Averages Module for Paris Simulator.

Introduction

As the name of the module indicates the Averages Module has been created with the idea of averaging and at the same time simplifying the data generated by Paris Simulator either for reducing output data storage or for physical analysis purposes. Any additional application aligned with the previous stated idea should be included in this module.

Use

In order to use the Average Module functionalities in Paris Simulator, it is only required to add two additional lines to the code.

First, in the Initialization part of the code, after reading the general parameters of Paris Simulator, it is necessary to add: *call ReadAveParameters*

On the other hand, in the output part of Paris Simulator the following line must be added: *call ComputeAverages(itimestep)*

Input file

The input file for the Averages module is called 'inputaverages' this List Script file is necessary to make the module compute any average, in the case that there is no input file Paris Simulator with keep working properly without giving any error, however any average will be computed.

Necessary fields in the Input File

AVERAGES_TO_DO

Until now it's possible to compute 'P' pressure, 'UDarcy' average of u component of velocity and 'Saturation' that is the average of the ratio between the color function and the pore volume.

$$avPressure = \frac{\sum_{i=1}^{i=N} p_i \cdot \chi_{ph} \cdot vol_i}{poreVol \sum_{i=\&}^{i=N} \chi_{ph}}$$

$$UDarcy = \frac{\sum_{i=1}^{i=N} u_i \cdot \chi_{ph} \cdot vol_i}{totalVol}$$

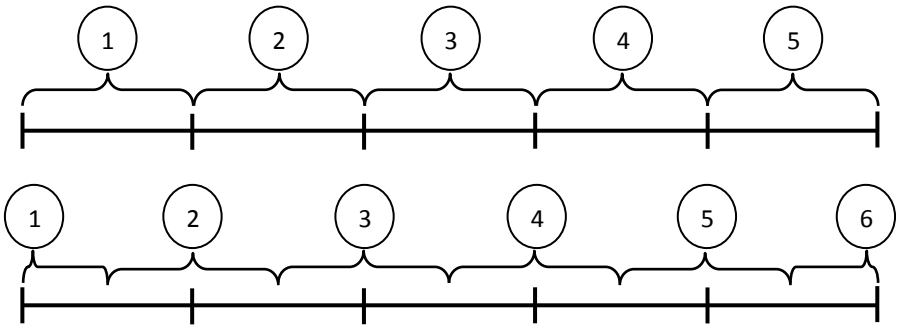
$$Saturation = \frac{\sum_{i=1}^{i=N} \chi_{ph} \cdot vol_i}{poreVol}$$

Where, p is the pressure, u is the x projection of the velocity, χ is the color function, N is the number of samples taken to make the average, and the sub index ph designates the phase which is selected. Ph = 0 (both phaces), ph = 1 (phase 1) and ph = 2 (phase 2).

AV_ON_NODES

This variable determines if the averages are done in the nodes of the new sub-discretized average mesh or if the averages are done on its faces. The idea can be easily understood through the two

following figures, where in the first one the averages are taken in the nodes of the sub-discretized mesh and in the second one, in the faces.



Moreover, Paris Simulator is a three dimensional flow solver, consequently the previous idea is extrapolated to the three dimensional case. Therefore, the user will have to introduce for each wanted average the directions where the averages will be done in the nodes. For example if we want to have the averages centered in the nodes for all three directions it will be necessary to introduce 'xyz', but if we want to have the averages in the faces in x-direction, then in the input is necessary to write 'yz'. The arrangement of the input coordinates must always follow the x-y-z order, so if in the previous case 'zy' is written instead of 'yz' the input of the program won't be valid.

AV_DIV

It is the definition of the averages sub-discretized mesh in the x,y and z directions. It is important to keep in mind that the sub-discretization number of one specific direction must be multiple of the original number of notes of the same direction.

SELECTED_PHASE

That variable indicates from which phase the user wants to obtain the information, the physical averages can be taken from phase 1 '1' phase '2' or from both phases '0'. Logically, when working in single phase simulation the 'selected_phase' number will have to be '0'.

OUTPUT_VTK

Finally, it is possible to choose if the computed averages are exported in VTK format, this option can be very useful when working in very large simulations to reduce the data size of the files produced by the Simulator.

Subroutines

ReadAverageParameters: Subroutine used to read the input parameters of the module, any other subroutine of the module should not be called before reading the parameters since the program will fail.

ComputeAverages: Main subroutine of the Module, it performs all the operations stated in the input file. To compute the averages it uses mainly the subroutine AverageField.

AverageField: This is an important subroutine of the module from the point of view that a good understanding of it can help other developers to easily extend the module. Basically, this subroutine

is used to take an original data set and compute with it a sub-discretized averages mesh. The parameters of the subroutine are:

- ***original_field:*** real array containing the totality of the magnitude that will be averaged.
- ***localaveraged_field:*** real array that contains the averaged field computed by a single processor.
- ***solidFilter:*** boolean that indicates if the original field will be filtered by the solid, in other words the original field will take a null value in the regions that there is solid. (original_field*(1 – solid)).
- ***complementary:*** boolean that indicates if the average is computed using the original field or it is (1 – *original_field*).
- ***colourFilter:*** it can take three values (0, 1, 2) it is similar to the *solidFilter* but using the color function instead of the solid.
- ***bounds:*** integer array of three cells containing the dimensions of the averaged sub-discretized mesh.

Conclusion

It is important to remark that the module is in an early stage of development so, it has not been tested enough to guarantee a perfect operation especially in high parallel environments. Another important recommendation is not to over pass sub-discretized meshes of more than 1E6 nodes, because the processing of the average data is not parallel and it is only done with the front node.