New topic :

Direct and Iterative Methods for Solving Linear Systems

Goal : Solve $Ax = b$

where $A \in \mathbb{R}^{n\times n}$, $\operatorname{rank}(A) = n$

Easiest case : A is upper/lower triangular matrix

$$A \quad x = b$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{12} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ & & & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- Backward substitution:
$$x_n = b_n / a_{nn}$$
$$x_{n-1} = (b_{n-1} - a_{n-1,n} x_n) / a_{n-1,n-1}$$
$$\vdots$$
$$x_i = \left(b_i - \sum_{j=i+1}^{n} a_{ij} x_j\right) / a_{ii}$$

- Operation count :
$$\sum_{k=1}^{n} [2(n-k)+1] = n^2 \text{ FLOPs}$$

- Backward stability :

Backward substitution is very stable, analyzing its backward error is very similar to that of inner product.

Lemma. If $y = (c - \sum_{i=1}^{k-1} a_i b_i)/b_k$ is evaluated in floating point arithmetic, no matter what the order of evaluation, the computed $\hat{y}$ satisfies

$$b_k \hat{y}(1+\theta_k^{(0)}) = c - \sum_{i=1}^{k-1} a_i b_i (1+\theta_k^{(i)})$$

where $|\theta_k^{(i)}| \le \gamma_k := \dfrac{k\varepsilon_{mach}}{1-k\varepsilon_{mach}}$

Proof: exercise (Hint: Use binary tree) $\blacksquare$

Applying this lemma to triangular system, we have

Thm (Backward substitution is backward stable)

Let the triangular system $Ax=b$, where $A \in \mathbb{R}^{n\times n}$ is nonsingular, be solved by substitution, with any ordering. Then the computed solution $\hat{x}$ satisfies

$$(A+\Delta A)\hat{x} = b, \qquad |\Delta A| \le \gamma_n |A|$$

$|\Delta a_{ij}| \le \gamma_n |a_{ij}|, \forall i,j$
(implies $\|\Delta A\| \le C_n \|A\|$)

Triangular system is easy to solve and stable.

For general A, use different strategies to factor it into triangular matrices

Method 1: QR factorization

$$A = QR, \quad Q^T Q = I, \quad R \text{ upper triangular}$$

$$Ax = b \iff QRx = b \iff Rx = Q^T b \iff x = R^{-1} Q^T b$$

- Implementation:

Operation count:

Step 1: Householder QR $A = QR \longrightarrow \approx 2n^3 - \frac{2}{3}n^3$
Step 2: Compute $d = Q^T b \longrightarrow \approx O(n^2)$
Step 3: Solve $Rx = d \longrightarrow = n^2$

$\approx \frac{4}{3}n^3$ FLOPs

• Stability : Since 1) Householder QR is backward stable,

2) computed $\hat{Q}$ is very close to orthogonal matrix

3) solving a triangular system is stable.

solving linear system with Householder QR is also backward stable.

<u>thm</u> Let $A \in \mathbb{R}^{n \times n}$ be nonsingular. Suppose we solve the system $Ax = b$ with the aid of QR factorization computed by the Householder algorithm. The computed $\hat{x}$ satisfies

$$(A + \Delta A)\hat{x} = b + \Delta b$$

where $\dfrac{\|\Delta A\|_2}{\|A\|_2} \le C_n \varepsilon_{mach}$, $\dfrac{\|\Delta b\|_2}{\|b\|_2} \le C_n' \varepsilon_{mach}$

pf : From the backward stability of back substitution,

$$\text{(*)} \quad (\hat{R} + \Delta\hat{R})\hat{x} = \hat{Q}^T b , \quad \text{with } \|\Delta\hat{R}\|_2 \le C_n \varepsilon_{mach} \|\hat{R}\|_2 \cdots (1)$$

From the stability of Householder QR, $\exists\, Q$, $Q^T Q = I$ and

$$\|Q - \hat{Q}\|_2 \le C_n' \varepsilon_{mach} \cdots (2), \quad \|Q\hat{R} - A\|_2 \le C_n'' \varepsilon_{mach} \|A\| \cdots (3)$$

Hence, apply Q to (*). we get

$$\Big(A + \underbrace{(Q\hat{R} - A) + Q\Delta\hat{R}}_{=:\Delta A}\Big)\hat{x} = b + \underbrace{Q(Q^T - \hat{Q}^T)b}_{=:\Delta b}$$

$$\begin{aligned}
\|\Delta A\|_2 &\overset{(3)}{\le} C_n'' \varepsilon_{mach} \|A\|_2 + \|\Delta\hat{R}\|_2 \\
&\overset{(1)}{\le} C_n'' \varepsilon_{mach} \|A\|_2 + C_n \varepsilon_{mach} \big(\|Q\hat{R} - A\|_2 + \|A\|_2\big) \\
&\overset{(3)}{\le} (2C_n'' + C_n)\, \varepsilon_{mach} \|A\|_2
\end{aligned}$$

$$\|\Delta b\|_2 \le \|Q^T - \hat{Q}^T\|_2 \, \|b\|_2 \overset{(2)}{\le} C_n' \, \varepsilon_{mach} \, \|b\|_2$$

See Higham Thm 19.5 for a refined bound $\square$

Method 2: LU factorization

$$A = L \, U$$

(unit) lower triangular matrix — L

upper triangular matrix — U

- Idea: Gaussian elimination

$$[A \,\vdots\, b] \xrightarrow{\sum_{i=1}^{n-1} (n-i) + 2(n-i)^2 \,\approx\, \frac{2}{3} n^3} [U \,\vdots\, d] \xrightarrow{O(n^2)} [I \,\vdots\, x]$$

*: Reduce Linear system to a triangular system via elementary row operation

*: At the k^{th} stage, zero the element below the diagonal in the k^{th} column $\quad (A^{(k)} x = b^{(k)})$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \qquad i,j = k+1, \dots, n$$

$$b_i^{(k+1)} = b_i^{(k)} - m_{ik} b_k^{(k)}, \qquad i = k+1, \dots, n$$

where $m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$

*: operation count $\approx 2n^3/3$ FLOPs for each b

- GE in matrix form

ex. $A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & 3 & 3 \\ 8 & 7 & 9 \end{bmatrix}$

$$\overset{E_1}{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix}} \overset{A}{\begin{bmatrix} 2 & 1 & 1 \\ 4 & 3 & 3 \\ 8 & 7 & 9 \end{bmatrix}} = \overset{E_1A}{\begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 3 & 5 \end{bmatrix}}$$

$$\overset{E_2}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix}} \overset{E_1A}{\begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 3 & 5 \end{bmatrix}} = \overset{E_2E_1A}{\begin{bmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}} = U \leftarrow \text{upper triangular matrix}$$

$$E_2 E_1 A = U$$

$$\Rightarrow \quad A = \underbrace{E_1^{-1} E_2^{-1}}_{=:L} U$$

$=: L \leftarrow$ unit lower triangular matrix

$$L = \overset{E_1^{-1}}{\begin{bmatrix} 1 & & \\ 2 & 1 & \\ 4 & & 1 \end{bmatrix}} \overset{E_2^{-1}}{\begin{bmatrix} 1 & & \\ & 1 & \\ & 3 & 1 \end{bmatrix}} = \begin{bmatrix} 1 & & \\ 2 & 1 & \\ 4 & 3 & 1 \end{bmatrix}$$

- Computing an LU factorization $A = LU$ is equivalent to solving the equation

$$a_{ij} = \sum_{r=1}^{\min(i,j)} l_{ir} u_{rj} \qquad i,j = 1, \cdots, n$$

These nonlinear equation are easily solved if examined in the right order.

Suppose now row $1, \cdots, k-1$ of U are computed

column $1, \cdots, k-1$ of L are computed

Setting $l_{kk} = 1$,

$$a_{kj} = l_{k1} u_{1j} + l_{k2} u_{2j} + \cdots + l_{k,k-1} u_{k-1,j} + \boxed{u_{kj}}, \qquad j = k, \dots, n$$

$$a_{ik} = l_{i1} u_{1k} + l_{i2} u_{2k} + \cdots + l_{i,k-1} u_{k-1,k} + \boxed{l_{ik}} u_{kk}, \quad i = k+1, \dots, n$$

- Implementation (Doolittle's method)

same flops as GE

for $k = 1, \dots, n$ $\qquad$ operation count $= \sum_{k=1}^{n} 2(k-1)(n-k+1) + [2(k-1)+1](n-k)$

$\quad$ for $j = k, \dots, n$ $\qquad \approx \frac{2}{3}n^3$

(△) $\qquad u_{kj} = a_{kj} - \sum_{r=1}^{k-1} l_{kr} u_{rj} \longrightarrow 2(k-1)(n-k+1)$ FLOPs

$\quad$ end

$\quad$ for $i = k+1, \dots, n$

(△△) $\qquad l_{ik} = \left(a_{ik} - \sum_{r=1}^{k-1} l_{ir} u_{rk} \right) / u_{kk} \longrightarrow [2(k-1)+1](n-k)$ FLOPs

$\quad$ end

end

- Solve $Ax = b$ via LU factorization

Step 1 : Compute LU factorization $A = LU$

Step 2 : Solve $Ly = b$ and then $Ux = y$

Remark :

$[b_1, \dots, b_m]$. $b_i \in \mathbb{R}^n$

1) We can solve $Ax = b$ or $AX = B$ by applying GE to augmented matrix $[A \,\vdots\, b]$ or $[A \,\vdots\, B]$. This is faster and save memory.

2) If $b_1, \dots, b_m$ are not available at the same time, applying LU first then solve $Ly_i = b_i$, $Ux_i = y_i$ is faster.

($O(n^2)$ for each new b_i)

3) Doolittle's method is mathematically equivalent to GE

if we identify

$$a_{kj} - l_{k1}u_{1j} - l_{k2}u_{2j} - \cdots - l_{k,s}u_{s,j} = a_{kj}^{(s+1)}, \quad j=k+1,\dots,n$$

- Error Analysis

Note that GE & Doolittle simply correspond to different order of evaluating (△)(△△), which are of the form

$$y = \left(c - \sum_{i=1}^{k-1} a_i b_i\right) / b_k.$$

By the lemma at the beginning of the notes, the rounding error in evaluating such formula is independent of the ordering.

Hence, the worst case errors of GE is the same as Doolittle.

we focus on Doolittle in the following.

By the lemma at the begining of the notes,

the computed $\hat{L}$, $\hat{U}$ satisfy

$$\left| a_{kj} - \sum_{i=1}^{k-1} \hat{l}_{ki}\hat{u}_{ij} - \hat{u}_{kj} \right| \le \gamma_k \sum_{i=1}^{k} |\hat{l}_{ki}|\,|\hat{u}_{ij}|, \quad j \ge k$$

$$\left| a_{ik} - \sum_{j=1}^{k} \hat{l}_{ij}\hat{u}_{jk} \right| \le \gamma_k \sum_{j=1}^{k} |\hat{l}_{ij}|\,|\hat{u}_{jk}|, \quad i > k$$

<u>Thm</u> If Doolittle's algorithm runs to completion, then the computed $\hat{L}$, $\hat{U}$ satisfy

$$\hat{L}\hat{U} = A + \Delta A, \qquad |\Delta A| \le \gamma_n |\hat{L}|\,|\hat{U}|$$

Thm (Backward error of GE)

Let $A \in \mathbb{R}^{n \times n}$ and suppose GE produces computed LU factor $\hat{L}$, $\hat{U}$ and a computed solution $\hat{x}$ to $Ax = b$

then $(A + \Delta A)\hat{x} = b$, with $|\Delta A| \le \gamma_{3n} |\hat{L}| |\hat{U}|$ $(\ast)$