Solution Methods:

Iterative Techniques

Lecture 6

1 Motivation

SLIDE 1

Consider a standard second order finite difference discretization of

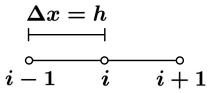
$$-\nabla^2 u = f,$$

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

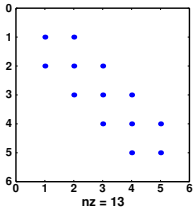
$$\Rightarrow \qquad \boxed{A\,\boldsymbol{u} = \boldsymbol{f}}$$

1.1 1D Finite Differences

SLIDE 2



$\rightarrow$



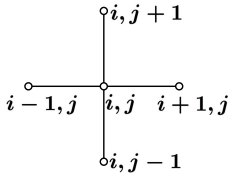
$n \times n$ matrix
bandwidth $b = 1$

n points

Cost of Gaussian elimination $O(b^2 n) = \boldsymbol{O(n)}$

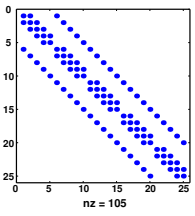
1.2 2D Finite Differences

SLIDE 3



$n \times n$ points

$\rightarrow$

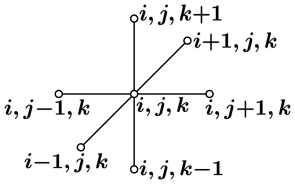


$n^2 \times n^2$ matrix
bandwidth $b = n$

Cost of Gaussian elimination $O(b^2 n^2) = \boldsymbol{O(n^4)}$

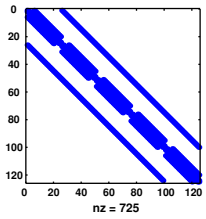
1.3 3D Finite Differences

SLIDE 4



$n \times n \times n$ points

$\rightarrow$



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

Cost of Gaussian elimination $O(b^2 n^3) = \boldsymbol{O(n^7)}$ **!**

This means that we we halve the grid spacing, we will have 8 times (2^3) more unknowns and the cost of solving the problem will increase by a factor of 128 (2^7). It is apparent that, at least for practical three dimensional problems, faster methods are needed.

2 Basic Iterative Methods

2.1 Jacobi

2.1.1 Intuitive Interpretation

SLIDE 5

Instead of solving

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

we solve

$$\frac{\partial u}{\partial t} = u_{xx} + f, \qquad \boxed{\text{N1}}$$

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

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

That is, we expect the solution of the time dependent problem to converge to the solution of the steady state problem.

Note 1 **Solution by relaxation**

By adding the time dependent term $\frac{\partial u}{\partial t}$, we have now a parabolic equation. If the boundary conditions are kept fixed, we expect the solution to "converge" to the steady state solution (i.e., $\frac{\partial u}{\partial t} = 0$). Recall that the time dependent heat transfer problem is modeled by this equation. In this case, we expect the temperature to settle to a steady state distribution provided that the heat source, f, and the boundary conditions, do not depend on t.

SLIDE 6

To solve

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

we use an inexpensive (explicit) method. *Thus avoiding the need to solve a linear system of equations.*

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

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

Here, we use the super index r to denote iteration (or time level). $\boldsymbol{u}$ will denote the solution vector. An approximation to $\boldsymbol{u}$ at iteration r will be denoted by $\boldsymbol{u}^r$.

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

SLIDE 7

Stability dictates that

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

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

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

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

2.1.2 Matrix Form

SLIDE 8

Split A

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

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

Iterative method

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

SLIDE 9

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

$$\boxed{\boldsymbol{u}^{r+1} = R_{\text{J}}\, \boldsymbol{u}^r + D^{-1}\, \boldsymbol{f}}$$

$R_{\text{J}} = (I - D^{-1}A)$: **Jacobi Iteration Matrix**

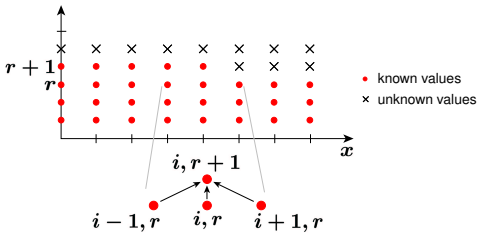
$$\boxed{D_{ii}^{-1} = h^2/2}$$

Note that, in order to implement this method we would typically not form any matrices, but update the unknowns, one component at a time, according to

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

2.1.3 Implementation

SLIDE 10



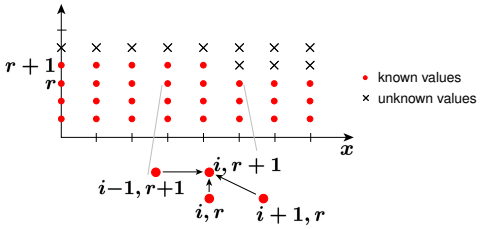
Jacobi iteration

$$u_i^{r+1} = \tfrac{1}{2}\,(u_{i+1}^r + u_{i-1}^r + h^2 f_i)$$

2.2 Gauss-Seidel

SLIDE 11

Assuming we are updating the unknowns according to their index i, at the time u_i^{r+1} needs to be calculated, we already know u_{i-1}^{r+1}. The idea of Gauss-Seidel iteration is to always use the most recently computed value.



Gauss-Seidel iteration (consider most recent iterate)

$$u_i^{r+1} = \tfrac{1}{2}\,(u_{i+1}^r + u_{i-1}^{\boldsymbol{r+1}} + h^2 f_i)$$

2.2.1 Matrix Form

SLIDE 12

Split A

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

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

Iterative method

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

The above matrix form assumes that we are updating through our unknowns in ascending order. If we were to update in reverse order, i.e., the last unknown first, the iteration matrix would become $R_{\mathrm{GS}} = (D - U)^{-1}L$.

Note 2 **Updating order**

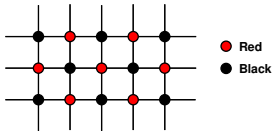
We see that, unlike in the Jacobi method, the order in which the unkowns are updated in Gauss-Seidel changes the result of the iterative procedure. One could sweep through the unknowns in ascending, descending, or alternate orders. The latter procedure is called symmetric Gauss-Seidel.
Another effective strategy, known as red-black Gauss-Seidel iteration, is to update the even unknowns first (red)

$$u_{2i}^{r+1} = \frac{1}{2}\,\left(u_{2i+1}^{r} + u_{2i-1}^{r} + h^2\,f_{2i}\right)\;,$$

and then the odd components (black)

$$u_{2i+1}^{r+1} = \frac{1}{2}\,\left(u_{2i+1}^{r+1} + u_{2i}^{r+1} + h^2\,f_{2i+1}\right)\;.$$

The red-black Gauss-Seidel iteration is popular for parallel computation since the red (black) points only require the black (red) points and these can be updated in any order. The method readily extends to multiple dimensions. In 2D, for instance, the red and black points are shown below.



SLIDE 13

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

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

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

2.3 Error Equation

SLIDE 14

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

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

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

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

Subtracting $A\,\boldsymbol{u}^r$ from both sides of $A\,\boldsymbol{u} = \boldsymbol{f}$,

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

ERROR EQUATION $\rightarrow$ $\boxed{A\,\boldsymbol{e}^r = \boldsymbol{r}^r}$ N3

We note that, whereas the iteration error is not an acessible quantity, since it requires $\boldsymbol{u}$, the residual is easily computable and only requires $\boldsymbol{u}^r$.

Note 3 **Relationship between error and residual.**

We have seen that the residual is easily computable and in some sense it measures the amount by which our approximate solution $\boldsymbol{u}^r$ fails to satisfy the original problem.
It is clear that if $\boldsymbol{r} = \boldsymbol{0}$, we have $\boldsymbol{e} = \boldsymbol{0}$. However, it is not always the case that when $\boldsymbol{r}$ is small in norm, $\boldsymbol{e}$ is also small in norm.
We have that $A\,\boldsymbol{u} = \boldsymbol{f}$ and $A^{-1}\boldsymbol{r} = \boldsymbol{e}$. Taking norms, we have

$$\|f\|_2 \le \|A\|_2\,\|u\|_2, \qquad \|e\|_2 \le \|A^{-1}\|_2\,\|r\|_2\ .$$

Combining these two expressions, we have

$$\frac{\|e\|_2}{\|u\|_2} \le \underbrace{\|A\|_2\,\|A^{-1}\|_2}_{\text{cond}\,(A)}\,\frac{\|r\|_2}{\|f\|_2}\ .$$

From here we see that if our matrix is not well conditioned, i.e., cond (A) is large, then small residuals can correspond to significant errors.

2.3.1 Jacobi

SLIDE 15

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

subtracting

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

2.3.2 Gauss-Seidel

SLIDE 16

Similarly,

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

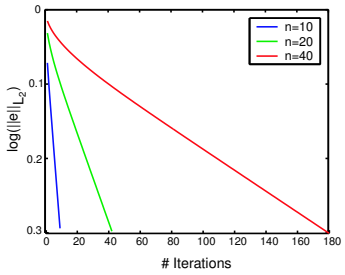
It is clear that the above equations satisfied by the error are not useful for practical purposes since $\boldsymbol{e}^0 = \boldsymbol{u} - \boldsymbol{u}^0$, will not be known before the problem is solved. We will see however, that by studying the equation satisfied by the error we can determine useful properties regarding the convergence of our schemes.

2.4 Examples

2.4.1 Jacobi

SLIDE 17

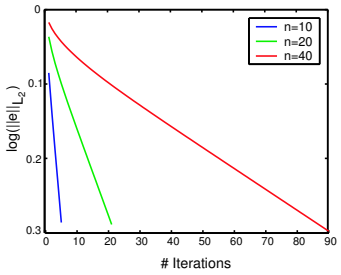
$-u_{xx} = 1$ $\qquad u(0) = u(1) = 0;$ $\qquad \boldsymbol{u}^0 = \mathbf{0}$



2.4.2 Gauss-Seidel

SLIDE 18

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



2.4.3 Observations

SLIDE 19

– 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?

3 Convergence Analysis

SLIDE 20

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

The iterative method will converge if

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

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

The spectral radius is the radius of the smallest circle centered at the origin that contains all the eigenvalues. The above condition indicates that the necessary and sufficient condition for the scheme to converge, for arbitrary intial conditions, is that the largest eienvalue, in modulus, lies within the unit circle.

Note 4 **Condition on $\rho(R)$**

The condition that

$$\lim_{r\to\infty} R^r = 0 \quad \text{if and only if} \quad \rho(R) < 1$$

can be easily shown for the case where R is diagonalizable. In this case $R = X^{-1}\,\Lambda\,X$, where Λ is the diagonal matrix of eigenvectors. We have that $R^r = X^{-1}\,\Lambda^r\,X$ and hence we require that

$$\lim_{r\to\infty}\ \Lambda^r = 0$$

which is equivalent to $\rho(R) < 1$.

3.1 Theorems

SLIDE 21

– 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

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

– If the matrix A is symmetric positive definite then Gauss-Seidel iteration converges for any initial solution. N6

Note 5 Convergence of Jacobi iteration for diagonally dominant matrices

---

For the Jacobi iteration, $R_{\rm J} = D^{-1}(L+U)$, it can be easily seen that R has zero diagonal entries. It can be shown (Gershgorin Theorem - see GVL) that a matrix with zeros in the main diagonal, $\rho(R_{\rm J}) \le \|R_{\rm J}\|_\infty$. Hence

$$\rho(R_{\rm J}) = \|R_{\rm J}\|_2 \le \|R_{\rm J}\|_\infty = \max_{1\le j\le n} \sum_{\substack{j=1\\ j\neq 1}}^{n} \left|\frac{a_{i\,j}}{a_{i\,i}}\right| < 1\ ,$$

the last inequality follows form the diagonal dominance of A.

Note 6 Convergence of Gauss-Seidel for SPD matrices

(See Theorem 10.1.2 of GVL)

3.2 Jacobi

SLIDE 22

Unfortunately none of the above convergence theorems helps us to show that Jacobi iteration will converge for our model problem. However, since for our model problem we can calculate the eigenvalues of A explicitly (see previous lectures), convergence can be shown directly.

$$R_{\mathrm{J}} = D^{-1}\,(L+U) = D^{-1}\,(D-A) = I - D^{-1}\,A = I - \frac{h^2}{2}\,A$$

If $\quad A\,\boldsymbol{v}^k = \lambda^k\,\boldsymbol{v}^k$,

then $\quad R_{\mathrm{J}}\,\boldsymbol{v}^K = \left(I - \dfrac{h^2}{2}\,A\right)\boldsymbol{v}^k = \underbrace{\left(1 - \dfrac{h^2}{2}\,\lambda^k(A)\right)}_{\lambda^k(R_{\mathrm{J}})}\,\boldsymbol{v}^k$

$$\boxed{\begin{array}{c}\lambda^k(R_{\mathrm{J}}) = 1 - \dfrac{h^2}{2}\,\lambda^k(A) \\ \text{Eigenvectors of } R_{\mathrm{J}} \equiv \text{ Eigenvectors of } A\end{array}}$$

SLIDE 23

Recall ...

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

$n \times n$ $\qquad$ *SPD*

$A\boldsymbol{v}^k = \lambda^k\boldsymbol{v}^k \qquad k = 1,\ldots,n$

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

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

Eigenvalues:

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

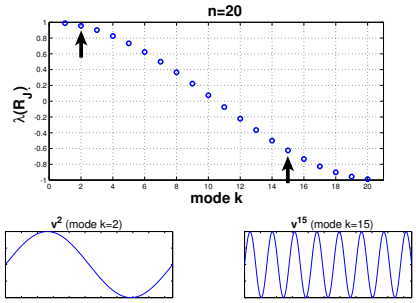
SLIDE 24

$$\begin{aligned}\lambda^k(R_{\mathrm{J}}) &= 1 - \frac{h^2}{2}\,\lambda^k(A) = 1 - [1 - \cos k\pi h] \\ &= \cos\frac{k\pi}{n+1} < \mathbf{1}, \qquad k = 1,\ldots,n\end{aligned}$$

Jacobi converges for our model problem.

$$R_{\mathrm{J}}\boldsymbol{v}^k = \lambda^k(R_{\mathrm{J}})\boldsymbol{v}^k$$

SLIDE 25



SLIDE 26

Since A has a complete set of eigenvectors $\boldsymbol{v}^k$, $k = 1, \ldots n$,

$$\text{Write} \quad \boldsymbol{e}^0 = \sum_{k=1}^{n} c_k \, \boldsymbol{v}^k \qquad \boldsymbol{v}^k, \; k\text{-th eigenvector of } R_{\mathrm{J}}$$

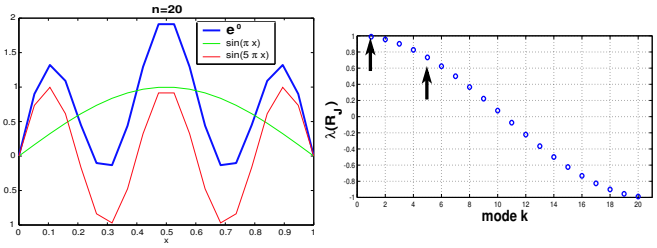
$$\boldsymbol{e}^1 = R\,\boldsymbol{e}^0 = \sum_{k=1}^{n} c_k \, \lambda^k(R_{\mathrm{J}}) \, \boldsymbol{v}^k$$

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

3.2.1 Example

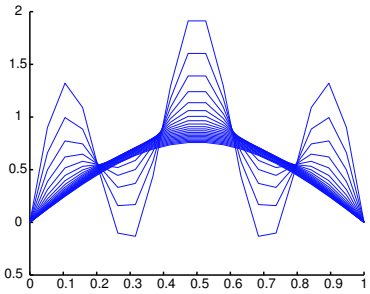
SLIDE 27

We consider the situation in which the initial error only contains to modal components corresponding to $k = 1$ and $k = 5$.



SLIDE 28

The figure below show the evolution of the error for each iteration. The faster decay of the $k = 5$ mode can be readily observed.



3.2.2 Convergence rate

SLIDE 29

$$\lambda^k(R_{\mathrm{J}}) = cos(k\pi h), \qquad k = 1, \ldots n$$

Largest $|\lambda^k(R_{\mathrm{J}})|$ for $k = 1, \ldots n$

Worst case $\boldsymbol{e}^0 = c_1 \boldsymbol{v}^1 \rightarrow \boldsymbol{e}^r = c_1\, \rho(R_{\mathrm{J}})^r\, \boldsymbol{v}^1$

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

SLIDE 30

For h small (n large) $\pi h \simeq 0$ and $\cos(\pi h)$ can be approximated as $1 - \dfrac{\pi^2 h^2}{2}$.

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

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

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

We have used the fact that for x small $\log(1 + x) = x - \dfrac{x^2}{2} + \dfrac{x^3}{3} \ \ldots$

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

The number of iterations required to obtain a given level of convergence is n^2.

3.2.3 Convergence rate (2D)

SLIDE 31

This analysis can be extended to two and three dimensions in a straightforward manner.

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

$$\lambda^{k\ell}(R_{\rm J}) = 1 - \frac{h^2}{4}\lambda(A) = \frac{1}{2}\left[\cos(k\pi h) + \cos(\ell\pi h)\right]$$

$$\rho(R_{\rm J}) = \cos(\pi h) = \cos\left(\frac{\pi}{h+1}\right) \qquad \boxed{h = \tfrac{1}{n+1}}$$

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

It is important to note that in 1D, 2D and also in 3D, $r = O(n^2)$. This, combined with the fact that the number of unknowns in 1, 2, and 3D is n, n^2 and n^3, makes iterative methods potentially attractive for 2, and specially, 3D problems.

3.3 Gauss-Seidel

SLIDE 32

The convergence analysis for the Gauss-Seidel method is analogous to that of the Jacobi method. The expressions for the eigenvectors and eigenvalues, for the model problems, can also be found in closed although the procedure is somewhat more involved.

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

It can be shown (try it !) that

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

Gauss-Seidel converges for our problem

The eigenvalues are always positive.

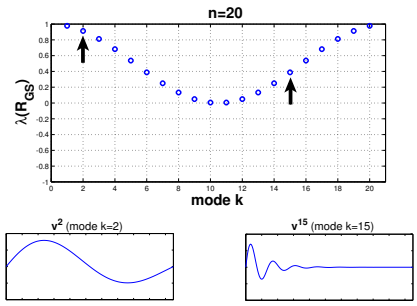
But, Eigenvectors of $R_{\rm GS} \not\equiv$ Eigenvectors of A

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

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

These eigenvectors, although not orthogonal, still form a complete basis and can be used to represent the error components.

SLIDE 33



3.3.1 Convergence rate

SLIDE 34

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

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

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

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

Same rate as Jacobi, but requires only half the number of iterations.

4 Comparative cost

SLIDE 35

		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

Only in 3D, give iterative methods some cost advantage over Gaussian elimination.

5 Over/Under Relaxation

5.0.2 Main Idea

SLIDE 36

Typically

$$\boldsymbol{u}^{r+1} = R\,u^r + f^* \qquad \begin{array}{lcll} f^* & = & D^{-1}f & \text{Jacobi} \\ f^* & = & (D-L)^{-1}f & \text{Gauss-Seidel} \end{array}$$

Can we "extrapolate" the changes?

$$\begin{aligned} \boldsymbol{u}^{r+1} &= \omega(R\boldsymbol{u}^r + f^*) + (1-\omega)\,\boldsymbol{u}^r \\ &= \underbrace{[\omega\,R + (1-\omega)\,I]}_{R_\omega}\,\boldsymbol{u}^r + \omega f^* \end{aligned} \qquad \boxed{\omega > 0}$$

5.1 How large can we choose ω?

SLIDE 37

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

Jacobi $\qquad \lambda^k(R_{\mathrm{J}}) = \cos k\pi h$

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

$\Rightarrow \qquad \rho(R_\omega) < 1 \Rightarrow \; 0 \le \omega_{\mathrm{J}} \le 1$ can only be under-relaxed

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

N7

Note 7 **Succesive Over-Relaxation (SOR)**

Over-relaxed Gauss-Seidel is about a factor of two faster than standard GS but still requires $O(n^2)$ iterations.

The idea of over-relaxation can be used to improve convergence further. If instead of extrapolating the changes for all the unknowns at the same time, we extrapolate the change of each unknown as soon as it has been computed, and use the updated value in subsequent unknown updates we obtain what is called successive over-relaxation (SOR). In a component form SOR is written as

$$u_i^{r+1} = \omega\left[\frac{1}{2}\,(u_{i+1}^r + u_{i-1}^{r+1}) + \frac{h^2}{2}\,fi\right] + (1-\omega)\,u_i^r \ .$$

with $1 < \omega < 2$ and typically $\omega \sim 1.5$.

In matrix form SOR can be written as we

$$\begin{aligned} \boldsymbol{u}^{r+1} &= (D-\omega L)^{-1}\,[((1-\omega)\,D + \omega U)\,\boldsymbol{u} + \omega f] \\ &= R_{\mathrm{SOR}}\,\boldsymbol{u}^r + (D-\omega L)^{-1} \ . \end{aligned}$$

It can be shown that the spectral radius of R_{SOR}, for our model problem, is $1 - O(h)$ and hence the number of iterations scales like n rather than n^2. SOR was very popular 30 years ago. Nowadays, we can do better.

REFERENCES

[GVL] G. Golub, C. van Loan, Matrix Computations, Johns Hopkins.