7. Solving linear equations

- triangular linear systems
- solution via QR factorization
- Gaussian elimination, LU factorization
- pivoted LU factorization
- condition of linear systems

Solution of triangular linear equations

- if $A \in \mathbb{R}^{n\times n}$ is lower/upper triangular with nonzero diagonals
- $Ax = b$ can be solved using forward/back substitution

Forward substitution algorithm: assume A is *lower triangular*

$$
\begin{aligned}
x_1 &= b_1/A_{11} \\
x_2 &= (b_2 - A_{21}x_1)/A_{22} \\
x_3 &= (b_3 - A_{31}x_1 - A_{32}x_2)/A_{33} \\
&\;\vdots \\
x_n &= (b_n - A_{n1}x_1 - A_{n2}x_2 - \cdots - A_{n,n-1}x_{n-1})/A_{nn}
\end{aligned}
$$

this can be written as

$$
x_1 = b_1/A_{11}, \quad x_i = \Big(b_i - \sum_{j=1}^{i-1} A_{ij}x_j\Big)/A_{ii},, \quad i = 2, \dots, n
$$

Back substitution algorithm: assume A is *upper triangular*

$$\begin{aligned}
x_n &= b_n/A_{nn} \\
x_{n-1} &= (b_{n-1} - A_{n-1,n}x_n)/A_{n-1,n-1} \\
x_{n-2} &= (b_{n-2} - A_{n-2,n-1}x_{n-1} - A_{n-2,n}x_n)/A_{n-2,n-2} \\
&\vdots \\
x_1 &= (b_1 - A_{12}x_2 - A_{13}x_3 - \cdots - A_{1n}x_n)/A_{11}
\end{aligned}$$

this can be written as

$$x_n = b_n/A_{nn}, \quad x_i = \big(b_i - \sum_{j=i+1}^{n} A_{ij}x_j\big)/A_{ii},, \quad i = n-1, \ldots, 1$$

Complexity

$$1 + 3 + 5 + \cdots + (2n-1) = \sum_{k=1}^{n}(2k-1) = n^2 \text{ flops}$$

Example

$$\begin{array}{llll} 5x_1 & & & = 15 \\ x_1 & +2x_2 & & = 7 \\ -x_1 & +3x_2 & +2x_3 & = 5 \end{array}, \quad A = \begin{bmatrix} 5 & 0 & 0 \\ 1 & 2 & 0 \\ -1 & 3 & 2 \end{bmatrix}, \ b = \begin{bmatrix} 15 \\ 7 \\ 5 \end{bmatrix}$$

applying the forward substitution:

$$x_1 = \frac{15}{5} = 3$$
$$x_2 = \frac{7-3}{2} = 2$$
$$x_3 = \frac{5+3-6}{2} = 1$$

Inverse of triangular matrix

a triangular matrix A with nonzero diagonal elements is nonsingular:

$$Ax = 0 \quad \Longrightarrow \quad x = 0$$

this follows from forward or back substitution applied to the equation $Ax = 0$

- inverse of A can be computed by solving $AX = I$ column by column

$$A[x_1 \;\; x_2 \;\; \cdots \;\; x_n] = [e_1 \;\; e_2 \;\; \cdots \;\; e_n] \quad (x_i \text{ is the } i\text{th column of } X)$$

  – inverse of lower/upper triangular matrix is lower/upper triangular

- complexity of computing inverse of $n \times n$ triangular matrix

$$n^2 + (n-1)^2 + \cdots + 2^2 + 1 = \frac{n(n+1)(2n+1)}{6} \approx \frac{1}{3}n^3 \text{ flops}$$

- conclusion: using back/forward substitution is more efficient than inverse way

Outline

- triangular linear systems
- **solution via QR factorization**
- Gaussian elimination, LU factorization
- pivoted LU factorization
- condition of linear systems

Solving linear equations via QR factorization

- assuming A is nonsingular, then $x = A^{-1}b$ solves $Ax = b$
- with QR factorization $A = QR$, we have $A^{-1} = (QR)^{-1} = R^{-1}Q^T$
- compute $x = R^{-1}(Q^Tb)$ by back substitution

QR factorization method: to solve $Ax = b$ with nonsingular $A \in \mathbb{R}^{n\times n}$

1. factor A as $A = QR$
2. compute $y = Q^Tb$
3. solve $Rx = y$ by back substitution

Complexity

- QR factorization $2n^3$ flops
- matrix-vector product $2n^2$
- back substitution n^2

total $= 2n^3 + 3n^2 \approx 2n^3$

Multiple right-hand sides

consider k sets of linear equations with the same coefficient matrix A:

$$Ax_1 = b_1, \quad Ax_2 = b_2, \quad \ldots, \quad Ax_k = b_k$$

- factor A once ($2n^3$ flops)
- solve $QRx_i = b_i$ for each $i = 1, \ldots, n$ ($3kn^2$ flops)

Complexity

- $2n^3 + 3kn^2$ flops if we reuse the factorization $A = QR$
- for $k \ll n$, cost is roughly equal to cost of solving one equation: $2n^3$

Computing the inverse

solving the matrix equation $AX = I$ gives A^{-1}

- equivalent to solving n equations $Ax_i = e_i$ ($i = 1, \ldots, n$) or:

$$Rx_1 = Q^T e_1, \quad Rx_2 = Q^T e_2, \quad \ldots, \quad Rx_n = Q^T e_n$$

- x_i is ith column of X and $Q^T e_i$ is the ith column of Q^T
- complexity is $2n^3 + n^3 = 3n^3$

Solving linear equations by computing the inverse

- compute inverse A^{-1} costs $3n^3$, then compute $A^{-1}b$ costs $2n^2$
- total complexity: $3n^3 + 2n^2 \approx 3n^3$
- more expensive than QR factorization method, which costs $2n^3$
- while inverse appears in many formulas, it is computed far less often

Solving general linear equations

suppose $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ with $\mathrm{rank}(A) = r$ and consider solving

$$Ax = b$$

- solution exists if $\mathrm{rank}(A) = \mathrm{rank}[A\ \ b] = r$ ($b \in \mathrm{range}(A)$)
- no solution exists if $\mathrm{rank}[A\ \ b] = r + 1$ ($b \notin \mathrm{range}(A)$)
- we start with the full pivoted QR factorization of A:

$$AP = \hat{Q}\hat{R} = \begin{bmatrix} Q & Q_0 \end{bmatrix} \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}$$

  $\hat{Q} \in \mathbb{R}^{m\times m}$ is orthogonal, $\hat{R} \in \mathbb{R}^{m\times n}$, $P \in \mathbb{R}^{n\times n}$ is a permutation matrix
  - $Q \in \mathbb{R}^{m\times r}$, $Q_0 \in \mathbb{R}^{m\times(m-r)}$
  - $R_1 \in \mathbb{R}^{r\times r}$ is upper triangular with nonzero diagonals, $R_2 \in \mathbb{R}^{r\times(n-r)}$
  - the zero submatrices in the bottom (block) row of $\hat{R}$ have $m - r$ rows

Solving general linear equations using QR factorization

- using $A = \hat{Q}\hat{R}P^T$ we can write $Ax = b$ as

$$\hat{Q}\hat{R}P^T x = \hat{Q}\hat{R}z = b, \quad \text{where} \quad z = P^T x$$

- multiplying both sides by $\hat{Q}^T$ gives the equivalent set of m equations $\hat{R}z = \hat{Q}^T b$
- expanding this into subcomponents gives

$$\hat{R}z = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix} z = \begin{bmatrix} Q^T b \\ Q_0^T b \end{bmatrix}$$

- we see that there is no solution of $Ax = b$, unless we have $Q_0^T b = 0$
- assuming $Q_0^T b = 0$, the equations reduce to a set r linear equations in n variables

$$R_1 z_1 + R_2 z_2 = Q^T b$$

- we can find a solution of these equations by setting z_2 arbitrary

Solving general linear equations using QR factorization

- solving for z_1:

$$R_1 z_1 = Q^T b - R_2 z_2 \iff z_1 = R_1^{-1}(Q^T b - R_2 z_2)$$

- now we have a z that satisfies $\hat{R} z = \hat{Q}^T b$

- we get the corresponding x from $x = Pz$:

$$x = P \begin{bmatrix} R_1^{-1}(Q^T b - R_2 z_2) \\ z_2 \end{bmatrix} = P \begin{bmatrix} R_1^{-1} Q^T b \\ 0 \end{bmatrix} + P \begin{bmatrix} R_1^{-1} R_2 \\ I \end{bmatrix} z_2$$

  this x satisfies $Ax = b$, provided we have $Q_0^T b = 0$

- right term is in $\mathrm{null}(A)$ – see page 6.20

- a particular solution is obtained by setting $z_2 = 0$:

$$x = P \begin{bmatrix} R_1^{-1} Q^T b \\ 0 \end{bmatrix}$$

- the construction outlined above is pretty much what `A\b` does in MATLAB

Outline

- triangular linear systems
- solution via QR factorization
- **Gaussian elimination, LU factorization**
- pivoted LU factorization
- condition of linear systems

Elementary row operations

suppose A is an $n \times n$ invertible matrix, b is an n-vector

solution of $Ax = b$ is invariant under the elementary row operations:

1. *interchanging any two rows of the matrix* $[A \mid b]$
2. *multiplying one of its rows by a real nonzero number*
3. *adding a scalar multiple of one row to another row*

Elementary elimination matrix

for n-vector u, we can zero out elements below kth entry as follows:

$$G^{(k)}u = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -L_{k+1,k} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -L_{n,k} & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_k \\ u_{k+1} \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} u_1 \\ \vdots \\ u_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- $L_{i,k} = u_i/u_k$ for $i = k+1, \ldots, n$
- the divisor u_k is called the *pivot*
- $G^{(k)}$ is unit lower triangular, and hence nonsingular
- $G^{(k)}$ called *elementary elimination matrix* or *Gauss transformation*

Gaussian elimination procedure

Iteration 1

- zero out the first column below the main diagonal
- subtract $\frac{A_{i1}}{A_{11}}\times$ the first row from the ith row for all $i = 2, 3, \ldots, n$

$$\underbrace{\begin{bmatrix} 1 & 0 \\ -L_{2:n,1} & I \end{bmatrix}}_{G^{(1)}} [A \mid b] = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} & b_1 \\ 0 & A_{22}^{(1)} & \cdots & A_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & A_{n2}^{(1)} & \cdots & A_{nn}^{(1)} & b_n^{(1)} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11} & A_{1,2:n} & b_1 \\ 0 & A_{2:n,2:n} - L_{2:n,1}A_{1,2:n} & b_{2:n} - L_{2:n,1}b_1 \end{bmatrix}$$

where $L_{2:n,1} = A_{2:n,1}/A_{11} = (A_{21}/A_{11}, \ldots, A_{n1}/A_{11})$

Iteration 2:

- zero out the second column below diagonal
- subtract $\frac{A_{i2}^{(1)}}{A_{22}^{(1)}} \times$ the second row from the ith row for all $i = 3, 4, \ldots, n$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -L_{3:n,2} & I \end{bmatrix}}_{G^{(2)}} [A^{(1)} | b^{(1)}] = \begin{bmatrix} A_{11} & A_{12} & \cdots & \cdots & A_{1n} & b_1 \\ 0 & A_{22}^{(1)} & A_{23}^{(1)} & \cdots & A_{2n}^{(1)} & b_2^{(1)} \\ \vdots & 0 & A_{33}^{(2)} & \cdots & A_{3n}^{(2)} & b_3^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & A_{n3}^{(2)} & \cdots & A_{nn}^{(2)} & b_n^{(2)} \end{bmatrix}$$

$$= \begin{bmatrix} A_{11} & A_{12} & A_{1,3:n} & b_1 \\ 0 & A_{22}^{(1)} & A_{2,3:n}^{(1)} & b_2^{(1)} \\ 0 & 0 & A_{3:n,3:n}^{(1)} - L_{3:n,2} A_{2,3:n}^{(1)} & b_{3:n}^{(1)} - L_{3:n,2} b_2^{(1)} \end{bmatrix}$$

where $L_{3:n,2} = A_{3:n,2}^{(1)} / A_{22}^{(1)} = (A_{32}^{(1)} / A_{22}^{(1)}, \ldots, A_{n2}^{(1)} / A_{22}^{(1)})$

Final iteration

- after $n-1$ iterations, we get the upper-triangular system

$$[A^{(n-1)}|b^{(n-1)}] = \begin{bmatrix} A_{11} & A_{12} & \cdots & \cdots & A_{1n} & b_1 \\ 0 & A_{22}^{(1)} & A_{23}^{(1)} & \cdots & A_{2n}^{(1)} & b_2^{(1)} \\ \vdots & 0 & A_{33}^{(2)} & \cdots & A_{3n}^{(2)} & b_3^{(2)} \\ \vdots & \vdots & \cdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & A_{nn}^{(n-1)} & b_n^{(n-1)} \end{bmatrix}$$

  where

$$U = A^{(n-1)} = G^{(n-1)} \cdots G^{(2)} G^{(1)} A$$
$$b^{(n-1)} = G^{(n-1)} \cdots G^{(2)} G^{(1)} b$$

- now, we solve $Ux = b^{(n-1)}$ using back substitution

Example

$$Ax = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 10 \end{bmatrix} = b$$

we subtract four times the first row from each of the second and third rows:

$$G^{(1)}A = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & -2 & -4 \end{bmatrix}$$

$$G^{(1)}b = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 10 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \\ -2 \end{bmatrix}$$

we subtract 0.5 times the second row from the third row:

$$G^{(2)}G^{(1)}A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & -2 & -4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & 0 & -1 \end{bmatrix}$$

$$G^{(2)}G^{(1)}b = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 3 \\ -6 \\ -2 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \\ 1 \end{bmatrix}$$

we have reduced the original system to the equivalent upper triangular system

$$Ux = \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \\ 1 \end{bmatrix}$$

which can now be solved by back-substitution to obtain $x = (-1, 3, -1)$

Inverse of elementary matrix

$$\begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -L_{k+1,k} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -L_{n,k} & 0 & \cdots & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & L_{k+1,k} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & L_{n,k} & 0 & \cdots & 1 \end{bmatrix} = L^{(k)}$$

- compactly: $(I - l_k e_k^T)^{-1} = I + l_k e_k^T$ where $l_k = (0, \ldots, 0, L_{k+1,k}, \ldots, L_{n,k})$
- inverse $L^{(k)}$ has same form as $G^{(k)}$ with subdiagonal entries negated
- for $k \leq j$, we have $e_k^T l_j = 0$ and thus

$$L^{(1)} \cdots L^{(n-2)} L^{(n-1)} = I + l_1 e_1^T + \cdots + l_{n-1} e_{n-1}^T$$

which is also lower triangular

LU factorization

Gaussian elimination produces

$$U = G^{(n-1)} \cdots G^{(2)} G^{(1)} A$$

or written equivalently

$$A = LU$$

- $L = L^{(1)} \cdots L^{(n-2)} L^{(n-1)}$ where $L^{(k)} = \left(G^{(k)}\right)^{-1}$
- L is lower triangular (see previous page)
- this is called *LU factorization* or *LU decomposition*
- requires pivots to be nonzero during Gaussian elimination procedure

Example

consider A from previous example

$$A = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix}$$

we have

$$G^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix}, \quad G^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.5 & 1 \end{bmatrix}$$

hence,

$$L = \left(G^{(1)}\right)^{-1}\left(G^{(2)}\right)^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 4 & 0.5 & 1 \end{bmatrix}$$

we thus have

$$A = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 4 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & 0 & -1 \end{bmatrix} = LU$$

Gaussian elimination algorithm

---

given $Ax = b$ with nonsingular $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$
set $U = A$ and $L = I$

for $k = 1, \ldots, n-1$

1. $L_{k+1:n,k} = U_{k+1:n,k}/U_{kk}$ then set $U_{k+1:n,k} = 0$
2. $U_{k+1:n,k+1:n} = U_{k+1:n,k+1:n} - L_{k+1:n,k}U_{k,k+1:n}$
3. $b_{k+1:n} = b_{k+1:n} - L_{k+1:n,k}b_k$

---

next, apply the algorithm of back substitution to $Ux = b$

algorithm gives factorization $A = LU$

Complexity

- cost is approximately $(2/3)n^3$
- back substitution costs n^2
- cost of the Gaussian elimination phase dominates

Recursive computation of $A = LU$

$$
\begin{bmatrix} A_{11} & A_{1,2:n} \\ A_{2:n,1} & A_{2:n,2:n} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ L_{2:n,1} & L_{2:n,2:n} \end{bmatrix} \begin{bmatrix} U_{11} & U_{1,2:n} \\ 0 & U_{2:n,2:n} \end{bmatrix}
$$

$$
= \begin{bmatrix} U_{11} & U_{1,2:n} \\ U_{11}L_{2:n,1} & L_{2:n,1}U_{1,2:n} + L_{2:n,2:n}U_{2:n,2:n} \end{bmatrix}
$$

1. find the first row of U and the first column of L:

$$
U_{11} = A_{11}, \quad U_{1,2:n} = A_{1,2:n}, \quad L_{2:n,1} = \frac{1}{A_{11}} A_{2:n,1}
$$

2. factor the $(n-1) \times (n-1)$-matrix

$$
L_{2:n,2:n}U_{2:n,2:n} = A_{2:n,2:n} - L_{2:n,1}U_{1,2:n} = A_{2:n,2:n} - \frac{1}{A_{11}} A_{2:n,1}A_{1,2:n}
$$

   this is an LU factorization of size $(n-1) \times (n-1)$

3. we can calculate $L_{2:n,2:n}$ and $U_{2:n,2:n}$ by repeating process on factored matrix

(this is basically Gaussian elimination on page 7.22)

Example

$$A = \left[\begin{array}{ccc} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{array}\right]$$

factor as $A = LU$ with L unit lower triangular, U upper triangular

$$A = \left[\begin{array}{ccc} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{array}\right] = \left[\begin{array}{ccc} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{array}\right] \left[\begin{array}{ccc} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{array}\right]$$

Solution

- first row of U, first column of L:

$$\begin{bmatrix} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 3/4 & L_{32} & 1 \end{bmatrix} \begin{bmatrix} 8 & 2 & 9 \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- second row of U, second column of L:

$$\begin{bmatrix} 9 & 4 \\ 7 & 9 \end{bmatrix} - \begin{bmatrix} 1/2 \\ 3/4 \end{bmatrix} \begin{bmatrix} 2 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ L_{32} & 1 \end{bmatrix} \begin{bmatrix} U_{22} & U_{23} \\ 0 & U_{33} \end{bmatrix}$$

$$\begin{bmatrix} 8 & -1/2 \\ 11/2 & 9/4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 11/16 & 1 \end{bmatrix} \begin{bmatrix} 8 & -1/2 \\ 0 & U_{33} \end{bmatrix}$$

- third row of U: $U_{33} = 9/4 + 11/32 = 83/32$

putting things together, we obtain

$$A = \begin{bmatrix} 8 & 2 & 9 \\ 4 & 9 & 4 \\ 6 & 7 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 3/4 & 11/16 & 1 \end{bmatrix} \begin{bmatrix} 8 & 2 & 9 \\ 0 & 8 & -1/2 \\ 0 & 0 & 83/32 \end{bmatrix}$$

Factorization $A = LU$ may not exists

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- first row of U, first column of L:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & L_{32} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- second row of U, second column of L:

$$\begin{bmatrix} 0 & 2 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ L_{32} & 1 \end{bmatrix} \begin{bmatrix} U_{22} & U_{23} \\ 0 & U_{33} \end{bmatrix}$$

- issue: $U_{22} = 0$, $U_{23} = 2$, $L_{32} = 1/0$! (can be fixed via pivoting)

Outline

- triangular linear systems
- solution via QR factorization
- Gaussian elimination, LU factorization
- **pivoted LU factorization**
- condition of linear systems

LU factorization with pivoting

LU factorization (no pivoting)

$$A = LU$$

- L unit lower triangular, U upper triangular
- does not always exist (even if A is nonsingular)
- sufficient existence condition: A is *diagonally dominant* $|A_{ii}| \geq \sum_{j \neq i} |A_{ij}|$

LU factorization with row pivoting

$$PA = LU$$

- P permutation matrix, L unit lower triangular, U upper triangular
- interpretation: permute the rows of A and factor $PA = LU$
- always exists if A is nonsingular
- not unique; there may be several possible choices for P, L, U

LU factorization and matrix inverse

let A is nonsingular and $n \times n$, with LU factorization

$$A = P^T LU$$

- inverse from LU factorization

$$A^{-1} = (P^T LU)^{-1} = U^{-1} L^{-1} P$$

- gives interpretation of solving $Ax = b$ steps: we evaluate

$$x = A^{-1} b = U^{-1} L^{-1} P b$$

  in three steps

$$z_1 = Pb, \quad z_2 = L^{-1} z_1, \quad x = U^{-1} z_2$$

Solving linear equations by LU factorization

---

given $Ax = b$ with nonsingular $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$

1. factor A as $A = P^T LU$
2. solve $(P^T LU)x = b$ in three steps
   (a) permutation: $z_1 = Pb$
   (b) forward substitution: solve $Lz_2 = z_1$
   (c) back substitution: solve $Ux = z_2$

Complexity:

- factorization requires $(2/3)n^3$flops
- forward and back substitution costs n^2 each
- total: $(2/3)n^3 + 2n^2 \approx (2/3)n^3$ flops

this is the standard method for solving $Ax = b$ with nonsingular A

Multiple right-hand sides

k sets of linear equations with same coefficient non-singular matrix $A \in \mathbb{R}^{n \times n}$:

$$Ax_1 = b_1, \quad Ax_2 = b_2, \quad \dots, \quad Ax_k = b_k$$

- factor A once
- forward/back substitution to get x_1
- forward/back substitution to get x_2
- ...etc

complexity: $(2/3)n^3 + 4kn^2 \approx (2/3)n^3$ if $k << n$

Computing the inverse

solve $AX = I$ column by column:

- one LU factorization of A: $(2/3)n^3$ flops
- n solve steps: $2n^3$ flops
- total: $(8/3)n^3$ flops

Conclusion: do not solve $Ax = b$ by multiplying A^{-1} with b

- $4\times$ more computationally expensive than using the LU factorization route
- forming A^{-1} is wasteful in storage
- it may give rise to a more pronounced presence of roundoff errors

Effect of rounding error

$$\begin{bmatrix} 10^{-5} & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

solution is:

$$x_1 = \frac{-1}{1 - 10^{-5}}, \quad x_2 = \frac{1}{1 - 10^{-5}}$$

- let us solve using LU factorization for the two possible permutations:

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{or} \quad P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- we round intermediate results to four significant decimal digits

First choice: $P = I$ (no pivoting)

$$\begin{bmatrix} 10^{-5} & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 10^5 & 1 \end{bmatrix} \begin{bmatrix} 10^{-5} & 1 \\ 0 & 1 - 10^5 \end{bmatrix}$$

- L, U rounded to 4 significant decimal digits

$$L = \begin{bmatrix} 1 & 0 \\ 10^5 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 10^{-5} & 1 \\ 0 & -10^5 \end{bmatrix}$$

- forward substitution

$$\begin{bmatrix} 1 & 0 \\ 10^5 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \Longrightarrow \quad z_1 = 1, \quad z_2 = -10^5$$

- back substitution

$$\begin{bmatrix} 10^{-5} & 1 \\ 0 & -10^5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -10^5 \end{bmatrix} \quad \Longrightarrow \quad x_1 = 0, \quad x_2 = 1$$

error in x_1 is 100%

Second choice: interchange rows

$$\begin{bmatrix} 1 & 1 \\ 10^{-5} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 10^{-5} & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 - 10^{-5} \end{bmatrix}$$

- L, U rounded to 4 significant decimal digits

$$L = \begin{bmatrix} 1 & 0 \\ 10^{-5} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

- forward substitution

$$\begin{bmatrix} 1 & 0 \\ 10^{-5} & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \Longrightarrow \quad z_1 = 0, \quad z_2 = 1$$

- back substitution

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \Longrightarrow \quad x_1 = -1, \quad x_2 = 1$$

error in x_1, x_2 is about 10^{-5}

Conclusion: rounding error and numerical instability

- for some P, small roundoff errors can cause very large errors in the solution
- this is called numerical instability:
  - for the first choice of P in the example, the algorithm is unstable
  - for the second choice of P, it is stable
- a simple rule for selecting a good permutation is via partial pivoting (see next)

Computing LU factorization with partial pivoting

Gaussian elimination with partial pivoting

---

given nonsingular $A \in \mathbb{R}^{n \times n}$
set $P = I$, $L = 0$, $U = A$

for $k = 1, 2, \ldots, n-1$

1. select $q \geq k$ to maximize $|U_{qk}|$
   $P_{k,:} \leftrightarrow P_{q,:}$ (swap rows)
   $U = PU$ (swap rows)
   $L = PL$ (swap rows if $k \geq 2$)
2. set $L_{kk} = 1$
3. $L_{k+1:n,k} = U_{k+1:n,k}/U_{kk}$ then set $U_{k+1:n,k} = 0$
   $U_{k+1:n,k+1:n} = U_{k+1:n,k+1:n} - L_{k+1:n,k}U_{k,k+1:n}$

---

algorithm produces factorization $PA = LU$

Example

$$A = \begin{bmatrix} 0 & 5 & 5 \\ 2 & 3 & 0 \\ 6 & 9 & 8 \end{bmatrix}$$

since $A_{11} = 0$, we swap rows 1 and 3 using

$$U = P_1 A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 5 & 5 \\ 2 & 3 & 0 \\ 6 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 9 & 8 \\ 2 & 3 & 0 \\ 0 & 5 & 5 \end{bmatrix}$$

set $L_{11} = 1$, $(L_{21}, L_{31}) = (\frac{2}{6}, \frac{0}{6})$, and

$$L^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 1/3 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad U^{(1)}_{2:n,2:n} = \begin{bmatrix} 3 & 0 \\ 5 & 5 \end{bmatrix} - \begin{bmatrix} 1/3 \\ 0 \end{bmatrix} \begin{bmatrix} 9 & 8 \end{bmatrix} = \begin{bmatrix} 0 & -8/3 \\ 5 & 5 \end{bmatrix}$$

we swap the second and third row of $U^{(1)}$

$$U^{(2)}_{2:n,2:n} = P_2 U^{(1)}_{2:n,2:n} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -8/3 \\ 5 & 5 \end{bmatrix} = \begin{bmatrix} 5 & 5 \\ 0 & -8/3 \end{bmatrix}$$

we also swap the second and third rows of $L^{(1)}$ and set $L_{22} = 1$

$$L^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/3 & 0 & 0 \end{bmatrix}$$

the matrix $U^{(2)}_{2:n,2:n}$ is upper triangular; hence $U^{(3)}_{3:n,3:n} = -8/3$ and

$$L^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/3 & 0 & 1 \end{bmatrix}$$

the permutation matrix is (I swap rows $1 \leftrightarrow 3$ then $2 \leftrightarrow 3$)

$$P = \begin{bmatrix} 1 & 0 \\ 0 & P_2 \end{bmatrix} P_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

the LU factorization $A = P^T LU$ can now be assembled follows

$$\underbrace{\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 0 & 5 & 5 \\ 2 & 3 & 0 \\ 6 & 9 & 8 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/3 & 0 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 6 & 9 & 8 \\ 0 & 5 & 5 \\ 0 & 0 & -8/3 \end{bmatrix}}_{U}$$

Sparse linear equations

if A is sparse, it is usually factored as

$$P_1 A P_2 = LU$$

P_1 and P_2 are permutation matrices

- interpretation: permute rows and columns of A and factor $\tilde{A} = P_1 A P_2$

$$\tilde{A} = LU$$

- choice of P_1 and P_2 greatly affects the sparsity of L and U
- several heuristic methods exist for selecting good permutations
- in practice: #flops $\ll (2/3)n^3$; exact value depends on n, number of nonzero elements, sparsity pattern

Outline

- triangular linear systems
- solution via QR factorization
- Gaussian elimination, LU factorization
- pivoted LU factorization
- **condition of linear systems**

Matrix 2-norm

a matrix norm $\|\cdot\|$ is any function satisfying the properties

- nonnegative: $\|A\| \geq 0$ for all A
- positive definiteness: $\|A\| = 0$ only if $A = 0$
- homogeneity: $\|\beta A\| = |\beta|\|A\|$
- triangle inequality: $\|A + B\| \leq \|A\| + \|B\|$

the **2-norm** or **spectral norm** is

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

- the norms $\|Ax\|$ and $\|x\|$ are Euclidean norms of vectors
- $\|Ax\|/\|x\|$ gives the amplification factor or gain of A in the direction x
- no simple explicit expression, except for special A
- in MATLAB: `norm(A)`

Special cases

sometimes it is easy to maximize $\|Ax\|/\|x\|$

- zero matrix: $\|0\|_2 = 0$
- identity matrix: $\|I\|_2 = 1$
- diagonal matrix:

$$A = \begin{bmatrix} A_{11} & 0 & \cdots & 0 \\ 0 & A_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{nn} \end{bmatrix}, \quad \|A\|_2 = \max_{i=1,\ldots,n} |A_{ii}|$$

- matrix with orthonormal columns: $\|A\|_2 = 1$

General matrices: $\|A\|_2$ must be computed by numerical algorithms

Additional properties satisfied by the 2-norm

- *submultiplicative* (*consistency* condition)
  - $\|Ax\| \leq \|A\|_2 \|x\|$ if the product Ax exists
  - $\|AB\|_2 \leq \|A\|_2 \|B\|_2$ if the product AB exists
- if A is nonsingular: $\|A\|_2 \|A^{-1}\|_2 \geq 1$
- if A is nonsingular: $1/\|A^{-1}\|_2 = \min_{x \neq 0} (\|Ax\|/\|x\|)$
- $\|A^T\|_2 = \|A\|_2$

Other matrix norms

the **infinity-norm** is the maximum absolute row sum:

$$\|A\|_\infty = \max_{1\le i\le m} \sum_{j=1}^{n} \left|a_{ij}\right|$$

the **1-norm** is the maximum absolute column sum:

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{m} \left|a_{ij}\right|$$

Example

$$A = \begin{bmatrix} 1 & 3 & 7 \\ -4 & 1.2725 & -2 \end{bmatrix}$$

we have

$$\|A\|_\infty = \max\{11, 7.2725\} = 11$$
$$\|A\|_1 = \max\{5, 4.2725, 9\} = 9$$

Condition of a set of linear equations

- assume A is nonsingular and $Ax = b$
- if we change b to $b + \Delta b$, the new solution is $x + \Delta x$ with

$$A(x + \Delta x) = b + \Delta b$$

- the change in x is

$$\Delta x = A^{-1} \Delta b$$

Condition

- well-conditioned if small Δb results in small Δx
- ill-conditioned if small Δb can result in large Δx

Example of ill-conditioned equations

$$A = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ 1 + 10^{-10} & 1 - 10^{-10} \end{bmatrix}, \quad A^{-1} = \begin{bmatrix} 1 - 10^{10} & 10^{10} \\ 1 + 10^{10} & -10^{10} \end{bmatrix}$$

- solution for $b = (1, 1)$ is $x = (1, 1)$
- change in x if we change b to $b + \Delta b$:

$$\Delta x = A^{-1} \Delta b = \begin{bmatrix} \Delta b_1 - 10^{10} (\Delta b_1 - \Delta b_2) \\ \Delta b_1 + 10^{10} (\Delta b_1 - \Delta b_2) \end{bmatrix}$$

small Δb can lead to very large Δx

Bound on absolute error

suppose A is nonsingular and define

$$x = A^{-1}b, \quad \Delta x = A^{-1}\Delta b$$

Upper bound on $\|\Delta x\|$:

$$\|\Delta x\| \leq \|A^{-1}\|_2 \|\Delta b\|$$

- small $\|A^{-1}\|_2$ means that $\|\Delta x\|$ is small when $\|\Delta b\|$ is small
- large $\|A^{-1}\|_2$ means that $\|\Delta x\|$ can be large, even when $\|\Delta b\|$ is small
- for every A, there exists nonzero Δb such that $\|\Delta x\| = \|A^{-1}\|_2 \|\Delta b\|$

Bound on relative error

suppose in addition that $b \neq 0$; hence $x \neq 0$

Upper bound on $\|\Delta x\|/\|x\|$:

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\|_2 \|A^{-1}\|_2 \frac{\|\Delta b\|}{\|b\|}$$

- follows from $\|\Delta x\| \leq \|A^{-1}\|_2 \|\Delta b\|$ and $\|b\| \leq \|A\|_2 \|x\|$
- $\|A\|_2 \|A^{-1}\|_2$ small means $\|\Delta x\|/\|x\|$ is small when $\|\Delta b\|/\|b\|$ is small
- $\|A\|_2 \|A^{-1}\|_2$ large means $\|\Delta x\|/\|x\|$ can be much larger than $\|\Delta b\|/\|b\|$
- for every A, there exist nonzero $b, \Delta b$ such that equality holds

Condition number

the *condition number* of a nonsingular matrix A is

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_2$$

- we have $1 = \|I\|_2 = \|A^{-1}A\|_2 \leq \kappa(A)$
- condition number is a measure of how close a matrix is to being singular
- matrix is ideally conditioned if its condition number equals 1
- A is a well-conditioned matrix if $\kappa(A)$ is small (close to 1):
  the relative error in x is not much larger than the relative error in b
- A is badly conditioned or ill-conditioned if $\kappa(A)$ is large (nearly singular):
  the relative error in x can be much larger than the relative error in b
- by convention $\kappa(A) = \infty$ if A is singular

Example

- A is blurring matrix, nonsingular with condition number $\approx 10^9$
- we apply A to image x

blurred image

$y_1 = Ax$

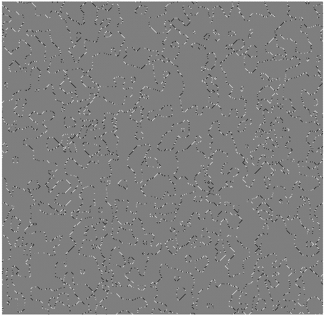
blurred and noisy image

$y_2 = Ax+$ small noise

Example

we solve $Ax = y$ for the two blurred images

$A^{-1}y_1$



$A^{-1}y_2$

- illustrates ill-conditioning of A (nearly singular)
- inverse amplifies the noise component

Residual and condition number

$$A(x + \Delta x) = b + \Delta b$$

- let $\hat{x}$ be an estimate solution of $Ax = b$
- residual $\hat{r} = b - A\hat{x}$; zero residual mean we get exact solution
- let $\Delta x = \hat{x} - x$ so $\hat{x} = x + \Delta x$
- we have

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

- hence from before

$$\frac{\|\Delta x\|}{\|x\|} \leq \kappa(A) \frac{\|\hat{r}\|}{\|b\|}$$

- error can be much larger than residual when condition number is large
- small residual does not imply small error in solution unless A is well-conditioned

Example

$$A = \begin{bmatrix} 0.913 & 0.659 \\ 0.457 & 0.330 \end{bmatrix}, \quad b = \begin{bmatrix} 0.254 \\ 0.127 \end{bmatrix}$$

- consider two approximate solutions

$$\hat{x}_1 = \begin{bmatrix} 0.6391 \\ -0.5 \end{bmatrix} \quad \text{and} \quad \hat{x}_2 = \begin{bmatrix} 0.999 \\ -1.001 \end{bmatrix}$$

  the norms of their respective residuals are

$$\|\hat{r}_1\| = 6.8721 \times 10^{-5} \quad \text{and} \quad \|\hat{r}_2\| = 1.8 \times 10^{-3}$$

- $\hat{x}_1$ has smaller residual but solution is $(1, -1)$, so $\hat{x}_2$ is more accurate
- this is due to A being ill-conditioned
- in practice we cannot expect to deliver much more than a small residual

References and further readings

- L. Vandenberghe, *EE133A Lecture Notes*, University of California, Los Angeles.
- M. T. Heath. *Scientific Computing: An Introductory Survey* (revised second edition). Society for Industrial and Applied Mathematics, 2018.
- U. M. Ascher. *A First Course on Numerical Methods*. Society for Industrial and Applied Mathematics, 2011.