MAT 228A Notes

Sam Fleischer

November 10, 2016

1 The Big Idea of Multigrid

- Use smoothers like GS or ω-Jacobi to get rid of high spacial frequency error.
- Use a coarser mesh to eliminate lower frequency errors.

2 Coarse Grid Correction

We will subscripting everything with the gridspacing since there are multiple grids... let u_h be the solution to $L_h u_h = f_h$ where L_h is the discrete Laplacian operator of a grid of size h. u_h^k is the approximate solution after k iterations. Then $e_h^k \coloneqq u_h - u_h^k$ is the error in the approximation solution and $r_h^k \coloneqq f_h - L_h u_h^k$ is the residual. Then $L_h e_h^k = r_h^k$. Rearranging the definition of the error gives us

$$\begin{aligned} u_h &= u_h^k + e_h^k \\ &= u_h^k + L_h^{-1} r_h^k \end{aligned}$$

This suggests how to generate an iterative scheme. Lets approximate L_h^{-1} to generate an iterative scheme.

For coarse-grid correction, use a coarse mesh to "solve" this equation. Let Ω_h be the original mesh with grid size h. This is Ω_h is the original fine mesh. Then one way to define a coarser grid is by Ω_{2h}, which is, in 1D, half the amount of grid points as Ω_h.

Then let $G(\Omega_h)$ be the set of grid functions of Ω_h. We need "transfer operators" which maps elements between grids.

- A restriction operator (from fine mesh to coarse mesh)

$$I_h^{2h} \ : \ G(\Omega_h) \to G(\Omega_{2H})$$

- An interpolation (or prolongation) operator (from coarse mesh to fine mesh)

$$I_{2h}^h \ : \ G(\Omega_{2h}) \to G(\Omega_h)$$

We have u_h^k. We must compute the fine grid residual

$$r_h^k = f_h - L_h u_h^k$$

Then we restrict the residual

$$r_{2h}^k = I_h^{2h} r_h^k$$

and solve for the error (however you solve doesn't really matter - it should be relatively cheap on a coarser mesh)

$$e_{2h}^k = L_{2h}^{-1} r_{2h}^k$$

Then interpolate the coarse grid error e_{2h}^k back on to the fine mesh

$$\tilde{e}_h^k = I_{2h}^h e_{2h}^k$$

Then correct the approximate solution

$$u_h^{k+1} = u_h^k + \tilde{e}_h^k$$

How is $\tilde{e}_h^k$ defined?

$$\begin{aligned} u_h^{k+1} &= u_h^k + \tilde{e}_h^k \\ &= u_h^k + I_{2h}^k e_{2h}^k \\ &= u_h^k + I_{2h}^k L_{2h}^{-1} r_{2h}^k \\ &= u_h^k + I_{2h}^k L_{2h}^{-1} I_h^{2h} r_h^k \\ &= u_h^k + I_{2h}^k L_{2h}^{-1} I_h^{2h} \big(f_h - L_h u_h^k\big) \end{aligned}$$

whew!

$$\begin{aligned} u_h^{k+1} &= (I - \underbrace{I_{2h}^h L_{2h}^{-1} I_h^{2h}}_{\text{coarse-grid inv.}} L_h) u_h^k + I_{2h}^h L_{2h}^{-1} I_h^{2h} f_h \\ &= K u_h^k + C \end{aligned}$$

where K is the "coarse grid operator" and C is the constant. This iteration does NOT converge since we cannot represent the high-frequency errors. In fact, the high-frequency errors are added to the low-frequency errors. So we MUST perform smoothing first in order for this converge. This will get rid of the high-frequency errors.

3 Smoothing

Let S denote the smoothing operator. Two-grid iteration:

1. Pre-smoothing: smooth ν_1 times.
2. Apply coarse-grid correction.
    - Compute residual
    - Restrict residual
    - Solve for coarse grid error (*** this is what turns 2-grid into multigrid)
    - Interpolate the error
    - Correct (add the error back in)
3. Post-smooth
    - Smooth ν_2 times.

So,

$$M = S^{\nu_2}(I - I_{2h}^h L_{2h}^{-1} I_h^{2h}) S^{\nu_1}$$

4 Questions?

- How do we pick ν_i?
- What are the transfer operators?
- What is L_{2h}?
- Which smoothing operator?
- How efficient is 2-grid?
    - Turns out it gives $\mathcal{O}(N \log(N))$ work (near optimal)

5 Transfer Operators

5.1 Restriction

- The simplest operator is to just throw out half the points. This is called **Injection**. So, just set the jth point of the coarse grid equal to the $2j$th point of the fine mesh.

$$(u_{2h})_j = (u_h)_{2j} \tag{1}$$

- Another common operator is called full-weighting (adjoint of the linear interpolation operator). Grab a local average of the points above.

$$(u_{2h})_j = \frac{1}{4}\left[(u_h)_{2j-1} + 2(u_h)_{2j} + (u_h)_{2j+1}\right] \tag{2}$$

Stencil for full-weighting is, in 1D,

$$I_h^{2h} = \frac{1}{4}\left[\begin{array}{ccc} 1 & 4 & 1 \end{array}\right] \tag{3}$$

Stencil for full-weighting is, in 2D,

$$I_h^{2h} = \frac{1}{16}\left[\begin{array}{ccc} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{array}\right] \tag{4}$$