Solution Methods:

Iterative Techniques

Lecture 6

Motivation

Consider a standard second order finite difference discretization of

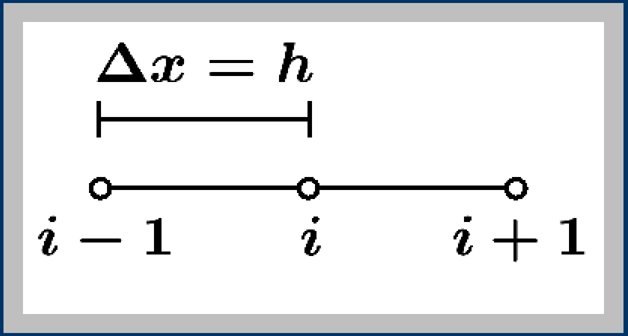
$$-\nabla^2 u = f,$$

on a regular grid, in 1, 2, and 3 dimensions.

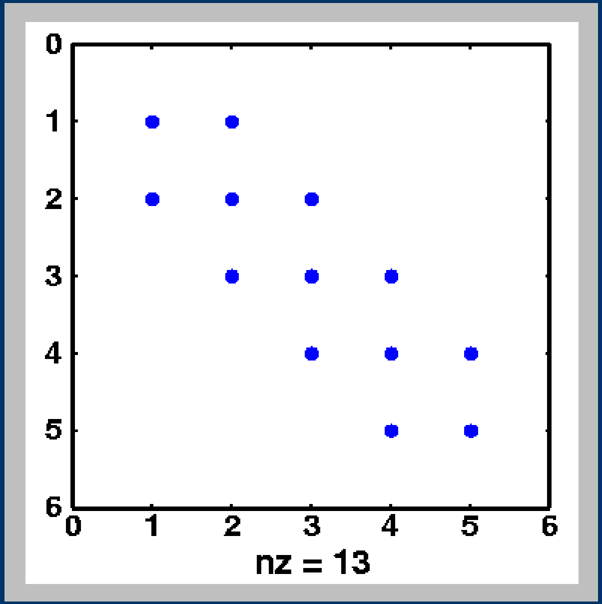
$$\Rightarrow \qquad \boxed{A\,u = f}$$

Motivation

1D Finite Differences



$\rightarrow$



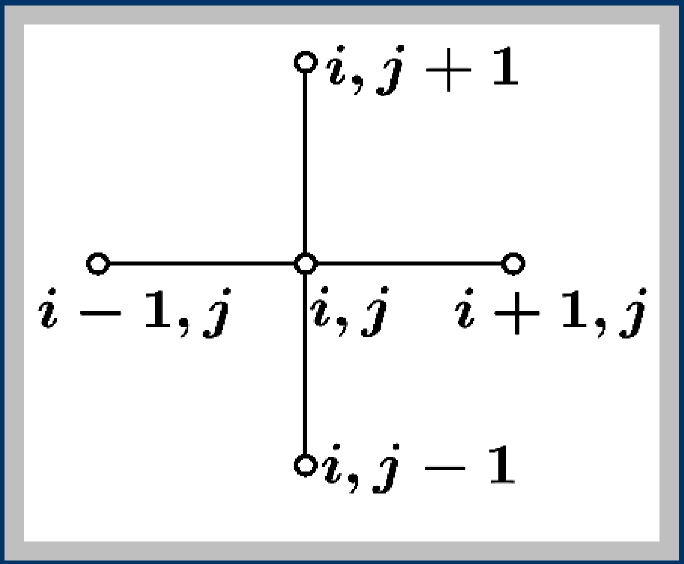
n points

$n \times n$ matrix
bandwidth $b = 1$

Cost of Gaussian elimination $O(b^2 n) = O(n)$

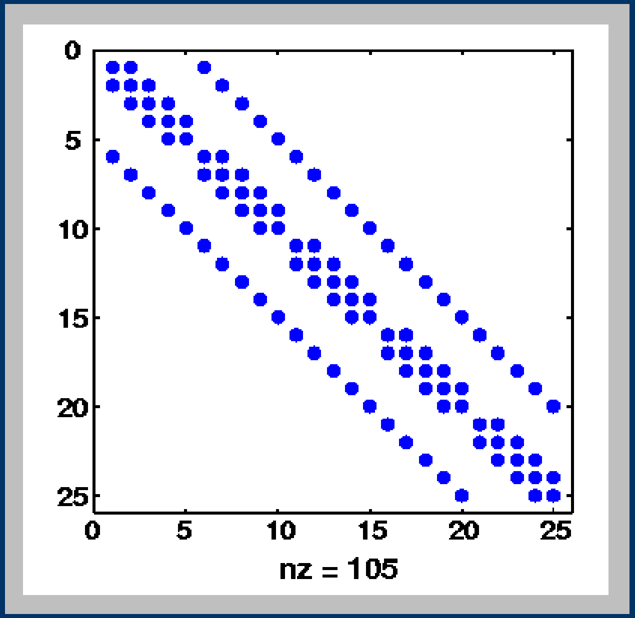
Motivation

2D Finite Differences



$n \times n$ points

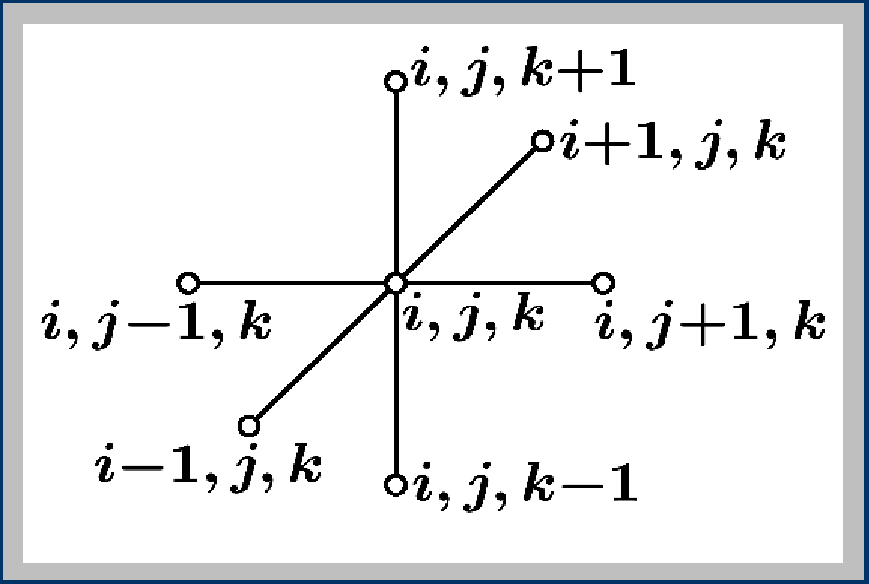
$\rightarrow$



$n^2 \times n^2$ matrix
bandwidth $b = n$

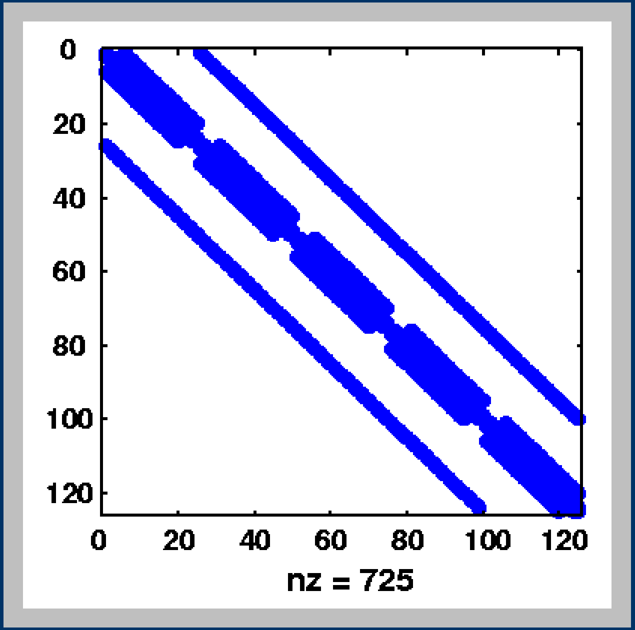
Cost of Gaussian elimination $O(b^2 n^2) = O(n^4)$

Motivation



$n \times n \times n$ points

→



$n^3 \times n^3$ matrix
bandwidth $b = n^2$

Cost of Gaussian elimination $O(b^2 n^3) = O(n^7)$!

Basic Iterative Methods

Jacobi

Intuitive Interpretation...

Instead of solving

$$-u_{xx} = f,$$

we solve

$$\frac{\partial u}{\partial t} = u_{xx} + f,$$

N1

starting from an arbitrary $u(x, 0)$.

We expect $u(x, t \to \infty) \to u(x)$.

Basic Iterative Methods

Jacobi

...Intuitive Interpretation...

To solve

$$\frac{\partial u}{\partial t} = u_{xx} + f$$

we use an inexpensive (explicit) method.

For instance, $$\frac{\hat{u}_i^{r+1} - \hat{u}_i^r}{\Delta t} = \frac{\hat{u}_{i+1}^r - 2\hat{u}_i^r + \hat{u}_{i-1}^r}{h^2} + \hat{f}_i$$

$$u = \{\hat{u}_i\}_{i=1}^n, \quad f = \{\hat{f}_i\}_{i=1}^n$$

$$u^{r+1} = u^r + \Delta t(f - Au^r) = (I - \Delta t A)\, u^r + \Delta t f\,.$$

Basic Iterative Methods

Jacobi

...Intuitive Interpretation

Stability dictates that

$$\Delta t \leq \frac{h^2}{2}$$

Thus, we take Δt as large as possible, i.e. $(\Delta t = h^2/2)$.

$$u^{r+1} = \left(I - \frac{h^2}{2} A\right) \hat{u}^r + \frac{h^2}{2} f$$

$$\Rightarrow \qquad \hat{u}_i^{r+1} = \frac{1}{2} (\hat{u}_{i+1}^r + \hat{u}_{i-1}^r + h^2 \hat{f}_i) \quad \text{for } i = 1, \ldots n.$$

Basic Iterative Methods

Jacobi

Matrix Form...

Split A

$$A = D - L - U \quad \begin{cases} D: \text{ Diagonal} \\ L: \text{ Lower triangular} \\ U: \text{ Upper triangular} \end{cases}$$

$$A\,u = f \quad \text{becomes} \quad (D - L - U)\,u = f$$

Iterative method

$$\boxed{D\,u^{r+1} = (L + U)\,u^{r} + f}$$

Basic Iterative Methods

Jacobi

...Matrix Form

$$u^{r+1} = D^{-1}(L+U)\,u^r + D^{-1}\,f$$
$$= D^{-1}(D-A)\,u^r + D^{-1}\,f$$
$$= (I - D^{-1}A)\,u^r + D^{-1}\,f$$

$$\boxed{u^{r+1} = R_J\,u^r + D^{-1}\,f}$$

$R_J = (I - D^{-1}A)$: **Jacobi Iteration Matrix**

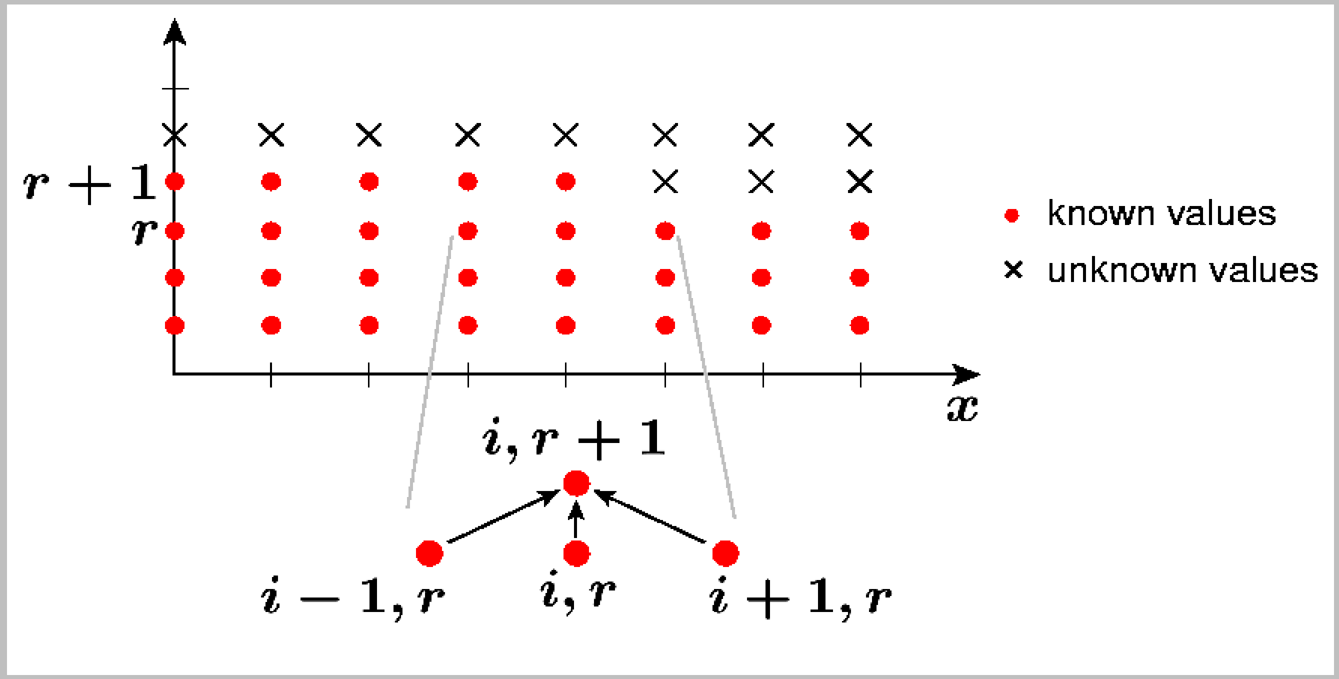
$D_{ii}^{-1} = h^2/2$

$\Rightarrow \quad \hat{u}_i^{r+1} = \frac{1}{2}\left(\hat{u}_{i+1}^r + \hat{u}_{i-1}^r + h^2\,\hat{f}_i\right)$ for $i = 1, \ldots n$

Basic Iterative Methods

Jacobi

Implementation

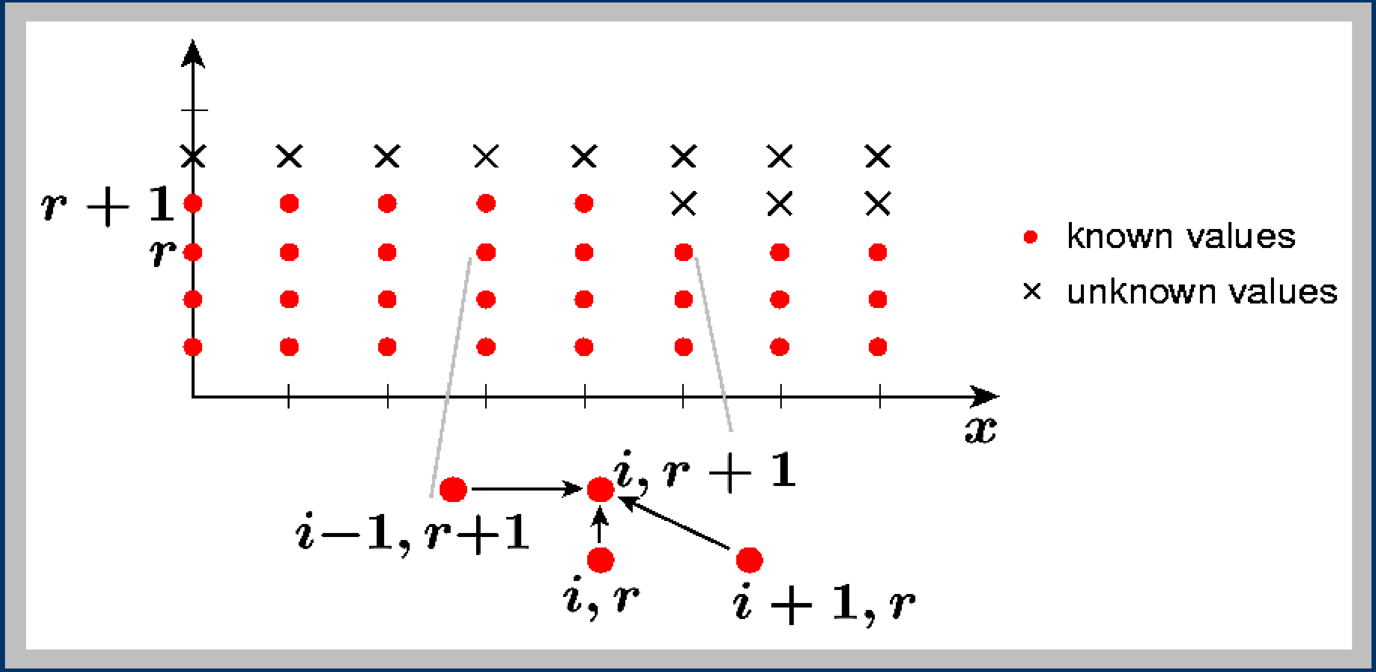


Jacobi iteration

$$u_i^{r+1} = \frac{1}{2}\left(u_{i+1}^r + u_{i-1}^r + h^2 f_i\right)$$

Basic Iterative Methods

Gauss-Seidel



Gauss-Seidel iteration (consider most recent iterate)

$$u_i^{r+1} = \frac{1}{2}\left(u_{i+1}^{r} + u_{i-1}^{r+1} + h^2 f_i\right)$$

Basic Iterative Methods

Gauss-Seidel

Matrix Form...

Split A

$$A = D - L - U \quad \begin{cases} D\text{: Diagonal} \\ L\text{: Lower triangular} \\ U\text{: Upper triangular} \end{cases}$$

$$A\, u = f \quad \text{becomes} \quad (D - L - U)\, u = f$$

Iterative method

$$\boxed{(D - L)\, u^{r+1} = U\, u^r + f}$$

Basic Iterative Methods

Gauss-Seidel

...Matrix Form

$$u^{r+1} = (D - L)^{-1} U\, u^r + (D - L)^{-1} f$$

$$\boxed{u^{r+1} = R_{\mathrm{GS}}\, u^r + (D - L)^{-1} f}$$

$R_{\mathrm{GS}} = (D - L)^{-1} U$: **Gauss-Seidel Iteration Matix**

Basic Iterative Methods

Error Equation

Let $\boldsymbol{u}$ be the soution of $\boldsymbol{A}\,\boldsymbol{u} = \boldsymbol{f}$.

For an approximate solution $\boldsymbol{u}^r$, we define

Iteration Error: $\quad \boldsymbol{e}^r = \boldsymbol{u} - \boldsymbol{u}^r$

Residual: $\quad \boldsymbol{r}^r = \boldsymbol{f} - \boldsymbol{A}\,\boldsymbol{u}^r$

$$\boldsymbol{A}\,\boldsymbol{u} \; - \boldsymbol{A}\,\boldsymbol{u}^r = \boldsymbol{f} \; - \boldsymbol{A}\,\boldsymbol{u}^r$$

ERROR EQUATION $\rightarrow$ $\boxed{\boldsymbol{A}\,\boldsymbol{e}^r = \boldsymbol{r}^r}$

N3

Basic Iterative Methods

Error Equation

Jacobi

$$\boldsymbol{u}^{r+1} = \boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})\,\boldsymbol{u}^{r} + \boldsymbol{D}^{-1}\boldsymbol{f}$$

$$\boldsymbol{u} \quad\; = \boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})\,\boldsymbol{u} + \boldsymbol{D}^{-1}\boldsymbol{f}$$

subtracting

$$\boldsymbol{e}^{r+1} = \underbrace{\boldsymbol{D}^{-1}(\boldsymbol{L}+\boldsymbol{U})}_{\boldsymbol{R}_{\mathrm{J}}}\,\boldsymbol{e}^{r} = \boldsymbol{R}_{\mathrm{J}}\,\boldsymbol{e}^{r}$$

Basic Iterative Methods

Error Equation

Gauss-Seidel

Similarly,

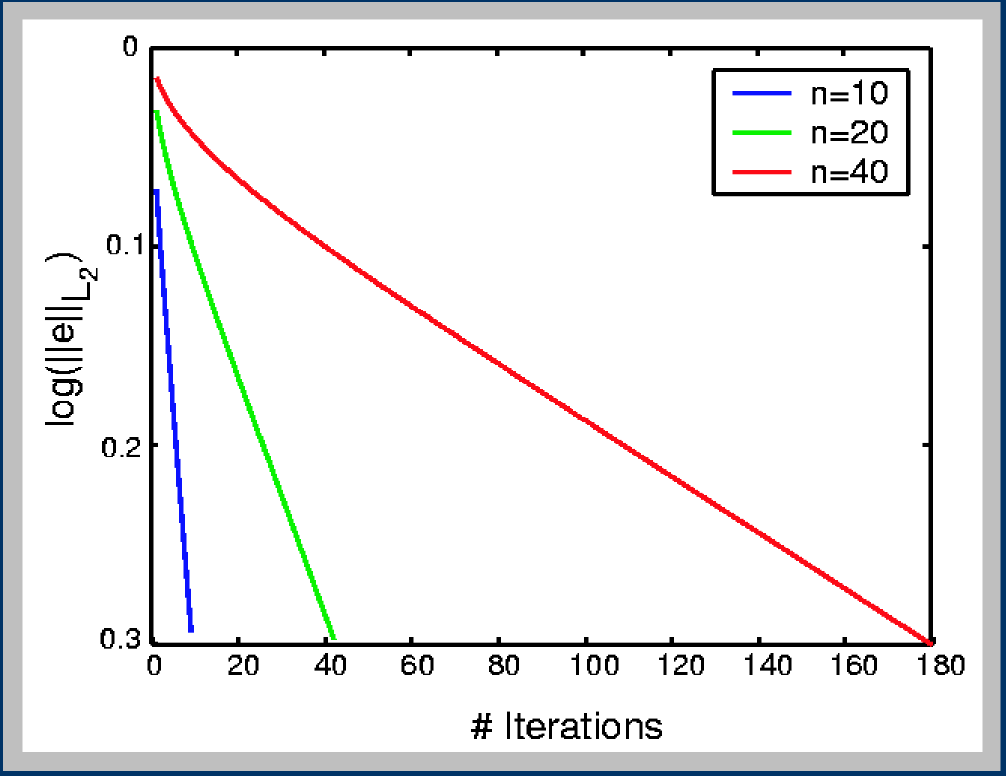
$$e^{r+1} = \underbrace{(D - L)^{-1}\,U}_{R_{\mathrm{GS}}}\; e^{r} = R_{\mathrm{GS}}\, e^{r}$$

Basic Iterative Methods

Examples

Jacobi

$-u_{xx} = 1 \qquad u(0) = u(1) = 0; \qquad u^0 = 0$

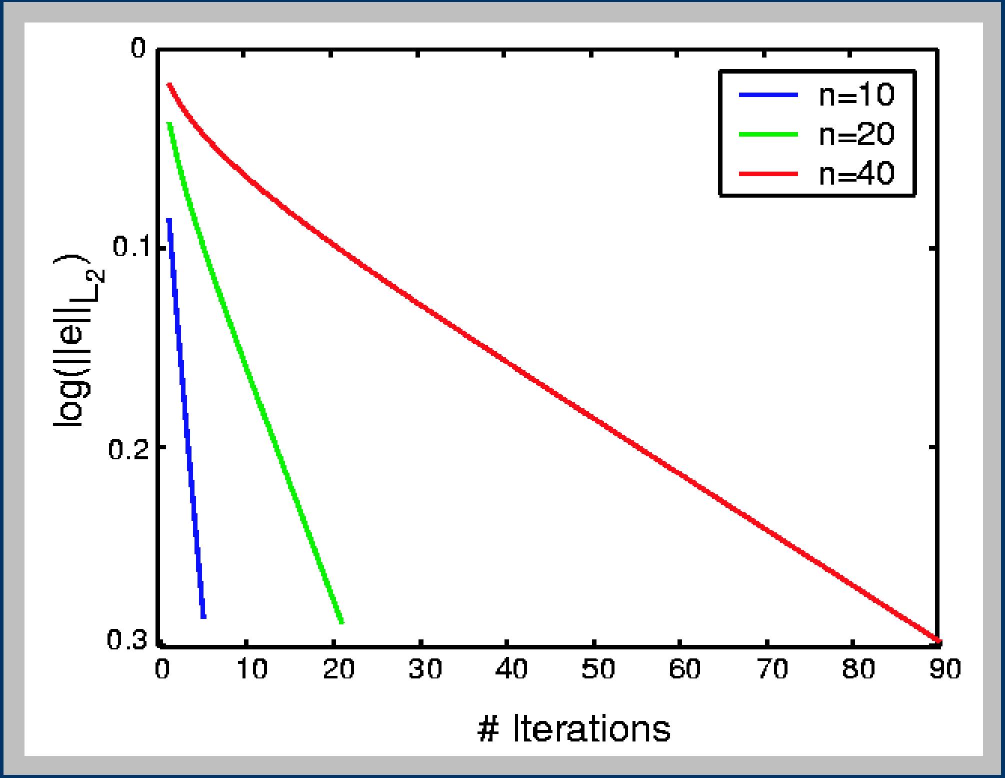


Basic Iterative Methods

Examples

Gauss-Seidel

$$-u_{xx} = 1 \qquad u(0) = u(1) = 0; \qquad u^0 = 0$$



Basic Iterative Methods

Examples

Observations

– The number of iterations required to obtain a certain level of convergence is $O(n^2)$.

– Gauss-Seidel is about twice as fast as Jacobi.

Why?

Convergence Analysis

$$e^r = R\, e^{r-1} = R\, R\, e^{r-2} = \cdots = R^r\, e^0 \,.$$

The iterative method will converge if

$$\lim_{r \to \infty} R^r = 0 \iff \rho(R) = \max |\lambda(R)| < 1 \,.$$ N4

$\rho(R)$ is the spectral radius.

Convergence Analysis

Theorems

– If the matrix A is *strictly diagonally dominant* then Jacobi and Gauss-Seidel iterations converge starting from an arbitrary initial condition.

N5

$A = \{a_{i\,j}\}$ is strictly diagonally dominant if

$$|a_{i\,j}| > \sum_{j=1}^{n} |a_{i\,j}| \text{ for all } i.$$

– If the matrix A is symmetric positive definite then Gauss-Seidel iteration converges for any initial solution.

N6

Convergence Analysis

Jacobi

$$R_J = D^{-1}(L+U) = D^{-1}(D-A) = I - D^{-1}A = I - \frac{h^2}{2}A$$

If $\quad A\,v^k = \lambda^k\,v^k$,

then $R_J\,v^K = \left(I - \frac{h^2}{2}A\right)v^k = \underbrace{\left(1 - \frac{h^2}{2}\lambda^k(A)\right)}_{\lambda^k(R_J)}v^k$

$$\lambda^k(R_J) = 1 - \frac{h^2}{2}\lambda^k(A)$$

Eigenvectors of $R_J \equiv$ Eigenvectors of A

Convergence Analysis

Jacobi

Recall . . .

$$Av^k = \lambda^k v^k \qquad k = 1, \ldots, n$$

$$A = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \vdots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

$n \times n$ *SPD*

Eigenvectors: $h = \frac{1}{n+1}$

$$v_j^k = \sin(k\pi hj) = \sin\left(\frac{k\pi j}{n+1}\right)$$

Eigenvalues:

$$\lambda^k(A) = \frac{2}{h^2}\left[1 - \cos(k\pi h)\right]$$

Convergence Analysis

Jacobi

$$\lambda^k(R_J) = 1 - \frac{h^2}{2}\lambda^k(A) = 1 - [1 - \cos k\pi h]$$

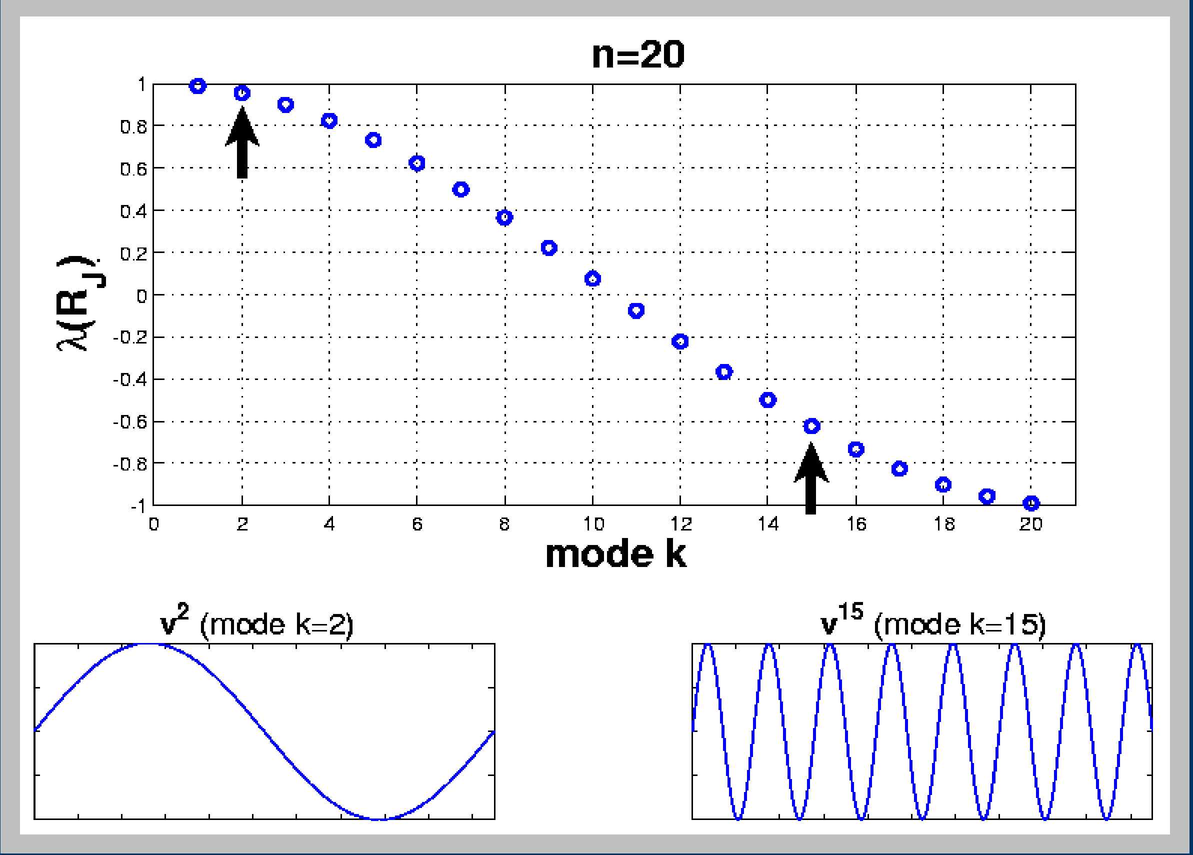
$$= \cos\frac{k\pi}{n+1} < 1, \qquad k = 1, \ldots, n$$

Jacobi converges for our model problem.

$$R_J v^k = \lambda^k(R_J) v^k$$

Convergence Analysis

Jacobi



Convergence Analysis

Jacobi

Write $e^0 = \sum_{k=1}^{n} c_k \, v^k$ $\qquad v^k$, k-th eigenvector of R_J

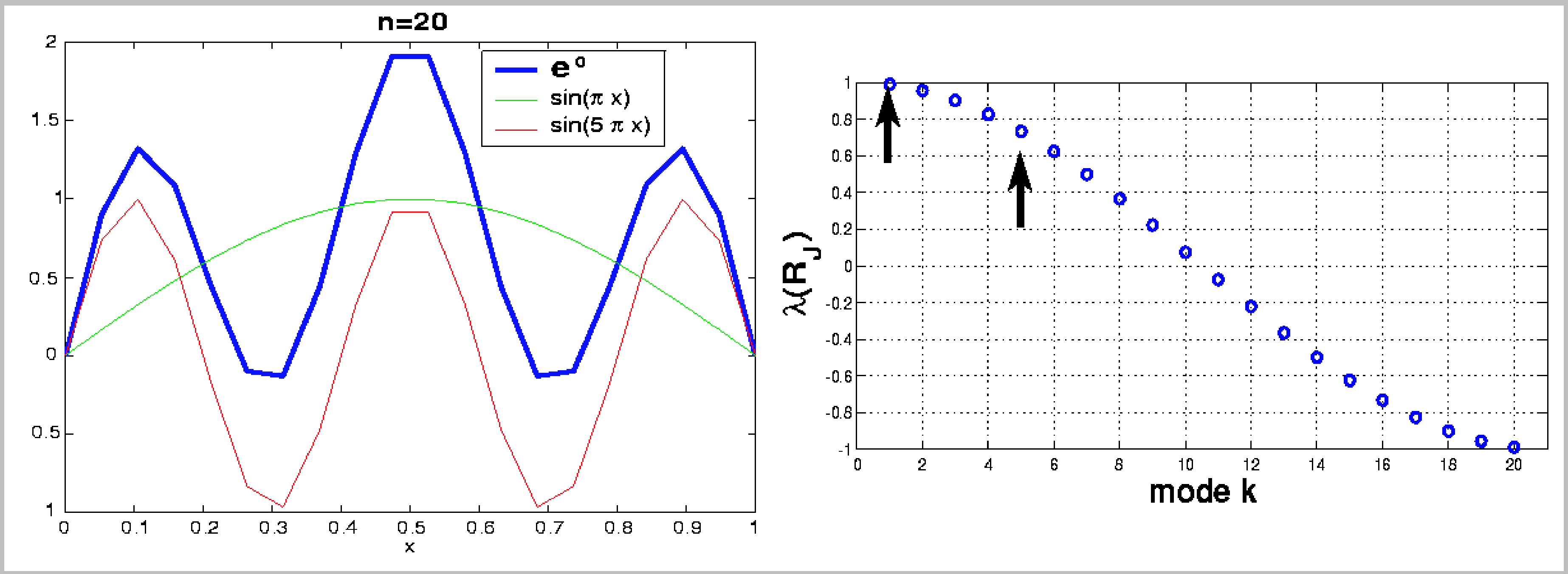
$$e^1 = R\, e^0 = \sum_{k=1}^{n} c_k \, \lambda^k(R_J)\, v^k$$

$$e^r = R^r\, e^0 = \sum_{k=1}^{n} c_k \, (\lambda^k(R_J))^r\, v^k$$

Convergence Analysis

Jacobi

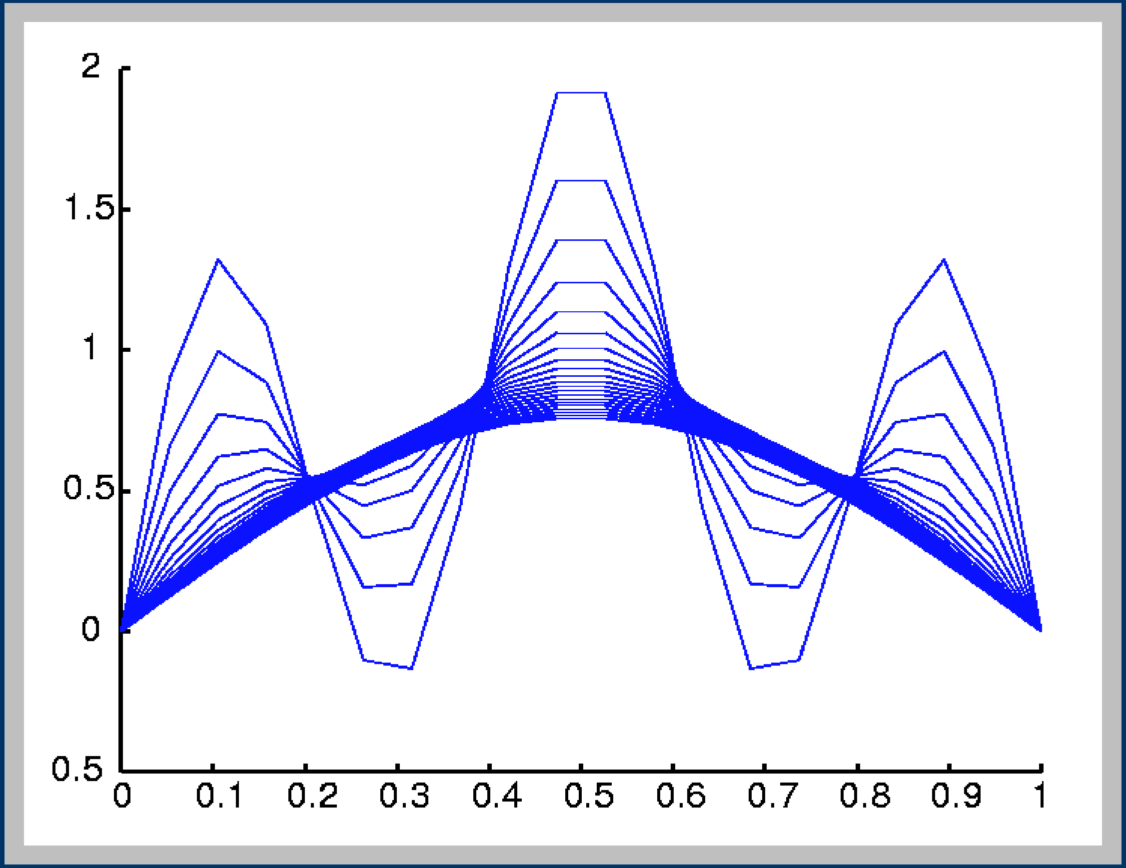
Example...



Convergence Analysis

Jacobi

...Example



Convergence Analysis

Jacobi

Convergence rate...

$$\lambda^k(R_J) = cos(k\pi h), \qquad k = 1, \ldots n$$

Largest $|\lambda^k(R_J)|$ for $k = 1, \ldots n$

Worst case $e^0 = c_1 v^1 \rightarrow e^r = c_1\, \rho(R_J)^r\, v^1$

$$\frac{||e^r||}{||e^0||} = (\cos(\pi h))^r \simeq \left(1 - \frac{\pi^2 h^2}{2}\right)^r$$

Convergence Analysis

Jacobi

...Convergence rate

To obtain a given level of convergence; e.g., $\mathbf{10^{-\delta}}$

$$\frac{\|e^r\|}{\|e^0\|} < 10^{-\delta}$$

$$\Rightarrow \left(1 - \frac{\pi^2 h^2}{2}\right)^r < 10^{-\delta} \rightarrow r = \frac{-\delta}{\log(1 - \frac{\pi^2 h^2}{2})} \simeq \frac{2\delta}{\pi^2 h^2} = \frac{2\delta(n+1)^2}{\pi^2}$$

$$\rightarrow \quad \boxed{r = O(n^2)}$$

Convergence Analysis

Jacobi

Convergence rate (2D)

In two dimensions, and for a uniform, $n \times n$, grid,

we have

$$\lambda^{k\ell}(R_J) = 1 - \frac{h^2}{4}\lambda(A) = \frac{1}{2}[\cos(k\pi h) + \cos(\ell\pi h)]$$

$$\rho(R_J) = \cos(\pi h) = \cos\left(\frac{\pi}{h+1}\right)$$

$h = \frac{1}{n+1}$

Therefore, $\boxed{r = O(n^2)}$

Convergence Analysis

Gauss-Seidel

$$R_{\text{GS}} = (D - L)^{-1} U$$

$$\boxed{\lambda^k(R_{\text{GS}}) = \cos^2(k\pi h) = [\lambda^k(R_{\text{J}})]^2 < 1}$$

Gauss-Seidel converges for our problem

But,

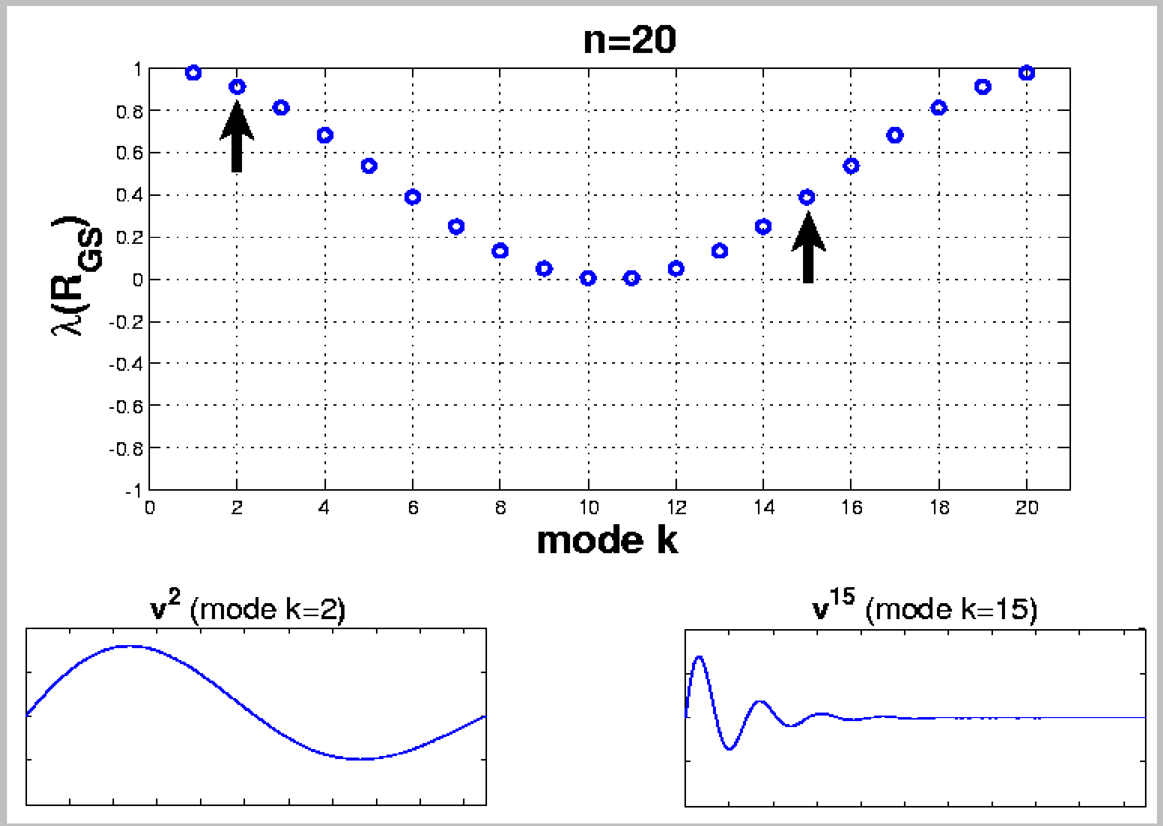
Eigenvectors of $R_{\text{GS}} \neq$ Eigenvectors of A

The eigenvectors $v^k = \{v_j^k\}_{j=1}^n$ of R_{GS} are

$$v_j^k = [\sqrt{\lambda^k(R_{\text{GS}})}]^j \sin(k\pi h j)$$

Convergence Analysis

Gauss-Seidel



Convergence Analysis

Gauss-Seidel

Convergence rate

To obtain a given level of convergence; e.g., $10^{-\delta}$

$$\frac{\|e^r\|}{\|e^0\|} < 10^{-\delta}$$

$$\Rightarrow \left(1 - \frac{\pi^2 h^2}{2}\right)^{2r} < 10^{-\delta} \rightarrow r = \frac{-\delta}{2\log(1 - \frac{\pi^2 h^2}{2})} \simeq \frac{\delta}{\pi^2 h^2} = \frac{\delta(n+1)^2}{\pi^2}$$

$$\rightarrow \quad \boxed{r = O(n^2)}$$

Comparative cost

		Iteration	Gauss Elimination
1D	$n \times n$	$O(n^2\, n) = O(n^3)$	$O(n)$
2D	$n^2 \times n^2$	$O(n^2\, n^2) = O(n^4)$	$O(n^4)$
3D	$n^3 \times n^3$	$O(n^2\, n^3) = O(n^5)$	$O(n^7)$

red # iters
green cost/iter

Over/Under Relaxation

Gauss-Seidel

Main Idea

Typically

$$u^{r+1} = R\,u^r + f^*$$

$f^* = D^{-1} f$ Jacobi

$f^* = (D - L)^{-1} f$ Gauss-Seidel

Can we "extrapolate" the changes?

$$u^{r+1} = \omega(Ru^r + f^*) + (1-\omega)\,u^r$$

$$= \underbrace{[\omega\,R + (1-\omega)\,I]}_{R_\omega}\,u^r + \omega f^*$$

$\omega > 0$

Over/Under Relaxation

How large can we choose ω?

$$\lambda^k(R_\omega) = \omega\,\lambda^k(R) + (1-\omega)$$

Jacobi $\quad \lambda^k(R_J) = \cos k\pi h$

GS $\quad \lambda^k(R_{GS}) = \cos^2 k\pi h$

$\Rightarrow \quad \rho(R_\omega) < 1 \Rightarrow 0 \leq \omega_J \leq 1$ can only be under-relaxed

$0 \leq \omega_{GS} \leq 2$ can be over-relaxed

N7