chapter 3 : numerical linear algebra

3.1 review of linear algebra

$$\left.\begin{array}{c} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array}\right\} : \text{system of linear equations for } x_1, \ldots, x_n$$

We can write the system in 3 other forms.

1. $\sum_{j=1}^{n} a_{ij}x_j = b_i$, $i = 1 : n$, i : row index , j : column index

2. $\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$

3. $Ax = b$

basic problem : Given A and b, find x.

solution : $x = b/A$: no, but $x = A\backslash b$ does work in Matlab (what is it doing?)

thm : The following conditions are equivalent.

1. The equation $Ax = b$ has a unique solution for any vector b.
2. A is invertible, i.e. there exists a matrix A^{-1} such that $AA^{-1} = I$
3. $\det A \neq 0$
4. The equation $Ax = 0$ has the unique solution $x = 0$.
5. The columns of A are linearly independent.
6. The eigenvalues of A are nonzero.

pf : Math 214/417/419

note

1. If A is invertible, then $x = A^{-1}b$ (pf : $Ax = A(A^{-1}b) = (AA^{-1})b = Ib = b$), but this is not the best way to compute x in practice.

2. There are two types of methods for solving $Ax = b$, direct methods and iterative methods. We will begin with direct methods.

3.2 Gaussian elimination

First consider the special case in which A is upper triangular.

$$\begin{array}{ccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\
 & & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\
 & & & \ddots & & & & & \\
 & & & & a_{n-1,n-1}x_{n-1} & + & a_{n-1,n}x_n & = & b_{n-1} \\
 & & & & & & a_{nn}x_n & = & b_n
\end{array}$$

$$\begin{aligned}
\Rightarrow \quad x_n &= b_n/a_{nn} \\
x_{n-1} &= (b_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1} \\
&\vdots \\
x_1 &= (b_1 - (a_{12}x_2 + \cdots + a_{1n}x_n))/a_{11}
\end{aligned}$$

back substitution

1. $x_n = b_n/a_{nn}$
2. for $i = n-1 : -1 : 1$ % i : row index
3. $sum = b_i$
4.   for $j = i+1 : n$ % j : column index
5. $sum = sum - a_{ij} \cdot x_j$
6. $x_i = sum/a_{ii}$

operation count

$\#$ divisions $= n$

$\#$ mults $= \#$ adds $= \frac{1}{2}n(n-1) = \frac{1}{2}n^2 - \frac{1}{2}n \sim \frac{1}{2}n^2$ for large n

pf

$\#$ mults $= 1 + 2 + \cdots + (n-1) = S$

$2S = (1+2+\cdots+(n-1)) + ((n-1)+\cdots+2+1) = n+n+\cdots+n = n(n-1)$

$\Rightarrow \; S = \frac{1}{2}n(n-1)$ ok

Hence the leading order term in the operation count for back substitution is n^2.

note : Similar considerations apply if A is lower triangular.

<u>note</u>

6 Tues 1/29

In case A is a non-triangular matrix, we use <u>elementary row operations</u> to reduce $Ax = b$ to upper triangular form and then apply back substitution to find x.

elementary row operation : $\left\{\begin{array}{l}\text{multiply an equation by a nonzero constant and} \\ \text{subtract the result from another equation}\end{array}\right.$

<u>ex</u> : $n = 3$

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

$$\left(\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right)$$

<u>step 1</u> : eliminate variable x_1 from eqs. 2 and 3

$$m_{21} = \frac{a_{21}}{a_{11}} \Rightarrow \begin{array}{lcl} a_{22} & \to & a_{22} - m_{21}a_{12} \\ a_{23} & \to & a_{23} - m_{21}a_{13} \\ b_2 & \to & b_2 - m_{21}b_1 \end{array}$$

% m_{21} is called a <u>multiplier</u>

$$m_{31} = \frac{a_{31}}{a_{11}} \Rightarrow \begin{array}{lcl} a_{32} & \to & a_{32} - m_{31}a_{12} \\ a_{33} & \to & a_{33} - m_{31}a_{13} \\ b_3 & \to & b_3 - m_{31}b_1 \end{array}$$

$$\left(\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & a_{32} & a_{33} & b_3 \end{array}\right)$$

these elements have changed

<u>step 2</u> : eliminate variable x_2 from eq. 3

$$m_{32} = \frac{a_{32}}{a_{22}} \Rightarrow \begin{array}{lcl} a_{33} & \to & a_{33} - m_{32}a_{23} \\ b_3 & \to & b_3 - l_{32}b_2 \end{array}$$

$$\left(\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & 0 & a_{33} & b_3 \end{array}\right) : \text{ upper triangular}$$

ex

$$
\begin{aligned}
2x_1 - x_2 &= 1 \\
-x_1 + 2x_2 - x_3 &= 0 \\
-x_2 + 2x_3 &= 1
\end{aligned}
$$

$$
\left(\begin{array}{rrr:r} 2 & -1 & 0 & 1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & 1 \end{array}\right) \qquad \begin{array}{l} m_{21} = -1/2 \\ m_{31} = 0 \end{array}
$$

$$
\left(\begin{array}{rrr:r} 2 & -1 & 0 & 1 \\ 0 & 3/2 & -1 & 1/2 \\ 0 & -1 & 2 & 1 \end{array}\right) \qquad m_{32} = -1/(3/2) = -2/3
$$

$$
\left(\begin{array}{rrr:r} 2 & -1 & 0 & 1 \\ 0 & 3/2 & -1 & 1/2 \\ 0 & 0 & 4/3 & 4/3 \end{array}\right)
$$

$x_3 = 1$, $x_2 = (\frac{1}{2} - (-1) \cdot 1)/\frac{3}{2} = 1$, $x_1 = (1 - (-1) \cdot 1)/2 = 1$ check : ok

general $n \times n$ case

reduction to upper triangular form

1. for $k = 1 : n-1$ % k : step index
2. for $i = k+1 : n$
3. $\quad m_{ik} = a_{ik}/a_{kk}$ % assume $a_{kk} \neq 0$, more later
4. $\quad$ for $j = k+1 : n$
5. $\qquad a_{ij} = a_{ij} - m_{ik} \cdot a_{kj}$
6. $\quad b_i = b_i - m_{ik} \cdot b_k$

note

The element a_{kk} in step k is called a pivot (these are the diagonal elements in the last step). In the previous example, the pivots are 2, $\frac{3}{2}$, $\frac{4}{3}$.

operation count

The leading order term comes from line 5.

$$
\left.\begin{array}{l} k = 1 \;\Rightarrow\; 2(n-1)^2 \text{ ops} \\ k = 2 \;\Rightarrow\; 2(n-2)^2 \text{ ops} \\ \quad\vdots \\ k = n-2 \;\Rightarrow\; 2 \cdot 2^2 \text{ ops} \\ k = n-1 \;\Rightarrow\; 2 \cdot 1^2 \text{ ops} \end{array}\right\} \Rightarrow\; 2 \cdot \sum_{k=1}^{n-1} k^2 = 2 \cdot \tfrac{1}{6}(n-1)n(2n-1) \quad , \quad \text{pf : soon}
$$

Hence the operation count for Gaussian elimination is $\frac{2}{3}n^3$.

<u>note</u>

$$\sum_{k=1}^{n} k = \tfrac{1}{2}n(n+1) \quad , \quad \sum_{k=1}^{n} k^2 = \tfrac{1}{6}n(n+1)(2n+1)$$

<u>pf</u> : 1. already done

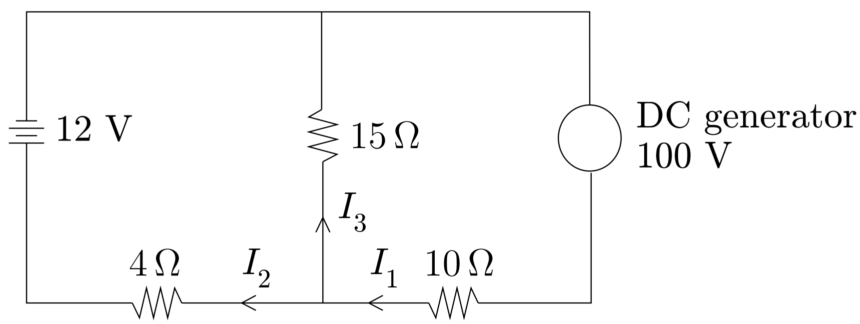
2. $n^3 = n^3 - (n-1)^3 + (n-1)^3 + \cdots - 2^3 + 2^3 - 1^3 + 1^3 = \sum_{k=1}^{n}(k^3 - (k-1)^3)$

$k^3 - (k-1)^3 = k^3 - (k^3 - 3k^2 + 3k - 1) = 3k^2 - 3k + 1$

$n^3 = \sum_{k=1}^{n}(3k^2 - 3k + 1) = 3\sum_{k=1}^{n} k^2 - 3\sum_{k=1}^{n} k + \sum_{k=1}^{n} 1 = 3S - 3 \cdot \tfrac{1}{2}n(n+1) + n$

$3S = n^3 + \tfrac{3}{2}n(n+1) - n = n(n^2 + \tfrac{3}{2}n + \tfrac{1}{2}) = n(n+1)(n + \tfrac{1}{2})$ <u>ok</u>

<u>ex</u> : electric circuit for charging a car battery



To determine the currents, we will apply Kirchoff's voltage law and current law.

1. The sum of the voltage drops around any closed loop is zero.

Ohm's law : $V = IR \;\Rightarrow\; 10I_1 + 15I_3 - 100 = 0$, $4I_2 + 12 - 15I_3 = 0$

2. The sum of the currents flowing into a junction equals the sum flowing out.

$\Rightarrow\; I_1 = I_2 + I_3$

7 Thurs 1/31

$$\Rightarrow \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = \begin{pmatrix} 100 \\ -12 \\ 0 \end{pmatrix}$$

Then we can apply Gaussian elimination. But if we write the first 2 equations in reverse order, then we obtain the following system.

$$\begin{pmatrix} 0 & 4 & -15 \\ 10 & 0 & 15 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix} = \begin{pmatrix} -12 \\ 100 \\ 0 \end{pmatrix}$$

In this case Gaussian elimination breaks down because the 1st pivot is zero.

3.3 pivoting

There are various strategies that can be applied if one of the pivots is zero.

partial pivoting

Consider the reduced matrix at the beginning of step k.

$$\left(\begin{array}{cccccc|c} a_{11} & \cdots & \cdots & a_{1k} & \cdots & a_{1n} & b_1 \\ & \ddots & & \vdots & & \vdots & \vdots \\ & & \ddots & \vdots & & \vdots & \vdots \\ & & & a_{kk} & \cdots & a_{kn} & b_k \\ & & & \vdots & & \vdots & \vdots \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{nk} & \cdots & a_{nn} & b_n \end{array}\right)$$

If $a_{kk} = 0$, find index l such that $|a_{lk}| = \max\{|a_{ik}|; k \le i \le n\}$, then interchange row l and row k and proceed with the elimination.

1. If A is invertible, then Gaussian elimination with partial pivoting does not break down. (pf : Math 571)

2. In practice, pivoting is often applied even if the pivot element is nonzero.

ex

$$\left(\begin{array}{cc|c} \epsilon & 1 & 1+\epsilon \\ 1 & 1 & 2 \end{array}\right) \to \left(\begin{array}{cc|c} \epsilon & 1 & 1+\epsilon \\ 0 & 1-\frac{1}{\epsilon} & 1-\frac{1}{\epsilon} \end{array}\right) \Rightarrow \left.\begin{array}{l} x_1 = \dfrac{1+\epsilon-1}{\epsilon} = 1 \\ \\ x_2 = \dfrac{1-\frac{1}{\epsilon}}{1-\frac{1}{\epsilon}} = 1 \end{array}\right\} : \text{ exact solution}$$

$m_{21} = \frac{1}{\epsilon}$

Now consider the effect of roundoff error.

$$\left(\begin{array}{cc|c} \epsilon & 1 & 1 \\ 0 & -\frac{1}{\epsilon} & -\frac{1}{\epsilon} \end{array}\right) \Rightarrow \left.\begin{array}{l} \tilde{x}_1 = \frac{1-1}{\epsilon} = 0 \\ \tilde{x}_2 = \dfrac{-\frac{1}{\epsilon}}{-\frac{1}{\epsilon}} = 1 \end{array}\right\} : \text{ computed solution , inaccurate}$$

Now apply pivoting in the presence of roundoff error.

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ \epsilon & 1 & 1 \end{array}\right) \to \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right) \Rightarrow \left.\begin{array}{l} \tilde{x}_1 = 1 \\ \tilde{x}_2 = 1 \end{array}\right\} : \text{ new computed solution , accurate}$$

$m_{21} = \frac{\epsilon}{1} = \epsilon$

This is an issue of stability. (more later)

3.4 vector and matrix norms

To prepare for error analysis, we need a way to measure the size of a vector.

def : A vector norm is a function $||x||$ satisfying the following properties.

1. $||x|| \geq 0$ and $||x|| = 0 \Leftrightarrow x = 0$
2. $||\alpha x|| = |\alpha| \cdot ||x||$, α : scalar
3. $||x + y|| \leq ||x|| + ||y||$: triangle inequality

ex

$$||x||_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2} \quad : \text{ Euclidean length}$$

$$||x||_\infty = \max\{|x_i| : i = 1, \ldots, n\}$$

pf ...

ex : $x = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \Rightarrow ||x||_2 = \sqrt{5}$, $||x||_\infty = 2$

def : Given a matrix A, consider the operator $x \to Ax$ as input $\to$ output. Then $\dfrac{||Ax||}{||x||}$ is the amplification factor for a given input vector x, and we define the matrix norm to be the maximum amplification factor over all nonzero input vectors, $||A|| = \max\limits_{x \neq 0} \dfrac{||Ax||}{||x||}$. The matrix norm satisfies the following properties.

1. $||A|| \geq 0$ and $||A|| = 0 \Leftrightarrow A = 0$
2. $||\alpha A|| = |\alpha| \cdot ||A||$
3. $||A + B|| \leq ||A|| + ||B||$
4. $||Ax|| \leq ||A|| \cdot ||x||$
5. $||AB|| \leq ||A|| \cdot ||B||$

pf : just 5

$$||AB|| \underset{\substack{\uparrow \\ \text{def}}}{=} \max_{x \neq 0} \frac{||ABx||}{||x||} \underset{\substack{\uparrow \\ \text{prop 4}}}{\leq} \max_{x \neq 0} \frac{||A|| \cdot ||Bx||}{||x||} \underset{\substack{\uparrow \\ \text{prop 4}}}{\leq} \max_{x \neq 0} \frac{||A|| \cdot ||B|| \cdot ||x||}{||x||} = ||A|| \cdot ||B||$$

ok

note : Computing $||A||$ by the definition is difficult and there are more convenient formulas that can be used in practice.

<u>thm</u> : $||A||_\infty = \max_{x\neq 0} \frac{||Ax||_\infty}{||x||_\infty} = \max_i \sum_j |a_{ij}|$: max row sum

<u>pf</u> : omit (Math 571)

<u>ex</u> : $A = \begin{pmatrix} 3 & -4 \\ 1 & 0 \end{pmatrix} \Rightarrow ||A||_\infty = \max\{|3| + |-4|, |1| + |0|\} = 7$

$$x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow Ax = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \Rightarrow \frac{||Ax||_\infty}{||x||_\infty} = \frac{3}{1} = 3$$

$$x = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow Ax = \begin{pmatrix} -4 \\ 0 \end{pmatrix} \Rightarrow \frac{||Ax||_\infty}{||x||_\infty} = \frac{4}{1} = 4$$

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow Ax = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \Rightarrow \frac{||Ax||_\infty}{||x||_\infty} = \frac{1}{1} = 1$$

$$x = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \Rightarrow Ax = \begin{pmatrix} 7 \\ 1 \end{pmatrix} \Rightarrow \frac{||Ax||_\infty}{||x||_\infty} = \frac{7}{1} = 7 : \text{ max amp factor by thm}$$

<u>3.5 error analysis</u>

8 Tues 2/5

$Ax = b$

x : exact solution , $\tilde{x}$: approximate solution

$e = x - \tilde{x}$: <u>error</u> (usually unknown) , $r = b - A\tilde{x}$: <u>residual</u> (can be computed)

<u>question</u> : What is the relation between e and r?

<u>ex</u> : $\left(\begin{array}{cc|c} 1.01 & 0.99 & 2 \\ 0.99 & 1.01 & 2 \end{array}\right) \Rightarrow x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

$$\tilde{x}_1 = \begin{pmatrix} 1.01 \\ 1.01 \end{pmatrix} \Rightarrow e_1 = x - \tilde{x}_1 = \begin{pmatrix} -0.01 \\ -0.01 \end{pmatrix} \Rightarrow ||e_1|| = 0.01$$

$$r_1 = b - A\tilde{x}_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 2.02 \\ 2.02 \end{pmatrix} = \begin{pmatrix} -0.02 \\ -0.02 \end{pmatrix} \Rightarrow ||r_1|| = 0.02$$

$$\tilde{x}_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \Rightarrow e_2 = x - \tilde{x}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \Rightarrow ||e_2|| = 1$$

$$r_2 = b - A\tilde{x}_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 2.02 \\ 1.98 \end{pmatrix} = \begin{pmatrix} -0.02 \\ 0.02 \end{pmatrix} \Rightarrow ||r_2|| = 0.02$$

Hence if $||r||$ is small, there is no guarantee that $||e||$ is also small.

<u>question</u> : How large can $||e||$ be?

<u>thm</u> : $\dfrac{||e||}{||x||} \le \kappa(A)\dfrac{||r||}{||b||}$, where $\kappa(A) = ||A|| \cdot ||A^{-1}||$: <u>condition number</u>

<u>ex</u> : $A = \begin{pmatrix} 1.01 & 0.99 \\ 0.99 & 1.01 \end{pmatrix} \Rightarrow \; ||A|| = 2$

$$A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \frac{1}{0.04}\begin{pmatrix} 1.01 & -0.99 \\ -0.99 & 1.01 \end{pmatrix}$$

$$= \begin{pmatrix} 25.25 & -24.75 \\ -24.75 & 25.25 \end{pmatrix} \;\Rightarrow\; ||A^{-1}|| = 50 \;\Rightarrow\; \kappa(A) = 100 \quad \underline{\text{ok}}$$

<u>pf</u>

1. $||b|| = ||Ax|| \le ||A|| \cdot ||x|| \;\Rightarrow\; ||x|| \ge ||b||/||A||$
2. $Ae = A(x - \tilde{x}) = Ax - A\tilde{x} = b - A\tilde{x} = r \;\Rightarrow\; Ae = r$
3. $e = A^{-1}r \;\Rightarrow\; ||e|| = ||A^{-1}r|| \le ||A^{-1}|| \cdot ||r||$
4. $\dfrac{||e||}{||x||} \le \dfrac{||A^{-1}|| \cdot ||r||}{||b||/||A||} = \dfrac{||A|| \cdot ||A^{-1}|| \cdot ||r||}{||b||} = \kappa(A) \cdot \dfrac{||r||}{||b||}$ <u>ok</u>

<u>alternative viewpoint</u>

1. $\left.\begin{array}{r} Ax = b \\ A\tilde{x} = \tilde{b} \end{array}\right\} \Rightarrow \dfrac{||x - \tilde{x}||}{||x||} \le \kappa(A)\dfrac{||b - \tilde{b}||}{||b||}$: perturbation of RHS , pf : <u>ok</u>
2. $\left.\begin{array}{r} Ax = b \\ \tilde{A}\tilde{x} = b \end{array}\right\} \Rightarrow \dfrac{||x - \tilde{x}||}{||\tilde{x}||} \le \kappa(A)\dfrac{||A - \tilde{A}||}{||A||}$: perturbation of matrix , pf : . . .

Hence $\kappa(A)$ controls the change in x due to changes in A and b.

<u>ex</u> (recall)

$$\left(\begin{array}{cc|c} \epsilon & 1 & 1+\epsilon \\ 1 & 1 & 2 \end{array}\right) \to \left(\begin{array}{cc|c} \epsilon & 1 & 1+\epsilon \\ 0 & 1-\frac{1}{\epsilon} & 1-\frac{1}{\epsilon} \end{array}\right) \Rightarrow \left.\begin{array}{r} x_1 = 1 \\ x_2 = 1 \end{array}\right\} : \text{exact solution}$$

Now consider the effect of roundoff error.

$$\left(\begin{array}{cc|c} \epsilon & 1 & 1 \\ 0 & -\frac{1}{\epsilon} & -\frac{1}{\epsilon} \end{array}\right) \Rightarrow \left.\begin{array}{r} \tilde{x}_1 = 0 \\ \tilde{x}_2 = 1 \end{array}\right\} : \text{computed solution , inaccurate}$$

<u>explanation</u>

$$A = \begin{pmatrix} \epsilon & 1 \\ 1 & 1 \end{pmatrix} \;,\; A^{-1} = \frac{1}{\epsilon - 1}\begin{pmatrix} 1 & -1 \\ -1 & \epsilon \end{pmatrix} \Rightarrow \kappa(A) = 2 \cdot \frac{1}{|\epsilon - 1|} \cdot 2 \approx 4$$

However, Gaussian elimination reduces the system to upper triangular form.

$$U = \begin{pmatrix} \epsilon & 1 \\ 0 & 1 - \frac{1}{\epsilon} \end{pmatrix} \;,\; U^{-1} = \frac{1}{\epsilon - 1}\begin{pmatrix} 1 - \frac{1}{\epsilon} & -1 \\ 0 & \epsilon \end{pmatrix}$$

$$\Rightarrow \; \kappa(U) = |1 - \tfrac{1}{\epsilon}| \cdot \tfrac{1}{|\epsilon - 1|} \cdot (|1 - \tfrac{1}{\epsilon}| + 1) \;\approx\; \tfrac{1}{\epsilon^2} \;:\; \text{larger than } \kappa(A)$$

Hence a small change in the matrix or RHS of the reduced system (e.g. due to roundoff error) can produce a large change in the computed solution (as in the example). This means that Gaussian elimination is an unstable method for solving $Ax = b$, because it replaced a well-conditioned matrix A by an ill-conditioned matrix U. However, pivoting produces a different reduced system.

$$\left(\begin{array}{cc|c} 1 & 1 & 2 \\ \epsilon & 1 & 1+\epsilon \end{array}\right) \to \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1-\epsilon & 1-\epsilon \end{array}\right) \Rightarrow \left.\begin{array}{l} \tilde{x}_1 = 1 \\ \tilde{x}_2 = 1 \end{array}\right\} : \text{exact solution}$$

$$U = \begin{pmatrix} 1 & 1 \\ 0 & 1-\epsilon \end{pmatrix} , \ U^{-1} = \frac{1}{1-\epsilon}\begin{pmatrix} 1-\epsilon & -1 \\ 0 & 1 \end{pmatrix} \Rightarrow \kappa(U) \approx 4 \approx \kappa(A)$$

Hence, pivoting preserves the condition number of the original matrix, and therefore Gaussian elimination + pivoting is stable (in most cases).

3.6 LU factorization : matrix form of Gaussian elimination 9 Thurs 2/7

Consider the 3×3 case (but the $n \times n$ case is similar).

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

step 1 : eliminate variable x_1 from eqs. 2 and 3

$$m_{21} = \frac{a_{21}}{a_{11}} , \ m_{31} = \frac{a_{31}}{a_{11}}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{pmatrix}$$

step 2 : eliminate variable x_2 from eq. 3

$$m_{32} = \frac{a_{32}}{a_{22}}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{pmatrix} = U : \text{ upper triangular}$$

$$\Rightarrow E_2 E_1 A = U \Rightarrow E_1 A = E_2^{-1} U \Rightarrow A = E_1^{-1} E_2^{-1} U$$

$$E_1 = \begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix} \Rightarrow E_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{pmatrix} , \text{ check : } E_1 E_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix} \Rightarrow E_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{pmatrix} \text{ , check : } \ldots$$

$$E_1^{-1}E_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{pmatrix} = L \text{ : lower triangular}$$

final result : $A = LU$

<u>ex</u> : $\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & -1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix}$

$m_{21} = \frac{-1}{2}$ $\qquad$ $m_{32} = \frac{-1}{3/2} = -\frac{2}{3}$

$m_{31} = \frac{0}{2} = 0$

check : $LU = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & -\frac{2}{3} & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} = A$ <u>ok</u>

<u>note</u> : The following steps are used to solve $Ax = b$.

1. factor $A = LU$, op count $= \frac{2}{3}n^3$
2. solve $Ly = b$ by forward substitution , op count $= n^2$
3. solve $Ux = y$ by back substitution , op count $= n^2$

check : $Ax = LUx = Ly = b$ <u>ok</u>

<u>ex</u> : $A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$, $b = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \Rightarrow x = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$

Previously we used Gaussian elimination, but now we'll use LU factorization.

$$Ly = b \Rightarrow \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & -\frac{2}{3} & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{4}{3} \end{pmatrix}$$

$$Ux = y \Rightarrow \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{4}{3} \end{pmatrix} \Rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad \underline{\text{ok}}$$

question : So what's the point of LU factorization?

answer : Some applications require solving $Ax = b$ for a given matrix A and a sequence of vectors b, e.g. a time-dependent problem. Once the LU factorization of A is known, we can apply forward and back substitution to the sequence of vectors b; it's not necessary to repeat the LU factorization.

3.7 two-point boundary value problem

Find $y(x)$ on $0 \le x \le 1$ satisfying the differential equation $-y'' = r(x)$, subject to boundary conditions $y(0) = \alpha, y(1) = \beta$. This problem is a model for 1D steady state heat diffusion, where $y(x)$ is a temperature profile and $r(x)$ is a distribution of heat sources. (Think of $r(x), \alpha, \beta$ as input and $y(x)$ as output.)

finite-difference scheme

10 Tues 2/12

choose $n \ge 1$ and set $h = \frac{1}{n+1}$: mesh size

set $x_i = ih$ for $i = 0, 1, \dots, n+1$: mesh points ($x_0 = 0$, $x_{n+1} = 1$)

$$x_0 \quad x_1 \quad \cdots\cdots \quad x_{i-1} \quad x_i \quad x_{i+1} \quad \cdots\cdots \quad x_n \quad x_{n+1}$$

$y(x_i) = y_i$: exact solution , $r_i = r(x_i)$

recall : $D_+ y_i = \dfrac{y_{i+1} - y_i}{h}$, $D_- y_i = \dfrac{y_i - y_{i-1}}{h}$

$$D_+ D_- y_i = D_+ (D_- y_i) = D_+ \left(\frac{y_i - y_{i-1}}{h} \right) = \frac{1}{h} (D_+ y_i - D_+ y_{i-1})$$

$$= \frac{1}{h} \left(\frac{y_{i+1} - y_i}{h} - \left(\frac{y_i - y_{i-1}}{h} \right) \right) = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \approx y''(x_i)$$

question : How accurate is the approximation?

$y_{i+1} = y(x_{i+1}) = y(x_i + h)$: expand in a Taylor series about $x = x_i$

$$y_{i+1} = y_i + hy_i' + \tfrac{h^2}{2} y_i'' + \tfrac{h^3}{3!} y_i''' + \tfrac{h^4}{4!} y_i^{(4)} + \tfrac{h^5}{5!} y_i^{(5)} + O(h^6)$$

$$y_{i-1} = y_i - hy_i' + \tfrac{h^2}{2} y_i'' - \tfrac{h^3}{3!} y_i''' + \tfrac{h^4}{4!} y_i^{(4)} - \tfrac{h^5}{5!} y_i^{(5)} + O(h^6)$$

$$D_+ D_- y_i = \underbrace{\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}}_{\text{approximation}} = \underset{\substack{\uparrow \\ \text{exact} \\ \text{value}}}{y_i''} + \underbrace{\frac{h^2}{12} y_i^{(4)} + O(h^4)}_{\text{discretization error}} : \text{2nd order accurate}$$

w_i : numerical solution , $w_i \approx y_i$, $w_0 = \alpha$, $w_{n+1} = \beta$

$$-\left(\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} \right) = r_i \ , \ i = 1, \dots, n \ : \ \text{finite-difference equations}$$

$$\tfrac{1}{h^2} (-w_{i+1} + 2w_i - w_{i-1}) = r_i$$

$$i = 2 \ \Rightarrow \ \tfrac{1}{h^2} (-w_3 + 2w_2 - w_1) = r_2$$

$$i = 1 \ \Rightarrow \ \tfrac{1}{h^2} (-w_2 + 2w_1 - \alpha) = r_1$$

$$i = n \ \Rightarrow \ \tfrac{1}{h^2} (-\beta + 2w_n - w_{n-1}) = r_n$$

$$\frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_n \end{pmatrix} = \begin{pmatrix} r_1 + \alpha/h^2 \\ r_2 \\ \vdots \\ r_{n-1} \\ r_n + \beta/h^2 \end{pmatrix} \Rightarrow A_h w_h = r_h \qquad A_h : \begin{cases} \text{symmetric,} \\ \text{tridiagonal} \end{cases}$$

<u>questions</u>

1. Is A_h invertible?
2. Can w_h be computed efficiently?
3. Does $w_h \to y_h$ as $h \to 0$, i.e. does the numerical solution converge to the exact solution as the mesh is refined? If so, what is the order of accuracy?

<u>LU factorization for a tridiagonal system</u> (Thomas algorithm)

$$\begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & a_n & b_n \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{pmatrix}\begin{pmatrix} u_1 & c_1 & & & \\ & u_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & c_{n-1} \\ & & & & u_n \end{pmatrix}$$

<u>special case</u> : $n = 3$

$$\begin{pmatrix} b_1 & c_1 & 0 \\ a_2 & b_2 & c_2 \\ 0 & a_3 & b_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_2 & 1 & 0 \\ 0 & l_3 & 1 \end{pmatrix}\begin{pmatrix} u_1 & c_1 & 0 \\ 0 & u_2 & c_2 \\ 0 & 0 & u_3 \end{pmatrix}$$

<u>find L, U</u>

$b_1 = u_1 \quad \Rightarrow \quad u_1 = b_1$

$a_2 = l_2 u_1 \quad \Rightarrow \quad l_2 = a_2/u_1$

$b_2 = l_2 c_1 + u_2 \quad \Rightarrow \quad u_2 = b_2 - l_2 c_1 \quad , \quad \ldots$

<u>general case</u>

<u>find L, U</u>

$b_1 = u_1 \quad \Rightarrow \quad u_1 = b_1$

$$\left.\begin{array}{ll} a_k = l_k u_{k-1} & \Rightarrow \quad l_k = a_k/u_{k-1} \\ b_k = l_k c_{k-1} + u_k & \Rightarrow \quad u_k = b_k - l_k c_{k-1} \end{array}\right\} \text{ for } \quad k = 2 : n$$

<u>solve $Lz = r$</u>

$z_1 = r_1$

$l_k z_{k-1} + z_k = r_k \quad \Rightarrow \quad z_k = r_k - l_k z_{k-1} \quad \text{for} \quad k = 2 : n$

<u>solve $Uw = z$</u>

$u_n w_n = z_n \quad \Rightarrow \quad w_n = z_n/u_n$

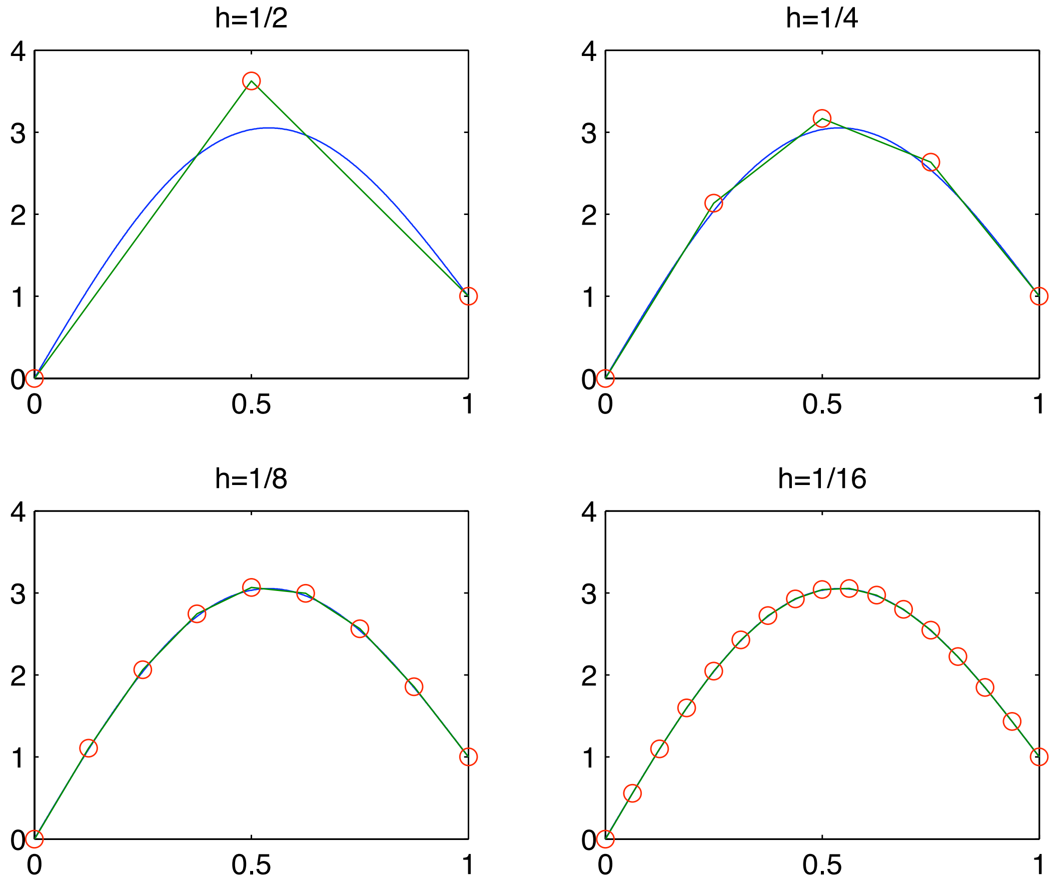
$u_k w_k + c_k w_{k+1} = z_k \quad \Rightarrow \quad w_k = (z_k - c_k w_{k+1})/u_k \quad \text{for} \quad k = n-1 : -1 : 1$

<u>note</u> : operation count $= O(n)$
memory $= O(n)$ if vectors are used instead of full matrices

11
Thurs
2/14

two-point bvp : $-y'' = 25\sin \pi x\,,\ 0 \le x \le 1\,,\ y(0) = 0\,,\ y(1) = 1$

solution : $y(x) = \dfrac{25}{\pi^2}\sin \pi x + x$, check ...



exact solution : $y(x)$ is plotted as a solid curve

numerical solution : w_h is plotted as circles connected by straight lines

The error is $||y_h - w_h||$, where y_h denotes the exact solution at the mesh points.

h	$\|\|y_h - w_h\|\|$	$\dfrac{\|\|y_h - w_h\|\|}{h}$	$\dfrac{\|\|y_h - w_h\|\|}{h^2}$	$\dfrac{\|\|y_h - w_h\|\|}{h^3}$
0.50000000	0.591970401	1.18394082	2.36788164	4.73576327
0.25000000	0.134324755	0.53729902	2.14919607	8.59678429
0.12500000	0.032804625	0.26243700	2.09949598	16.7959678
0.06250000	0.008153732	0.13045971	2.08735544	33.3976870

<u>note</u>

1. If h decreases by $\frac{1}{2}$, then the error decreases by approximately $\frac{1}{4}$.
2. We see that $||y_h - w_h|| = O(h^2)$, so the method is 2nd order accurate.

3.8 iterative methods

Gaussian elimination is a direct method for solving $Ax = b$, because it yields the exact solution x after a finite number of steps. In practice, the $O(n^3)$ operation count is an obstacle when n is large and memory is an issue too. Now we consider iterative methods, an alternative class of methods which generate a sequence of approximate solutions x_k such that $\lim\limits_{k\to\infty} x_k = x$. As we shall see, iterative methods have some advantages over direct methods.

$Ax = b \;\Leftrightarrow\; x = Bx + c$: equivalent linear system

$$x_{k+1} = Bx_k + c \;:\; \text{fixed-point iteration : given } x_0, \text{ compute } x_1, \ldots$$

B : iteration matrix

Jacobi method

$A = L + D + U$: this is different than LU factorization

$D = \mathrm{diag}(a_{11}, \ldots, a_{nn})$, assume $a_{ii} \neq 0$, $i = 1 : n$

$$L = \begin{pmatrix} 0 & & & & \\ a_{21} & 0 & & & \\ \vdots & \ddots & \ddots & & \\ \vdots & & \ddots & \ddots & \\ a_{n1} & \cdots & \cdots & a_{n,n-1} & 0 \end{pmatrix} \;,\quad U = \begin{pmatrix} 0 & a_{12} & \cdots & \cdots & a_{1n} \\ & 0 & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & a_{n-1,n} \\ & & & & 0 \end{pmatrix}$$

$$\begin{aligned} Ax = b \;&\Leftrightarrow\; (L + D + U)x = b \\ &\Leftrightarrow\; Dx = -(L + U)x + b \\ &\Leftrightarrow\; x = -D^{-1}(L + U)x + D^{-1}b \;,\quad B_J = -D^{-1}(L + U) \end{aligned}$$

$Dx_{k+1} = -(L + U)x_k + b$: easy to solve for x_{k+1}

component form

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \;\Rightarrow\; a_{11}x_1^{(k+1)} = b_1 - \left(a_{12}x_2^{(k)} + a_{13}x_3^{(k)}\right)$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \;\Rightarrow\; a_{22}x_2^{(k+1)} = b_2 - \left(a_{21}x_1^{(k)} + a_{23}x_3^{(k)}\right)$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \;\Rightarrow\; a_{33}x_3^{(k+1)} = b_3 - \left(a_{31}x_1^{(k)} + a_{32}x_2^{(k)}\right)$$

ex

$$2x_1 - \ \ x_2 = 1 \quad \Rightarrow \quad 2x_1^{(k+1)} = 1 + x_2^{(k)}$$
$$-x_1 + 2x_2 = 1 \quad \Rightarrow \quad 2x_2^{(k+1)} = 1 + x_1^{(k)}$$

The exact solution is $x_1 = x_2 = 1$. Let the initial guess be $x_1^{(0)} = x_2^{(0)} = 0$.

k	$x_1^{(k)}$	$x_2^{(k)}$
0	0	0
1	1/2	1/2
2	3/4	3/4
3	7/8	7/8

Hence the numerical solution converges to the exact solution as $k \to \infty$.

def : $e_k = x - x_k$: error at step k

In the example we have $||e_0|| = 1$, $||e_1|| = \frac{1}{2}$, $||e_2|| = \frac{1}{4}$, $\ldots$, $||e_{k+1}|| = \frac{1}{2}||e_k||$.

question : What determines the factor $\frac{1}{2}$?

thm

Consider a linear system $Ax = b$ and fixed-point iteration $x_{k+1} = Bx_k + c$.

1. $e_{k+1} = Be_k$ for all $k \geq 0$
2. If $||B|| < 1$, then $x_k \to x$ as $k \to \infty$ for any initial guess x_0.

pf

1. $e_{k+1} = x - x_{k+1} = (Bx + c) - (Bx_k + c) = B(x - x_k) = Be_k$
2. $||e_{k+1}|| = ||Be_k|| \leq ||B|| \cdot ||e_k|| = ||B|| \cdot ||Be_{k-1}|| \leq ||B|| \cdot ||B|| \cdot ||e_{k-1}||$

$\Rightarrow ||e_{k+1}|| \leq ||B||^2 \cdot ||e_{k-1}||$

$\ldots$

$\Rightarrow ||e_{k+1}|| \leq ||B||^{k+1} \cdot ||e_0|| \to 0$ as $k \to \infty$ ok

ex

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \Rightarrow B_J = -D^{-1}(L+U) = -\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$$

$\Rightarrow \ ||B_J|| = \frac{1}{2}$

Hence since $||B_J|| = \frac{1}{2} < 1$, the theorem implies that Jacobi's method converges, and the proof shows that $||e_k||$ decreases by a factor of at least $\frac{1}{2}$ in each step.

12 Tues 2/19

Gauss-Seidel method

$A = L + D + U$: as before

$$Ax = b \;\Leftrightarrow\; (L+D+U)x = b$$
$$\Leftrightarrow\; (L+D)x = -Ux + b$$
$$\Leftrightarrow\; x = -(L+D)^{-1}Ux + (L+D)^{-1}b \;\;,\;\; B_{GS} = -(L+D)^{-1}U$$

$(L+D)x_{k+1} = -Ux_k + b$: solve by forward substitution

component form

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \;\Rightarrow\; a_{11}x_1^{(k+1)} = b_1 - \left(a_{12}x_2^{(k)} + a_{13}x_3^{(k)}\right)$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \;\Rightarrow\; a_{22}x_2^{(k+1)} = b_2 - \left(a_{21}x_1^{(k+1)} + a_{23}x_3^{(k)}\right)$$
$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \;\Rightarrow\; a_{33}x_3^{(k+1)} = b_3 - \left(a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)}\right)$$

Hence $x_i^{(k+1)}$ is used as soon as it's computed, in contrast to Jacobi.

ex

$$2x_1 - x_2 = 1 \;\Rightarrow\; 2x_1^{(k+1)} = 1 + x_2^{(k)}$$
$$-x_1 + 2x_2 = 1 \;\Rightarrow\; 2x_2^{(k+1)} = 1 + x_1^{(k+1)}$$

k	$x_1^{(k)}$	$x_2^{(k)}$
0	0	0
1	1/2	3/4
2	7/8	15/16
3	31/32	63/64

Hence Gauss-Seidel converges faster than Jacobi.

$$||e_0|| = 1 \;,\; ||e_1|| = \tfrac{1}{2} \;,\; ||e_2|| = \tfrac{1}{8} \;,\; ||e_3|| = \tfrac{1}{32} \;,\; \ldots \;,\; ||e_{k+1}|| = \tfrac{1}{4}||e_k|| \text{ for } k \geq 1$$

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \Rightarrow B_{GS} = -(L+D)^{-1}U = -\tfrac{1}{4}\begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix}\begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}$$

$$\Rightarrow \;\; ||B_{GS}|| = \tfrac{1}{2}$$

Since $||B_{GS}|| = \frac{1}{2} < 1$, the theorem implies that Gauss-Seidel converges, but we see that $||e_k||$ decreases by a factor of $\frac{1}{4} < ||B_{GS}||$ in each step.

summary

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \;\Rightarrow\; B_J = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \;\Rightarrow\; ||B_J|| = \tfrac{1}{2}\;,\; ||e_{k+1}|| = \tfrac{1}{2}||e_k||$$

$$B_{GS} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix} \;\Rightarrow\; ||B_{GS}|| = \tfrac{1}{2}\;,\; ||e_{k+1}|| = \tfrac{1}{4}||e_k||$$

question : What determines the factor by which $||e_k||$ decreases in each step?

To answer this question, we need to recall some facts about eigenvalues and eigenvectors.

def : If $Ax = \lambda x$, where $x \neq 0$ is a vector and λ is a scalar (real or complex), then λ is an eigenvalue of A and x is a corresponding eigenvector.

ex : $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$$A\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \;\Rightarrow\; \lambda = 1 \text{ is an e-value with e-vector } x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$A\begin{pmatrix} -1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \;\Rightarrow\; \lambda = 1\;,\; x = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

$$A\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \;\Rightarrow\; \lambda = -1\;,\; x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

note

$$Ax = \lambda x\,,\; x \neq 0 \;\Leftrightarrow\; (A - \lambda I)x = 0\,,\; x \neq 0 \;\Leftrightarrow\; \det(A - \lambda I) = 0$$

$f_A(\lambda) = \det(A - \lambda I)$: characteristic polynomial of A

Hence the e-values of A are the roots of the characteristic polynomial $f_A(\lambda)$.

ex : $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$$f_A(\lambda) = \det(A - \lambda I) = \det\begin{pmatrix} -\lambda & 1 \\ 1 & -\lambda \end{pmatrix} = \lambda^2 - 1 = 0 \;\Rightarrow\; \lambda = \pm 1 \qquad \underline{\text{ok}}$$

thm : If A is upper triangular, then the e-values are the diagonal elements.

pf

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ & \ddots & \vdots \\ 0 & & a_{nn} \end{pmatrix} \;\Rightarrow\; A - \lambda I = \begin{pmatrix} a_{11} - \lambda & \cdots & a_{1n} \\ & \ddots & \vdots \\ 0 & & a_{nn} - \lambda \end{pmatrix}$$

$$f_A(\lambda) = \det(A - \lambda I) = (a_{11} - \lambda)\cdots(a_{nn} - \lambda) = 0 \;\Rightarrow\; \lambda = a_{ii} \text{ for some } i \qquad \underline{\text{ok}}$$

<u>recall</u> : $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \Rightarrow B_{GS} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}$

$\lambda_1 = 0$ is an e-value of B_{GS} with e-vector $v_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, check : $Bv_1 = \lambda v_1$

$\lambda_2 = \frac{1}{4}$ ...." .... $v_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$, check : $Bv_2 = \lambda v_2$

$e_0 = x - x_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = v_2 - v_1$

$e_1 = Be_0 = B(v_2 - v_1) = Bv_2 - Bv_1 = \lambda_2 v_2 - \lambda_1 v_1$

$e_2 = Be_1 = B(\lambda_2 v_2 - \lambda_1 v_1) = \lambda_2^2 v_2 - \lambda_1^2 v_1$

$\vdots$

$e_k = \lambda_2^k v_2 - \cancel{\lambda_1^k} v_1 = \left(\frac{1}{4}\right)^k v_2 \Rightarrow ||e_k|| = \left(\frac{1}{4}\right)^k ||v_2||$

This explains why $||e_{k+1}|| = \frac{1}{4}||e_k||$, even though $||B_{GS}|| = \frac{1}{2}$.

<u>question</u>

What determines the convergence rate of an iterative method? 13 Thurs 2/21

<u>def</u> : $\rho(B) = \max\{|\lambda| : \lambda \text{ is an e-value of } B\}$: <u>spectral radius</u> of B

<u>thm</u>

1. $||e_{k+1}|| \le ||B|| \cdot ||e_k||$ for all $k \ge 0$: error bound
2. $||e_{k+1}|| \sim \rho(B) \cdot ||e_k||$ as $k \to \infty$: asymptotic relation

This means that $\lim_{k\to\infty} \frac{||e_{k+1}||}{||e_k||} = \rho(B)$.

Hence the spectral radius of the iteration matrix $\rho(B)$ determines the convergence rate of an iterative method.

<u>pf</u>

1. recall : $e_{k+1} = Be_k \Rightarrow ||e_{k+1}|| = ||Be_k|| \le ||B|| \cdot ||e_k||$
2. Math 571 (but the idea is the same as in the example above)

$e_0 = \alpha_1 v_1 + \alpha_2 v_2 \Rightarrow e_k = B^k e_0 = \alpha_1 \lambda_1^k v_1 + \alpha_2 \lambda_2^k v_2 = \lambda_1^k \left(\alpha_1 v_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^k \alpha_2 v_2 \right)$ <u>ok</u>

<u>recall</u> : $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \Rightarrow B_J = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \Rightarrow \rho(B_J) = \frac{1}{2}$

$B_{GS} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix} \Rightarrow \rho(B_{GS}) = \frac{1}{4}$ <u>ok</u>

<u>question</u> : Are there faster methods?

Jacobi (1804-1851) , Gauss (1777-1855) , Seidel (1821-1896)

Richardson (1881-1953) : numerical weather forecasting

$Ax = b\ ,\ A = L + D + U$

Recall the Gauss-Seidel method.

$(L+D)x_{k+1} = -Ux_k + b \ \Leftrightarrow\ Dx_{k+1} = Dx_k - (Lx_{k+1} + (D+U)x_k - b)$

Now let ω be a free parameter and consider a modified iteration.

$Dx_{k+1} = Dx_k - \omega(Lx_{k+1} + (D+U)x_k - b)$

$\omega = 1 \Rightarrow$ GS , $\omega > 1$: <u>successive over-relaxation</u> (SOR)

<u>component form</u>

$$a_{11}x_1^{(k+1)} = a_{11}x_1^{(k)} + \omega(b_1 - (a_{11}x_1^{(k)} + a_{12}x_2^{(k)} + a_{13}x_3^{(k)}))$$
$$a_{22}x_2^{(k+1)} = a_{22}x_2^{(k)} + \omega(b_2 - (a_{21}x_1^{(k+1)} + a_{22}x_2^{(k)} + a_{23}x_3^{(k)}))$$
$$a_{33}x_3^{(k+1)} = a_{33}x_3^{(k)} + \omega(b_3 - (a_{31}x_1^{(k+1)} + a_{32}x_2^{(k+1)} + a_{33}x_3^{(k)}))$$

<u>ex</u>

$$\begin{aligned} 2x_1 - x_2 = 1 \quad &\Rightarrow \quad 2x_1^{(k+1)} = 2x_1^{(k)} + \omega(1 - (2x_1^{(k)} - x_2^{(k)})) \\ -x_1 + 2x_2 = 1 \quad &\Rightarrow \quad 2x_2^{(k+1)} = 2x_2^{(k)} + \omega(1 - (x_1^{(k+1)} + 2x_2^{(k)})) \end{aligned}$$

<u>matrix form</u>

$$(\omega L + D)x_{k+1} = ((1-\omega)D - \omega U))x_k + \omega b \ \Rightarrow\ B_\omega = (\omega L + D)^{-1}((1-\omega)D - \omega U))$$

<u>ex</u>

$$\begin{pmatrix} 2 & 0 \\ -\omega & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{k+1} = \begin{pmatrix} 2(1-\omega) & \omega \\ 0 & 2(1-\omega) \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_k + \omega \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$B_\omega = \begin{pmatrix} 2 & 0 \\ -\omega & 2 \end{pmatrix}^{-1} \begin{pmatrix} 2(1-\omega) & \omega \\ 0 & 2(1-\omega) \end{pmatrix} = \begin{pmatrix} 1-\omega & \frac{1}{2}\omega \\ \frac{1}{2}\omega(1-\omega) & \frac{1}{4}\omega^2 + 1 - \omega \end{pmatrix}$$

check : $\omega = 1 \ \Rightarrow\ B_\omega = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}$: GS , $\rho(B_\omega) = \frac{1}{4}$ <u>ok</u>

<u>question</u> : Can we choose ω so that $\rho(B_\omega)$ is smaller?

<u>thm</u> (Young 1950)

14 Tues 2/26

1. If $\rho(B_\omega) < 1$, then $0 < \omega < 2$.

2. Assume A is symmetric, block tridiagonal, and positive definite (defined later).

Then $\omega_* = \dfrac{2}{1+\sqrt{1-\rho(B_J)^2}}$ is the <u>optimal SOR parameter</u> in the sense that

$\rho(B_{\omega_*}) = \min\limits_{0<\omega<2} \rho(B_\omega) = \omega_* - 1 < \rho(B_{GS}) < \rho(B_J) < 1.$

<u>pf</u> : Math 571 (sometimes)

return to example : $\omega_* = \dfrac{2}{1+\sqrt{1-\rho(B_J)^2}} = \dfrac{2}{1+\sqrt{1-(\frac{1}{2})^2}} = \dfrac{4}{2+\sqrt{3}} = 1.0718$

k	$x_1^{(k)}$	$x_2^{(k)}$	$\|\|e_k\|\|$	$\|\|e_k\|\|/\|\|e_{k-1}\|\|$
0	0.0000	0.0000	1.0000	$\cdots$
1	0.5359	0.8231	0.4641	0.4641
2	0.9385	0.9798	0.0615	0.1325
3	0.9936	0.9980	0.0064	0.1047
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
∞	1	1	0	$\rho(B_{\omega_*}) = \omega_* - 1 = 0.0718$

Hence optimal SOR converges faster than GS.

<u>def</u> : A is <u>positive definite</u> if $x^TAx > 0$ for all $x \neq 0$

<u>ex 1</u> : $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$ is positive definite

<u>pf</u> : $x^TAx = (\,x_1\,,\,x_2\,)\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = (\,x_1\,,\,x_2\,)\begin{pmatrix} 2x_1 - x_2 \\ -x_1 + 2x_2 \end{pmatrix}$

$$= 2(x_1^2 + x_2^2) - 2x_1x_2 = x_1^2 + x_2^2 + (x_1 - x_2)^2 \geq 0$$

If $x \neq 0$, then either $x_1 \neq 0$ or $x_2 \neq 0$, but in any case we have $x^TAx > 0$. <u>ok</u>

<u>ex 2</u> : $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ is positive definite : hw

<u>ex 3</u> : $A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$ is <u>not</u> positive definite

<u>pf</u> : $x^TAx = (\,x_1\,,\,x_2\,)\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = x_1^2 + x_2^2 + 4x_1x_2$: indefinite

for example : $x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow x^TAx = 1$, $x = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \Rightarrow x^TAx = -2$ <u>ok</u>

<u>ex 4</u>

$$A_h = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} \ : \ \text{dimension } n \times n \ , \ h = \frac{1}{n+1}$$

The matrix A_h represents the finite difference operator $-D_+D_-$; A_h is symmetric, tridiagonal, and positive definite, and hence Young's theorem applies.

<u>note</u> : The real advantage of iterative methods, in comparison with direct methods, is for BVPs in more than one dimension.

<u>3.9 two-dimensional BVP</u>

15 Tues 3/12

<u>problem</u> : A metal plate has a square shape. The plate is heated by internal sources and the edges are held at a given temperature. Find the temperature at points inside the plate.

$D = \{(x,y) : 0 \le x, y \le 1\}$: plate domain

$\phi(x,y)$: temperature

$f(x,y)$: heat sources , $g(x,y)$: boundary temperature

Then $\phi(x,y)$ satisfies the following two equations.

1. $-\Delta\phi = -\nabla^2\phi = -\left(\dfrac{\partial^2\phi}{\partial x^2} + \dfrac{\partial^2\phi}{\partial y^2}\right) = f$ for (x,y) in D : <u>Poisson equation</u>
   $\uparrow$
   Laplace operator

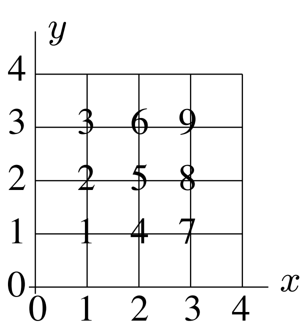
(note : This equation arises in many areas, e.g. if f is a charge/mass distribution, then ϕ is the electrostatic/gravitational potential.)

2. $\phi = g$ for (x,y) on ∂D : <u>Dirichlet boundary condition</u>

<u>finite-difference scheme</u>

$h = \dfrac{1}{n+1}$: mesh size , $(x_i, y_j) = (ih, jh)$, $i, j = 0, \ldots, n+1$: mesh points

<u>ex</u> : $n = 3$, $h = \frac{1}{4}$



$\phi(x_i, y_j)$: exact solution

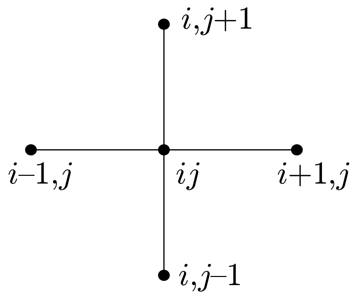
w_{ij} : numerical solution

ordering of mesh points : $w_{11}, w_{12}, \ldots$

$-\left(D_+^x D_-^x w_{ij} + D_+^y D_-^y w_{ij}\right) = f_{ij}$: finite-difference equations

$$-\left(\frac{w_{i+1,j} - 2w_{ij} + w_{i-1,j}}{h^2} + \frac{w_{i,j+1} - 2w_{ij} + w_{i,j-1}}{h^2}\right) = f_{ij}$$

$$\tfrac{1}{h^2}\left(4w_{ij} - w_{i+1,j} - w_{i-1,j} - w_{i,j+1} - w_{i,j-1}\right) = f_{ij}$$



: <u>5-point stencil</u>

Consider what happens near the boundary.

$$(i,j) = (1,1) \;\Rightarrow\; \tfrac{1}{h^2}\left(4w_{11} - w_{21} - w_{01} - w_{12} - w_{10}\right) = f_{11}$$

$$\Rightarrow\; \tfrac{1}{h^2}\left(4w_{11} - w_{21} - w_{12}\right) = f_{11} + \tfrac{1}{h^2}\left(g_{01} + g_{10}\right)$$

Write the equations for w_{ij} in matrix form.

1	2	3	4	5	6	7	8	9
w_{11}	w_{12}	w_{13}	w_{21}	w_{22}	w_{23}	w_{31}	w_{32}	w_{33}
4	−1		−1					
−1	4	−1		−1				
	−1	4			−1			
−1			4	−1		−1		
	−1		−1	4	−1		−1	
		−1		−1	4			−1
			−1			4	−1	
				−1		−1	4	−1
					−1		−1	4

$$A_h w_h = f_h \;,\; A_h = \begin{pmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & T \end{pmatrix}$$

$T : n \times n$, symmetric , tridiagonal

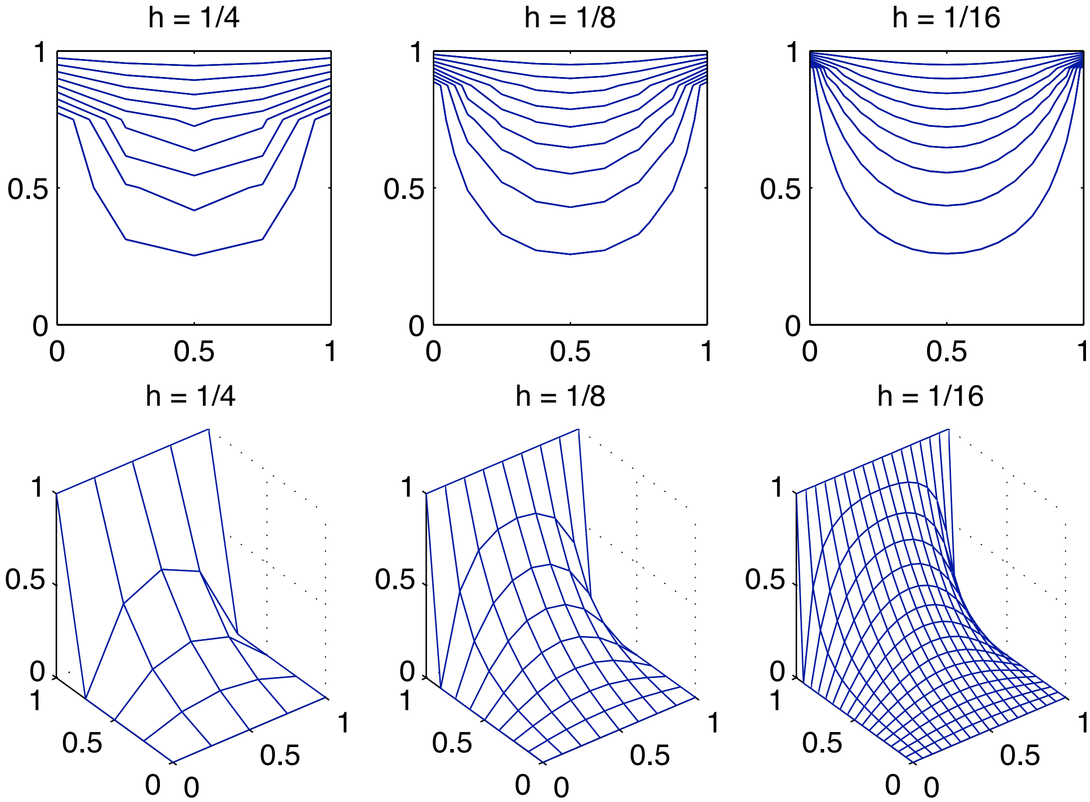
$A_h : n^2 \times n^2$, symmetric , block tridiagonal , positive definite (pf : omit)

<u>temperature distribution on a metal plate</u> : no heat sources, one side heated

differential equation : $\phi_{xx} + \phi_{yy} = 0$

boundary conditions : $\phi(x,1) = 1$, $\phi(x,0) = \phi(0,y) = \phi(1,y) = 0$

finite-difference scheme : $D_+^x D_-^x w_{ij} + D_+^y D_-^y w_{ij} = 0$



<u>above</u> : solution of linear system $A_h w_h = f_h$ for given mesh size h

<u>below</u> : number of iterations k required for each method

initial guess = zero vector, stopping criterion : $||r_k||/||r_0|| \leq 10^{-4}$

Jacobi

h	k	$\rho(B)$
1/4	26	0.7071
1/8	96	0.9239
1/16	334	0.9808

Gauss-Seidel

h	k	$\rho(B)$
1/4	15	0.5000
1/8	51	0.8536
1/16	172	0.9619

optimal SOR

h	k	$\rho(B)$
1/4	9	0.1716
1/8	18	0.4465
1/16	34	0.6735

16 Thurs 3/14

<u>note</u>

1. For each method, more iterations are needed as the mesh size $h \to 0$. Hence refining the mesh yields a more accurate solution of the BVP, but the computational cost increases.

2. For a given mesh size h, SOR converges the fastest, then GS, and then J.

3. Explicit formulas for $\rho(B)$ can be derived in this example. (Math 571)

$$\rho(B_J) = \cos \pi h \sim 1 - \tfrac{1}{2}\pi^2 h^2$$

$$\rho(B_{GS}) = \cos^2 \pi h \sim 1 - \pi^2 h^2$$

$$\rho(B_{\omega_*}) = \frac{2}{1+\sqrt{1-\rho(B_J)^2}} - 1 = \frac{1-\sin \pi h}{1+\sin \pi h} \sim \frac{1-\pi h}{1+\pi h} \sim 1 - 2\pi h$$

This shows that $\rho(B) \to 1$ as $h \to 0$ (confirming that the iteration slows down as the mesh is refined). The formulas also show that $\rho(B_{\omega_*}) < \rho(B_{GS}) < \rho(B_J) < 1$ (confirming that SOR converges the fastest, then GS, and then J).

4. Consider what happens if Gaussian elimination is used instead of J/GS/SOR.

$$\begin{pmatrix}
4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & \cancel{0} & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & \cancel{0} & \cancel{0} & -1 & 0 & 0 & 0 \\
-1 & \cancel{0} & \cancel{0} & 4 & -1 & \cancel{0} & -1 & 0 & 0 \\
0 & -1 & \cancel{0} & -1 & 4 & -1 & \cancel{0} & -1 & 0 \\
0 & 0 & -1 & \cancel{0} & -1 & 4 & \cancel{0} & \cancel{0} & -1 \\
0 & 0 & 0 & -1 & \cancel{0} & \cancel{0} & 4 & -1 & \cancel{0} \\
0 & 0 & 0 & 0 & -1 & \cancel{0} & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & \cancel{0} & -1 & 4
\end{pmatrix}$$

a) A_h is a <u>band matrix</u>, i.e. $a_{ij} = 0$ for $|i-j| > m$, where m is the <u>bandwidth</u> (in this example we have $m = 3$).

b) As the elimination proceeds, zeros inside the band can become non-zero (this is called <u>fill-in</u>), but zeros outside the band are preserved. Hence we can adjust the limits on the loops to reduce the operation count for Gaussian elimination from $O(n^3)$ to $O(nm^2)$.

c) Due to fill-in, more memory needs to be allocated than is required for the original matrix A_h. This is a disadvantage in comparison with iterative methods like J/GS/SOR which preserve the <u>sparsity</u> of A_h.

final comments on linear systems

1. comparison of operation counts : two-dimensional BVP

mesh size : $h = \frac{1}{n+1}$

typical equation : $\frac{1}{h^2}(4w_{ij} - w_{i+1,j} - w_{i-1,j} - w_{i,j+1} - w_{i,j-1}) = f_{ij}$

vector w_{ij} has length n^2

matrix A_h has dimension $n^2 \times n^2$ and bandwidth $m = n$

a) Gaussian elimination : $O((n^2)^3) = O(n^6)$ ops

banded Gaussian elimination : $O(n^2 m^2) = O(n^4)$ ops

b) iterative methods

cost per iteration : $O(n^2)$ ops (roughly the same for J/GS/SOR)

stopping criterion : $\dfrac{||r_k||}{||r_0||} = \epsilon \ \Rightarrow\ \rho(B)^k = \epsilon \ \Rightarrow\ k = \dfrac{\log \epsilon}{\log \rho(B)}$

J , GS $\Rightarrow$ $\rho(B) \sim 1 - ch^2 \ \Rightarrow\ \log \rho(B) \sim \log(1 - ch^2) \sim -ch^2$

$$\Rightarrow\ k \sim \frac{\log \epsilon}{-ch^2} = O(n^2) \text{ iterations}$$

$$\Rightarrow\ \text{total cost} = O(n^2) \times O(n^2) = O(n^4) \text{ ops}$$

17 Tues 3/19

SOR $\Rightarrow$ $\rho(B) \sim 1 - ch$

$$\Rightarrow\ k \sim \frac{\log \epsilon}{-ch} = O(n) \text{ iterations}$$

$$\Rightarrow\ \text{total cost} = O(n^2) \times O(n) = O(n^3) \text{ ops}$$

2. developments after SOR
   - conjugate gradient method
   - FFT = fast Fourier transform
   - multigrid
   - GMRES
   - preconditioning : $Ax = b \to PAx = Pb$
   - software
   - parallel