section 3.0 : 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, 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.

note

1. If A is invertible, then $x = A^{-1}b$ (because then $Ax = A(A^{-1}b) = (AA^{-1})b = Ib = b$), 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.

section 3.1 : 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. $\quad sum = b_i$
4. $\quad$ for $j = i+1 : n$ % j : column index
5. $\qquad sum = sum - a_{ij} \cdot x_j$
6. $\quad x_i = sum/a_{ii}$

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 $= 1 + 2 + \cdots + (n-1) = \sum_{i=1}^{n-1} i = S$

$$2S = \sum_{i=1}^{n-1} i + \sum_{i=1}^{n-1} (n-i) = \sum_{i=1}^{n-1} (i + (n-i)) = \sum_{i=1}^{n-1} n = n(n-1)$$

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

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

note : Similar considerations hold if A is lower triangular.

<u>note</u>

In the case of 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: multiply an equation by a nonzero constant and subtract from another equation

<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

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

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

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

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

---- these elements have changed

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

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

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

: upper triangular

ex

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

$$\left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & 1 \end{array}\right) \quad \begin{aligned} m_{21} &= -1/2 \\ m_{31} &= 0 \end{aligned}$$

$$\left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ 0 & 3/2 & -1 & 1/2 \\ 0 & -1 & 2 & 1 \end{array}\right) \quad m_{32} = -1/(3/2) = -2/3$$

$$\left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ 0 & 3/2 & -1 & 1/2 \\ 0 & 0 & 4/3 & 4/3 \end{array}\right)$$

$x_3 = 1$, $x_2 = (\frac{1}{2} - (-1) \cdot 1)/\frac{3}{2} = 1$, $x_1 = (1 - (-1) \cdot 1)/2 = 1$ 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. $\quad\quad a_{ij} = a_{ij} - m_{ik} \cdot a_{kj}$
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.

$$\left.\begin{aligned} k = 1 &\Rightarrow 2(n-1)^2 \text{ ops} \\ k = 2 &\Rightarrow 2(n-2)^2 \text{ ops} \\ &\vdots \\ k = n-2 &\Rightarrow 2 \cdot 2^2 \text{ ops} \\ k = n-1 &\Rightarrow 2 \cdot 1^2 \text{ ops} \end{aligned}\right\} \Rightarrow 2 \cdot \sum_{k=1}^{n-1} k^2 = 2 \cdot \frac{(n-1)n(2n-1)}{6}$$

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

<u>note</u>

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

<u>pf</u>

1. already done

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

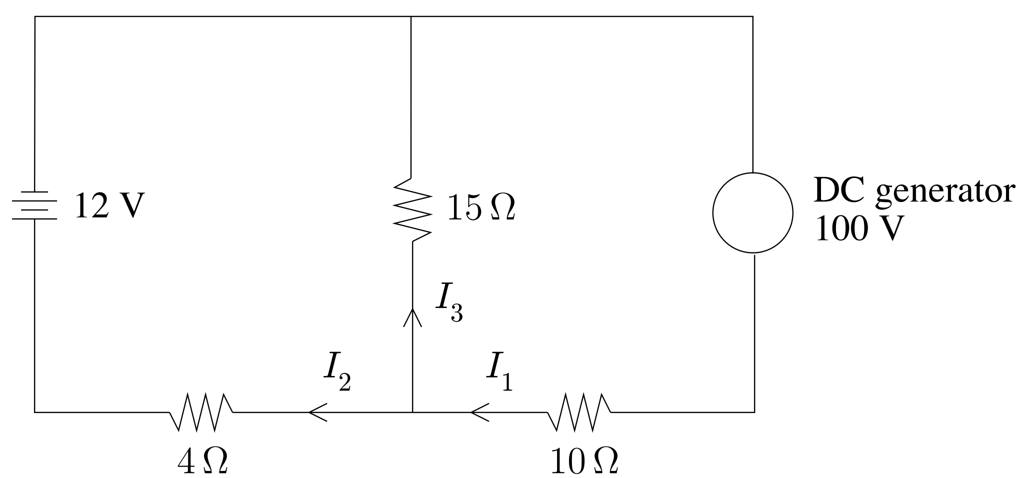
$k^3 - (k-1)^3 = k^3 - (k^3 - 3k^2 + 3k - 1) = 3k^2 - 3k + 1$

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

$$S = \sum_{k=1}^{n} k^2 = \frac{1}{3}\left(n^3 + \frac{3}{2}n(n+1) - n\right) = \frac{1}{3}n\left(n^2 + \frac{3}{2}(n+1) - 1\right)$$

$$= \frac{1}{3}n\left(n^2 + \frac{3}{2}n + \frac{1}{2}\right) = \frac{1}{3}n \cdot \frac{1}{2}(2n^2 + 3n + 1) = \frac{1}{3}n \cdot \frac{1}{2}(2n+1)(n+1) \quad \underline{\text{ok}}$$

<u>ex</u> : electric circuit for charging a car battery (page 129, problem 13)



Apply Kirchoff's voltage law and current law to determine the currents.

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

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

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

$\Rightarrow$ $I_1 = I_2 + I_3$

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

Hence we can't apply Gaussian elimination because the 1st pivot is zero.

section 3.2 : pivoting

There are various pivoting 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. Interchanging the rows can be done implicitly, using an index array, to avoid the expense of moving the matrix elements in memory.

3. Other strategies are scaled partial pivoting and complete pivoting, but we won't consider these.

4. 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.

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

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) section 3.3 : vector and matrix norms

def : A vector norm is a function $x \to ||x||$ 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

note : The vector norm measures the size of a vector.

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 transformation $x \to Ax$ as input $\to$ output. Then $\frac{||Ax||}{||x||}$ 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, $||A|| = \max_{x \neq 0} \frac{||Ax||}{||x||}$. 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 property 5

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

ok

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

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

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

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

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

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

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

<u>pf</u> (thm)

$$||Ax||_\infty = \max_i |(Ax)_i| = \max_i \left| \sum_j a_{ij}x_j \right| \le \max_i \sum_j |a_{ij}||x_j| \le \max_i \sum_j |a_{ij}| \cdot ||x||_\infty$$

$\Rightarrow \dfrac{||Ax||_\infty}{||x||_\infty} \le \max_i \sum_j |a_{ij}| = \sum_j |a_{lj}|$ for some index l (this holds for all $x \neq 0$)

Define $y = (\text{sign}(a_{l1}), \ldots, \text{sign}(a_{ln}))^T$.

$$\Rightarrow ||y||_\infty = 1 \ , \ \sum_j |a_{lj}| = \sum_j a_{lj}y_j = (Ay)_l \le ||Ay||_\infty$$

$\Rightarrow ||A||_\infty = \max_{x\neq 0} \dfrac{||Ax||_\infty}{||x||_\infty} \le \sum_j |a_{lj}| \le \dfrac{||Ay||_\infty}{||y||_\infty} \le \max_{x\neq 0} \dfrac{||Ax||_\infty}{||x||_\infty} = ||A||_\infty$ <u>ok</u>

<u>thm</u> : $||A||_2 = \max_{x\neq 0} \dfrac{||Ax||_2}{||x||_2} = \max\{\sqrt{\lambda} : \lambda \text{ is an eigenvalue of } A^TA\}$

<u>ex</u> : $A^TA = \begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix}\begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 8 \end{pmatrix} \Rightarrow ||A||_2 = 2.9208$ (Matlab)

$$x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow Ax = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow \frac{||Ax||_2}{||x||_2} = \frac{1}{1} = 1$$

$$x = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow Ax = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \Rightarrow \frac{||Ax||_2}{||x||_2} = \frac{\sqrt{8}}{1} = 2.8284$$

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow Ax = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \Rightarrow \frac{||Ax||_2}{||x||_2} = \frac{\sqrt{13}}{\sqrt{2}} = 2.5495$$

<u>pf</u> (thm) Math 571

section 3.4 : error analysis

$Ax = b$, A : invertible

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

$e = x - \tilde{x}$: error , $r = b - A\tilde{x}$: residual

note : $Ae = r$, pf : $Ae = A(x - \tilde{x}) = Ax - A\tilde{x} = b - A\tilde{x} = r$ ok

Then $e = 0$ if and only if $r = 0$, but if $||r||$ is small, there's no guarantee that $||e||$ is also small.

ex

$$\left(\begin{array}{cc:c} 1.01 & 0.99 & 2 \\ 0.99 & 1.01 & 2 \end{array}\right) \Rightarrow x = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\tilde{x}_1 = \begin{pmatrix} 1.01 \\ 1.01 \end{pmatrix} \Rightarrow e_1 = \begin{pmatrix} -0.01 \\ -0.01 \end{pmatrix}, ||e_1||_\infty = 0.01 , r_1 = \begin{pmatrix} -0.02 \\ -0.02 \end{pmatrix}, ||r_1||_\infty = 0.02$$

$$\tilde{x}_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \Rightarrow e_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad ||e_2||_\infty = 1 , \quad r_2 = \begin{pmatrix} -0.02 \\ 0.02 \end{pmatrix}, ||r_2||_\infty = 0.02$$

Hence $||r||$ is small in both cases, while $||e||$ is small in case 1 and 100 times larger in case 2. How large can $||e||$ be?

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

ex

$$A = \begin{pmatrix} 1.01 & 0.99 \\ 0.99 & 1.01 \end{pmatrix} \Rightarrow ||A||_\infty = 2$$

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

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

pf

1. $||b|| = ||Ax|| \leq ||A|| \cdot ||x|| \Rightarrow ||x|| \geq ||b||/||A||$

2. $Ae = r \Rightarrow e = A^{-1}r \Rightarrow ||e|| = ||A^{-1}r|| \leq ||A^{-1}|| \cdot ||r||$

3. $\dfrac{||e||}{||x||} \leq \dfrac{||A^{-1}|| \cdot ||r||}{||b||/||A||} = \kappa(A) \cdot \dfrac{||r||}{||b||}$ ok

note

1. $\left.\begin{array}{l} Ax = b \\ A\tilde{x} = \tilde{b} \end{array}\right\} \Rightarrow \dfrac{||x-\tilde{x}||}{||x||} \le \kappa(A)\dfrac{||b-\tilde{b}||}{||b||}$: perturbation in RHS , pf: ok

2. $\left.\begin{array}{l} Ax = b \\ \tilde{A}\tilde{x} = b \end{array}\right\} \Rightarrow \dfrac{||x-\tilde{x}||}{||\tilde{x}||} \le \kappa(A)\dfrac{||A-\tilde{A}||}{||A||}$: perturbation in matrix , pf: hw

Hence $\kappa(A)$ controls the error in the solution due to perturbations in A and b.

ex (recall)

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

Now consider the effect of roundoff error.

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

explanation

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

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

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

$$\Rightarrow \kappa_\infty(U) = |1-\tfrac{1}{\epsilon}|\cdot\frac{1}{|\epsilon-1|}\cdot(|1-\tfrac{1}{\epsilon}|+1) \approx \frac{1}{\epsilon^2} : \text{can be larger than } \kappa_\infty(A)$$

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 previous example). This means that Gaussian elimination is an unstable method for solving $Ax = b$ because it can replace a well-conditioned matrix A by an ill-conditioned matrix U. Pivoting however produces a different reduced system.

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

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

In fact, Gaussian elimination + partial pivoting + IEEE arithmetic is stable (in most cases).

section 3.5 : LU factorization

matrix form of Gaussian elimination

We consider the 3×3 case (but the general $n \times n$ 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

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

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

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

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

$$E_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix} \ , \ E_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{pmatrix}$$

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

final result : $A = LU$

ex

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & -1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix}$$

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

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

check :

$$LU = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & -\frac{2}{3} & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} = A \quad \underline{\text{ok}}$$

note

To solve $Ax = b$.

step 1. factor $A = LU$

step 2. solve $Ly = b$ by forward substitution

step 3. solve $Ux = y$ by back substitution

check : $Ax = LUx = Ly = b$ ok

ex

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}, \ b = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \Rightarrow x = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

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 different vectors b (e.g. in 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.

<u>LU factorization and partial pivoting</u>

To perform partial pivoting we need to interchange rows and this can be represented using a <u>permutation matrix</u>. Instead of $A = LU$, the final result is $PA = LU$, where P is a permutation matrix.

<u>ex</u>

$$\begin{pmatrix} 0 & 4 & -15 \\ 10 & 0 & 15 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -12 \\ 100 \\ 0 \end{pmatrix}$$

We want to interchange rows 1 and 2.

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 4 & -15 \\ 10 & 0 & 15 \\ 1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 1 & -1 & -1 \end{pmatrix} , \quad P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 1 & -1 & -1 \end{pmatrix} \rightarrow \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 0 & -1 & -2.5 \end{pmatrix} \rightarrow \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 0 & 0 & -6.25 \end{pmatrix}$$

$m_{21} = \frac{0}{10} = 0 \qquad m_{32} = \frac{-1}{4} = -0.25$

$m_{31} = \frac{1}{10}$

check : $PA = LU \ \ldots$

Then $Ax = b \Rightarrow PAx = Pb \Rightarrow LUx = Pb$ and we can apply forward and back substitution to find x.

$$Ly = Pb \ \Rightarrow \ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.1 & -0.25 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 100 \\ -12 \\ 0 \end{pmatrix} \ \Rightarrow \ \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 100 \\ -12 \\ -13 \end{pmatrix}$$

$$Ux = y \ \Rightarrow \ \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 0 & 0 & -6.25 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 100 \\ -12 \\ -13 \end{pmatrix} \ \Rightarrow \ \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6.88 \\ 4.80 \\ 2.08 \end{pmatrix}$$

<u>note</u>

If pivoting is required in more than one step, we proceed as follows.

$E_2 P_2 E_1 P_1 A = U$, but it can be shown that $P_2 E_1 = \tilde{E}_1 P_2$ (hw)

$\Rightarrow \ E_2 \tilde{E}_1 P_2 P_1 A = U \ \Rightarrow \ PA = LU$, where $P = P_2 P_1$, $L = \tilde{E}_1^{-1} E_2^{-1}$

section 8.1 : 2-point boundary value problem

Find $y(x)$ on $0 \le x \le 1$ satisfying the differential equation $-y'' + q(x)y = r(x)$ subject to boundary conditions $y(0) = \alpha, y(1) = \beta$, where $q(x), r(x)$ are given. The equation models a steady state convection-reaction-diffusion system, where $y(x)$ represents a velocity or temperature profile (for example).

finite-difference scheme

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

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

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

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

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 $x = x_i$

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

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

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

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

$$-\left(\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2}\right) + q_i w_i = r_i \ , \ i = 1, \ldots, n$$

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

$$i = 1 \ \Rightarrow \ \tfrac{1}{h^2}\left(-w_2 + \left(2 + q_1 h^2\right) w_1 - \alpha\right) = r_1$$

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

$$\frac{1}{h^2}\begin{pmatrix} 2+q_1h^2 & -1 & & & & \\ -1 & 2+q_2h^2 & -1 & & & \\ \ddots & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2+q_{n-1}h^2 & -1 \\ & & & & -1 & 2+q_nh^2 \end{pmatrix}\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ \vdots \\ \vdots \\ w_{n-1} \\ w_n \end{pmatrix} = \begin{pmatrix} r_1+\alpha/h^2 \\ r_2 \\ \vdots \\ \vdots \\ \vdots \\ r_{n-1} \\ r_n+\beta/h^2 \end{pmatrix}$$

$A_h w_h = r_h$, A_h : tridiagonal , symmetric

<u>questions</u>

1. Is A_h invertible for all $h, q(x), r(x)$?

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>find L, U</u>

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

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

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

$$z_1 = r_1$$

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

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

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

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

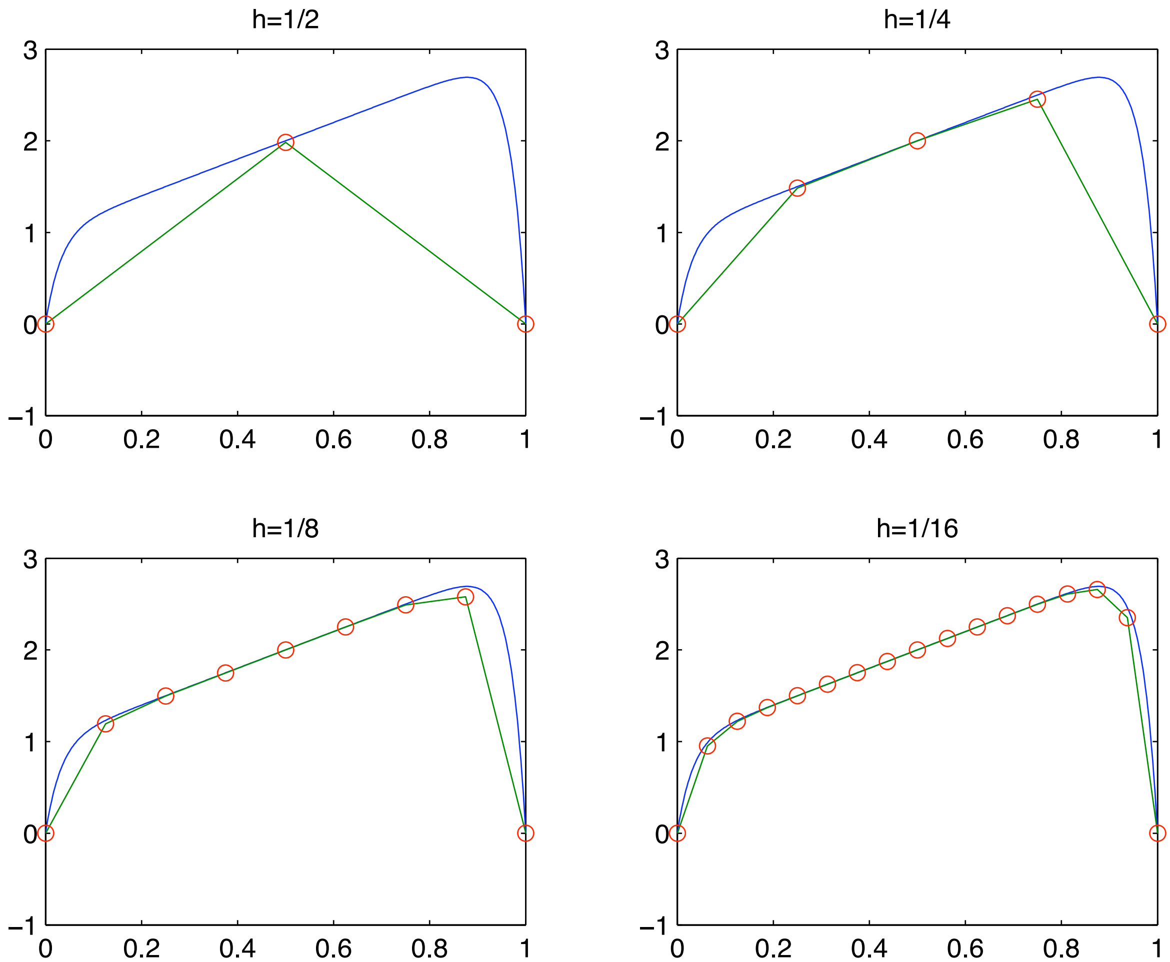
<u>note</u>

operation count $= O(n)$

memory $= O(n)$ if vectors are used instead of full matrices

2-point bvp : $-\epsilon y'' + y = 2x + 1\,,\ 0 \le x \le 1\,,\ y(0) = 0\,,\ y(1) = 0\,,\ \epsilon = 10^{-3}$

solution : $y(x) = 2x + 1 - (\sinh \frac{1-x}{\sqrt{\epsilon}} + 3\sinh \frac{x}{\sqrt{\epsilon}})/\sinh \frac{1}{\sqrt{\epsilon}}$, check : hw



h	$\|\|y_h - w_h\|\|_\infty$	$\dfrac{\|\|y_h - w_h\|\|_\infty}{h}$	$\dfrac{\|\|y_h - w_h\|\|_\infty}{h^2}$	$\dfrac{\|\|y_h - w_h\|\|_\infty}{h^3}$
0.50000000	0.015872	0.031744	0.0634890	0.126979
0.25000000	0.045420	0.181682	0.7267300	2.906920
0.12500000	0.113164	0.905315	7.2425200	57.94016
0.06250000	0.107705	1.723287	27.572593	441.1615
0.03125000	0.041333	1.322659	42.325086	1354.402
0.01562500	0.010982	0.702852	44.982586	2878.885
0.00781250	0.002790	0.357233	45.725862	5852.910
0.00390625	0.000701	0.179364	45.917351	11754.84

1. For $\frac{1}{8} \le h \le \frac{1}{2}$, the error increases as h decreases. This is due to the presence of boundary layers (look closely at the plots).

2. For $h \le \frac{1}{32}$, if h decreases by $\frac{1}{2}$, then the error decreases by approximately $\frac{1}{4}$.

3. We see that $||y_y - w_h||_\infty = O(h^2)$, so the method is 2nd order accurate.

section 3.8 : iterative methods

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

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

$x_{k+1} = Bx_k + c$: fixed-point iteration

B : iteration matrix

Jacobi method

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

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

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

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

$Dx_{k+1} = -(L + U)x_k + b$

component form

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

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

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

<u>ex</u>

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

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

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 $k \to \infty$.

<u>def</u>: $e_k = x - x_k$: error at step k

$$||e_0||_\infty = 1 \;,\; ||e_1||_\infty = \tfrac{1}{2} \;,\; ||e_2||_\infty = \tfrac{1}{4} \;,\; ||e_3||_\infty = \tfrac{1}{8} \;\Rightarrow\; ||e_{k+1}||_\infty = \tfrac{1}{2}||e_k||_\infty$$

<u>thm</u>

Consider a fixed-point iteration of the form $x_{k+1} = Bx_k + c$.

1. $e_{k+1} = Be_k$
2. If $||B|| < 1$ for any matrix norm, then $x_k \to x$ for any initial guess x_0.

<u>pf</u>

1. $e_{k+1} = x - x_{k+1} = (Bx + c) - (Bx_k + c) = B(x - x_k) = Be_k$
2. $e_k = Be_{k-1} = B^2 e_{k-2} = \cdots = B^k e_0 \;\Rightarrow\; ||e_k|| = ||B^k e_0|| \le ||B^k|| \cdot ||e_0||$

$$||B^2|| = ||B \cdot B|| \le ||B|| \cdot ||B|| = ||B||^2 \;\Rightarrow\; ||B^k|| \le ||B||^k$$

$$\Rightarrow\; ||e_k|| \le ||B||^k \cdot ||e_0|| \to 0 \quad \text{as} \quad k \to \infty$$ <u>ok</u>

<u>ex</u>

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

$$\Rightarrow\; ||B_J||_\infty = \tfrac{1}{2}$$

1. Since $||B_J||_\infty < 1$, the theorem implies that Jacobi's method converges.
2. In each step the norm of the error decreases by $\frac{1}{2}$.

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 with 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.

$$\|e_0\|_\infty = 1 \;,\; \|e_1\|_\infty = \tfrac{1}{2} \;,\; \|e_2\|_\infty = \tfrac{1}{8} \;,\; \|e_3\|_\infty = \tfrac{1}{32} \;\Rightarrow\; \|e_{k+1}\|_\infty = \tfrac{1}{4}\|e_k\|_\infty$$

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

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

1. Since $\|B_{GS}\|_\infty < 1$, the theorem implies that Gauss-Seidel converges.

2. In each step the norm of the error decreases by $\frac{1}{4}$.

summary

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$$

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

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

We see that $||e_{k+1}||_\infty \leq ||B||_\infty \cdot ||e_k||_\infty$ in both cases, but the bound is not sharp in the case of GS (because $\frac{1}{4} < ||B_{GS}||_\infty$). To explain this, we need to consider the eigenvalues of the iteration matrix.

def

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

ex

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \;:\; \text{permutation matrix}$$

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

$$A\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \;\Rightarrow\; \lambda = -1 \;\ldots\ldots\ldots\;\text{''}\;\ldots\ldots\ldots\; x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

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

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

pf

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

$f_A(\lambda) = \det(A - \lambda I) = (a_{11} - \lambda)\cdots(a_{nn} - \lambda) = 0 \Rightarrow \lambda = a_{ii}$ for some i ok recall : $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \Rightarrow B_{GS} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}$

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

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

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

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

$$e_2 = Be_1 = B(\lambda_2 v_2) = \lambda_2 B v_2 = \lambda_2 \cdot \lambda_2 v_2 = \lambda_2^2 v_2$$

$$\vdots$$

$$e_k = \lambda_2^k v_2 = \left(\tfrac{1}{4}\right)^k v_2 \;\Rightarrow\; ||e_k||_\infty = \left(\tfrac{1}{4}\right)^k ||v_2||_\infty$$

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

question : What determines the rate of convergence of an iterative method?

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

thm

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

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

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

pf

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

recall

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \Rightarrow B_J = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \Rightarrow \rho(B_J) = \tfrac{1}{2}$$

$$B_{GS} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix} \Rightarrow \rho(B_{GS}) = \tfrac{1}{4} \qquad \underline{\text{ok}}$$

question : Can we design 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.

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

$\omega > 1$ is called successive over-relaxation (SOR) , $\omega = 1 \Rightarrow$ GS

component form

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

ex

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

matrix form

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

ex

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

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

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

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

thm (Young 1950)

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

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

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

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

pf : 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	$x_1^{(k)}$	$x_2^{(k)}$	$\|\|e_k\|\|_\infty$	$\|\|e_k\|\|_\infty / \|\|e_{k-1}\|\|_\infty$
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.

def : A is positive definite if $x^TAx > 0$ for all $x \neq 0$ (section 3.7)

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

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

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

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

ex 2 : $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ is positive definite , hw

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

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

for example : $x = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow x^TAx = 1$, $x = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \Rightarrow x^TAx = -2$ ok ex 4

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

$\Rightarrow (A_h w)_i = \frac{1}{h^2}(w_{i-1} - 2w_i + w_{i+1}) = -D_+D_-w_i$ (where $w_0 = w_{n+1} = 0$), so A_h represents the finite difference operator $-D_+D_-$. A_h is tridiagonal, symmetric, and positive definite, and hence Young's theorem applies.

note : The real advantage of iterative methods, in comparison with direct methods, occurs for BVPs in more than one dimension.

section 9.1 : two-dimensional BVP

problem : A metal plate has the shape of a square. The plate is heated by internal sources and the edges of the plate 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

$\phi(x, y)$: plate temperature

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

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

1. $-\Delta\phi = -\left(\dfrac{\partial^2\phi}{\partial x^2} + \dfrac{\partial^2\phi}{\partial y^2}\right) = f$ for (x, y) in D : Poisson equation

   ↑ Laplace operator

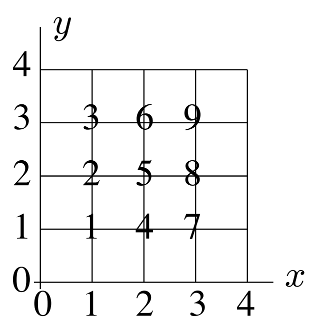
(note : This equation arises in many areas, e.g. if f is a charge/mass distribution, then ϕ is the electrostatic/gravitational potential.)

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

finite-difference scheme

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

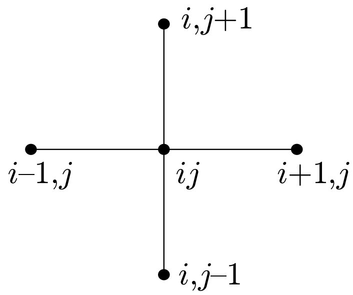
ex : $n = 3$, $h = \frac{1}{4}$



$\phi(x_i, y_j)$: exact solution , w_{ij} : approximation

ordering of mesh points : $(1,1), (1,2), \ldots$

$-\left(D_+^x D_-^x + D_+^y D_-^y\right) w_{ij} = f_{ij}$: 2nd order accurate



: 5-point stencil

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

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

Now consider what happens near the boundary.

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

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

Now 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

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

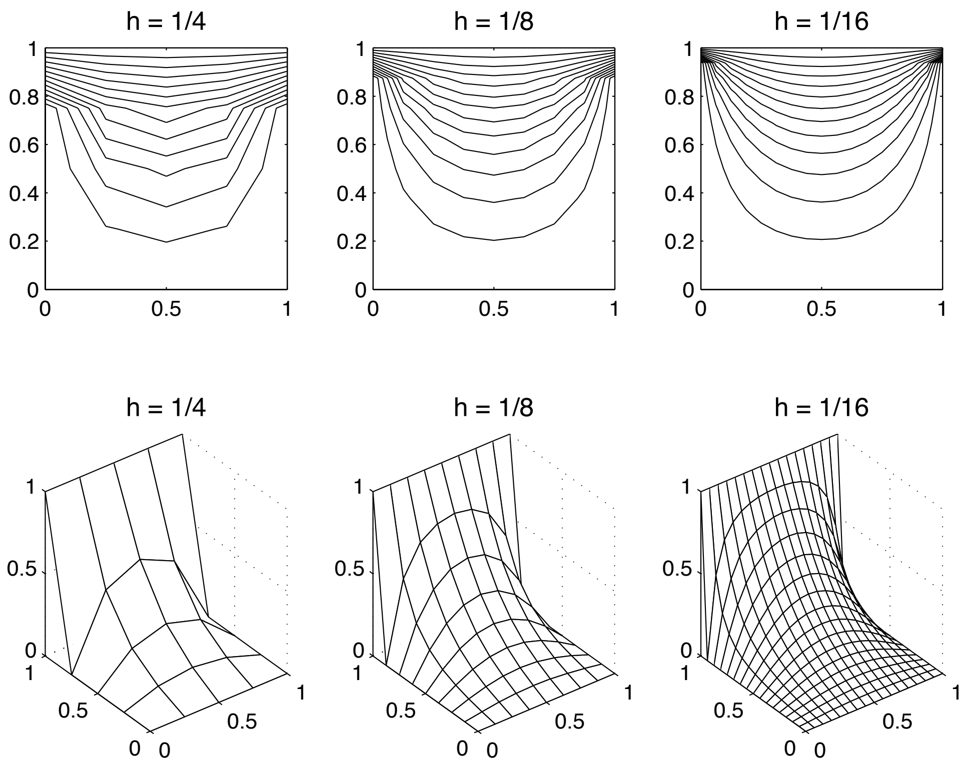
T : $n \times n$, symmetric , tridiagonal

A_h : $n^2 \times n^2$, block tridiagonal , symmetric , positive definite (pf : omit) temperature distribution on a metal plate

no heat sources : $\phi_{xx} + \phi_{yy} = 0$

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

finite-difference scheme : $4w_{ij} - w_{i+1,j} - w_{i-1,j} - w_{i,j+1} - w_{i,j-1} = 0$



The plots above show the solution of the linear system $A_h w_h = f_h$ for given mesh size h. The results below show the behaviour of the iterative methods; the initial guess was the zero vector and the stopping criterion was $||r_k||_\infty / ||r_0||_\infty \le 10^{-2}$.

Jacobi

h	k	$\lVert r_k\rVert_\infty / \lVert r_{k-1}\rVert_\infty$	$\rho(B)$
1/4	13	0.7071	0.7071
1/8	38	0.9238	0.9239
1/16	97	0.9804	0.9808

Gauss-Seidel

h	k	$\lVert r_k\rVert_\infty / \lVert r_{k-1}\rVert_\infty$	$\rho(B)$
1/4	7	0.4997	0.5000
1/8	19	0.8521	0.8536
1/16	47	0.9600	0.9619

optimal SOR

h	k	$\lVert r_k\rVert_\infty / \lVert r_{k-1}\rVert_\infty$	$\rho(B)$
1/4	5	0.2645	0.1716
1/8	8	0.5124	0.4465
1/16	11	0.6855	0.6735

<u>note</u>

1. For a given value of h, GS requires fewer iterations than J, and SOR requires fewer iterations than GS, but whichever method is used, more iterations are needed as the mesh size $h \to 0$. Hence decreasing h leads to smaller truncation error, but the computational cost increases.

2. The ratio of successive residuals converges to the spectral radius of the iteration matrix as $h \to 0$.

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

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

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

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

This confirms the observation above that the iteration converges more slowly as $h \to 0$, since $\rho(B) \to 1$ in this limit. SOR is least affected by this, followed by GS, and then J, i.e. $\rho(B_{\omega_*}) < \rho(B_{GS}) < \rho(B_J) < 1$.

4. Now consider what happens if Gaussian elimination is used to solve $A_h w_h = f_h$.

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

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. This is a disadvantage in comparison with iterative methods of the form $x_{k+1} = Bx_k + c$ (e.g. J, GS, SOR) which preserve the <u>sparsity</u> of A.

final comments on linear systems

1. comparison of operation counts

recall : For a two-dimensional BVP, the matrix A_h has dimension $n^2 \times n^2$, where $h = \frac{1}{n+1}$ is the mesh size, the bandwidth of A_h is $m = n$, and the typical equation is $\frac{1}{h^2}(4w_{ij} - w_{i+1,j} - w_{i-1,j} - w_{i,j+1} - w_{i,j-1}) = f_{ij}$.

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

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

b) iterative methods

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

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

$\Rightarrow k \sim \dfrac{\log \epsilon}{-ch^2} = O(n^2)$ iterations , cost per iteration $= O(n^2)$ ops

$\Rightarrow$ total cost $= O(n^4)$ ops

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

$\Rightarrow k \sim \dfrac{\log \epsilon}{-ch} = O(n)$ iterations , cost per iteration $= O(n^2)$ ops

$\Rightarrow$ total cost $= O(n^3)$ ops

2. developments after SOR

multigrid : large h (fast, low accuracy) + small h (slow, high accuracy)

conjugate gradient method , GMRES : energy minimization

preconditioning : $Ax = b \rightarrow MAx = Mb$

software

parallel algorithms