Institut für Informatik — TU München
Scientific **C**omputing in **C**omputer **S**cience
K. Rahnema, M. Schreiber

WS 2012

Tutorial (Advanced Programming) Worksheet 9:

Assignment 1: Multigrid

Implement the components of a simple multigrid solver:

1) Implement a Gauss-Seidel smoother performing two Gauss-Seidel iterations as a function of the current approximation of the solution, the right hand side, and N_x and N_y on the respective grid level.
2) Implement a function computing the residual as a function of the current approximation of the solution, the right hand side, N_x and N_y.
3) Implement the restriction (injection) as a function of the fine grid function and the grid resolutions N_x and N_y of the coarse grid.
   Hint: Injection means taking every second point.
4) Implement the interpolation (bilinear) as a function of the coarse grid function and the grid resolutions N_x and N_y of the fine grid.

Implement a function performing one iteration of a multigrid solver (v-cycle) as a function of the current approximation of the solution, the right hand side, and the grid resolutions N_x and N_y. The output of the function is the new approximation of the solution and the residual norm after the iteration.
Hint: recursivity!!!

Implement a multigrid solver as a function of the right hand side and the grid resolutions N_x and N_y. The output of the function is the approximate solution. Iterate up to an accuracy of 10^{-4} (measured by the residual norm).

Solve the stationary boundary value problem from Worksheet 6 with the multigrid solver. Use $T(x, y) = 1$ for all (x, y) in $]0; 1[^2$ as an initial guess for the solution. Record the number of iterations, the runtime and the storage requirements to achieve an accuracy of 10^{-4} for different grid resolutions:

$N_x = N_y$	3	7	15	31	63	127	255
# iterations							
runtime (sec)							
memory (floats)							

By which factor could we reduce the runtime for the highest resolution ($N_x = N_y = 255$) in comparison to the Gauss-Seidel?

Questions:

1) Examine the computational costs for the Gauss-Seidel method: By which factor are the costs multiplied when you double the number of grid points in each coordinate direction? Is this an optimal behaviour for a solver? Which factor would be optimal?

2) Can you give upper bounds for the memory requirements of the multigrid solver implemented (if N is the number of unknowns on the finest grid)?

3) Can you give an upper bound for the computational costs (in floating point operations) of one multigrid v-cycle (as implemented above) in dependence on the number N of unknowns on the finest grid?

4) Is the multigrid method an optimal solver in the sense of Worksheet 6?