IPVS – Universität Stuttgart
Abteilung Simulation großer Systeme
Prof. Miriam Mehl
Malte Brunn, Amin Totounferoush, Dr.-Ing. Nehzat Emamy

WS 17/18

Numerical Simulation
Exercise sheet 3

Extension: Further boundary conditions and general geometries
Hand in via email to amin.totounferoush@ipvs.uni-stuttgart.de until 19.12.2017, 18:00
Contact for quesions about the framework malte.brunn@ipvs.uni-stuttgart.de

3.1 Note

Note: In case your parallel implementation is erroneous or you are more confident with your serial implementation, you can accomplish the tasks of this exercise sheet in your serial implementation of the flow solver (sheet 1).
In case you would like to use the implementation of the first sheet, you must take the extension for visualization ($\rightsquigarrow$ the computation and output of the stream function and vorticity) from sheet 2 for the actual exercise sheet.

3.2 General boundary conditions

Until now, you could only use the NOSLIP boundary condition with your simulations. Implement the missing boundary conitions SLIP, INFLOW, OUTFLOW and PRESSURE in the first part of this exercise. To do this you should specify an appropriate coding and assigning of different boundary conditions to the corresponding cells of the grid of the simlation domain. This can be done, for example, with using a flag-field, in which the corresponding boundary conditions are registered.

3.3 General geometries

So far, a rectangular domain Ω was necessary for the simulations. Now the simulations will be so extended that arbitrary two-dimensional domains could be used. Extend your implementation as explained in the lecture notes using the Marker-and-Cell approach. If one uses a suitable coding (Bit-coding) the Marker (flags) for fluid, obstacle and boundary could be saved in the same flag-field as for the general boundary conditions.

3.4 Examples

3.4.1 Channel flow

The length and width of the channel should be given as parameters (`xlength`, `ylength`). Also the number of cells in the corresponding directions (`iMax`, `jMax`). Take the length of the channel at least five-times of its width. A pressure-driven flow would be simulated. Therefore, at the left and right boundaries, Dirichlet conditions for the pressure (higher value at the left and lower value at the right) and homogeneous Neumann conditions (=OUTFLOW) for the velocity should be applied. For the upper and lower boundaries, consider the NO-SLIP condition for the velocity, $(u, v)^T = 0$. More information could be found in the lecture notes. Set the pressure difference of the inflow and outflow to $\Delta p = 0.1$. You can find an analytical formula in the lecture notes to evaluate the velocity and validate your results.

3.4.2 Flow over a backward step

For the flow over a backward step, a square-shaped step with half width of the channel is put at the inflow (left end of domain). Figure 9 should describe the geometry clearly. ☺
Here, the boundary conditions are the same as for the channel flow. However, with the additional NO-SLIP boundary condition on the surface of the step. Take the pressure difference of $\Delta p = 0.1$ as well.

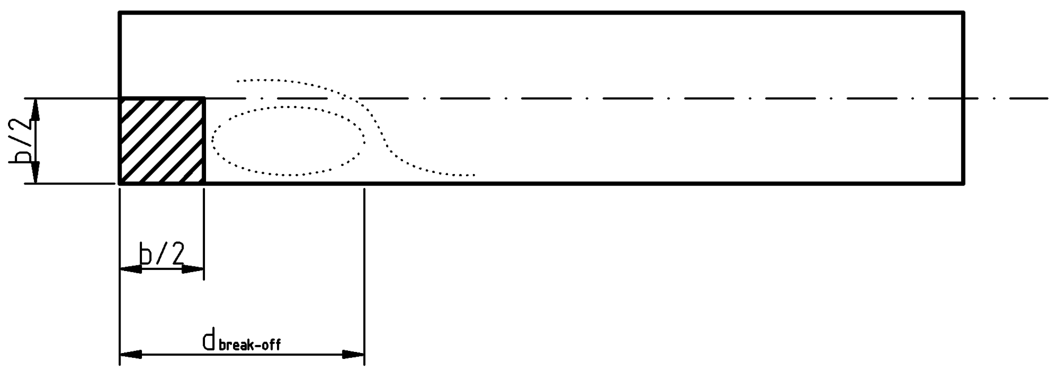

Figure 9: Geometry of the backward step flow

3.4.3 Von Kármán vortex street

In the von Kármán vortex street, the flow comes to an inclined beam, which its length is half of the channel width. The beam is placed in the channel according to figure 10 and rotates with an angle α around its middle point. You can set α as 45° or give it as a parameter together with your geometry data. Take care that to set the correct boundary conditions maximum 2 out of 4 cell faces should be restricted by the fluid. This means that the width of the beam *br* should be at least 2 cells. The boundary conditions are the same as the ones corresponding to the channel flow.

3.5 Input files for geometry and parameters

The required program and files will be introduced in the tutorial and provided via ILIAS. The program creates the geometry data for the three introduced examples in the lecture notes, which are more specified below. Also, it creates the data for different discretizations and sets the corresponding boundary conditions. An appropriate text-based format would be considered that is easy to parse. For example, one can think of using the csv-files, for which rows of the file represent cells of the discretization. The payload in this case would be cell type together with the corresponding boundary conditions, coded as integers for example. Besides the parameters, which you require for accomplishing your geometry data, use the values in the table below. In order that your code is still capable of computing the driven cavity flow, without the requirement

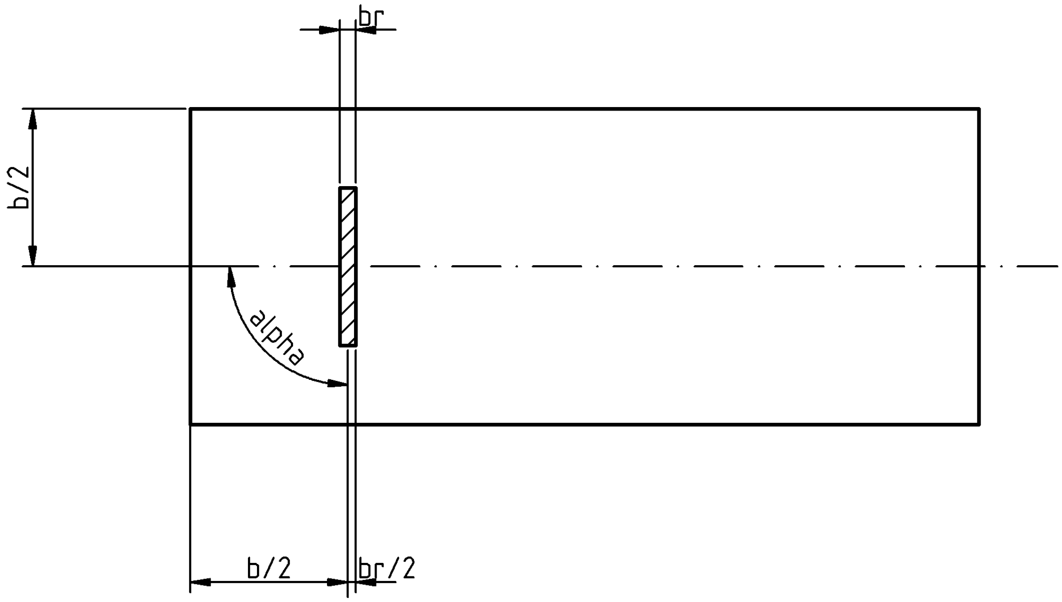

Figure 10: Geometry of the Von Kármán vortex street flow

Geometry	iMax= __	jMax= __	xLength=__	yLength=__
Timing	tEnd=50	tau=0.5	deltaVec=0.5	
Solver	eps=0.001	omega=1.7	alpha=0.9	iterMax=500
Forces and RE	GX=0.0	GY=0.0	RE=10000	$RE_{Stufe} = 100$
Initial values	UI=0.0	VI=0.0	PI=0.0	

Table 4: Parameters for the flow

that you create a geometry file, you need a flag in the geometry file that shows if a geometry should be used (for example `geometry=free`).

3.6 Extended visulaization

Extend your IO-class so that you can trace the particles and visualize streaklines as described in section 10.4 in the lecture notes. You will find an example VTK-file in ILIAS to download.

3.7 Questions

- Compare your solution of the channel flow with the analytical solution. Make a plot of errors for different discretizations.
- Using the vorticity, measure length of the vortex for the flow over a step. Measure at the bottom of the channel (see the schematic sketch of the channel above on the sheet). Plot the length of the vortex for different discretizations.
- Visualize the streamlines, streaklines and pathlines (particle tracing) for the von Kármán vortex street.
- Destroy by purpose the 2-Cells Criteria. What do you observe?

- Imagine that in a three-dimensional fluid domain with curved boundaries, you want to increase the approximation of the domain to two such that respecting to the boundary you refine faster as in the inner of the domain. In the inner of the domain you have $O(N^3)$ cells, where N is the number of cells per each direction. How many cells do you need on the curved boundaries? (tip: two-dimensional manifold)