Summer Term
2015

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

Simulation and Scientific Computing 2
Seminar

Organization Details

1. **General:** For the seminar, each group of **two** students should work on the topic below, and present the corresponding implementation and the results in a seminar presentation. The task is to optimize the performance of your multigrid solver for the problem specified below. It is open ended and meant to give you room to be creative.

2. **Presentation:** The result should be presented in one of the seminar sessions on 2 slides which should be designed like a poster (i.e., concise and catchy). The file format must be PDF! Each group has 3 minutes to present the results.

3. **Submission:** Please hand in your presentation and the final solution until Friday, June 5^{th} 2015, 4 p.m. which is a **HARD** deadline. More details are specified below.

4. **Date:** The seminar will take place on June, 8^{th} and June, 12^{nd} 2015 as specified on the schedule.

5. **Successful participation:** Each student is required to give a part of his/her group's presentation and to attend **every** presentation session in order to successfully pass the seminar. Note that passing the seminar is a prerequisite to pass the course.

Tasks

1. Your task is to modify the multigrid (MG) solver from assignment 1 to a MG solver that computes the approximate solution for the following elliptic partial differential equation (PDE):

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

for the domain $\Omega = (-1,1) \times (-1,1) \setminus [0,1] \times \{0\}$ with 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:

$$g(x,y) = r^{1/2}\sin(\frac{1}{2}\varphi), \tag{3}$$

$$f(x,y) = 0, \tag{4}$$

where (r,φ) are the polar coordinates centered at the origin $(0,0)$.

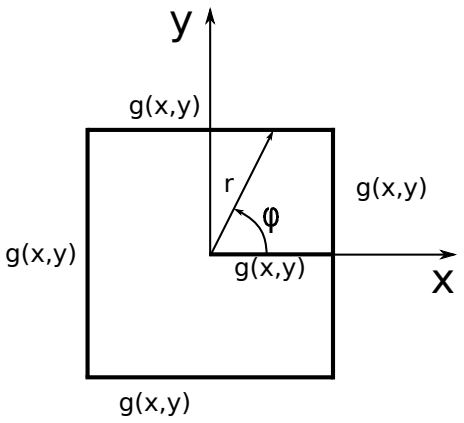

Figure 1: Slit domain.

For the solution of this PDE choose a suitable finite difference discretization with a mesh size $h = 2/n$ such that your slit domain grid consists of $(n+1) \times (n+1) = (2^l+1) \times (2^l+1)$ points (including boundaries).

As solver choose a suitable MG variant with $u = 0$ as initial guess inside the domain.

Optimize your code and choose algorithmic variants, such that your code becomes fast. The goal is to construct the most efficient solver for (1) with BCs (2) for problem size $l = 11$. The L_2-error which is calculated by your results must be smaller than $9.18 \cdot 10^{-5}$.

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

```
./mgsolve l,
```

where l is the number of levels.

3. We will provide a tool that compiles your code, runs it for $l = 11$ and ranks your runtime together with an identifier (denoted in the following by 'Your_Alias') on a separate webpage[1]. Furthermore, it checks the error of your computed solution. Additionally, a check is done on your initial guess file. For that purpose, we need an output file in the format specified below.

   The tool can be executed on any *i10cip*$\{1 \ldots 12\}$ machine at LSS by entering the shell command `verifier`. You can then specify the tarball with your solution that is automatically unpacked and compiled by means of the Makefile you provide.

4. Your program must create two files: `init.dat` with the initial guess plus boundary conditions and `solution.dat` with the solution. The files must contain the approximation u_{ij} of PDE (1) at the coordinates (x_i, y_j) of the grid points in row-major format, respectively:

$$\begin{array}{ccc} x_0 & y_0 & u_{00} \\ x_1 & y_0 & u_{10} \\ \vdots & \vdots & \vdots \\ x_n & y_0 & u_{n0} \\ x_0 & y_1 & u_{01} \\ \vdots & \vdots & \vdots \\ x_n & y_n & u_{nn} \end{array}$$

5. Your main function has to contain the following fragment:

```
std::cout<<"Your Alias: "<<your_alias<<std::endl;
struct timeval t0, t;
gettimeofday(&t0, NULL);
solveMG( ... );
gettimeofday(&t, NULL);
std::cout << "Wall clock time of MG execution: " <<
((int64_t)(t.tv_sec - t0.tv_sec) * (int64_t)1000000 +
(int64_t)t.tv_usec - (int64_t)t0.tv_usec) * 1e-3
<< " ms" << std::endl;
```

   Where `solveMG( ... )` is the call of your multigrid solver. The other code is required for the ranking functionality and has to be contained in your code in the exact way as specified. In case of MPI-parallel execution, make sure that the overall time among all processes is measured (i.e., the time at which the first process starts executing `solveMG( ... )` MG until the last process has finished).

6. Make sure the following requirements are met when submitting your solution:

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

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

[1] `https://www10.informatik.uni-erlangen.de/Teaching/Courses/SS2015/SiWiR2/seminar.shtml`

```
-O3 -Wall -Winline -Wshadow -std=c++11
```

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

(d) The solution should contain well commented source files and a fitting Makefile that satisfies all the conditions specified above.

(e) Do not forget to add the presentation slides!

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

```
tar -cjf seminar.tar.bz2 seminar/
```

where `seminar/` is the directory containing your files. Then upload your solution to StudOn as a team submission.

Challenge

There will be a performance challenge, in which the three groups with the fastest solutions for a given problem size and accuracy of the solution will be awarded prizes. The fastest solutions will be measured for levels $l = 11$ after verifying the correct L_2-error condition.
The results on the webpage at the submission deadline are considered for the ranking. Groups that violate any of the conditions above will be excluded from the challenge.