Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
Carsten Uphoff, Nikola Tchipev

SS 2018
Sheet 3P
04.05.2018

# Scientific Computing II

## Multigrid Methods

### Exercise 1: Multigrid for Convection-Diffusion

Consider the one-dimensional convection-diffusion equation

$$-\epsilon u_{xx} + u_x = f(x) \tag{1}$$

and its discrete representation

$$\frac{\epsilon}{h^2}(-u_{n-1} + 2u_n - u_{n+1}) + \frac{1}{h}(u_n - u_{n-1}) = f_n, \quad n = 1, ..., N-1, \tag{2}$$

with Dirichlet conditions $u_0 = u_N = 0$ and $\epsilon \geq 0$. In this exercise, we assume a right hand side $f(x) = 2x - (2\epsilon + 1)$ and a respective discrete representation $f_n = 2(nh) - (2\epsilon + 1)$. The number of grid points is given by $N := 2^l$ plus one, i.e. $N + 1$ points with $l \in \mathbb{N}$, and the mesh size is $h := 1/N$, respectively. In the following, we want to solve the discrete system by a multigrid method implemented in Matlab. A Gauss-Seidel smoother is provided on the webpage ($\rightarrow$ smooth.m).

(a) Show that the analytical solution to the problem from Eq. (1) is given by $u(x) = x^2 - x$.

(b) Implement a function `restrict(stencil,residual)` which constructs a matrix-dependent restriction (see the lecture slides). The vector `stencil`$= [s_l \;\; s_c \;\; s_r] \in \mathbb{R}^3$ corresponds to the three-point stencil on the fine grid, `residual` to the residual vector of the current grid level. The function should return the coarsened residual vector.

(c) Implement a function `interpolate(stencil,eCoarse)` which interpolates the error `eCoarse` from the coarse grid to the fine grid. The interpolation should be carried out based on the matrix-dependent interpolation rule as presented in the multigrid slides. The stencil `stencil` should correspond to the three-point stencil on the fine grid. The function should return the interpolated error vector.

(d) Implement a function `computeCoarseGridStencil(stencil)` which computes and returns the coarse grid stencil for a given fine grid stencil `stencil`. The function should again be based on the matrix-dependent restriction/ prolongation from the multigrid slides.

(e) Put the steps together into a function `wCycle(u,rhs,level,stencil)` which implements the w-cycle algorithm. The vector `u` corresponds to the current solution and

should also be returned by this function. The right hand side of the linear problem is given by `rhs`, `level` corresponds to the current grid level and `stencil` denotes the three-point stencil for the current grid level.
Test your implementation by

- using a two-grid algorithm (e.g. remove the recursive call in wCycle(...) and use a sufficient number of smoothing steps on the coarse grid)
- comparing your results with results obtained from pure Gauss-Seidel solving
- comparing your results with the exact solution.

(f) Use the w-cycle implementation to solve the discrete system from Eq. (2) for $\epsilon = 1, 0.1, 0.01$. The simulation should stop when the maximum norm of the residual drops below a tolerance $tol = 1e - 10$. Measure the number of w-cycle iterations for each simulation setup and initial grid levels $l = 4, 6, 8, 10$. Besides, plot the numerical solution as well as the error $e = u - u^{analytic}$ where $u^{analytic}$ denotes the discrete representation of $u(x) = x^2 - x$. What do you observe?

| | $l = 4$ | $l = 6$ | $l = 8$ | $l = 10$ |
|---|---|---|---|---|
| $\epsilon = 1$ | 7 | 7 | 7 | 7 |
| $\epsilon = 0.1$ | 6 | 7 | 7 | 7 |
| $\epsilon = 0.01$ | 5 | 7 | 7 | 7 |

Table 1: Number of iterations of the w-cycle algorithm for different grid resolutions and diffusion levels $\epsilon$.

**Solution:**

(a) Differentiating the function $u(x) = x^2 - x$ and inserting the results for the first and second derivative into the convection-diffusion equation from Eq. (1) shows the respective statement.

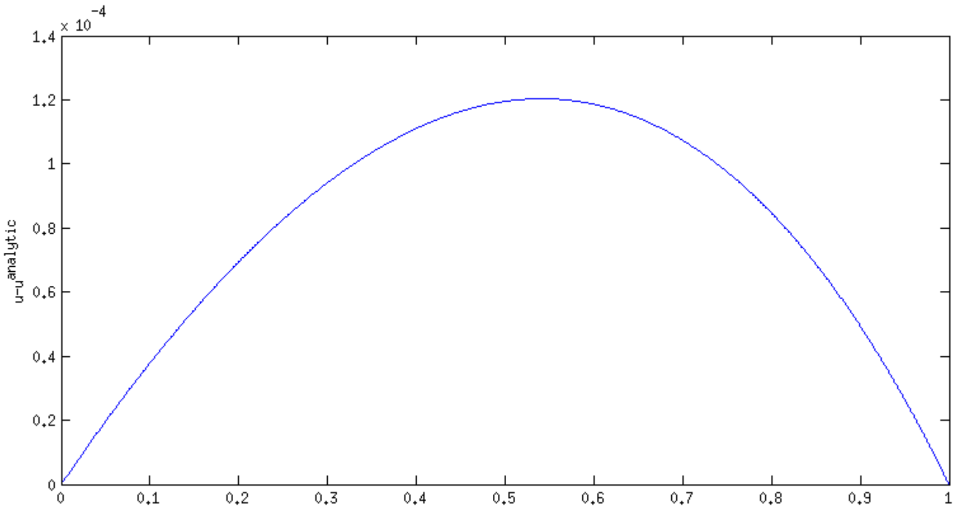
(b) See restrict.m.

(c) See interpolate.m.

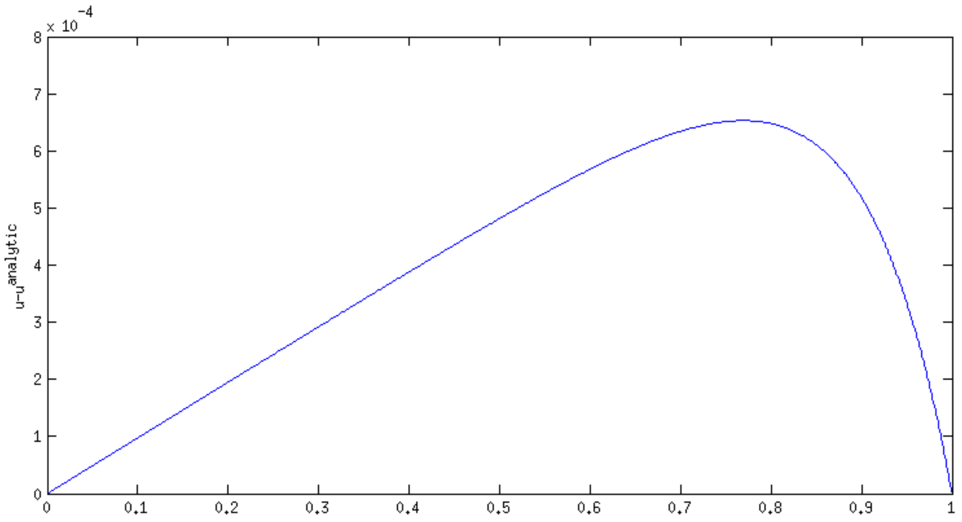
(d) See computeCoarseGridStencil.m.

(e) See wCycle.m.

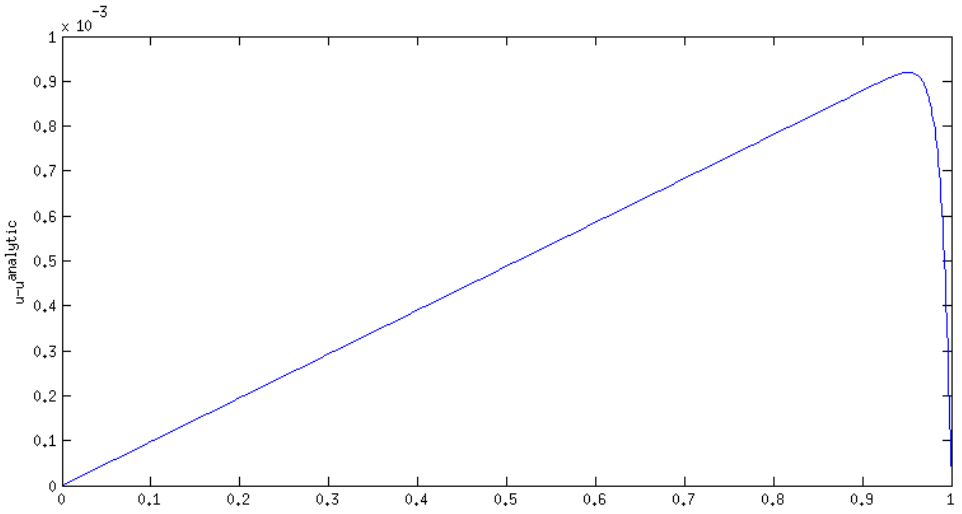
(f) See conDiff.m. The results for different values of $\epsilon$ and grid levels $l$ are shown in Tab. 1 ($tol = 1e - 10$). The number of iterations stays—similar to the multigrid scheme for the Poisson equation from the previous worksheet—approx. constant, even for higher grid resolutions.
Exemplary error plots $u - u^{analytic}$ are shown in Fig. 1. The error increases towards the middle of the domain, but it is shifted to the right for increasing values of $\epsilon$, that is for decreasing diffusion and increasing convection. Since we do not have any error on the Dirichlet boundaries, the error must be bigger in the inner part of the domain. Besides, due to the convection, the error is "transported" and increased from left to right.



(a)



(b)



(c)

Figure 1: Plot of error $u - u^{analytic}$ over space. (a) $\epsilon = 1$, $l = 10$. (b) $\epsilon = 0.1$, $l = 10$. (c) $\epsilon = 0.01$, $l = 10$.