Introduction of Multigrid

Zichao Di

November 7, 2011

Outline

- *Model Problems*
- Development of Multigrid
  - Coarse-grid correction
  - Nested Iteration
  - Restriction and Interpolation
  - Standard cycles: MV, FMG
- Summary and discussion

Model Problems

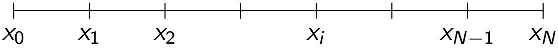
- *One-dimensional boundary value problem:*

$$-u''(x)+\sigma u(x) = f(x) \qquad 0 < x < 1, \quad \sigma > 0$$

$$u(0) = u(1) = 0$$

- Grid: $h = \frac{1}{N}$, $\qquad x_i = ih, \qquad i = 0, 1, \ldots, N$

Ω^h :$x = 0$ $\qquad\qquad$ $x = 1$



- Let $v_i \approx u(x_i)$ and $f_i \approx f(x_i)$ for $i = 0, 1, \ldots, N$

We approximate the equation with a finite difference scheme

- We approximate the BVP

$$-u''(x) + \sigma u(x) = f(x) \qquad 0 < x < 1, \quad \sigma > 0$$

$$u(0) = u(1) = 0$$

with the finite difference scheme:

$$\frac{-v_{i-1} + 2v_i - v_{i+1}}{h^2} + \sigma v_i = f_i, i = 1, 2, \ldots, N-1$$

$$v_0 = v_N = 0$$

The discrete model problem

- Letting $v = v_1, v_2, \ldots, v_{N-1}{}^T$ and

$$f = (f_1, f_2, \ldots, f_{N-1})^T$$

we obtain the matrix equation $Av = f$ where A is $(N-1)x(N-1)$, symmetric, positive definite, and

$$A = \frac{1}{h^2}\begin{pmatrix} 2+\sigma h^2 & -1 & & & & \\ -1 & 2+\sigma h^2 & -1 & & & \\ & -1 & 2+\sigma h^2 & -1 & & \\ & & & \ldots & & \\ & & & -1 & 2+\sigma h^2 & -1 \\ & & & & -1 & 2+\sigma h^2 \end{pmatrix}$$

$$v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \ldots \\ v_{N-2} \\ v_{N-1} \end{pmatrix}, \quad f = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \ldots \\ f_{N-2} \\ f_{N-1} \end{pmatrix}$$

Solution Methods

- Direct
  - Gaussian elimination
  - Factorization
- Iterative
  - Let $A = (D - L - U)$ where D is diagonal and L and U are the strictly lower and upper parts of A
  - Jacobi
    - Let $R_J = D^{-1}(L + U)$, then the iteration is :

$$v^{new} = R_J v^{old} + D^{-1} f$$

  - Gauss-Seidel (1D)
    - Let $R_G = (D - L)^{-1} U$, then the iteration is:

$$v^{new} = R_G v^{old} + (D - L)^{-1} f$$

  - Conjugate Gradient, etc.

Analysis of stationary iterations

- Let $v^{new} = Rv^{old} + g$. The exact solution is unchanged by the iteration, i.e., $u = Ru + g$
- Subtracting, we have the error propagation:

$$e^{new} = Re^{old}$$

- With e^0 be the initial error, after *ith* iteration, we have

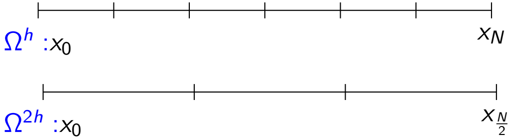
$$e^n = R^n e^0$$

Fundamental theorem of iteration

- R is convergent ($R^n \to 0$ as $n \to \infty$) if and only if $\rho(R) < 1$, where $\rho(R) = max\{|\lambda_1|, |\lambda_2|, \ldots, |\lambda_N|\}$
- Therefore, for any initial vector v^0, wer see that $e^n \to 0$ as $n \to \infty$ if and only if $\rho(R) < 1$
- $\rho(R) < 1$ assures the convergence of the iteration given by R and $\rho(R)$ is called the convergence factor for the iteration.
- Since $\frac{\|e^M\|}{\|e^0\|} \leq \|R^M\|$, so we use $(\frac{\|e^M\|}{\|e^0\|})^{\frac{1}{M}}$ to measure the convergence factor

First observation toward multigrid

- Many relaxation schemes have the smoothing property, where oscillatory modes of the error are eliminated effectively, but smooth modes are damped very slowly.
- This might seem like a limitation, but by using coarse grids we can use the smoothing property to good advantage.



- Why use coarse grids ?

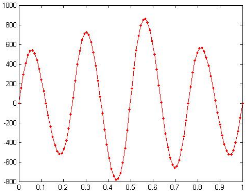
Reason #1 for using coarse grids: Nested Iteration

- Coarse grids can be used to compute an improved initial guess for the fine-grid relaxation. This is advantageous because:
  - Relaxation on the coarse-grid is much cheaper (1/2 as many points in 1D, 1/4 in 2D, 1/8 in 3D)
  - Relaxation on the coarse grid has a marginally better convergence factor, for example

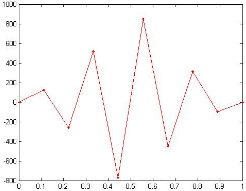
$$1 - O(4h^2) \qquad \textit{instead} \quad \textit{of} \qquad 1 - O(h^2)$$

Reason #2 for using a coarse grid: smooth error is (relatively) more oscillatory there!

- A smooth function:



- Can be represented by liner interpolation from a coarser grid:



Second observation toward multigrid:

- Recall the residual correction idea: Let v be an approximation to the solution of $Au = f$, where the residual $r = f - Av$. The the error $e = u - v$ satisfies $Ae = r$.
- After relaxing on $Au = f$ on the fine grid, the error will be smooth. On the coarse grid, however, this error appears more oscillatory, and relaxation will be more effective
- Therefore we go to a coarse grid and relax on the residual equation $Ae = r$, with an initial guess of $e = 0$.

Idea! Coarse-grid correction

- Relax on $Au = f$ on Ω^h to obtain an approximation v^h
- Compute $r = f - Av^h$
- Relax on $Ae = r$ on Ω^{2h} to obtain an approximation to the error, e^{2h}
- Correct the approximation $v^h \leftarrow v^h + e^{2h}$
- Clearly, we need methods for the mappings $\Omega^h \rightarrow \Omega^{2h}$ and $\Omega^{2h} \rightarrow \Omega^h$

1D Interpolation (Prolongation)

- Mapping from the coarse grid to the fine grid:

$$I_{2h}^{h} : \Omega^{2h} \rightarrow \Omega^{h}$$

- Let v^h, v^{2h} be defined on Ω^h, Ω^{2h}. Then

$$I_{2h}^{h} v^{2h} = v^{h}$$

where

$$\left\{ \begin{array}{c} v_{2i}^{h} = v_{i}^{2h} \\ v_{2i+1}^{h} = \frac{1}{2}(v_{i}^{2h} + v_{i+1}^{2h}) \end{array} \right.$$

for $0 \leq i \leq \frac{N}{2} - 1$

The prolongation operator (1D)

- We may regard I_{2h}^{h} as a linear operator from $R^{N/2-1} \rightarrow R^{N-1}$
- e.g., for N=8,

$$\begin{pmatrix} 1/2 & & \\ 1 & & \\ 1/2 & 1/2 & \\ & 1 & \\ & 1/2 & 1/2 \\ & & 1 \\ & & 1/2 \end{pmatrix}_{7x3} \begin{pmatrix} v_1^{2h} \\ v_2^{2h} \\ v_3^{2h} \end{pmatrix}_{3x1} = \begin{pmatrix} v_1^h \\ v_2^h \\ v_3^h \\ v_4^h \\ v_5^h \\ v_6^h \\ v_7^h \end{pmatrix}_{7x1}$$

- I_{2h}^{h} has full rank, and thus null space $\{\emptyset\}$

How well does v^{2h} approximate u

- If the exact solution u of error equation is smooth, a coarse-grid interpolant of v^{2h} may do very well.
- If u is oscillatory, a coarse-grid interpolant of v^{2h} may not work well.
- Therefore, nested iteration is most effective when the error is smooth!

1D Restriction by full-weighting

- Mapping from the fine grid to the coarse grid:

$$I_h^{2h} : \Omega^h \to \Omega^{2h}$$

- Let v^h, v^{2h} be defined on Ω^h, Ω^{2h}. Then

$$I_h^{2h} v^h = v^{2h}$$

 where $v_i^{2h} = \frac{1}{4}(v_{2i-1}^h + 2v_{2i}^h + v_{2i+1}^h)$
- Regard I_h^{2h} as a linear operator from $R^{N-1} \to R^{N/2-1}$
- e.g., for $N = 8$,

$$\begin{pmatrix} 1/4 & 1/2 & 1/4 & & & & \\ & & 1/4 & 1/2 & 1/4 & & \\ & & & & 1/4 & 1/2 & 1/4 \end{pmatrix} \begin{pmatrix} v_1^h \\ v_2^h \\ v_3^h \\ v_4^h \\ v_5^h \\ v_6^h \\ v_7^h \end{pmatrix} = \begin{pmatrix} v_1^{2h} \\ v_2^{2h} \\ v_3^{2h} \end{pmatrix}$$

Coarse Grid Correction Scheme (V-cycle)

1. Relax α_1 times on $A^h u^h = f^h$ on Ω^h with arbitrary initial guess v_h.
2. Compute $r^h = f^h - A^h v^h$
3. Compute $r^{2h} = I_h^{2h} r^h$
4. Solve $A^{2h} e^{2h} = r^{2h}$ on Ω^{2h}
5. Correct fine-grid solution $v^h \leftarrow v^h + I_{2h}^h e^{2h}$
6. Relax α_2 times on $A^h u^h = f^h$ on Ω^h with initial guess v^h

Building A^{2h}

- Assume that $e^h \in Range(I^h_{2h})$. Then the residual equation can be written

$$r^h = A^h e^h = A^h I^h_{2h} u^{2h}$$

- Then the residual equation on the coarse grid is:

$$I^{2h}_h A^h I^h_{2h} u^{2h} = I^{2h}_h r^h$$

- Therefore,we identify the coarse-grid operator A^{2h} as

$$A^{2h} = I^{2h}_h A^h I^h_{2h}$$

.

- It turns out that the *ith* row of A^{2h} is $\frac{1}{(2h)^2}[-1 \quad 2 \quad -1]$ which is the Ω^{2h} version of A^h

The variational properties

- Recall that for the 1D examples, linear interpolation and full-weighting are related by :

$$I_{2h}^{h} = 2(I_h^{2h})^T$$

- The definition for A^{2h} that resulted from the foregoing line of reasoning is useful for both theoretical and practical reasons. Together with the commonly used relationship between restriction and prolongation we have the following variational properties:
- A commonly used, and highly useful, requirement is that $I_{2h}^{h} = c(I_h^{2h})^T$ for $c \in R$

Summary and Discussion

- Multigrid has been proven on a wide variety of problems, especially elliptic PDEs, but has also found application among parabolic hyperbolic PDEs, integral equations, evolution problems, geodesic problems, etc.
- With the right setup, multigrid is frequently an optimal solver
- Multigrid is of great interest because it is one of the very few scalable algorithms, and can be parallelized readily and efficiently!