

section 3.0 : review of linear algebra

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

We can write the system in 3 other forms.

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

$$2. \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

$$3. Ax = b, \quad A = (a_{ij})$$

basic problem : Given A, b , find x .

solution : $x = b/A$: no, but $x = b \backslash A$ does work in Matlab

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^{-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 linear systems, direct methods and iterative methods. We will begin with direct methods.

section 3.1 : Gaussian elimination for solving $Ax = b$

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

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

Then $x_n = b_n/a_{nn}$

$$x_{n-1} = (b_{n-1} - a_{n-1,n}x_n)/a_{n-1,n-1}$$

\vdots

$$x_1 = (b_1 - (a_{12}x_2 + \cdots + a_{1n}x_n))/a_{11}$$

back substitution

1. $x_n = b_n/a_{nn}$
2. for $i = n - 1 : -1 : 1$
3. $sum = b_i$
4. for $j = i + 1 : n$
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

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

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

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

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

note : Similar considerations hold if A is lower triangular.

note

In the general case we use elementary row operations to reduce $Ax = b$ to an equivalent upper triangular system and then apply back substitution to find x .

1. multiply an equation by a nonzero number and subtract from another equation
2. interchange two equations

ex : $n = 3$

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$

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

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

$$m_{21} = \frac{a_{21}}{a_{11}} \Rightarrow \begin{array}{l} a_{22} \rightarrow a_{22} - m_{21}a_{12} \\ a_{23} \rightarrow a_{23} - m_{21}a_{13} \\ b_2 \rightarrow b_2 - m_{21}b_1 \end{array} \quad \% m_{21} \text{ is called a } \underline{\text{multiplier}}$$

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

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

step 2 : eliminate variable x_2 from eq. 3

$$m_{32} = \frac{a_{32}}{a_{22}} \Rightarrow \begin{array}{l} a_{33} \rightarrow a_{33} - m_{32}a_{23} \\ b_3 \rightarrow b_3 - m_{32}b_2 \end{array}$$

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

ex

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

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

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

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

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

general $n \times n$ case

reduction to upper triangular form

1. for $k = 1 : n - 1$
2. for $i = k + 1 : n$
3. $m_{ik} = a_{ik} / a_{kk}$
4. for $j = k + 1 : n$
5. $a_{ij} = a_{ij} - m_{ik} \cdot a_{kj}$
6. $b_i = b_i - m_{ik} \cdot b_k$

note

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

operation count

The leading order term comes from line 5.

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

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

note

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

pf

1. already done

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

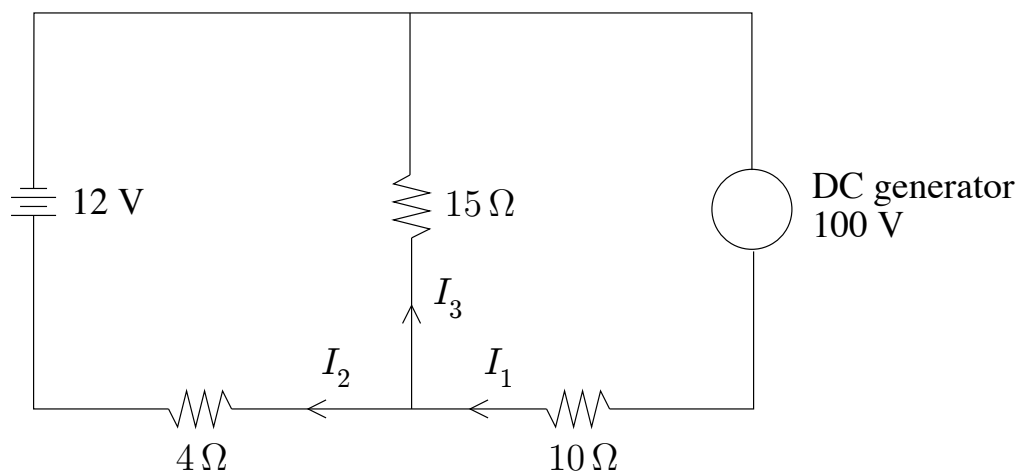
$$k^3 - (k-1)^3 = k^3 - (k^3 - 3k^2 + 3k - 1) = 3k^2 - 3k + 1$$

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

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

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

ex : 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 changes around any closed loop is zero.

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

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

We saw that Gaussian elimination breaks down if a pivot element is zero. There are various pivoting strategies that overcome this problem.

partial pivoting

Consider the reduced matrix at the beginning of step k in Gaussian elimination.

$$\begin{pmatrix} a_{11} & \cdots & \cdots & a_{1k} & \cdots & a_{1n} & b_1 \\ & \ddots & & \vdots & & \vdots & \vdots \\ & & \ddots & \vdots & & \vdots & \vdots \\ & & & a_{kk} & \cdots & a_{kn} & b_k \\ & & & \vdots & & \vdots & \vdots \\ & & & \vdots & & \vdots & \vdots \\ & & & a_{nk} & \cdots & a_{nn} & b_n \end{pmatrix}$$

If $a_{kk} = 0$, find index l such that $|a_{lk}| = \max\{|a_{ik}|; k \leq i \leq n\}$. 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 in detail.
4. In practice, pivoting is often applied even if the pivot element is nonzero.

ex

$$\left. \begin{aligned} \left(\begin{array}{ccc|c} \epsilon & 1 & 1 & 1 + \epsilon \\ 1 & 1 & 2 & 2 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} \epsilon & 1 & 1 & 1 + \epsilon \\ 0 & 1 - \frac{1}{\epsilon} & 1 - \frac{1}{\epsilon} & 1 - \frac{1}{\epsilon} \end{array} \right) \Rightarrow \left. \begin{aligned} x_2 &= \frac{1 - \frac{1}{\epsilon}}{1 - \frac{1}{\epsilon}} = 1 \\ x_1 &= \frac{1 + \epsilon - 1}{\epsilon} = 1 \end{aligned} \right\} : \text{exact solution} \\ m_{21} &= \frac{1}{\epsilon} \end{aligned}$$

Now consider the effect of roundoff error.

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

Now apply pivoting in the presence of roundoff error.

$$\left. \begin{aligned} \left(\begin{array}{ccc|c} 1 & 1 & 2 \\ \epsilon & 1 & 1 \end{array} \right) \rightarrow \left(\begin{array}{ccc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array} \right) \Rightarrow \left. \begin{aligned} \tilde{x}_2 &= 1 \\ \tilde{x}_1 &= 1 \end{aligned} \right\} : \text{new computed solution , accurate} \\ m_{21} &= \frac{\epsilon}{1} = \epsilon \end{aligned}$$

We'll explain this soon, but first we need a way to measure the size of a vector.

section 3.3 : vector and matrix norms

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

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

ex

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

$$\|x\|_\infty = \max\{|x_i| : i = 1, \dots, n\} \quad , \quad \text{pf} \dots$$

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

note : Given a vector norm $\|x\|$, we can define a matrix norm by $\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$.

The following properties are satisfied.

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 : ...

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

$$\text{ex} : A = \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} \Rightarrow \|A\|_\infty = 3$$

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

$$x = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow Ax = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \Rightarrow \frac{\|Ax\|_\infty}{\|x\|_\infty} = \frac{2}{1} = 2$$

$$x = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \Rightarrow Ax = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \Rightarrow \frac{\|Ax\|_\infty}{\|x\|_\infty} = \frac{3}{1} = 3 : \text{largest possible ratio}$$

thm : $\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \max\{\sqrt{\lambda} : \lambda \text{ is an eigenvalue of } A^T A\}$

$$\text{ex} : A^T A = \begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 2 & 8 \end{pmatrix} \Rightarrow \|A\|_2 = 2.9208 \quad (\text{Matlab})$$

section 3.4 : error analysis

$Ax = b$, A : invertible

x : exact solution , $x = A^{-1}b$

\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 $r = 0$. However, if $\|r\|$ is small, there's no guarantee that $\|e\|$ is also small.

ex

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

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

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

Hence we see that $\|r\|$ 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 : $\frac{\|e\|}{\|x\|} \leq \kappa(A) \frac{\|r\|}{\|b\|}$, where $\kappa(A) = \|A\| \cdot \|A^{-1}\|$: condition number

ex

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

$$\begin{aligned} A^{-1} &= \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \frac{1}{0.04} \begin{pmatrix} 1.01 & -0.99 \\ -0.99 & 1.01 \end{pmatrix} \\ &= \begin{pmatrix} 25.25 & -24.75 \\ -24.75 & 25.25 \end{pmatrix} \Rightarrow \|A^{-1}\|_\infty = 50 \Rightarrow \kappa_\infty(A) = 100 \quad \text{ok} \end{aligned}$$

pf

$$1. \|b\| = \|Ax\| \leq \|A\| \cdot \|x\| \Rightarrow \frac{1}{\|x\|} \leq \frac{\|A\|}{\|b\|}$$

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

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

note

1. $\left. \begin{array}{l} Ax = b \\ A\tilde{x} = \tilde{b} \end{array} \right\} \Rightarrow \frac{\|x - \tilde{x}\|}{\|x\|} \leq \kappa(A) \frac{\|b - \tilde{b}\|}{\|b\|} : \text{perturbation of RHS}$
2. $\left. \begin{array}{l} Ax = b \\ \tilde{A}\tilde{x} = b \end{array} \right\} \Rightarrow \frac{\|x - \tilde{x}\|}{\|\tilde{x}\|} \leq \kappa(A) \frac{\|A - \tilde{A}\|}{\|A\|} : \text{perturbation of matrix}$

pf

1. follows from previous theorem , 2. hw

Hence the condition number controls the size of perturbations in the solution due to perturbations of the RHS and the matrix.

ex (recall)

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

Now consider the effect of roundoff error.

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

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.

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

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

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 $Ax = b$. Pivoting however produces a different reduced system.

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

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

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

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

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

step 2 : eliminate variable x_2 from eq. 3

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

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

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

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

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

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

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 $Ax = b$.

1. factor $A = LU$
2. solve $Ly = b$ by forward substitution
3. solve $Ux = y$ by back substitution

check : $Ax = LUx = Ly = b \quad \underline{\text{ok}}$

ex

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

Using Gaussian elimination before we found $x_1 = x_2 = x_3 = 1$, 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.

LU factorization and pivoting

To perform partial pivoting we need to interchange rows, and this is done here using an elementary permutation matrix. The final result is that instead of $A = LU$, we obtain $PA = LU$, where P is a permutation matrix.

ex

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

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 : $PA = LU \dots$

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

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

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

note

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 $P_2 E_1 = \tilde{E}_1 P_2$ (see review sheet)

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

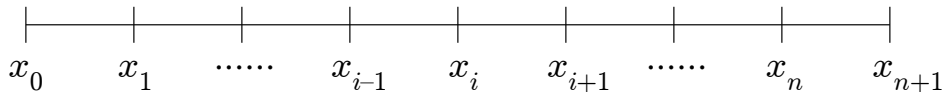
section 8.1 : 2-point boundary value problem

Find $y(x)$ on $0 \leq x \leq 1$ 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 (for example) a steady state convection-reaction-diffusion system, where $y(x)$ represents a velocity or temperature profile.

finite-difference scheme

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

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



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

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

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

question : How accurate is the approximation?

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

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2}y''_i + \frac{h^3}{3!}y'''_i + \frac{h^4}{4!}y^{(4)}_i + \frac{h^5}{5!}y^{(5)}_i + O(h^6)$$

$$y_{i-1} = y_i - hy'_i + \frac{h^2}{2}y''_i - \frac{h^3}{3!}y'''_i + \frac{h^4}{4!}y^{(4)}_i - \frac{h^5}{5!}y^{(5)}_i + O(h^6)$$

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

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

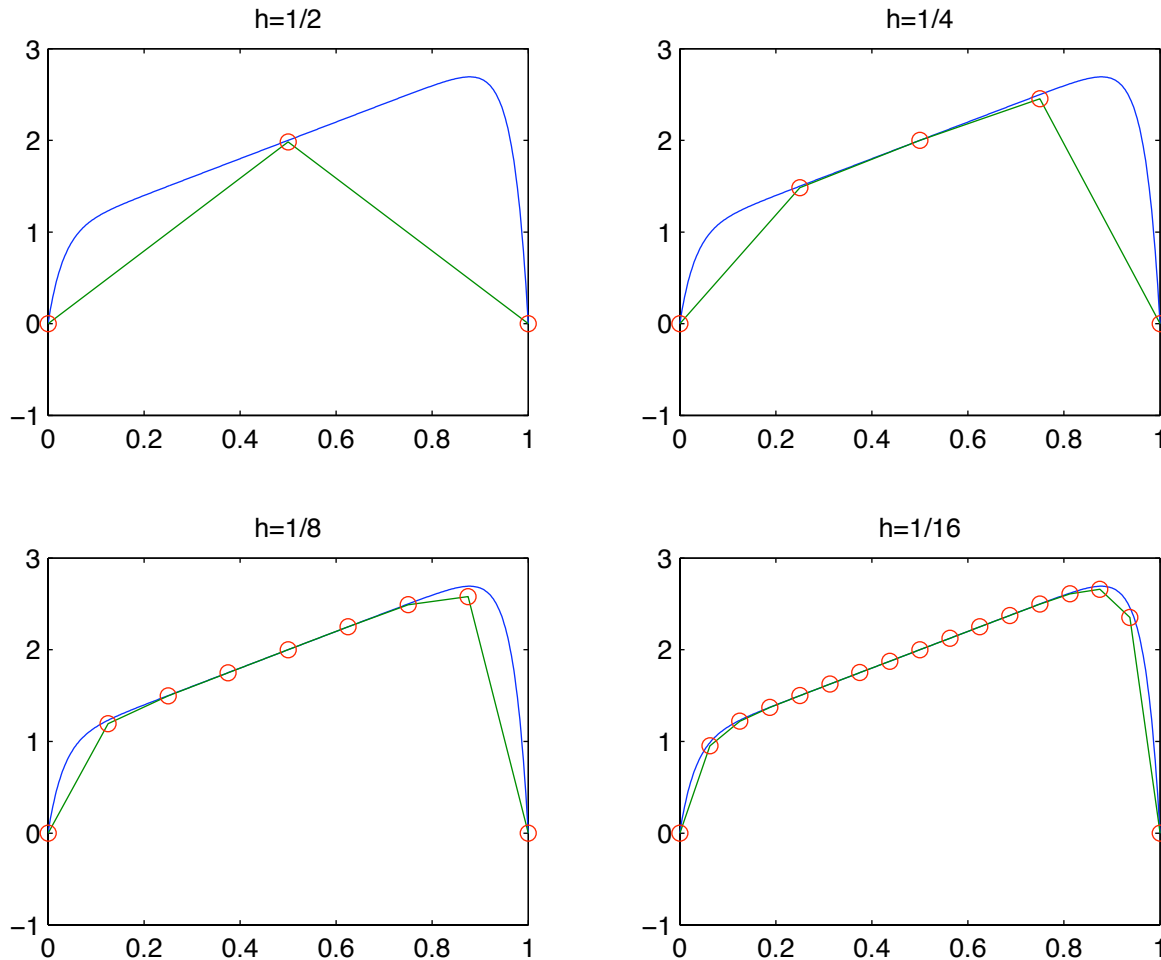
$$-\left(\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2}\right) + q_iw_i = r_i , i = 1, \dots, n$$

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

$$i = 1 \Rightarrow \frac{1}{h^2}(-w_2 + (2 + q_1h^2)w_1 - \alpha) = r_1$$

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

2-point bvp : $-\epsilon y'' + y = 2x + 1$, $0 < x < 1$, $y(0) = 0$, $y(1) = 0$, $\epsilon = 10^{-3}$
 solution : $y(x) = 2x + 1 - (\sinh \frac{1-x}{\sqrt{\epsilon}} + 3 \sinh \frac{x}{\sqrt{\epsilon}}) / \sinh \frac{1}{\sqrt{\epsilon}}$, check ...



h	$\ y_h - w_h\ _\infty$	$\frac{\ y_h - w_h\ _\infty}{h}$	$\frac{\ y_h - w_h\ _\infty}{h^2}$	$\frac{\ y_h - w_h\ _\infty}{h^3}$
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 $0.125 \leq h \leq 0.5$, the error increases as h decreases. This is due to the presence of boundary layers at $x = 0, 1$.
2. For $h \leq 0.03125$, decreasing h by 1/2 causes the error to decrease by 1/4.
3. We see that $\|y_h - w_h\|_\infty = O(h^2)$, so the method is 2nd order accurate.

review of two important properties of norms

first recall : $\|x\|_\infty = \max\{|x_1|, \dots, |x_n|\}$

$$\|A\|_\infty = \max_{x \neq 0} \frac{\|Ax\|_\infty}{\|x\|_\infty} = \max_i \sum_j |a_{ij}|$$

property 1 : $\|Ax\|_\infty \leq \|A\|_\infty \cdot \|x\|_\infty$

property 2 : $\|AB\|_\infty \leq \|A\|_\infty \cdot \|B\|_\infty$

pf

1. the proof is a bit abstract, so instead we'll give an example (which shows the idea of the proof)

$$x = \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \quad A = \begin{pmatrix} 3 & -4 \\ 1 & 0 \end{pmatrix}$$

$$Ax = \begin{pmatrix} 3 & -4 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \end{pmatrix} = \begin{pmatrix} (3)(1) + (-4)(-2) \\ (1)(1) + (0)(-2) \end{pmatrix} = \begin{pmatrix} 11 \\ 1 \end{pmatrix}$$

$$\|x\|_\infty = 2, \quad \|Ax\|_\infty = 11, \quad \|A\|_\infty = \max\{|3| + |-4|, |1| + |0|\} = 7$$

$$\|Ax\|_\infty = 11 \leq 14 = 7 \cdot 2 = \|A\|_\infty \cdot \|x\|_\infty \quad \underline{\text{ok}}$$

physical meaning of property 1

If x is the input to a system and Ax is the output, then the norm of the output is bounded by $\|A\|_\infty$ times the norm of the input.

2. here we give the actual proof

$$\begin{aligned} \|AB\|_\infty &= \max_{x \neq 0} \frac{\|ABx\|_\infty}{\|x\|_\infty} \quad : \text{ by definition} \\ &\leq \max_{x \neq 0} \frac{\|A\|_\infty \cdot \|Bx\|_\infty}{\|x\|_\infty} \quad : \text{ by property 1} \\ &\leq \max_{x \neq 0} \frac{\|A\|_\infty \cdot \|B\|_\infty \cdot \|x\|_\infty}{\|x\|_\infty} \quad : \text{ by property 1 again} \\ &= \|A\|_\infty \cdot \|B\|_\infty \quad \underline{\text{ok}} \end{aligned}$$

note

it follows that $\|A^2\|_\infty \leq \|A\|_\infty^2$, $\|A^k\|_\infty \leq \|A\|_\infty^k$ for any $k \geq 1$

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). Here we consider an alternative class of methods called iterative methods which generate a sequence of approximate solutions x_k such that $\lim_{k \rightarrow \infty} x_k = x$. As we shall see, iterative methods have some advantages over direct methods.

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

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

B : iteration matrix

Jacobi method

$A = L + D + U$: splitting

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

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

$$Ax = b \Leftrightarrow (L + D + U)x = b$$

$$\Leftrightarrow Dx = -(L + U)x + b$$

$$\Leftrightarrow x = -D^{-1}(L + U)x + D^{-1}b \quad , \quad B_J = -D^{-1}(L + U)$$

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

component form

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

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

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

ex

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

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

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 $k \rightarrow \infty$.

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

$$\|e_0\|_\infty = 1, \|e_1\|_\infty = \frac{1}{2}, \|e_2\|_\infty = \frac{3}{4}, \|e_3\|_\infty = \frac{7}{8} \Rightarrow \|e_{k+1}\|_\infty = \frac{1}{2}\|e_k\|_\infty$$

thm

Consider a fixed-point iteration of the form $x_{k+1} = Bx_k + c$. If $\|B\| < 1$ for any matrix norm, then $x_k \rightarrow x$ for any initial guess x_0 .

pf

$$e_k = x - x_k = (Bx + c) - (Bx_{k-1} + c) = B(x - x_{k-1}) = Be_{k-1}$$

$$e_k = Be_{k-1} = B^2e_{k-2} = \dots = B^ke_0 \Rightarrow \|e_k\| = \|B^ke_0\| \leq \|B^k\| \cdot \|e_0\|$$

$$\|B^2\| = \|B \cdot B\| \leq \|B\| \cdot \|B\| = \|B\|^2 \Rightarrow \|B^k\| \leq \|B\|^k$$

$$\Rightarrow \|e_k\| \leq \|B\|^k \cdot \|e_0\| \rightarrow 0 \text{ as } k \rightarrow \infty \quad \text{ok}$$

$$\text{important : } e_{k+1} = Be_k \Rightarrow \|e_{k+1}\| \leq \|B\| \cdot \|e_k\|$$

ex

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

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

1. Since $\|B_J\|_\infty < 1$, the theorem implies that Jacobi's method converges.

2. The error norm decreases by a factor of $\frac{1}{2} = \|B_J\|_\infty$ in each step.

Gauss-Seidel method

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

$$Ax = b \Leftrightarrow (L + D + U)x = b$$

$$\Leftrightarrow (L + D)x = -Ux + b$$

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

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

component form

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

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

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

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

ex

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

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

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.

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

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

$$\|B_{GS}\|_\infty = \frac{1}{2}$$

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

2. The error norm decreases by a factor of $\frac{1}{4}$ in each step, which is less than $\frac{1}{2} = \|B_{GS}\|_\infty$.

summary

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

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

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

We see that $\|e_{k+1}\|_\infty \leq \|B\|_\infty \cdot \|e_k\|_\infty$ in both cases (as required by the previous theorem), but the bound is not sharp in the case of GS. To explain the actual behavior of the error, we need to examine 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

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

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

$$\lambda = -1 \text{” } x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

note

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

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

The e-values of A are the roots of the characteristic polynomial.

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

thm

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

pf

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

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

note

Now we can explain why $\|e_{k+1}\|_\infty = \frac{1}{4}\|e_k\|_\infty$ when Gauss-Seidel is applied to

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}. \text{ In this case we have } B_{GS} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}.$$

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

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

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

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

$$e_2 = Be_1 = B(\lambda_2 v_2) = \lambda_2 Bv_2 = \lambda_2^2 v_2$$

⋮

$$e_k = \lambda_2^k v_2 = \left(\frac{1}{4}\right)^k v_2 \Rightarrow \|e_k\|_\infty = \left(\frac{1}{4}\right)^k \|v_2\|_\infty \quad \underline{\text{ok}}$$

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. $\|e_{k+1}\|_\infty \leq \|B\|_\infty \cdot \|e_k\|_\infty$ for all $k \geq 0$: error bound

2. $\|e_{k+1}\|_\infty \sim \rho(B) \cdot \|e_k\|_\infty$ as $k \rightarrow \infty$: asymptotic relation

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

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

pf

1. ok

2. Math 571 (but the idea is the same as in the example above)

recall

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

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

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.

$$Dx_{k+1} = Dx_k - \omega(Lx_{k+1} + (D + U)x_k - b) : \text{relaxation } (\omega = 1 \Rightarrow \text{GS})$$

component form

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

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

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

note

1. $\omega > 1$ is called successive over-relaxation (SOR)

$$2. (\omega L + D)x_{k+1} = ((1-\omega)D - \omega U)x_k + \omega b \Rightarrow B_\omega = (\omega L + D)^{-1}((1-\omega)D - \omega U)$$

ex

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

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

matrix form

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

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

$$\text{note : } \omega = 1 \Rightarrow B_\omega = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix} : \text{Gauss-Seidel , } \rho(B_\omega) = 0.25 \quad \text{ok}$$

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 that A is block tridiagonal, symmetric, and positive definite, and let

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

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

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

k	$x_1^{(k)}$	$x_2^{(k)}$	$\ e_k\ _\infty$	$\ e_k\ _\infty / \ e_{k-1}\ _\infty$
0	0.0000	0.0000	1.0000	...
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
4	0.9994	0.9998	0.0006	0.0944
\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
∞	1	1	0	$\rho(B_{\omega_*}) = \omega_* - 1 = 0.0718$

Hence we see that optimal SOR converges faster than GS in this example.

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

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

$$\begin{aligned} \text{pf} : x^T A x &= (x_1, x_2) \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = (x_1, x_2) \begin{pmatrix} 2x_1 - x_2 \\ -x_1 + 2x_2 \end{pmatrix} \\ &= 2(x_1^2 + x_2^2) - 2x_1x_2 = x_1^2 + x_2^2 + (x_1 - x_2)^2 \geq 0 \end{aligned}$$

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

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

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

note

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

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}$$

2. The real advantage of iterative methods, in comparison with direct methods, is seen for BVPs in more than one dimension.

section 9.1 : two-dimensional BVP

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.

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

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

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

$$-\underset{\uparrow}{\Delta}\phi = -\left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}\right) = f \text{ for } (x, y) \text{ in } D : \text{ Poisson equation}$$

Laplace operator

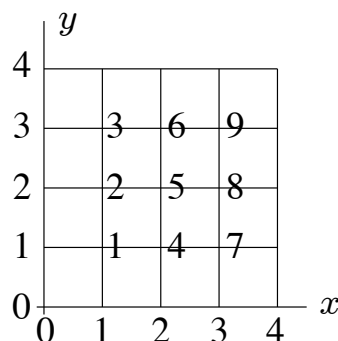
(note : The Poisson equation arises in many areas, e.g. if f is a charge distribution, then ϕ is the electrostatic potential.)

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

finite-difference scheme

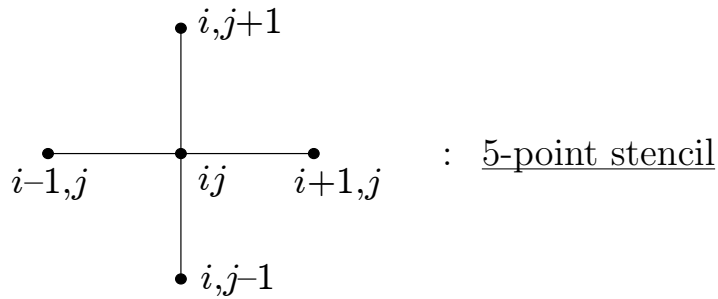
$h = \frac{1}{n+1}$: mesh size , $(x_i, y_j) = (ih, jh)$, $i, j = 0, \dots, n+1$: mesh points

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



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

$-(D_+^x D_-^x + D_+^y D_-^y) w_{ij} = f_{ij}$: the truncation error is $O(h^2)$



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

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

Now consider what happens near the boundary.

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

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

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

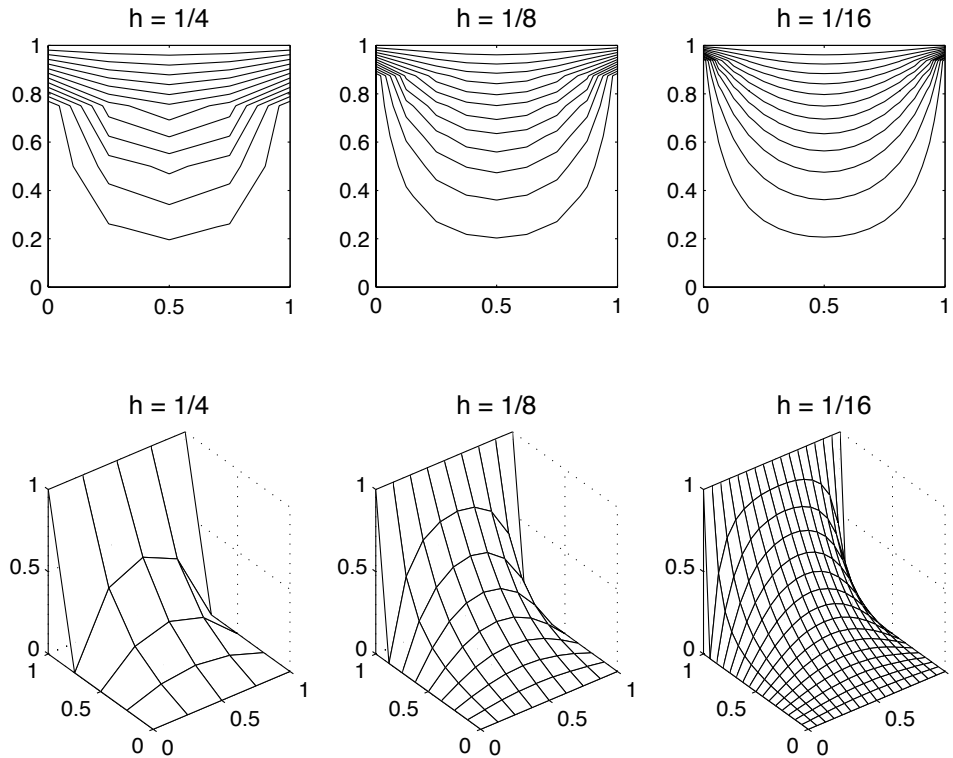
A_h : $n^2 \times n^2$, block tridiagonal , symmetric , positive definite (pf : omit)

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 $\|r_k\|_\infty / \|r_{k-1}\|_\infty \leq 10^{-2}$.

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

note

1. For a given value of h , GS requires fewer iterations than J, and SOR requires fewer iterations than GS. Whichever method is used, more iterations are required as the mesh size $h \rightarrow 0$. Hence there is a tradeoff; decreasing h leads to smaller truncation error, but the computational cost is increased.
2. The ratio of successive residuals converges to the spectral radius of the iteration matrix as $h \rightarrow 0$.
3. Explicit formulas can be derived for $\rho(B)$ in this example.

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

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

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

This reconfirms the observation above that the iteration converges more slowly as $h \rightarrow 0$, since $\rho(B) \rightarrow 1$ 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 $A_h w_h = f_h$.

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

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

b) As the elimination proceeds, zeros inside the band can become non-zero (this is called fill-in), 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 sparsity 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 : $O(n^2m^2) \Rightarrow O(n^4)$ ops

b) iterative methods

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

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

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

cost per iteration : $O(n^2)$ ops \Rightarrow total cost = $O(n^4)$ ops

SOR $\Rightarrow \rho(B) \sim 1 - ch \Rightarrow k \sim O(n)$ iterations

cost per iteration : $O(n^2)$ ops \Rightarrow total cost = $O(n^3)$ ops

2. developments after SOR

multigrid

conjugate gradient method , GMRES

preconditioning

software

parallel algorithms