Interdisziplinäres Zentrum
für Wissenschaftliches Rechnen (IWR)
Ruprecht–Karls–Universität Heidelberg

Prof. Dr. Vincent Heuveline
Dr. Philipp Gerstner

Exercise Sheet 5

21.05.2021 – 28.05.2021

Computational Fluid Dynamics (Summer Term 2021)

Exercise 1 (Programming). Instationary Lid-driven Cavity (15 Points)
In this exercise you will solve the instationary Stokes equations:

$$\begin{aligned} \partial_t \mathbf{v} - \nu \Delta \mathbf{v} + \nabla p &= \mathbf{f} \ \text{ in } [0,T] \times \Omega \\ \nabla \cdot \mathbf{v} &= 0 \ \text{ in } [0,T] \times \Omega \\ \mathbf{v} &= \mathbf{v}_D \ \text{ on } [0,T] \times \partial\Omega \\ \mathbf{v}(0) &= 0 \ \text{ on } \Omega \end{aligned}$$

with $\Omega = [0,1] \times [0,1]$, $\partial\Omega = \Gamma_D + \Gamma_0$, with $\Gamma_D = [0,1] \times \{1\}$ and $\Gamma_0 = \partial\Omega \setminus \Gamma_D$. Further,

$$\mathbf{f}(t,x,y) = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$$

$$\mathbf{v}_D(t,x,y) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot \sin(10t) \cdot \begin{cases} 0, & (x,y) \in \Gamma_0 \\ v_X, & (x,y) \in \Gamma_D, x \in [\epsilon, 1-\epsilon] \\ \frac{x}{\epsilon} v_X, & (x,y) \in \Gamma_D, x \in [0, \epsilon] \\ \frac{1-\epsilon-x}{\epsilon} v_X + v_X, & (x,y) \in \Gamma_D, x \in [1-\epsilon, 1] \end{cases}$$

for some constants $\epsilon, v_X \geq 0$, $\nu > 0$.

By using a simple θ-scheme, the above system can be discretized in time to obtain a sequence of stationary PDEs for $n = 1, \ldots, N$:

$$\begin{aligned} \frac{1}{k}(\mathbf{v}_n - \mathbf{v}_{n-1}) - \nu \Delta(\theta \mathbf{v}_n + (1-\theta)\mathbf{v}_{n-1}) + \nabla p_n &= \mathbf{f}(t_n) \ \text{ in } \Omega \\ \nabla \cdot \mathbf{v}_n &= 0 \ \text{ in } \Omega \\ \mathbf{v}_n &= \mathbf{v}_D(t_n) \ \text{ on } \partial\Omega \\ \mathbf{v}_0 &= 0 \end{aligned} \tag{1}$$

Here, $(\mathbf{v}_n, p_n)$ denotes the approximation to $(\mathbf{v}, p)(t_n)$ at time $t_n = nk$ with time step size $k > 0$ and $\theta \in [0,1]$. Note that $\theta = 0$ corresponds to the forward Euler method, $\theta = 1$ corresponds to the backward Euler method and $\theta = 0.5$ corresponds to the Crank-Nicolson method. The time-discretized system has to be solved for $(\mathbf{v}_n, p_n)$ while $\mathbf{v}_{n-1}$ is used from the previous time step.

a. Modify the evaluate(face, pt_coord, vals) routine of the struct VelocityDirichletBC at the location marked by TODO exercise A in ex4_InstationaryStokes.h to evaluate the Dirichlet function $\mathbf{v}_D$.

b. Modify the assembly routines of the class LocalStokesAssembler at the location marked by TODO exercise B in ex4_InstationaryStokes.h to compute the matrix and vector corresponding to the variational formulation of the time-discrete system (1).

c. Modify the time_loop routine of the class CavityStokes at the location marked by TODO exercise C in ex4_InstationaryStokes.cc. This routine should perform a loop over all time steps $n = 1, \dots, N = \frac{T}{k}$. Within each loop iteration, the boundary conditions have to be computed, the system has to be assembled and to be solved.

d. Modify the assembly routine of the class DivergenceIntegral at the location marked by TODO exercise D in ex4_InstationaryStokes.h to compute the scalar quantity $\int_\Omega |\nabla \cdot \mathbf{v}|^2 \mathrm{d}x$.

Hint: It might be helpful to have a close look on the class PressureIntegral in the file ex4_InstationaryStokes.h, which is used to compute $\int_\Omega p \, \mathrm{d}x$.

e. Run the code on up to 8 MPI processes for $\nu = 1$, $f_z = 1$, $\epsilon = 10^{-1}$, $v_X = 1$, $T = 1$, $k = 0.01$, $\theta = 0.5$. Note that all parameters are defined in the parameter file ex4_InstationaryStokes.xml and can be changed without recompilation.

The program output is given by a series of pvtu files ex4_solution6_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

$$\big(t_n \quad \|\nabla \cdot \mathbf{v}(t_n)\|_2 \quad \mathbf{v}_x(t_n, x^{(1)}) \quad \mathbf{v}_y(t_n, x^{(1)}) \quad p(t_n, x^{(1)}) \quad \mathbf{v}_x(t_n, x^{(2)}) \quad \mathbf{v}_y(t_n, x^{(2)}) \quad p(t_n, x^{(2)})\big)$$

Plot these quantities over time t_n and submit it.

You can create an animation with paraview, if you load the file ex4_solution6_....pvtu and click on the play button.

The corresponding code framework is ex4_InstationaryStokes. Don't forget to the modify the file exercises/CMakeLists.txt accordingly, see the first exercise sheet for details.

Hint: It might be helpful to have a look on the solution of the previous exercise sheet.

Submission: until 28.05.2021, 11 am, moodle upload