Today’s Agenda

- Iterative methods for solving linear systems
  - (Gauss) Jacobi (GJ)
  - Gauss Seidel (GS)
  - Successive Over-Relaxation (SOR)

- Case study: Solving PDE

Golub and Van Loan, *Matrix Computations 4th edition*, Johns Hopkins Press

GJ v.s. GS

- GJ: $x^{(k)} = D^{-1}b - D^{-1}(L+U)\,x^{(k-1)}$

$$
\begin{aligned}
&\textbf{for } i = 1{:}n \\
&\qquad x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k-1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right) \Big/ a_{ii} \\
&\textbf{end}
\end{aligned}
$$

- GS: $x^{(k)} = (L+D)^{-1}b - (L+D)^{-1}Ux^{(k-1)}$

$$
\begin{aligned}
&\textbf{for } i = 1{:}n \\
&\qquad x_i^{(k)} = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} \right) \Big/ a_{ii} \\
&\textbf{end}
\end{aligned}
$$

Splitting and convergence

- A splitting $A = M - N$ is proposed when $Mz = d$ is “easy” to solve.

- For GJ and GS, diagonal elements of A should be nonzero; otherwise not invertible.

- Spectral radius of the matrix $M^{-1}N$ must be bounded above 1 for convergence: $\rho(M^{-1}N) < 1$ and the smaller the better.

GJ vs GS

Consider four systems, with matrix A_i given below.
Consider their iteration matrices G_J and G_S.

$$A_1 = \begin{bmatrix} 3 & 0 & 4 \\ 7 & 4 & 2 \\ -1 & 1 & 2 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} -3 & 3 & -6 \\ -4 & 7 & -8 \\ 5 & 7 & -9 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 4 & 1 & 1 \\ 2 & -9 & 0 \\ 0 & -8 & -6 \end{bmatrix}, \qquad A_4 = \begin{bmatrix} 7 & 6 & 9 \\ 4 & 5 & -4 \\ -7 & -3 & 8 \end{bmatrix}.$$

Matlab:

```
D = diag(diag(A));
L = tril(A,-1);
U = triu(A,1);

GJ = -inv(D)*(L+U);
GS = -inv(L+D)*U;

max(abs(eig(GJ)))
max(abs(eig(GS)))
```

GJ vs GS

- For the system given by A_1, $\rho(G_J)>1$ but $\rho(G_S)<1$ → GJ diverges but GS converges
- For the system given by A_2, $\rho(G_J)<1$ but $\rho(G_S)>1$ → GJ converges but GS diverges
- For the system given by A_3, $\rho(G_J)=0.44$ and $\rho(G_S)=0.018$ → both converge but GJ typically converges slower than GS
- For the system given by A_4, $\rho(G_J)=0.64$ and $\rho(G_S)=0.77$ → both converge but GJ typically converges faster than GS

→ Take-away: there isn't a "one size fits all" answer for algorithm selection.

Motivation of a new scheme

- An example

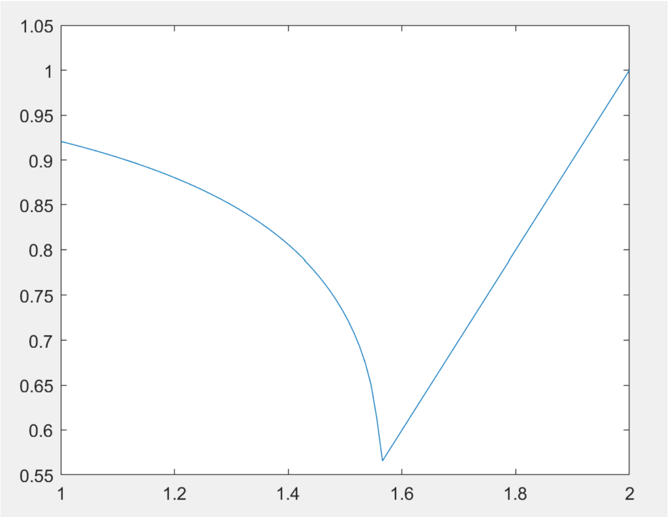
```
N = 11; A = toeplitz([2 -1 zeros(1,N-3)])
D=diag(diag(A)); U= triu(A,1); L=tril(A,-1);

max(abs(eig(-inv(D+L) *U)))
```

- Observe, A = M – N = (L + D) – (-U) = $(L + \frac{1}{w} D) - ((\frac{1}{w} - 1)D - U)$ for any w…

- In fact

```
G = @(w) (D/w+L)\((1./w-1).*D-U);
SR = @(w) max(abs(eig(G(w))));
w = linspace(1,2);
for j=1:length(w)
   f(j) = SR(w(j));
end
plot(w,f)
```



SOR

- Successive over relaxation (SOR)
- Write $A = L + D + U + \frac{1}{w}D - \frac{1}{w}D$
- $Ax = b \rightarrow (L + D + U + \frac{1}{w}D - \frac{1}{w}D)x = b \rightarrow$

  $(L+\frac{1}{w}D)x = b - (D - \frac{1}{w}D + U)x \rightarrow$

$$x^{(k)} = (L+\frac{1}{w}D)^{-1}b - (L+\frac{1}{w}D)^{-1}(U+D-\frac{1}{w}D)\, x^{(k-1)}$$

SOR (cont'd)

- Recall

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

- Written iteratively, we get…

GS-update

for $i = 1{:}n$

$$x_i^{(k)} = \omega \boxed{\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}\right) \Big/ a_{ii}} + (1-\omega) x_i^{(k-1)}$$

end

Relaxation term

Momentum term

SOR convergence

- SOR reduces to GS when w=1.
- A necessary condition for SOR to converge is 0<w<2.
- When the matrix A is SPD, then the condition becomes sufficient!

A case study

Solving the Poisson equation

Problem setup

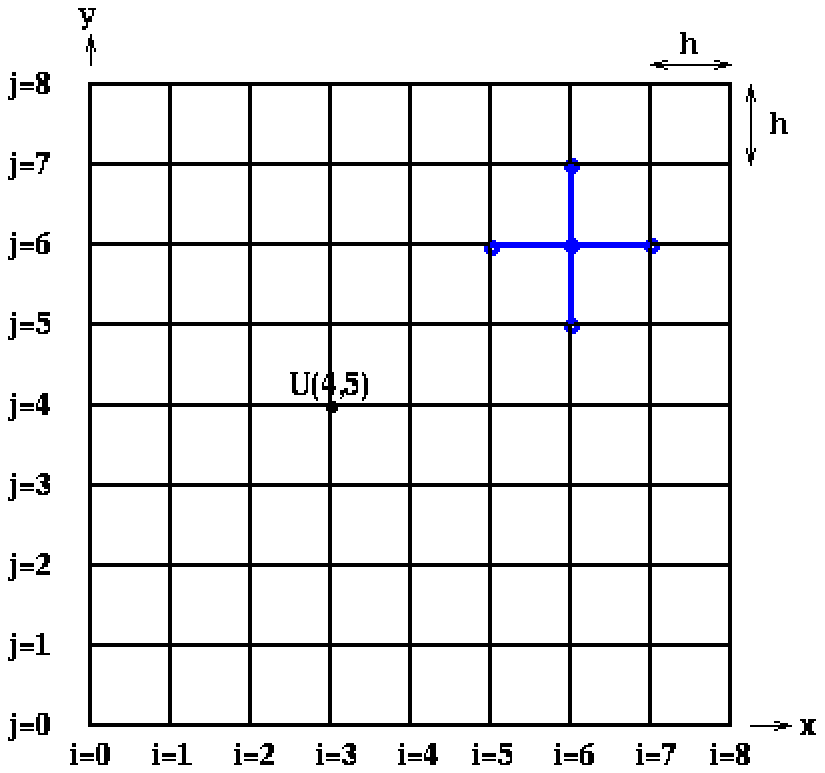
- Poisson equation

```
d^2 u(x,y)   d^2 u(x,y)
---------- + ----------  = f(x,y)
  d x^2        d y^2
```

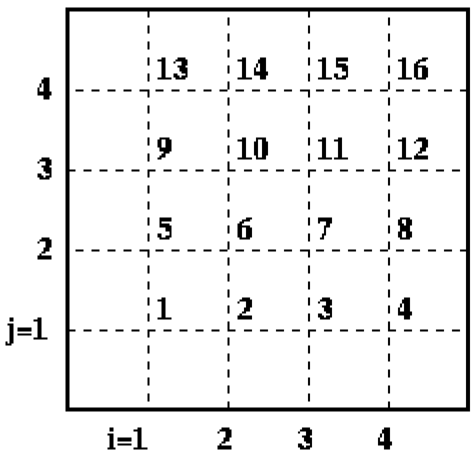
```
u(x,y) = 0 if (x,y) is on the boundary of Omega
```

- Discretization

4*U(i,j) – U(i–1,j) – U(i+1,j) – U(i,j–1) – U(i,j+1) = b(i,j)



Small-scale example



$$
\left[\begin{array}{cccccccccccccccc}
4 & -1 & & & -1 \\
-1 & 4 & -1 & & & -1 \\
& -1 & 4 & -1 & & & -1 \\
& & -1 & 4 & & & & -1 \\
-1 & & & & 4 & -1 & & & -1 \\
& -1 & & & -1 & 4 & -1 & & & -1 \\
& & -1 & & & -1 & 4 & -1 & & & -1 \\
& & & -1 & & & -1 & 4 & & & & -1 \\
& & & & -1 & & & & 4 & -1 & & & -1 \\
& & & & & -1 & & & -1 & 4 & -1 & & & -1 \\
& & & & & & -1 & & & -1 & 4 & -1 & & & -1 \\
& & & & & & & -1 & & & -1 & 4 & & & & -1 \\
& & & & & & & & -1 & & & & 4 & -1 \\
& & & & & & & & & -1 & & & -1 & 4 & -1 \\
& & & & & & & & & & -1 & & & -1 & 4 & -1 \\
& & & & & & & & & & & -1 & & & -1 & 4
\end{array}\right]
*
\begin{bmatrix}
U(1,1) \\ U(2,1) \\ U(3,1) \\ U(4,1) \\ U(1,2) \\ U(2,2) \\ U(3,2) \\ U(4,2) \\ U(1,3) \\ U(2,3) \\ U(3,3) \\ U(4,3) \\ U(1,4) \\ U(2,4) \\ U(3,4) \\ U(4,4)
\end{bmatrix}
=
\begin{bmatrix}
b(1,1) \\ b(2,1) \\ b(3,1) \\ b(4,1) \\ b(1,2) \\ b(2,2) \\ b(3,2) \\ b(4,2) \\ b(1,3) \\ b(2,3) \\ b(3,3) \\ b(4,3) \\ b(1,4) \\ b(2,4) \\ b(3,4) \\ b(4,4)
\end{bmatrix}
$$

Solving the Poisson equation

Recall

4*U(i,j) – U(i–1,j) – U(i+1,j) – U(i,j–1) – U(i,j+1) = b(i,j)

- GS: U(i,j) = [U(i-1,j)+U(i+1,j)+U(i,j+1)+U(i,j-1) + b(i,j)]/4

- SOR: V(i,j) = [U(i-1,j)+U(i+1,j)+U(i,j+1)+U(i,j-1) + b(i,j)]/4
  U(i,j) = U(i,j) + w(V(i,j)-U(i,j))