Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
Dipl.-Tech. Math. P. Neumann
Dipl.-Inf. W. Eckhardt

SS 2013
Sheet 4
13.05.2013

Scientific Computing II

Multigrid Methods

Exercise 5: 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 vanishing right hand side $f(x) = 0$. The solution of the differential equation is thus given by $u(x) = 0$.

(a) Show that for $\epsilon << 1$, the Jacobi method applied to Eq. (2) converges very slowly whereas the Gauss-Seidel method is expected to converge very fast. The Gauss-Seidel method is used from left to right, that is the updated values $u_1, ..., u_{n-1}$ are immediately used to compute the new value at u_n.
Hint: insert an error frequency $u_n = \sin(\pi k(nh))$ into the respective iterative scheme and analyse the behaviour for $\epsilon \to 0$.

(b) In the following programming exercise 4, we want to use a matrix-dependent restriction/ interpolation technique to set up a multigrid for solving Eq. (2), cf. slide 29 of the lecture slides (multigrid.pdf). Show that the Galerkin coarsening $A_{2h} := R_h^{2h} A_h P_{2h}^h$ applied to an arbitrary three-point stencil $[s_l \quad s_c \quad s_r]$ with $s_c = -(s_l + s_r)$ on the fine grid yields the coarse grid stencil

$$\frac{1}{s_c}[-s_l^2 \quad (s_l^2 + s_r^2) \quad -s_r^2]. \tag{3}$$

Programming Exercise 4: Multigrid for Convection-Diffusion

Consider the one-dimensional convection-diffusion equation from Eq. (1) and its discrete counterpart Eq. (2). 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 $N := 2^l + 1, l \in \mathbb{N}$, and the mesh size is $h := 1/N$, respectively. In the following, we want to solve the discrete system from Eq. (2) 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 slide 29 of multigrid.pdf from the lecture). 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 (cf. slide 29). 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?