Documentation MATLAB - Code

What is out there? - LSFEM only

1. *GMG simple heat eqn*
2. *AMG simple heat eqn*
3. *modulated MG*
4. *modulated direct*
5. *modulated old*
6. *eigenfunctions*
7. *comparison*
8. *other*

1 GMG simpe heat equation

we are solving LSFEM version of heat equation using a geometric multigrid. Let's talk about the multigrid setting first, for the **smoother** there are mainly two different options even though all other ones would theoretically work as well, but probably don't make much sense.

- GaussSeidelSolve LS
- **JacobiSolve extended space time**

The first one, does a regular Gauss-Seidel iteration but couples σ and u. Not so good though, not parallelisable. Then there are different Jacobi Solve versions, but the above one can do everything the others can, because there you get to choose the size of the patch in space and time. Patch size needs to be manually set in smoother function though, is not an input parameter. Improves performance a bit but not so much.

The set up of the grid has two options, you can start with the fine mesh, but then it has to have a size 2^k so that we can coarsen it appropriately or you start with a coarse grid that is then recursively refined or we take this λ business into account (however meaningful that is, current idea would be to keep $\frac{d\Delta t}{\Delta x} = \lambda \approx 1$, not well tested yet, there is an implementation for keeping $\frac{d\Delta t}{\Delta x^2} = \lambda \approx 1$ if d diffusion constant is small. Hence as set ups we have

- heat eqn zero rhs MG call
- run heat eqn lsfem adaptive mesh ref
- **run from coarse heat eqn zero rhs**

Hence we now need that appropriately sized interpolation operators are being generated

- heat eqn zero rhs MG call
- **set up interpolation op sp time** - coarsens in space and time
- set up interpolation op sp - only coarsens in space but not in time

- set up interpolation op

They all get called in the V-cycle function. Probably most important function, pretty everything happens there and often needs to be adapted manually but it already has so many input parameters. In **V-cylce**

- set up interpolation operators
- set up coarse grid operators (constructing them explicitly, including boundary conditions)
- then loop over maximal number of iterations and then in there over levels
- here we do a step from fine to coarse, that is smoothen + computing the restricted residual, which we return
- after we have done this for all levels except last one, solve directly on coarsest grid
- interpolate back to finer grid, set boundary terms to zero, and smoothen
- when on finest grid check if residual smaller than ϵ, if yes, stop otherwise contiunue unless max iteration reached

two V-cycle versions, one with comments, figures that are generated, etc. to debug, one plain one to see the overall structure, that is

- **V-cycle**
- V-cycle plain

4 Modulated Direct

Not using a multigrid, don't necessarily have positive definiteness, hence use a direct solve instead and as a nonlinear iteration scheme a trust region dogleg method and a backtracking line search with wolfe conditions