Elements of Multigrid


• However, the first major article on multigrid by one of its main developers: Achi Brandt, Multi-Level Adaptive Solutions to Boundary-Value Problems, Math Comp 31, April 1977, pp. 333-390 is a good candidate for the most exciting paper ever written in numerical analysis.

• We’ll continue to introduce the relevant ideas in the context of the Poisson Equation

\[ u_{xx} + u_{yy} = f(x, y) \]

and its discretization on a rectangular grid of the unit square.

• Let’s first discuss several major ideas:
• How accurately do we need to solve our linear system?

Error in the application of the solution of the linear system should be the same size as the GTE
• How do we start?

\[ A x = b \]

\[ x^{[0]} \text{ given} \]

\[ x^{[k+1]} = \omega b + c \omega x^{[k]} \]
- Relaxation does not much reduce the error, but it **smoothes** it.
Use several nested grids. Figure 1 shows several grids. Color indicates the depth of the grids. The coarsest is black, followed by red, green, blue.

Deal with each Fourier component of the error on the grid most appropriate for that component.

Several grids give us several approximations that can be used for error estimation.
• Every grid requires roughly $1/4$ the memory of the next finer grids.

• If the finest grid has $N^2$ points, say, the total memory requirement is approximately

$$N^2 \left(1 + \frac{1}{4} + \frac{1}{16} + \frac{1}{64} + \ldots\right) = \frac{4}{3} N^2,$$

hardly more than $N^2$. 
• We’ll now introduce Brandt’s notation, ignoring boundary conditions.

• In Brandt’s notation boundary conditions are written in the same form as the PDE, except that he uses Greek letters.

• We consider grids

$$G^0, G^1, G^2, \ldots, G^M$$

with corresponding step-sizes

$$h_0 > h_1 > h_2 > \ldots > h_M$$

• typically

$$h_{k+1} = h_k / 2.$$ 

• The grids cover a domain $$\Omega$$ (which for our purposes is the unit square).

• The PDE can be written as

$$LU(x) = F(x)$$

where $$L$$ is a linear differential operator and $$x$$ is a point in $$\Omega$$. (Thus $$x$$ is now a vector.)

• On each grid, we approximate the PDE by a linear system

$$L_k U_k = F^k(x), \quad x \in G^k.$$ 

• We wish to solve the linear equations on the finest grid.
• We denote exact solutions of the linear equations by capital letters, and any approximations by lower case letters.

• For example, \( u^M \) is an approximation of \( U^M \).

• Let \( f^M \) be the residual

\[
f^M = F^M - L^M u^M = L^M (U^M - u^M).
\]  

(1)

• We get from (1) that

\[
U^M = u^M + v^M
\]

where

\[
L^M v^M = f^M
\]

• This is always the same for linear problems: the error satisfies a linear system with the same matrix and the right hand side being the residual:

\[
L^M U^M = F^M
\]

\[
L^M u^M = f^M + F^M
\]

\[
L^M (u^M - U^M) = L^M u^M - f^M
\]

• We need to be able to move back and forth between the grids. The corresponding operators are:

\( I_{k+1}^k \) go from fine, \( k + 1 \), to coarse, \( k \).

\( I_{k}^{k+1} \) go from coarse, \( k \), to fine, \( k + 1 \).
$56.10:\,

\begin{align*}
Ax &= b \quad \Rightarrow \quad \hat{x} \\
&= x - \hat{x} \\
r &= b - A\hat{x} \\
&= Ax - A\hat{x} \\
&= A(x - \hat{x}) \\
&= 4e
\end{align*}$
Figure 2. Nested Grids.

Cycle C

- Cycle C is what I think of as the simplest version of multigrid.
- On each grid we consider the residual equation for the next finer grid.
- On the finest grid we can think of the residual with respect to the zero approximation of the solution.
• $k$ is always the current grid. $v^k$ is the approximation of the solution of the linear equation on the $k$-th grid.

• Brandt has a flow chart for the algorithm.

• Here is an outline:

1. Set 
   \[ k \leftarrow M, \quad f^k \leftarrow F^M, \quad v^k \leftarrow u^M. \]
   Go to 2.

2. If $k = 0$ solve
   \[ L^k v^k = f^k \]
   directly. Otherwise apply one relaxation sweep. This could be a Gauss-Seidel or SOR sweep. Go to 3.

3. Check if $v^k$ has converged. If so, and $k = M$ we are done. If so and $k < M$ then go to 4. If not, go to 5.

• How do we decide on convergence? On the coarser grids ($k < M$) convergence has occurred when the residual on the $k$-th grid is small compared to the residual on the $(k+1)$-th grid. Precise definitions of “small” depend on the DE and the relaxation technique.
On the finest grid, convergence has occurred when the residual is approximately equal to the truncation error.

The truncation error can be approximated by comparing approximations on different grids.

There is no reason to solve the linear system on the finest grid more accurately than that system approximates the solution of the DE.

4. Set
\[ k \leftarrow k + 1 \quad \text{and} \quad v^k \leftarrow v^k + I^k_{k-1}v^{k-1} \]

Go to 2.

5. Check if convergence is slow. If not go to 2. Else go to 6.

Convergence is slow if the residual does not get reduced by a sufficiently large factor.

6. Set
\[ f^k \leftarrow I^{k+1}_{k+1}(f^{k+1} - L^{k+1}v^{k+1}) \]

Go to 2.

There are lots of details in the above algorithm. A central role is played by Fourier Analysis.
• See the example for $M = 6$ in Brandt’s paper. The total number of sweeps is 143.9, including 10 on the finest grid. The total reduction in the error is $10^{-4}$.

**More Ideas**

• Since an estimate of the LTE is available it can be used to refine the grid locally, giving rise to **adaptive techniques**.

• Applying these ideas to general linear systems gives rise to a subject called **Algebraic Multigrid**.

• It’s clear how to transfer these ideas to more than two variables, or to parabolic problems.

• When all goes well the numerical effort is proportional to the number of grid points, which is remarkable. Compare this with the third power of the number of grid points, required for Gaussian Elimination.

• In modern implementations the number of sweeps on each grid is usually predetermined.

• Many versions of multigrid are based on applying a two level multigrid version recursively.