M.Sc. Simulation Sciences, Summer Semester 2023

Fast Iterative Solvers

Prof. Georg May

Work Package for Multigrid Assignment – Part 1

Summary

This is preparatory work for multigrid methods. You will implement a (single-grid) Gauss-Seidel relaxation method to solve the Poisson problem discussed in class. Here you will *not* assemble sparse matrices explicitly, so there is no need to use `MSR` storage format.

Poisson Solver

Consider the Poisson equation with homogeneous boundary conditions,

$$\begin{aligned} -\nabla^2 u &= f, && \text{in } \Omega, \\ u &= 0, && \text{on } \partial\Omega, \end{aligned}$$

where $\Omega = (0,1) \times (0,1)$, Use a finite difference discretization on a Cartesian Grid

$$\mathcal{G}_h := \{(ih, jh) : i, j = 0, \dots, N;\, hN = 1\} \tag{1}$$

to find $u_{i,j} := u(x_i, y_j) = u(ih, jh)$, such that

$$\begin{aligned} -f_{i,j} &= \frac{1}{h^2}\left(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}\right) + \frac{1}{h^2}\left(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}\right), && i, j = 1, \dots, N-1 \\ u_{i,j} &= 0, && \text{otherwise.} \end{aligned}$$

As discussed in class, implement a Gauss-Seidel Relaxation with lexicographical ordering (GS-LEX) to solve the problem.
You should implement this as a function, i.e., $\mathbf{u} = GS(\mathbf{u}_0, \mathbf{f}; \nu)$, where $\mathbf{u}$ and $\mathbf{f}$ are two-dimensional arrays for the solution and right-hand sides, respectively. $\mathbf{u}_0 = 0$ is the initial guess, and ν is the number of iterations. You will re-use this function later when implementing the multigrid method for this problem.

- **Note:** We don't iterate over the boundary points! These should be initialized to zero, and then not be touched again.
- Test your implementation for the right-hand side $f(x, y) = 8\pi^2 \sin(2\pi x)\sin(2\pi y)$. For this choice, the exact solution is $u(x, y) = \sin(2\pi x)\sin(2\pi y)$.
- Choose ν large enough so that $||\mathbf{u}_\nu - \mathbf{u}_{\nu-1}||_\infty < 10^{-10}$.
- Measure the *converged* maximum error, i.e. $\max_{i,j} |u_{i,j} - u(x_i, y_j)|$ for $N_x = N_y = N$, and $N = 10$, as well as $N = 100$. For the latter, you should observe a reduction in error by two orders of magnitude.
- Note: For this Cartesian grid, you should not assemble *any* matrix to implement the Gauss-Seidel iteration!