Numerical Solutions to PDEs

Lecture Notes #22 — Elliptic Equations, Iterative Schemes

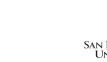
Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Spring 2018

Outline



Last Time: Finite Difference Schemes for Elliptic Problems

We looked the model problem $\nabla^2 u = f$ in the unit square in introduced the 5- and 9-point discrete (2D) Laplacians ($\nabla^2_{h_5}$, $\nabla^2_{h_9}$), which provide 2nd and 4th order accuracy, respectively.

In seeking numerical solutions we discovered that we quickly ended up with a matrix problem $A\bar{\mathbf{x}} = \bar{\mathbf{b}}$, where the entries in the matrix A are determined by the coefficients from the discrete Laplacian, and the entries in $\bar{\mathbf{b}}$ are due to the boundary conditions (and f, when $f \not\equiv 0$).

We introduced the Jacobi, Gauss-Seidel, and the Successive Over-Relaxation (SOR) methods for iteratively finding the solution; we showed how these methods can be interpreted as operation on either directly on the grid function (somewhat useful for implementation), or as a matrix operation (useful for analysis).

Linear Iterative Schemes $\nabla^2_{h_5}$

We restate the Jacobi, Gauss-Seidel, and SOR iterations for

$$A\bar{\mathbf{x}} = \bar{\mathbf{b}}.$$

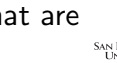
It is useful to think of A in terms of its diagonal, strictly lower triangular, and strictly upper triangular parts, *i.e.*

$$A = D - L - U.$$

If we consider Dirichlet boundary conditions, then we enumerate the interior points ($0 \le i < n_x$, $0 \le j < n_y$), and have

$$A_{(i+n_x\cdot j),(i+n_x\cdot j)} = 1, \quad A_{(i+n_x\cdot j),((i\pm 1)+n_x\cdot(j\pm 1))} = -\frac{1}{4},$$

when the $((i \pm 1) + n_x \cdot (j \pm 1))$-elements refer to points that are neighbors of (x_i, y_j), *i.e.* non-boundary points.



The Discrete Laplacian — Rectangular Grid

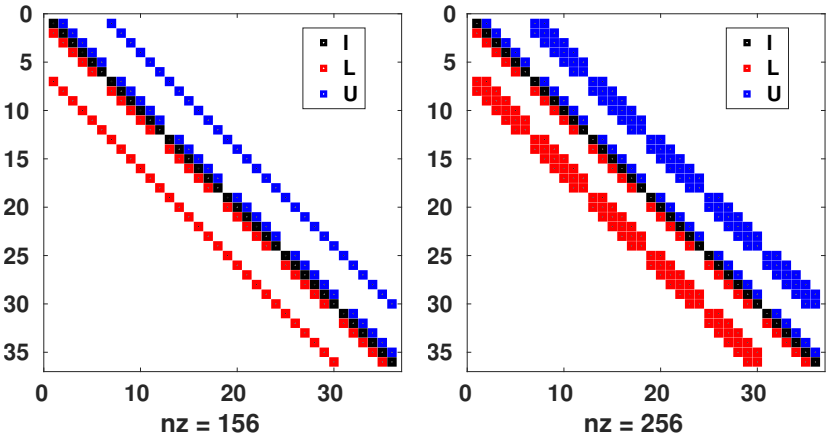


Figure: [LEFT] The Matrix $A = \mathbf{I} - \mathbf{U} - \mathbf{L}$ corresponding to the discrete 5-point Laplacian on a grid with 6×6 interior points and Dirichlet boundary conditions; [RIGHT] The Matrix $A = \mathbf{I} - \mathbf{U} - \mathbf{L}$ corresponding to the discrete compact 9-point Laplacian on a grid with 6×6 interior points and Dirichlet boundary conditions;

Jacobi, Gauss-Seidel, and SOR (5-Point Laplacian)

In terms of these matrix-splittings, we have

Jacobi

$$D\bar{\mathbf{x}}^{k+1} = (L+U)\bar{\mathbf{x}}^{k} + \bar{\mathbf{b}};$$

Gauss-Seidel

$$(D-L)\bar{\mathbf{x}}^{k+1} = U\bar{\mathbf{x}}^{k} + \bar{\mathbf{b}};$$

SOR

$$\left(\frac{1}{\omega}D - L\right)\bar{\mathbf{x}}^{k+1} = \left(\frac{1-\omega}{\omega}D + U\right)\bar{\mathbf{x}}^{k} + \bar{\mathbf{b}}.$$

The methods are convergent since the iteration matrices

$$M_J = D^{-1}(L+U),\ M_{GS} = (D-L)^{-1}U,\ M_{SOR} = \left(\frac{1}{\omega}D - L\right)^{-1}\left(\frac{1-\omega}{\omega}D + U\right)$$

have spectral radii strictly less than 1 (for $\omega \in (0,2)$).

Convergence of Jacobi and Gauss-Seidel — Definitions-1

Definition (Diagonal Dominance)

A matrix is diagonally dominant if

$$\sum_{j \neq i} |a_{ij}| \leq |a_{ii}|$$

for each row i. A row is strictly diagonally dominant of the inequality holds strictly and a matrix is strictly diagonally dominant if each row is strictly diagonally dominant.

This definition is relevant to our discussion since many schemes for elliptic problems give rise to diagonally dominant matrices; the 5- and 9-point Laplacians are two examples.

Convergence of Jacobi and Gauss-Seidel — Definitions-2

Definition (Matrix Permutation)

The permutation of a matrix is the **simultaneous** permutation of the rows and columns of the matrix, *i.e.* $a_{ij} \to a_{\sigma(i),\sigma(j)}$.

Definition (Reducible Matrix)

A matrix is reducible if there is a permutation σ under which A has the structure

$$\begin{bmatrix} A_1 & 0 \\ A_{12} & A_2 \end{bmatrix}$$

where A_1 and A_2 are square matrices. A matrix is **irreducible** if it is not reducible.

Convergence of Jacobi and Gauss-Seidel — Result

We can perform the Jacobi and Gauss-Seidel iterative methods to a general linear system $A\bar{\mathbf{x}} = \bar{\mathbf{b}}$, where we express the matrix A in the form $A = D - L - U$:

$$\bar{\mathbf{x}}^{k+1} = D^{-1}((D - A)\bar{\mathbf{x}}^k + \bar{\mathbf{b}}) = (I - D^{-1}A)\bar{\mathbf{x}}^k - D^{-1}\bar{\mathbf{b}} \quad \textbf{Jacobi}$$

$$\bar{\mathbf{x}}^{k+1} = (D - L)^{-1}(U\bar{\mathbf{x}}^k + \bar{\mathbf{b}}) \quad \textbf{Gauss-Seidel}$$

We notice that the diagonal dominance of a matrix is unaffected by simultaneous row- and column-permutations.

The Gauss-Seidel **method** is dependent on permutations of the matrix, whereas the Jacobi method is not.

Theorem

If A is an irreducibly diagonally dominant matrix, then the Jacobi and Gauss-Seidel methods are convergent.

Successive Over-Relaxation (SOR) — 1 of 4

Without going into details, we summarize some key results for the SOR iteration applied to the finite difference discretization of Laplace's equation in two dimensions using the 5-point Laplacian

$$\left(\frac{1}{\omega}D - L\right)\bar{\mathbf{x}}^{k+1} = \left(\frac{1-\omega}{\omega}D + U\right)\bar{\mathbf{x}}^k + \bar{\mathbf{b}}.$$

The non-zero eigenvalues λ of $M_{\text{SOR}} = \left(\frac{1}{\omega}D - L\right)^{-1}\left(\frac{1-\omega}{\omega}D + U\right)$ are related to the eigenvalues μ of $M_{\text{Jac}} = D^{-1}(L + U)$, by a quadratic equation in $\sqrt{\lambda}$

$$\frac{\lambda + \omega - 1}{\omega\lambda^{1/2}} = \mu.$$

From this relation it can be shown that we must require

$$0 < \omega < 2,$$

in order for $\rho(M_{\text{SOR}}) < 1$.

Successive Over-Relaxation (SOR) — 2 of 4

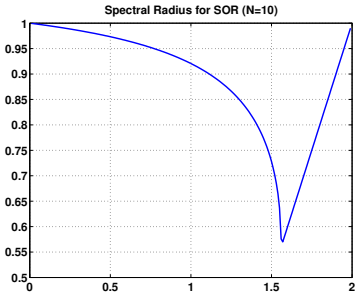
Minimizing $\rho(M_{\text{SOR}})$ with respect to ω gives

$$\omega^* = \frac{2}{1 + \sqrt{1 - \cos^2(\pi/N)}},$$

where the resulting optimal spectral radius

$$\rho^* = \omega^* - 1 \approx 1 - \frac{2\pi}{N},$$

is a dramatic improvement over Jacobi/Gauss-Seidel:



Successive Over-Relaxation (SOR) — 3 of 4

Comparison of spectral radii as a function of the problem size:

2D — 5-Point Laplacian				
n	n^2	$\rho(M_J)$	$\rho(M_{GS})$	$\rho(M_{SOR*})$
8	64	0.9397	0.8830	0.6460
16	256	0.9830	0.9662	0.7698
32	1024	0.9955	0.9910	0.8619
64	4096	0.9988	0.9977	0.9221
96	9216	0.9995	0.9990	0.9455

$$\omega^* = \frac{2}{1 + \sqrt{1 - \cos^2(\pi/n)}}$$

Successive Over-Relaxation (SOR) 4 of 4

It is possible to quantify how many iterations are necessary in order to achieve a prescribed error tolerance; given the spectral radius ρ, we need $\rho^k \approx \epsilon$ in order to reduce the error by a factor ϵ.

From this, we get

$$k_{\text{GS}} \approx \frac{N^2}{\pi^2}\log(\epsilon^{-1})$$

$$k^*_{\text{SOR}} \approx \frac{N}{\pi^2}\log(\epsilon^{-1}).$$

Each iteration requires $\mathcal{O}\left(N^2\right)$ operations, hence the overall work, which should be compared with $\mathcal{O}\left(N^6\right)$ for Gaussian Elimination, is

$$W_{\text{GS}} \approx \frac{N^4}{\pi^2}\log(\epsilon^{-1})$$

$$W^*_{\text{SOR}} \approx \frac{N^3}{\pi^2}\log(\epsilon^{-1}).$$

5-Point Laplacian ⇝ 9-point Laplacian ⇝ SPD Matrices

Unfortunately, the analysis which leads us to an exact expression for the optimal ω for the SOR iteration corresponding to the 5-point Laplacian is quite a bit messier for the fourth order accurate 9-point Laplacian (see next slide).

However, the corresponding matrix is **symmetric** $A = A^T$ and **positive definite** $\lambda(A) > 0$, and there are many useful results for this class of matrices, *e.g.*

Theorem

If A is symmetric positive definite, then the iterative method $B\bar{\mathbf{x}}^{k+1} = C\bar{\mathbf{x}}^k + \bar{\mathbf{b}}$ based on the splitting $A = B - C$ is convergent if

$$\text{Re}(B) > \frac{1}{2}A$$

or, equivalently, that $B^T + C$ is SPD ($B^T + C > 0$).

A Note on the 9-Point Laplacian

There are some results for the optimal relaxation parameter ω for the 9-Point Laplacian:

- [Garabadian-1956] *"Estimation of the Relaxation Factor for Small Mesh Size."* Mathematical Tables and Other Aids to Computation Vol. 10, No. 56 (Oct., 1956), pp. 183-185.

  $$\omega_1^* \approx 2 - 2.04\pi h, \quad \rho_1^* \approx 1 - 2.35\pi h$$

- [Adams-LeVeque-Young-1988] *"Analysis of the SOR Iteration for the 9-Point Laplacian."* SIAM Journal on Numerical Analysis Vol. 25, No. 5 (Oct., 1988), pp. 1156-1180.

  $$\omega_2^* \approx 2 - 2.116\pi h, \quad \rho_2^* \approx 1 - 1.791\pi h$$

  the paper also explores the effect of different orderings of the grid points.

Example #1: SOR for a General Symmetric $A\bar{\mathbf{x}} = \bar{\mathbf{b}}$

The SOR iteration applied to a symmetric matrix of the form

$$A = D - L - L^T, \quad (D > 0)$$

is the iteration $B\bar{\mathbf{x}}^{k+1} = C\bar{\mathbf{x}}^k + \bar{\mathbf{b}}$ where

$$B = \frac{1}{\omega}D - L, \quad C = \frac{1-\omega}{\omega}D + L^T,$$

and using the second form of the theorem, we have

$$B^T + C = \frac{2-\omega}{\omega}D,$$

which is positive definite as long as $\omega \in (0,2)$. Hence the SOR iteration is convergent $\forall \omega \in (0,2)$.

Example #2: Symmetric SOR

Symmetric SOR (SSOR) is the point SOR scheme applied with a forward and backward sweep. Described as a matrix splitting, SSOR is the iteration $B\bar{\mathbf{x}}^{k+1} = C\bar{\mathbf{x}}^k + \bar{\mathbf{b}}$ where

$$B = \frac{\omega}{2-\omega}\left(\frac{1}{\omega}D - L\right)\left(\frac{1}{\omega}D - L^T\right),$$

$$C = \frac{\omega}{2-\omega}\left(\frac{1-\omega}{\omega}D + L\right)\left(\frac{1-\omega}{\omega}D + L^T\right),$$

Since both B and C are symmetric, we apply the second form of the theorem, and with a little bit of algebra we get

$$B + C = \frac{2-\omega}{2\omega}D + \frac{\omega}{2-\omega}\left(\frac{1}{\sqrt{2}}D - \sqrt{2}L\right)\left(\frac{1}{\sqrt{2}}D - \sqrt{2}L\right)^T,$$

which is symmetric positive definite (and therefore the iteration is convergent) for $0 < \omega < 2$.

SSOR vs. SOR

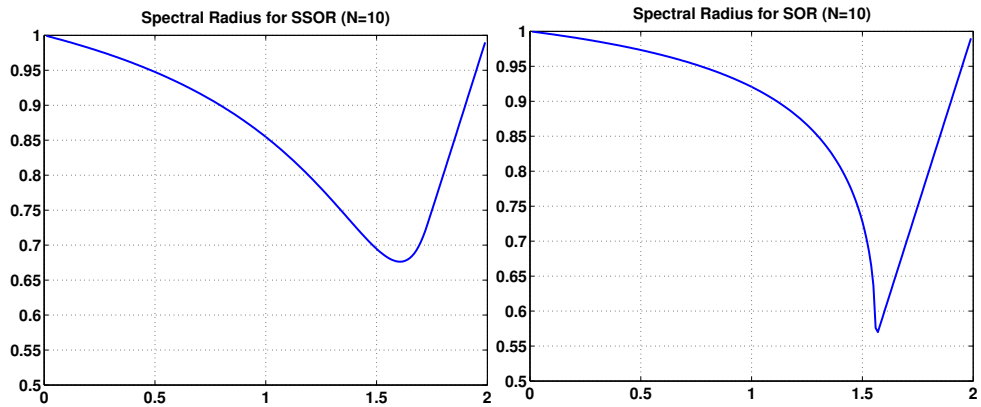


Figure: Surprisingly, the spectral radius for the SSOR iteration, even when optimal, is not as favorable as the one for the SOR iteration.

General Framework: Preconditioning 1 of 3

We "massage" the iterative scheme $B\bar{\mathbf{x}}^{k+1} = C\bar{\mathbf{x}}^k + \bar{\mathbf{b}}$, where $A = B - C$, to the equivalent form

$$\bar{\mathbf{x}}^{k+1} = \underbrace{B^{-1}C}_{G}\bar{\mathbf{x}}^k + \underbrace{B^{-1}\bar{\mathbf{b}}}_{\bar{\mathbf{f}}},$$

where

$$G = B^{-1}C = B^{-1}(B - A) = I - B^{-1}A.$$

The iteration $\bar{\mathbf{x}}^{k+1} = G\bar{\mathbf{x}}^k + \bar{\mathbf{f}}$, can be viewed as a technique for solving **the preconditioned system**

$$(I - G)\bar{\mathbf{x}} = \bar{\mathbf{f}} \quad \Leftrightarrow \quad B^{-1}A\bar{\mathbf{x}} = B^{-1}\bar{\mathbf{b}},$$

where B is the **preconditioner** ($B \approx A$, and the effect of B^{-1} easily computed.)

Preconditioning 2 of 3

Finding good preconditioners is an art/science in itself; here we have talked about the Jacobi, Gauss-Seidel, SOR, and SSOR preconditioners.

The general goals of a preconditioner $B \approx A$ is that

- B captures most of the "action" of A:
  this means $\rho(B^{-1}(B - A)) \ll 1 \Leftrightarrow B^{-1}A \approx I$.
  The theorem on slide 14 quantifies the minimal amount of "action" B must capture for an SPD matrix A.
- The effect of B^{-1} should be significantly easier to compute than the effect of A^{-1}.

Since the **Thomas Algorithm** for tri-diagonal matrices solves $T\bar{\mathbf{v}} = \bar{\mathbf{b}}$ in $\mathcal{O}(n)$ operations, letting B be the tri-diagonal part of A is sometimes a useful preconditioner. This is equivalent to the **Line SOR** approach described in Strikwerda (p.359).

Preconditioning

Many of our "old" algorithms, *e.g.* the Peaceman-Rachford alternating direction implicit (ADI) scheme, can be viewed from a matrix-centric point of view as a preconditioned iteration with tri-diagonal preconditioners.

The **alternating direction** part corresponds to the numbering-order of the grid points:

- When we solve in the x-direction, we enumerate the grid-points along the x-axis first, so that the neighboring points in that direction correspond to the first super- and sub-diagonal elements in the matrix.
- When we solve in the y-direction, we enumerate the grid-points along the y-axis first, so that the neighboring points in that direction correspond to the first super- and sub-diagonal elements in the matrix.