Prof. Dr. Peter Bastian, Linus Seelinger Submission date: 17 May 2018

IWR, Universität Heidelberg

Exercise 1 *Updating dune-parsolve*

As we update the *dune-parsolve* module during the semester, you need to get the current state before starting to solve a new programming exercise:

- Navigate to your *dune-parsolve* directory in a terminal
- Execute the commands

```
git stash
git pull
git stash pop
```

These *git* commands temporarily move your local changes to the stash, download the updates and apply your changes to the new version again.

Note: In order to avoid merge conflicts during the update procedure, we highly recommend that you do not modify files that belong to exercises that have not officially started yet.

(0 Points)

Exercise 2 *Adjacency matrix*

Let I be an index set and $R \subset I \times I$ a symmetric and reflexive relation with

$$\max_i |\{j : (i,j) \in R\}| \leq K$$

Define the associated Matrix E via

$$(E)_{ij} := \begin{cases} 1 & (i,j) \in R \\ 0 & \text{sonst} \end{cases}$$

Show that $||E||_2 \leq K$. **(5 Points)**

Exercise 3 *Optimal parameter for the Richardson iteration*

Let A be a symmetric and positive definite Matrix. The Richardson iteration is given by

$$x_{k+1} = x_k + \omega\,(b - Ax_k)\,.$$

Assume that the minimal and maximal eigenvalue λ_{min} and λ_{max} of A are known.

1. How can the eigenvalues of the iteration matrix be bounded?
2. Determine the optimal relaxation parameter ω_{opt} and the corresponding spectral radius.

(5 Points)

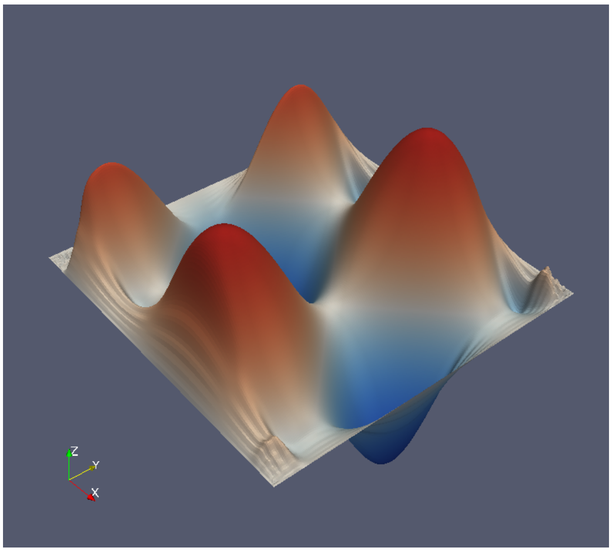
Exercise 4 *Dependence of linear solver convergence on the initial data*

In this exercise we want to consider the Laplace equation with homogeneous Dirichlet boundary conditions

$$
\begin{aligned}
-\Delta u &= 0 && \text{in } \Omega = (0,1)^2 \subset \mathbb{R}^2, \\
u &= 0 && \text{auf } \partial\Omega.
\end{aligned}
$$

The solution obviously reads $u = 0$. We are going to make use of it in order to study the convergence behaviour of the linear solvers by specifying different starting vectors $\neq 0$.

The code skeleton for the exercise is provided in the file `uebungen/uebung02/uebung02.cc`. The program solves the Laplace equation with the starting vector $u_1(x, y)$ defined below doing one Jacobi iteration. The program has been already prepared such that it creates VTK-outputs of the iterates calculated by the linear solvers in order to visualize the distribution of the error with ParaView.



In the file `src/tutorial/istl.cc` of the `dune-parsolve` module you will find some more linear solver and preconditioner objects created. In general, linear iterative methods such as the Jacobi and Gauss-Seidel iteration can be used as stand-alone iterations or as preconditioners. Using them as stand-alone iterations is accomplished by the class `LoopSolver` which simply applies the method in every iteration and checks the convergence criterion.

Modify the file `uebung02.cc` such that it creates VTK-outputs for the following combination of linear solvers, starting vectors and number of iterations:

- Solvers: Jacobi, Gauss-Seidel, Steepest Descent and Conjugate Gradient
- Starting vectors: Given by the functions

$$
\begin{aligned}
u_1(x, y) &= 1, \\
u_2(x, y) &= cos(10x) + sin(10y), \\
u_3(x, y) &= cos(100x) + sin(100y)
\end{aligned}
$$

- Number of iterations: 1, 10 or 100 iterations

Examine the convergence rate of the linear solvers. Does the initial data have an influence on the convergence? **(10 Points)**