Geometric multigrid methods

Lukas Einkemmer

Department of Mathematics
University of Innsbruck

May 18, 2015

Problem

Let us consider the model problem

$$-\partial_{xx} u = f, u(0) = 0, u(1) = 0$$

discretized using finite differences on a grid of N elements on $[0, 1]$.

The discretization results in a linear system of equations written in matrix form as

$$Au = f$$

with a large matrix A.

We can use fast Poisson solvers such as FFT based methods that are $\mathcal{O}(n \log n)$ but can only be applied to simple geometries and linear problems. Direct solvers (such as LU factorization) take $\mathcal{O}(N^3)$ operations. Basic iterative solvers (such as Jacobi or Gauss-Seidel) take $\mathcal{O}(N^3)$ operations. Krylov based iterative solvers (such as CG or GMRES) require $\mathcal{O}(N^2)$ operations.

Reminder

For $A = D + L + U$ the Jacobi iteration is given by

$$x_{n+1} = -D^{-1}(L + U)x_n + D^{-1}b$$

with iteration matrix $T = -D^{-1}(L + U)$.

The error is given by

$$\epsilon_n = u - x_n$$

and satisfies

$$\epsilon_n \leq \rho(T)^n \epsilon_0$$

Usually we have $\rho(T) = 1 - \mathcal{O}(N^2)$ which means that convergence is slow for large N.

A closer look on the error

For our problem we know the exact eigensystem which is given by

$$w^{(k)}(x) = \sin(xk\pi), \qquad \lambda^{(k)} = 4\sin^2(k\pi/(2N))$$

where k is a parameter (the wavenumber or frequency).
The corresponding discretization is also an eigenvalue of the discrete system; i.e.

$$w_j^{(k)} = \sin(jk\pi/N), \qquad Aw^{(k)} = \lambda^{(k)} w^{(k)}.$$

The iteration matrix (for our model problem)

$$T = I - \frac{1}{2}A$$

has the same eigenvectors with eigenvalue $1 - \frac{1}{2}\lambda^{(k)}$.

A closer look at the error

Let us consider $f = 0$. Then the exact solution is given by $u = 0$ and

$$Tw_j^{(k)} = (1 - \tfrac{1}{2}\lambda^{(k)})w_j^{(k)}$$

$$1 - \tfrac{1}{2}\lambda^{(k)} \approx 1 - k^2\pi^2/(2N^2).$$

Now for $N/4 < k < 3N/4$ we have

$$1 - \frac{1}{2}\lambda^{(k)} \leq 0.71$$

which is a bound independent of N.

Weighted Jacobi iteration

Let us conduct a Jacobi iteration with result $x^\star_{n+1}$ and define

$$x_{n+1} = (1-\omega)x_n + \omega x^\star_{n+1}$$

for which the iteration matrix is given by

$$T_\omega = (1-\omega)I + \omega T.$$

The corresponding eigenvalues are given by

$$1 - 2\omega \sin^2(k\pi/(2N))$$

Choose ω to damp the high frequencies as well. For $\omega = 2/3$ we have

$$1 - 2\omega \sin^2(k\pi/(2N)) \leq 1/3$$

Error in high frequencies is reduced by 10^{-2} in 4 iterations.
Error in low frequencies is reduced much slower.

A first idea

How can we exploit the fast convergence of high frequencies?

We know that

- the better the initial guess the faster the iteration converges
- low frequencies on a fine grid become high frequencies for a coarser grid.

Nested iteration

- Relax $Au = f$ on a very coarse grid to obtain initial guesses for the next finer grid
- ...
- Relax $Au = f$ on Ω^{2h} to obtain an initial guess for Ω^h
- Relax $Au = f$ on Ω^h to obtain the final solution

Interpolation/Prolongation $I_{2h}^h x^{2h} = x^h$

Nested iteration

- On the coarsest grid we can compute an exact solution.
- On the next finer grid the error is mainly in the high frequencies
- The high frequencies converge fast using only a few iterations
- Repeat

But if some smooth components remain (because the coarsest grid is not a good approximation to the problem, even for low frequencies) we obtain inaccurate solutions.
We look for some way to avoid this problem.

Residual equation

For the error $\epsilon = u - x$ we have

$$A\epsilon = r$$

with residual $r = f - Ax$.

Theorem: Relaxing x for a given initial value x_0 using the equation $Au = f$ is equivalent to relaxing ϵ for the initial value $\epsilon_0 = 0$ using the equation $Au = r_0$.

Two-grid scheme

The algorithm

- Relax $A^h u^h = f^h$ on Ω^h with some initial guess
- Compute the residual $r^h = f^h - A^h x^h$
- Restrict the residual to the coarse grid $r^{2h} = I_h^{2h} r^h$
- Solve $A^{2h} \epsilon^{2h} = r^{2h}$ on Ω^{2h}
- Interpolate the coarse grid error to the fine grid $\epsilon^h = I_h^{2h} r^h$
- Correct the approximation on the fine grid by $x^h \leftarrow x^h + \epsilon^h$
- Relax $A^h u^h = f^h$ on Ω^h

Restriction/Injection $I_h^{2h} x^h = x^{2h}$

- Error mainly in the high frequencies (Aliasing) $\Rightarrow$ Requires smooth function
- Complements the relaxation which efficiently reduces high frequencies in the error

Multigrid

How to solve the coarse problem on Ω^{2h}?

Idea of multigrid schemes is to apply recursion. For that we need a sequence of grids Ω^h, Ω^{2h}, Ω^{4h}, Ω^{8h}, Ω^{16h}

V-cycle: one sweep down one sweep up
W-cycle: alternate sweeps down and up

Computational complexity

For a sparse matrix A^h matrix-vector multiplication requires $\mathcal{O}(N)$ operations.

The V-cycle consists of $2 \log N$ steps each of which requires a number ν_i of matrix-vector multiplications.
If we assume that ν_i is bounded independent of N (which is at least suggested by the fact that at each level the high frequencies errors are reduced independent of N) the computational complexity is $\mathcal{O}(N \log N)$.

Under suitable assumptions on the problem this can be demonstrated both numerically as well as by giving convergence proofs.

Application of multigrid schemes

Multigrid methods have the same complexity as FFT based Poisson solvers but usually larger constants. But they are applicable to a larger class of problems.

- No tensor product structure of the grid is assumed
- Can be applied to problems with arbitrary boundary conditions
- Can be applied to problems with inhomogeneous coefficients or nonlinearities
- No all-to-all communication

As a consequence multigrid schemes are widely employed in practice.

Multigrid schemes as preconditioners

Remember a preconditioner is an efficiently invertible matrix P such that $A \approx B$. The iterative scheme under consideration is then applied to

$$Mu = P^{-1}f, \qquad M = P^{-1}A$$

instead of the original problem in order to accelerate convergence.

Note that in an iterative scheme only applications of $P^{-1}A$ to a vector have to be computed.

We can use the multigrid approach as a preconditioner. That is, for each iteration

- Compute $y = Ax$
- Use, for example, a V-cycle to compute an approximation to $P^{-1}y$

Combining a multigrid preconditioner with a Krylov subspace method is popular in practice.

Implementation

A hierarchy of grids needs to be defined.

- easy if tensor product grid

Suitable extension and restriction operators have to be implemented.

- A interpolation algorithm is usually employed

A relaxation scheme has to be chosen.

- Usually simple iterative methods (such as the modified Jacobi scheme considered here) give good results.

Limitations of multigrid methods

Multigrid methods usually require an underlying PDE problem in order to construct the sequence of grids. But so-called algebraic multigrid methods haven been developed to alleviate this problem.

Multigrid problems are not always efficient when complex geometries are of interest.

Multigrid methods are usually employed to elliptic or parabolic problems.

References

A Multigrid Tutorial, Briggs, Henson, McCormick, 2000
Available as pdf from the Universitätsbibliothek

Iterative methods for sparse linear systems, Saad, 2004