

MathCore: Discretization & Solution

Today, we

- measure running times of different solvers, direct (matlab backslash), Jacobi iteration, multigrid iteration
- Slides: <https://www.math.uni-magdeburg.de/~richter/mathcore/multigrid.pdf>

Matlab-Template

- Download and unzip <https://www.math.uni-magdeburg.de/~richter/mathcore/template2.zip>
 - Open Matlab and look at the files. Some parts are the same as yesterday `createmesh.m`, `setrhs.m`, `assemblematrix.m`, `exactsolution.m`, `computeerror.m`
 - Furthermore you have some empty templates to be finished by you.
1. We want to measure computing time. Have a look at `start1.m`. The most important change to yesterday: The function gets two parameters, `DIM` and `L` which stands for the number of mesh levels. It holds

$$M = 2^L.$$

`start1.m` sets up mesh, matrix and right hand side, it solves the linear system and it prints out some statistics including the time for assembly of matrix and for solving the problem.

Make a table like

L	N (2d)	time (2d)	N (3d)	time (3d)
3				
4				
5				
6				

including the time for solving the problem and compute the relation α between N and solving time

$$time \sim N^\alpha$$

Note: `tic`, `toc` are used for time measurements. The order α can be reconstructed from running times for different values of N .

2. Use the template `start2.m`. We want to replace the Matlab direct solver by the Jacobi iteration. Have a look at `jacobi.m`.
Instead of `xh=Ah\rhs` use the function `jacobi.m`

1. Jacobi needs an initial guess, this is already given.
2. Furthermore, Jacobi is only an approximation. We must implement a stopping criteria. In `start2.m` we compute the desired tolerance as

$$tol = 10^{-6} \cdot \|b\|_{\infty}$$

Program a loop. In every step perform one iteration of Jacobi `[xh,error] = jacobi(xh,rhs,Ah,1);`. Stop the loop once the Jacobi error reaches the tolerance.

Extend the table from part 1 with the corresponding values using Jacobi.

3. The template is `start3.m`. We implement the multigrid method. Some changes
 - If we have L mesh levels we need L meshes and L matrices. We use the Matlab cell-structure.
 - `start3.m` already has a loop over multigrid steps until we reach the tolerance

$$tol = 10^{-6} \cdot \|b\|_{\infty}$$

Your job is to finish the multigrid algorithm in `multigrid.m`. This is missing:

- Pre- and post-smoothing with `jacobi.m`
- Calculation of the residual `residual` before restricting it onto the coarse mesh
- Recursive call of `multigrid.m` to compute the coarse grid correction `correction`. You must call the same function on the level `level-1`. You will need an initial guess for the coarse mesh problem, this is already given in `multigrid.m`.

Extend the table with the multigrid values. What is the scaling

$$time \sim N^{\alpha}$$