Prof. Dr. Vincent Heuveline
Dr. Philipp Gerstner

Exercise Sheet 7

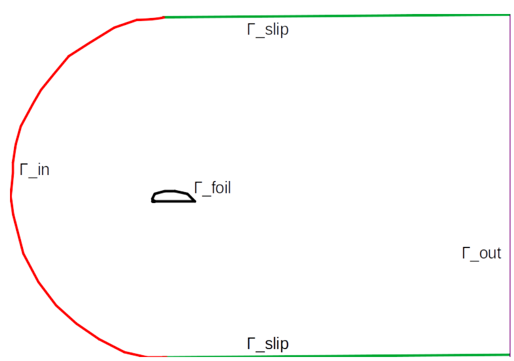
05.06.2021 – 11.06.2021

Computational Fluid Dynamics (Summer Term 2021)

Exercise 1 (Programming). Stokes equations for airfoil (15 Points)
In this exercise you will solve the instationary Stokes equations to compute the flow around an airfoil:

$$\begin{aligned}
\partial_t \mathbf{v} - \nu \Delta \mathbf{v} + \nabla p &= \mathbf{f} \ \text{ in } [0,T] \times \Omega \\
\nabla \cdot \mathbf{v} &= 0 \ \text{ in } [0,T] \times \Omega \\
\mathbf{v} &= (v_x, 0) \ \text{ on } [0,T] \times \Gamma_{in} \cup \Gamma_{slip} \\
\mathbf{v} &= 0 \ \text{ on } [0,T] \times \Gamma_{foil} \\
(\nu \nabla \mathbf{v} - pI)\vec{n} &= 0 \ \text{ on } [0,T] \times \Gamma_{out} \\
\mathbf{v}(0) &= 0 \ \text{ on } \Omega
\end{aligned}$$

with $\nu > 0$, $v_x > 0$, $\mathbf{f} = (0, -1)$ and domain Ω and boundaries given by the figure below.



In this setting, the foil is exposed to the force

$$\mathbf{f}_{foil} = \int_{\Gamma_{foil}} \sigma(\mathbf{v}, p)\vec{n}_{foil} \, \mathrm{dS}$$

with Cauchy stress tensor

$$\sigma(\mathbf{v}, p) = \mu(\nabla \mathbf{v} + \nabla \mathbf{v}^T) - pI, \text{ and } \mu = \nu \text{ (unit density)} ,$$

and $\vec{n}_{foil}$ denotes the unit normal on Γ_{foil} directing into the fluid domain.

a. Modify the evaluate(face, pt_coord, vals) routine of the struct VelocityDirichletBC at the location marked by TODO exercise A in ex6_FoilStokes.h to evaluate the Dirichlet boundary condtions. Note that the boundary condition on Γ_{out} is natural, i.e. it should not be explicitely imposed (see Exercise sheet 3, part d.). The corresponding material numbers for describing the boundary sections are given in the parameter file ex6_FoilStokes.xml.

b. Modify the assembly routines of the class LocalStokesAssembler at the locations marked by TODO exercise B in ex6_FoilStokes.h to compute the matrix and vector corresponding to the variational formulation of the time-discrete system (see exercise sheet 5). A θ scheme should be used for time discretization.

c. Modify the assembly routines of the class ForceIntegral at the locations marked by TODO exercise C in ex6_FoilStokes.h to compute the boundary integral $\mathbf{f}_{foil}$. Note that the global assembler loops over all facets on the boundary. Use the material number of Γ_{foil} to restrict the integration domain appropriately. The member variable force_dir_ determines which component of $\mathbf{f}_{foil}$ should be computed. (0: drag, 1: lift).

d. Modify the routine compute_forces of the class FoilStokes at the location marked by TODO exercise D in ex6_FoilStokes.cc to compute drag and lift component of the boundary integral $\mathbf{f}_{foil}$.

Hint: It might be helpful to have a close look on the routine compute_L2_divergence.

e. Run the code on up to 4 MPI processes for $\nu = 1$, $f_z = -1$, $v_X = 1$, $T = 10$, $k = 0.5$, $\theta = 0.5$. Note that all parameters are defined in the parameter file ex6_FoilStokes.xml and can be changed without recompilation.

The program output is given by a series of pvtu files ex6_solution0_n.pvtu with n denoting the time step and a csv file pp_values.csv. Each row in the csv file corresponds to one time step n and is of the form

$$\begin{pmatrix} t_n & \|\nabla \cdot \mathbf{v}(t_n)\|_2 & \mathbf{f}_{foil}[0] & \mathbf{f}_{foil}[1] & \mathbf{f}_{foil}[1]\mathbf{f}_{foil}[0]^{-1} & \end{pmatrix}$$

Plot these quantities over time t_n and submit it.

You can create an animation with paraview, if you load the file ex6_solution0_...pvtu and click on the play button.

f. The mesh file is created by the python script ex6_FoilStokes/mesh_generation/create_channel_mesh.py. You can run it by the terminal command python3 create_channel_mesh.py inside the directory mesh_generation. Afterwards, you have to run ccmake ../ inside the Hiflow build directory in order to copy the newly created mesh channel.inp into the example build directory.

In this script, the variable angle (in radians, $\in [-\pi, \pi]$) sets the angle of attack for the foil.

Vary the angle over the interval $[-\pi, \pi]$ and plot the resulting drag, lift and lift-to-grad ratio (at time $t = 10$) over the angle.

The corresponding code framework is ex6_FoilStokes. Don't forget to the modify the file exercises/CMakeLists.txt accordingly, see the first exercise sheet for details.

Submission: until 11.06.2021, 11 am, moodle upload