GUI).

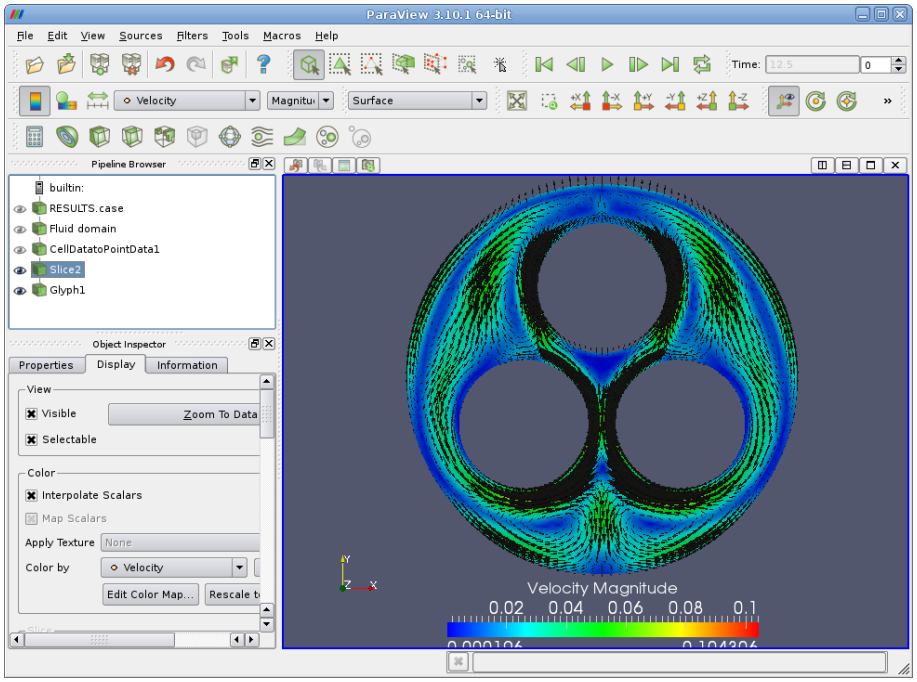


Figure VI.121: Visualization of the 2D fluid velocity field

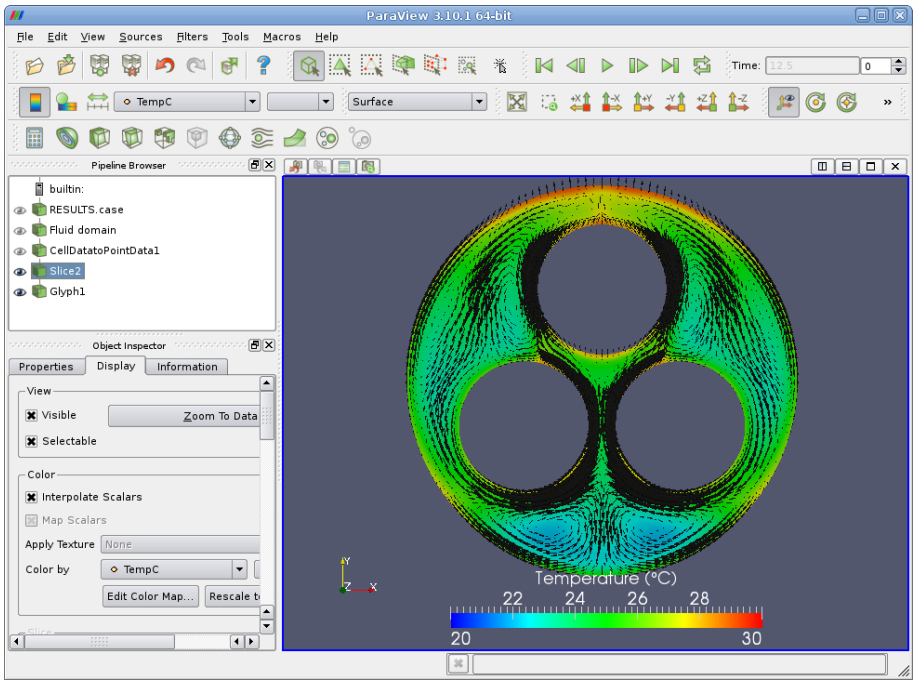


Figure VI.122: Visualization of the 2D fluid temperature field

6.3 Launching the *Code_Saturne*-SYRTHES coupling computation

The last modification to prepare the coupling computation are given below:
- **Step 1**: activate the conjugate heat transfer in the SYRTHES (Gui),
- **Step 2**: activate the conjugate heat transfer in the *Code_Saturne* (GUI),
- **Step 3**: give identical iterations number and time step for both codes,
- **Step 4**: check the `runcase_coupling` script and launch it.

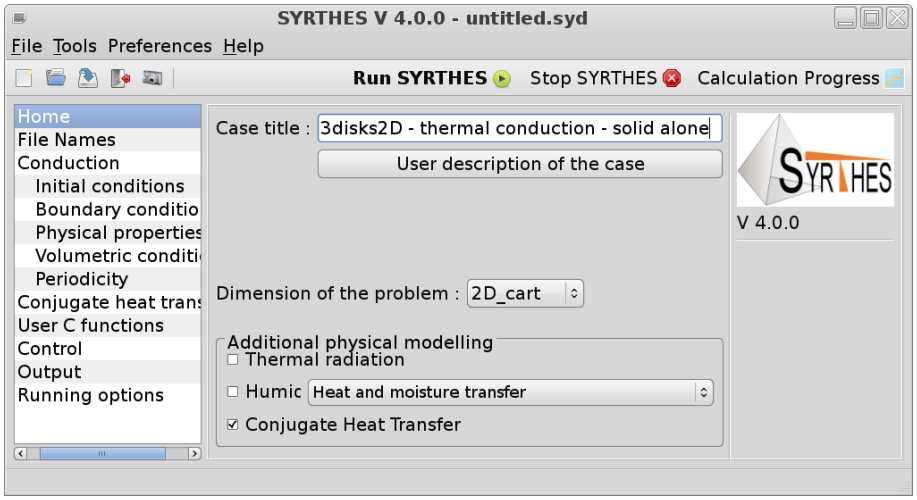


Figure VI.123: Activate the conjugate heat transfer for the solid domain.

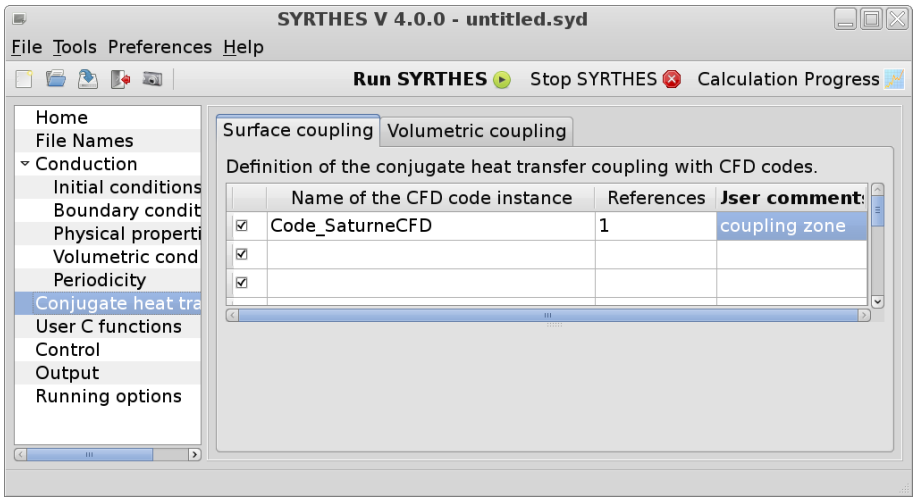


Figure VI.124: Specify the reference zone for the coupling surfaces with *Code_Saturne*.

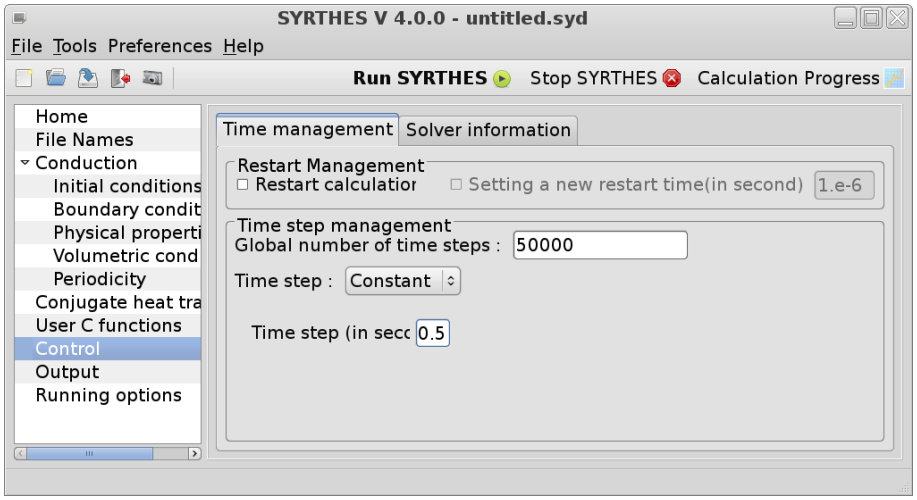


Figure VI.125: Change the iterations number and time step for the solid domain.

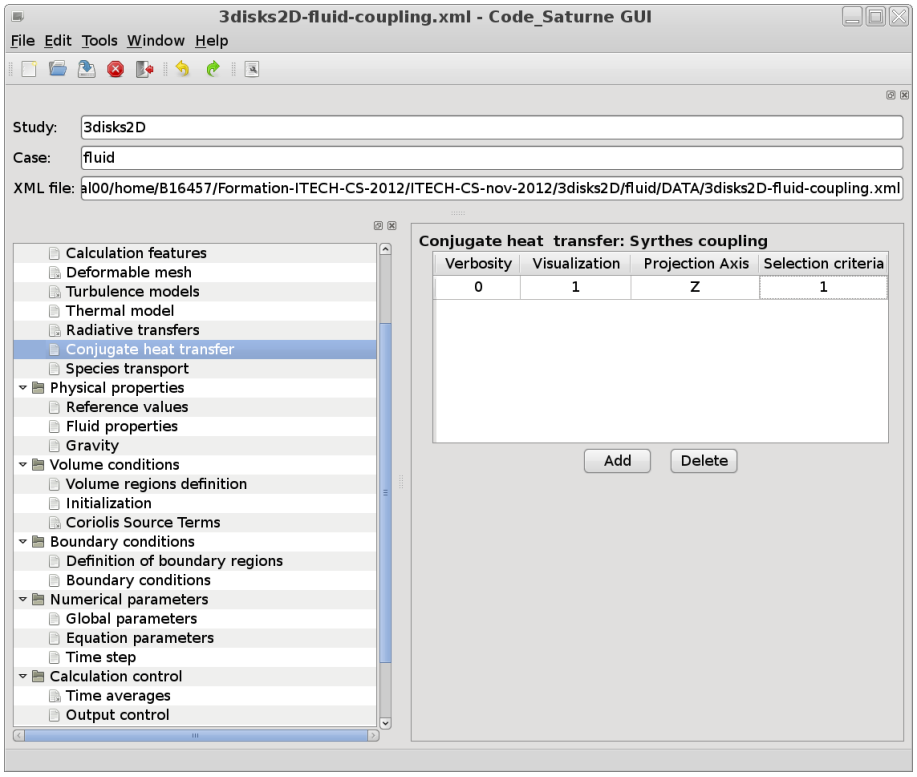


Figure VI.126: Activate the conjugate heat transfer for the fluid domain.

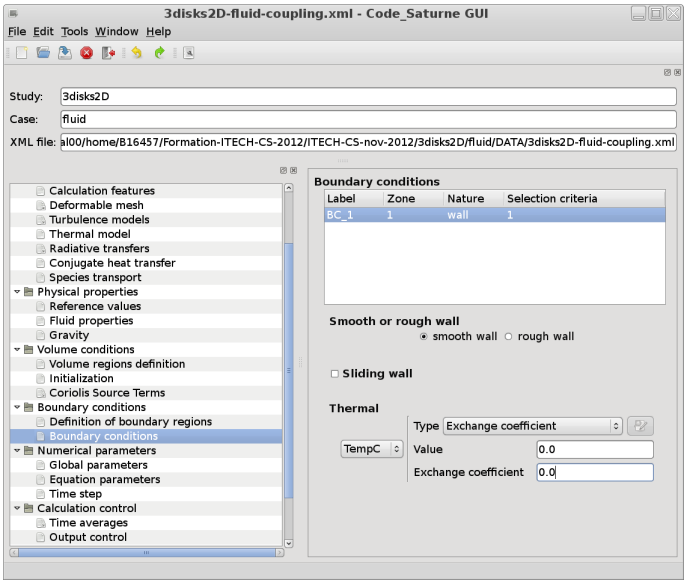


Figure VI.127: Change the boundary conditions for the wall temperature.

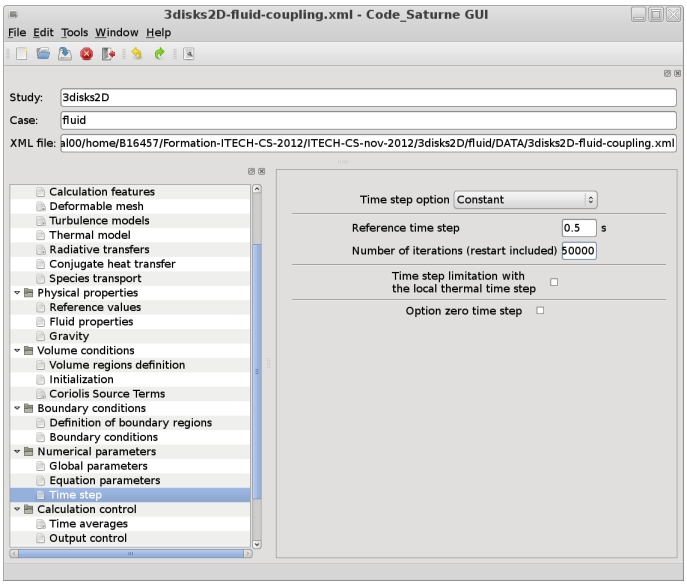


Figure VI.128: Change the iterations number and time step for the fluid computation.

• **Remark**: After having modified the data setting for the fluid and solid domains to activate the conjugate heat transfer on both sides, we just have to increase the iterations number and check the `runcase_coupling` script.

We just need to edit the `runcase_coupling` script and give the name of your SYRTHES script saved in the SYRTHES (Gui) as below:

```
$ vim runcase_coupling
> domains = [
>
> 'solver':  'Code_Saturne',
> 'domain':  'fluid',
> 'script':  'runcase',
> 'n_procs_weight':  None,
> 'n_procs_min':  4,
> 'n_procs_max':  4
>
> 'solver':  'SYRTHES',
> 'domain':  'solid',
> 'script':  'solid-coupling.syd',
> 'n_procs_weight':  None,
> 'n_procs_min':  2,
> 'n_procs_max':  2,
> 'opt' :  '-v ens'
>
> ]
```

You just have to launch the `runcase_coupling` present in the study directory (named in our case `3disks2d`) and run the coupling computation, as follows:

```
$ runcase_coupling
```

• **Remarks**: in the `runcase_coupling`, you can specify the processors number for each code (as this example with 4 processors for *Code_Saturne* and 2 processors for SYRTHES) in parallel or just one processor for each code in sequential.

You can specify the ouput results format for SYRTHES with an option (`opt`) which takes the value `-v ens` for a 3D fields output with a EnSight format or `-v med` for a 3D fields output with a SALOME format).