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

Prof. Dr. Vincent Heuveline
Dr. Philipp Gerstner

Programming Sheet 1

16.04.2021 – 23.04.2021

Computational Fluid Dynamics (Summer Term 2021)

Important Notes:
The programming exercises have to be solved within the provided C++ open-source package HiFlow^3. This package can be downloaded from the moodle page of the lecture. In order to compile HiFlow^3 on a Linux operating system, follow the instructions in hiflow3/INSTALL.txt of the unzipped directory.
For each programming exercise, we provide a framework in the form of a zip file exXY_NAME.zip on moodle. Within this framework, you have to solve the exercise.
In order to compile and run your solution, copy exXY_NAME.zip into the directory hiflow3/exercises, unzip it and add the line exXY_NAME to the file hiflow3/exercises/CMakeLists.txt at line 83, below the entry 'data'. Afterwards, you have to reconfigure and recompile HiFlow^3 according to the instructions hiflow3/INSTALL.txt.
Please add detailed comments to your code. We only evaluate codes, which compile without any bugs and do not exhibit run time errors. *We do not debug your code!*
After you have put your solution into the provided framework, create an zip or tar archive (only of the directory hiflow3/exercises/exXY_NAME) and upload it to moodle. In addition to your code, this archive should contain a PDF file, in which the terminal output (and plots) created by your program are stated.
We test your solution on Ubuntu 20.04. If you want to ensure that your solution compiles successfully on our computers, you should use this OS (possibly installing it on a virtual machine, e.g. VirtualBox).

Exercise 1 (Programming). Solving Poisson Equation (Points)
In this exercise, you learn how to solve the Poisson equation

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \\ u &= g \quad \text{on } \partial\Omega \end{aligned}$$

by using the Finite Element Method provided by HiFlow^3. Since this is the first programming exercise, the corresponding directory ex0_poisson is already included in the entire HiFlow^3 zip-archive, provided on moodle.

a. Compile and run the provided exercise on 4 cpu cores by following the instructions in hiflow3/INSTALL.txt. In this example, the right-hand side f is chosen such that an analytical solution u is known and $g = 0$. Note down the L^2 and H^1 error on each level of mesh refinement.

b. Change the code such that on each level of refinement, the number of CG iterations is printed on the terminal. Note down these iteration numbers.

c. Change the polynomial degree of the underlying finite element from 1 to 2. Note down the new L^2 and H^1 error on each level, as well as the number of CG iterations.

d. Change the code such that the following PDE is solved (with linear elements):

$$
\begin{aligned}
-0.01\Delta u + u &= 1 \quad \text{in } \Omega \\
u &= 2 \quad \text{on } \partial\Omega
\end{aligned}
$$

Note down the corresponding CG iteration numbers and provide a screenshot of the visualization (use Paraview for reading the output .pvtu files) for the largest level of refinement.
Hint: Uncomment line 28 (#define SUBEX_D) in ex0_poisson.h in order to enable the changes you've made for solving this sub-exercise.