Prof. Dr. Ulrich Rüde
Dominik Bartuschat
Kristina Pickl
Christoph Rettinger

Summer Term
2015

Simulation and Scientific Computing 2
Assignment 1

Exercise 1 (*Elliptic PDE with Multigrid*)

Organization Details

1. **Your Tutors:** Your tutors for the exercises are Dominik Bartuschat, Kristina Pickl and Christoph Rettinger. You can find their offices and email addresses on the official web page of the chair for System Simulation:

   http://www10.informatik.uni-erlangen.de

2. **Programming exercises:** The programming language for the implementation of the programming exercises is C++.

3. **Assignment sheets:** The assignment sheets and further information concerning this lecture will be available via the StudOn course. Also the submission of the solutions has to be done in StudOn.

4. **Team work:** In order to encourage team work we require you to form teams of three people submitting their solution together. The team formation is carried out in individual responsibility of the students. Submissions of groups with less than three people will **not** be accepted and corrected!

5. **Credits:** Each assignment will be awarded with either 2, 1 or 0 points. In order to successfully participate in the exercise classes, which is necessary for an admission to the exam, you have to acquire **at least 6 points**. This also means that you are allowed to receive zero points only once! However, every assignment will feature a bonus task that will grant you additional 0.5 points upon successful completion.

6. **Seminar:** In the context of the SiWiR 2 seminar, we have planned several presentation sessions. Each participant is required to give a presentation and to attend **every** presentation session in order to successfully participate in the seminar.

7. **CIP-Pool Access and Accounts:** Access to our computer exercise room 00.131 is possible during the computer exercise classes but also Monday to Friday between 8:00–17:00! After applying for an user account, you will receive a temporary password which you should change when logging in the first time via the command `passwd`. If you have problems with your accounts please write an email to

   `cs10-support@fau.de`

8. **Remote Access:** Remote access to our computers is possible via the *secure shell (ssh)*. For the login, you first have to either access a computer from the university network, or log in via VPN, from where you can access the LSS CIP pool machines *i10cip*{*1* . . . *12*}.

9. **General Remark:** It should go without saying that every team should provide us with their own solution, programmed by themselves. If a team ignores this, they will receive zero points for their assignment.

Tasks

1. Your task is to implement a multigrid solver that calculates the approximate solution for the following elliptic partial differential equation (PDE):

$$-\Delta u(x,y) = f(x,y) \qquad (x,y) \text{ on } \Omega \in \mathbb{R}^2 \tag{1}$$

for the domain $\Omega = (0,1) \times (0,1)$. Use Dirichlet boundary conditions (BCs)

$$u(x,y) = g(x,y) \qquad (x,y) \text{ on } \partial\Omega. \tag{2}$$

The Dirichlet BCs g and the right-hand side f are defined as follows:

$$\begin{aligned} g(x,y) &= \sin(\pi x)\sinh(\pi y), \\ f(x,y) &= 0. \end{aligned}$$

For the solution of this PDE choose a fitting finite difference discretization with a mesh size h such that your vertex centered square grid consists of $(2^l+1) \times (2^l+1)$ points (including boundaries). As solver choose the following multigrid method: Implement a V(2,1)-cycle using l levels with $u = 0$ as initial guess inside the domain. Use full weighting for restriction, bi-linear interpolation for prolongation, and (red-black) Gauss-Seidel for relaxation.

In order to increase the efficiency of your calculations, think about an optimal implementation in terms of data layout and update strategy.

We recommend a modular implementation of the different parts of your program in order to be easily adapted to other problems.

2. Your program should be able to be executed in the following form:

```
./mgsolve l n
```

where l is the number of levels and n the number of V-cycles.

3. After each V-cycle, the program should output the discrete L_2-norm of the residuum $\mathbf{r}$ and the residuum based estimation of the convergence rate q on the screen. They are given as:

$$\|\mathbf{r}\|_2 = \sqrt{\frac{1}{|\Omega_h|}\sum_{i=1}^{|\Omega_h|} r_i^2}, \quad q = \frac{\|\mathbf{r}^{(t+1)}\|_2}{\|\mathbf{r}^{(t)}\|_2}$$

For more information, see pages 34 and 46 in the SiWiR 1 lecture notes from Prof. Pflaum.

4. Perform a fixed number of V-cycles n (given on the command line) and measure the wall clock time it takes (e.g. using `gettimeofday()`). Print the time on the screen and write the computed solution to a file named `solution.txt`. Choose a file format, which can be visualized by `gnuplot`[1] such that the x-direction of the domain corresponds to the first dimension of the gnuplot output.

5. Finally, measure the error $e_h = \|u_h - I^h u^*\|_2$ in the L_2 norm for the grid sizes $h = \{\frac{1}{8}, \frac{1}{16}, \ldots, \frac{1}{256}\}$. Here, u^* is the exact solution, I^h is an operator that restricts the solution to the grid points at the finest level and u_h is the approximate solution computed by the MG solver.
   Show the error vs. the grid sizes as table and graph in a pdf file and describe the measurements.

6. Please hand in your solution to this exercise until Thursday, **May 21, 2015** at 3:00 am! Make sure the following requirements are met:

   (a) The program should be compilable with a Makefile.

   (b) The program should compile without errors or warnings with the following g++ compiler flags:

   ```
   -O3 -Wall -Winline -Wshadow -ansi
   ```

   (c) The program should be callable as specified above and output the required values.

   (d) The solution should contain well commented source files, a fitting Makefile that satisfies all the conditions specified above and instructions how to use your program (e.g. in form of a short README file).

   (e) When submitting the solution, remove all temporary files from your project directory and pack it using the following command:

   ```
   tar -cjf ex01.tar.bz2 ex01/
   ```

   where `ex01/` is the directory containing your solution. Then submit the archive on StudOn as demonstrated in the exercise class.

Credits

In this assignment the credits are awarded in the following way:

1. Up to 2 points are awarded if your program correctly performs the above tasks and fulfills all of the above requirements. Submissions with compile errors will lead to zero points!

2. Bonus task: Extra 0.5 points are awarded if additionally your multigrid solver is extended such that it computes the approximate solution of the PDE 1 on the corresponding domain for the following conditions. Use Neumann BCs on the vertical domain walls

$$\frac{\partial u}{\partial \vec{n}}(0, y) = \frac{\partial u}{\partial \vec{n}}(1, y) = -1,$$

   and Dirichlet BCs on the horizontal walls:

$$u(x, 0) = u(x, 1) = x\,(1 - x)\,.$$

   As right-hand side, use $f(x, y) = 2$.
   As for the main task, measure the L_2-norm of the error for the different grid sizes. Extend the table and graph by that measurements and describe the difference to the error from the main task.

References

[1] http://www.gnuplot.info/docs/gnuplot.pdf