M.Sc. Simulation Sciences, Summer Semester 2020

Fast Iterative Solvers

Suggested Work for Multigrid Assignment – Part 2

Restriction Operator

Implement a fully weighted restriction operation, as discussed in class. You will re-use this function when writing a complete multigrid solver!

For validation we approximate a square domain $\Omega = [0, 1]^2$ using a grid as defined in the previous handout. We define a "fine mesh" with $N = 2^n$; for some integer $n \geq 1$, and a coarse mesh using $N^c = 2^{n-1}$. This means that points in the coarse mesh will also be points in the fine mesh, while every other point in the fine mesh is deleted.

A function $\mathbf{u}_{2h} = RESTR(\mathbf{u}_h; N^c)$ should be most convenient to work with, i.e. the loop ought to be over the coarse mesh nodes. Please also see the class notes, where we discussed a possible loop layout.

- Test your implementation for two grids, using $n = 4$ and $n = 7$ for the fine grid, respectively.
- initialize $u_{i,j}$ on the *fine* grid such that $u_h[i, j] = u(x_i, y_j)$ for the function $u(x, y) = \sin(2\pi x) \sin(2\pi y)$.
- Then transfer to the coarse grid, and measure $||\mathbf{e}_{2h}||_\infty = \max_{i,j} |u_{2h}[i, j] - u(x_i, y_j)|$. (Note that here (i, j) are *coarse* grid indices!)

Prolongation Operator

Implement the bilinear prolongation operator, as discussed in class. You will re-use this function when writing a complete multigrid solver!

Define fine and coarse grids in the same way as for the restriction operator.

A Function $\mathbf{u}_h = PROLONG(\mathbf{u}_{2h}; N^c)$ should be most convenient to work with, i.e. the loop ought to be over the coarse mesh nodes.

- Test your implementation for two grids, using $n = 4$ and $n = 7$ for the fine grid, respectively.
- initialize $u_{i,j}$ on the *coarse* grid such that $u_{2h}[i, j] = u(x_i, y_j)$ for the function $u(x, y) = \sin(2\pi x) \sin(2\pi y)$.
- Then transfer to the fine grid, and measure $||\mathbf{e}_h||_\infty = \max_{i,j} |u_h[i, j] - u(x_i, y_j)|$. (Note that here (i, j) are *fine* grid indices!)