Examples of data set

Click here to come back to the previous page
The user finds here some examples illustrating different configurations related to the namelist "Domain_Features".
The data initialized by default, and not explicitly required, are generally not present for a sake of clarity.
Data values are showed for equations used in a dimensional form.

2D domain configuration

No parallel setting

Domain in cartesian geometry.
No OpenMP parallelization is considered.
No domain decomposition approach (MPI parallelization).
The grid is regular. The number of cells in each direction is 80.

```
 &Domain_Features Geometric_Layout                = 0,
                  Start_Coordinate_I_Direction   =-0.05 ,
                  End_Coordinate_I_Direction     = 0.05,
                  Start_Coordinate_J_Direction   =-0.05 ,
                  End_Coordinate_J_Direction     = 0.05,
                  Start_Coordinate_K_Direction   = 0.00 ,
                  End_Coordinate_K_Direction     = 0.00,
                  Cells_Number_I_Direction       = 80 ,
                  Cells_Number_J_Direction       = 60 ,
                  Cells_Number_K_Direction       = 1 /
```

Even in 2D geometrical configuration, the data about the K-direction must be set.

Parallel setting : OpenMP Only

Domain in cartesian geometry.
OpenMP parallelization is considered with 4 threads.
No domain decomposition approach (MPI parallelization).
The grid is regular. The number of cells in each direction is 80.

```
 &Domain_Features Geometric_Layout                = 0,
                  Start_Coordinate_I_Direction   =-0.05 ,
                  End_Coordinate_I_Direction     = 0.05,
                  Start_Coordinate_J_Direction   =-0.05 ,
                  End_Coordinate_J_Direction     = 0.05,
                  Start_Coordinate_K_Direction   = 0.00 ,
                  End_Coordinate_K_Direction     = 0.00,
                  Cells_Number_I_Direction       = 80 ,
                  Cells_Number_J_Direction       = 60 ,
```

```
                    Cells_Number_K_Direction       = 1,
                    Number_OMP_Threads= 4 /
```

The code must be compiled with the OpenMP options.

Parallel setting : MPI Only (in MPI cartesian topology)

Domain in cartesian geometry.
No OpenMP parallelization is considered .
Domain decomposition approach (MPI parallelization) in MPI cartesian topology. The domain is divided on 8 subdomains :

- 4 along the I-direction
- 2 along the J-direction
- 1 along the K-direction (default)

The grid is regular. The number of cells in each direction is 80 **for each subdomain**.

```
 &Domain_Features Geometric_Layout                = 0,
                  Start_Coordinate_I_Direction   =-0.05 ,
                  End_Coordinate_I_Direction     = 0.05,
                  Start_Coordinate_J_Direction   =-0.05 ,
                  End_Coordinate_J_Direction     = 0.05,
                  Start_Coordinate_K_Direction   = 0.00 ,
                  End_Coordinate_K_Direction     = 0.00,
                  Cells_Number_I_Direction       = 80 ,
                  Cells_Number_J_Direction       = 60 ,
                  Cells_Number_K_Direction       = 1,
                  MPI_Cartesian_Topology         = .false. ,
                  Total_Number_MPI_Processes     = 8,
                  Max_Number_MPI_Proc_I_Direction= 4,
                  Max_Number_MPI_Proc_J_Direction= 2,
                  Max_Number_MPI_Proc_K_Direction= 1 /
```

The code must be compiled with the MPI options.

Parallel setting : MPI Only (in MPI graph topology)

Domain in cartesian geometry.
No OpenMP parallelization is considered .
Domain decomposition approach (MPI parallelization) in MPI graph topology. The domain is divided on 4 subdomains :

- 4 along the I-direction (maximum value)
- 2 along the J-direction (maximum value)
- 1 along the K-direction (default)

The grid is regular. The number of cells in each direction is 80 **for each subdomain**.

```
 &Domain_Features Geometric_Layout                = 0,
                  Start_Coordinate_I_Direction   =-0.05 ,
                  End_Coordinate_I_Direction     = 0.05,
```

```
                    Start_Coordinate_J_Direction   =-0.05 ,
                    End_Coordinate_J_Direction     = 0.05,
                    Start_Coordinate_K_Direction   = 0.00 ,
                    End_Coordinate_K_Direction     = 0.00,
                    Cells_Number_I_Direction       = 80 ,
                    Cells_Number_J_Direction       = 60 ,
                    Cells_Number_K_Direction       = 1,
                    MPI_Graphic_Topology           = .true. ,
                    Total_Number_MPI_Processes     =  Value depending on the
geometrical configuration of the domain,
                    Max_Number_MPI_Proc_I_Direction= 4,
                    Max_Number_MPI_Proc_J_Direction= 2,
                    Max_Number_MPI_Proc_K_Direction= 1 /
```

The code must be compiled with the MPI options.
The MPI graph topology is used in cases where the domain configuration have got large immersed bodies.
The aim is to build a domain decomposition in a way that takes the solid parts out of the domain and ensure that the numerical domain is mainly fluid.
In a first step, the domain decomposition is carried out as if the MPI cartesian decomposition was used. The number of processes “Total_Number_MPI_Processes” will be equal to the multiplication of the “Max_Number_MPI_Proc_I_Direction” by “Max_Number_MPI_Proc_J_Direction” by “Max_Number_MPI_Proc_K_Direction”.
When some subdomains are totally occupied by solid parts, they are useless. They must therefore be removed in order to reduce the MPI process number. As the MPI topology is no longer cartesian due to the “holes” arisen in the MPI topology, the subdomain decomposition is handled with the MPI graph topology.

The software “mpi_subdomain_decomposition” has been developed for helping the user to build the subdomain decomposition in these cases :

- The user sets the MPI data in the data file as for a cartesian subdomain splitting.
- The user then runs the software that :
  - reads the data file
  - analyses the geometrical layout of the domain with respect to the subdomain decomposition following a MPI cartesian topology proposed by the user
  - reformulates the subdomain splitting by excluding the pointless subdomains (covering the solid parts of the domain).
  - build the data file named “data_mpi_subdomain_layout.dat” that contains :
    - a map of the enabled MPI processes related to the new subdomain layout
    - the maximum MPI process number implied in the MPI graph topology

The user must set the variable “Total_Number_MPI_Processes” to this new value and keep the other data in the same state. During the run, sunfluidh will read this own data file and the new data file “data_mpi_subdomain_layout.dat” in order to know the MPI process layout related to the MPI graph topology.

Click here to come back to the previous page