Section 2.1 - 2.3

- Polynomial interpolation
- Matrix operations
- Matlab \ and / (video)
- Solving triangular systems by forward and backward substitutions

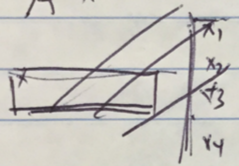
---

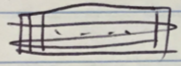
$$\left[\begin{array}{cc|c} 3 & -1 & 2 \\ 0 & 2 & 2 \end{array}\right] * \left[\begin{array}{cc|c} 2 & 1 & 1 \\ 0 & 3 & 4 \\ \hline -1 & -1 & -1 \end{array}\right]$$

$$\underset{2\times 2}{[\, D} \;|\; \underset{2\times 1}{E\,]} * \left[\begin{array}{c|c} \overset{2\times 2}{F} & \overset{2\times 1}{G} \\ \hline \underset{1\times 2}{H} & \underset{1\times 1}{J} \end{array}\right]$$

① Ax

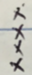
$$\begin{matrix} 1 & 6 & 11 \\ 2 & 7 & 12 \\ 3 & 8 & 13 \\ 4 & 9 & 14 \\ 5 & 10 & 15 \end{matrix} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$





~~View~~

view 1



view 2 $\quad x \,\big|\, + x \,\big|\, + x \,\big|\, + \cdots + x \,\big|$

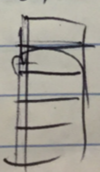
linear combination of the columns of A

② $y^T A$

$[y_1 \ y_2 \ y_3 \ y_4 \ y_5]$

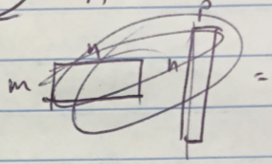
1×5

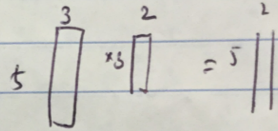
5×3



$= \quad y_1 * \text{—} + y_2 * \text{—} \ \vdots \ y_5 * \text{—} \quad =$ l.c. of rows

③ AB





④ $\begin{bmatrix} Ae_1 \\ Ae_2 \\ \vdots \\ Ae_n \end{bmatrix}$

- Identity matrix $\begin{bmatrix}1 & & \\ & \ddots & \\ & & 1\end{bmatrix}$

  $AI = IA = A$

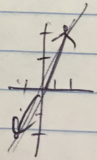
- Inverse matrix

  $\begin{cases} ZA = I & \text{left inverse} \\ AZ = I & \text{right inverse}\end{cases}$ $Z = A^{-1}$

- Singular matrix (no inverse exists)

  $\begin{bmatrix}0&0\\0&0\end{bmatrix}, \begin{bmatrix}0&0\\0&1\end{bmatrix}, \underbrace{\begin{bmatrix}1&-1\\2&-2\end{bmatrix}}_{\text{rank } 1}$

  $\begin{pmatrix}1&-1\\2&-2\end{pmatrix}\begin{pmatrix}x_1 & y_1\\ x_2 & y_2\end{pmatrix} = \begin{pmatrix}1 & \\ & 1\end{pmatrix}$



- $A, B \in \mathbb{R}^{n\times n}$ nonsingular $\Rightarrow (AB)^{-1} = B^{-1}A^{-1}$

- Don't do that inv(A)

  $AB^{-1} = A/B$ slash (e.g. 2/3 ?)

  $A^{-1}B = A\backslash B$ — Back slash

Section 2.4 LU Factorization

- Go through Gaussian elimination (Ex. 2.4.1)
  - Augmented matrix $S = [A \; b]_{n \times (n+1)}$
  - Row operations

The algebra of G.E.

- row 2 = row 2 + α row 1

$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ \alpha & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix} = \begin{bmatrix} x & x & x & x \\ \bar{x} & \bar{x} & \bar{x} & \bar{x} \\ x & x & x & x \\ x & x & x & x \end{bmatrix} = \begin{bmatrix} e_1^T \\ e_2^T + \alpha e_1^T \\ e_3^T \\ e_4^T \end{bmatrix} \begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}$$

$$\begin{bmatrix} e_1^T & \\ e_2^T + \alpha\, e_1^T & \\ e_3^T & \\ e_4^T & \end{bmatrix} A = \left(I + \alpha \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} e_1^T \right) A = \underbrace{\left(I + \alpha\, e_2 e_1^T \right)}_{\text{Matrix Representation (of } L_{21})} A$$

$$\begin{bmatrix} 1 & & & \\ \alpha & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} + \begin{bmatrix} & & & \\ \alpha & & & \\ & & & \\ & & & \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} + \begin{bmatrix} 0 \\ \alpha \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Adding α times row j of A to row i in place

- outer product
- rank-1 matrix

$$\underbrace{\left(I + \alpha\, e_i e_j^T \right)}_{\text{Elementary matrix}} A = \boxed{L_{ij}}\, A$$

(拿 j 去消 i，其他不变)

row row row

$$L_{ij} = \begin{array}{c} \\ i \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} j & & & \end{array} \\ \begin{bmatrix} 1 & 0 & & \\ \alpha & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \end{array} \leftarrow L_{21} \text{ example}$$

$$\left[L_{43}\left[L_{42}\,L_{32}\left(L_{41}\left(L_{31}\left(L_{21}A\right)\right)\right)\right]\right] \Rightarrow \begin{bmatrix} \times & \times & \times & \times \\ 0 & \bar{\times} & \bar{\times} & \bar{\times} \\ 0 & 0 & \tilde{\times} & \tilde{\times} \\ 0 & 0 & \tilde{\times} & \tilde{\times} \end{bmatrix} \Rightarrow \begin{bmatrix} \times & \times & \times & \times \\ 0 & \bar{\times} & \bar{\times} & \bar{\times} \\ 0 & 0 & \tilde{\times} & \tilde{\times} \\ 0 & 0 & 0 & \hat{\times} \end{bmatrix} = U$$

$$\begin{matrix} \times & \times & \times & \times \\ 0 & \bar{\times} & \bar{\times} & \bar{\times} \\ 0 & \bar{\times} & \bar{\times} & \bar{\times} \\ 0 & \bar{\times} & \bar{\times} & \bar{\times} \end{matrix} \qquad \begin{matrix} \times & \times & \times & \times \\ 0 & \bar{\times} & \bar{\times} & \bar{\times} \\ 0 & \bar{\times} & \bar{\times} & \bar{\times} \\ 0 & \times & \times & \times \end{matrix} \qquad \begin{matrix} \times & \times & \times & \times \\ 0 & \bar{\times} & \bar{\times} & \bar{\times} \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{matrix}$$

$$\underline{L_{43}}\;\underline{L_{42}\,L_{32}}\;\underline{L_{41}\,L_{31}\,L_{21}}\,A = U$$

$$\Rightarrow \quad A = \boxed{\underline{L_{21}^{-1}\,L_{31}^{-1}\,L_{41}^{-1}}\;\underline{L_{32}^{-1}\,L_{42}^{-1}}\;\underline{L_{43}^{-1}}}\;U$$

$$\Rightarrow \quad A = \boxed{L}\cdot U$$

Note 1 The inverse of L_{ij}. $(I+\alpha e_i e_j^T)$

$$(I+\alpha e_i e_j^T)(I-\alpha e_i e_j^T)$$

$$= I - \cancel{\alpha e_i e_j^T} + \cancel{\alpha e_i e_j^T} - \alpha^2 e_i \underbrace{\boxed{e_j^T e_i}}_{=0} e_j^T$$

$$= I$$

Note 2 Product of L_{ij}'s

$$L_{21} \cdot L_{31} = (I + 2 e_2 e_1^T)(I + 2 e_3 e_1^T)$$

$$= I + 2e_3 e_1^T + 2 e_2 e_1^T + 4 e_2 \underbrace{\boxed{e_1^T e_3}}_{=0} e_1^T$$

$$\begin{bmatrix} 1 & & & \\ \alpha & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ \beta & & 1 & \\ & & & 1 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ \alpha & 1 & & \\ \beta & & 1 & \\ & & & 1 \end{bmatrix}$$

$$A = \boxed{\underline{L_{21}^{-1}\, L_{31}^{-1}\, L_{41}^{-1}}\;\; \underline{L_{32}^{-1}\, L_{42}^{-1}}\;\; \underline{L_{43}^{-1}}}\; U$$

$$\underline{\begin{bmatrix} 1 & & & \\ -\alpha & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}\begin{bmatrix} 1 & & & \\ & 1 & & \\ -\beta & & 1 & \\ & & & 1 \end{bmatrix}\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ -\gamma & & & 1 \end{bmatrix}}\;\; \underline{\begin{bmatrix} 1 & & & \\ & 1 & & \\ & -a & 1 & \\ & & & 1 \end{bmatrix}\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & -b & & 1 \end{bmatrix}}\;\; \underline{\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -A & 1 \end{bmatrix}}$$

$$A = \begin{bmatrix} 1 & & & \\ -\alpha & 1 & & \\ -\beta & -a & 1 & \\ -\gamma & -b & -A & 1 \end{bmatrix}\begin{bmatrix} x & x & x & x \\ & x & x & x \\ & & x & x \\ & & & x \end{bmatrix} = \begin{bmatrix} x & x & x & x \\ -\alpha & x & x & x \\ -\beta & -a & x & x \\ -\gamma & -b & -A & x \end{bmatrix}$$

In-place operations

$$A = \boxed{L} \cdot \boxed{U}$$

upper triangular matrix

unit lower triangular matrix

Section 2.5 Efficency

Let $f(n) > 0$ & $g(n) > 0$.

- If $\lim_{n\to\infty} \frac{f(n)}{g(n)} < \infty$, then $f(n) = O(g(n))$ f is big-O of g
- If $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$, then $f(n) \sim g(n)$ f is asymptotic to g
- $f(n) \sim g(n) \Rightarrow f(n) = O(g(n))$

Example 2.5.1

$$\begin{cases} f(n) = a_1 n^3 + b_1 n^2 + c_1 n \\ g(n) = a_2 n^3 \end{cases} \Rightarrow \begin{array}{l} ① \ f(n) = O(g(n)) \\ ② \ f \sim g \text{ if } a_1 = a_2 \end{array}$$

Example 2.5.2

Homework. Hint: Taylor's theorem.

Flops Counts

- FLOP: Floating point operations
- Assumptions
  - $+, -, *, /, \sqrt{}$ ... are all counted as 1 FLOP

    slower expensive
  - "$a+bc$" is treated as 1 FLOP sometimes.
  - Only consider "computation" cost,

    ignore "communication" cost and "storage" costs

Example 2.5.3

Flops of $A \cdot x$ is $O(n^2)$

```
n = 6;
A = magic(n);
x = ones(n,1);
y = zeros(n,1);
for i = 1:n
    for j = 1:n
        y(i) = y(i) + A(i,j)*x(j);
    end
end
```

$$\sum_{i=1}^{n} \sum_{j=1}^{n} 2 = \sum_{i=1}^{n} 2n = 2n^2.$$

Ex. 2.5.4 Timing performance of $Ax = b$ in MATLAB

```
t_ = [];
n_ = 400:400:4000;
for n = n_
    A = randn(n,n);   x = randn(n,1);
    tic
    for j = 1:10
        A*x;
    end
    t = toc;
    t_ = [t_,t/10];
end
```

```
fprintf('    n     time (sec)\n')
fprintf('
fprintf('\n\n')
```

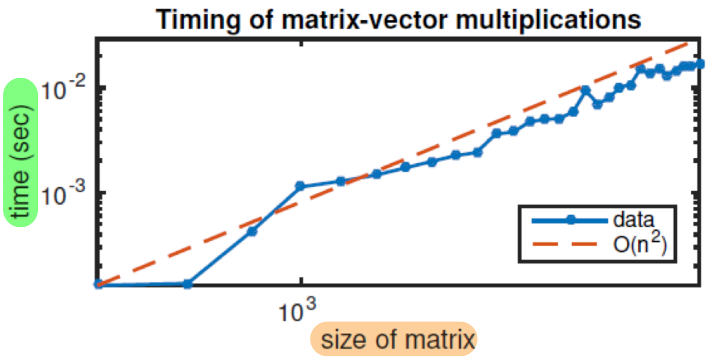
n	time (sec)
400	1.47e-04
800	7.49e-04
1200	1.80e-03
1600	1.88e-03
2000	2.69e-03
2400	4.14e-03
2800	5.29e-03
3200	4.81e-03
3600	7.75e-03
4000	7.40e-03

If we expect the timing is $O(n^p)$ (or cn^p), use log-log plot.

$\because \; t = cn^p \Rightarrow \log t = \log(cn^p)$

$$\Rightarrow \log t = \underbrace{p}_{\text{slope}} \log(n) + \underbrace{\log(c)}_{\text{intercept}}$$

```
hold on, loglog(n_,t_(1)*(n_/n_(1)).^2,'--')
axis tight
legend('data','O(n^2)','location','southeast')
```



Flops of GE


```
8   n = length(A);
9   L = eye(n);    % ones on diagonal
10
11  % Gaussian elimination
12  for j = 1:n-1
13    for i = j+1:n
14      L(i,j) = A(i,j) / A(j,j);    % row multiplier
15      A(i,j:n) = A(i,j:n) - L(i,j)*A(j,j:n);
16    end
17  end
18
19  U = triu(A);
```

+/-	*	/	
		1	(line 14)
$n-j+1$	$n-j+1$		(line 15)

$$2(n-j+1)+1 = 2(n-j)+3$$

$$\sum_{j=1}^{n-1}\sum_{i=j+1}^{n}[2(n-j)+3] = \sum_{j=1}^{n-1}(n-j)[2(n-j)+3]$$

Let $k = n-j$. We have $\displaystyle\sum_{k=1}^{n-1} k(2k+3) = 2\sum_{k=1}^{n-1}k^2 + 3\sum_{k=1}^{n-1}k$

By calculus,

$$\sum_{k=1}^{n} k \sim \frac{n^2}{2} = O(n^2), \text{ as } n \to \infty,$$

$$\sum_{k=1}^{n} k^2 \sim \frac{n^3}{3} = O(n^3), \text{ as } n \to \infty,$$

$$\vdots$$

$$\sum_{k=1}^{n} k^p \sim \frac{n^{p+1}}{p+1} = O(n^{p+1}), \text{ as } n \to \infty,$$

$$\sim \frac{2}{3}(n-1)^3 + \frac{3}{2}(n-1)^2$$

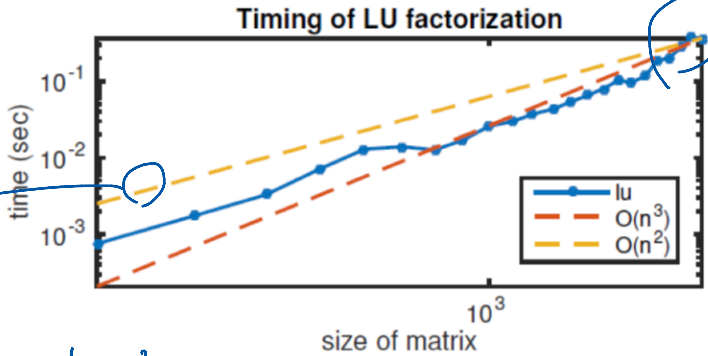
$$\sim \frac{2}{3}n^3$$

Ex. 2.5.5 Timing (Flops) of LU in MATLAB

```matlab
t_ = [];
n_ = 200:100:2400;
for n = n_
    A = randn(n,n);
    tic
    for j = 1:6,  [L,U] = lu(A);  end
    t = toc;
    t_ = [t_,t/6];
end
```

Operation Counts: Matlab

```matlab
hold on, loglog(n_,t_(end)*(n_/n_(end)).^2,'--')
legend('lu','O(n^3)','O(n^2)','location','southeast')
```



why $O(n^2)$ is higher?

Hint: constants before n^2 and n^3.

what happen for larger matrix?

§ 2.6 Row pivoting

(in exact arithmatics)

Q1: What if the pivot element is zero ?

Q2: What if the pivot element is a small number ?

(in floating-point arith.)

Example 2.6.1

- Show MATLAB

- $$\begin{bmatrix} 2 & 0 & 4 & 3 \\ 0 & \boxed{0} & 6 & -10 \\ 0 & 15 & 0 & -6 \\ 0 & 5 & 1 & -4 \end{bmatrix}$$ zero pivot element

$\Rightarrow$ Solution : row pivoting (partial pivoting) by exchange rows

Thm 2.6.1

If a pivot element and all the elements below it are zero, then the original matrix A is singular. In other words, if A is nonsingular, then Gaussian elimination with row pivoting will run to completion.

Algebra of partial pivoting

- Permutation matrix

  - $P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [e_2 \ e_1 \ e_3]$ $\Rightarrow$
    - PA : switch the first 2 rows
    - AP : " columns
    - $P^T P A = A$ $(P^T = P^{-1})$

- PLU factorization

$$U = \left[L_{43} P_3 \left[(L_{42} L_{32}) P_2 \boxed{(L_{41} L_{31} L_{21}) P_1 A} \right] \right]$$

Why?

$\Rightarrow$ $PA = LU$

or $A = P^T L U$

Example 2.6.2

- To solve $Ax = b \Rightarrow (P^T L U) x = b \Rightarrow L \underbrace{(Ux)}_{y} = Pb \Rightarrow Ux = y$
- $[L, U, P] = lu(A)$

Cost of PLU

- $O(n^2)$ comparison : $(n-1) + \cdots + 1$
- $O(n^3)$ flop

Q2: What if the pivot element is a small number?

(in floating-point arith.)

$$\mathbf{A} = \begin{bmatrix} -\epsilon & 1 \\ 1 & -1 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 1-\epsilon \\ 0 \end{bmatrix}$$

Method 1:

$$\begin{bmatrix} -\epsilon & 1 & 1-\epsilon \\ 0 & -1+\epsilon^{-1} & \epsilon^{-1}-1 \end{bmatrix} \Rightarrow \begin{array}{l} x_2 = 1 \\ x_1 = \dfrac{(1-\epsilon)-1}{-\epsilon} \end{array}$$

Substractive cancellation!

small denominator

Method 2:

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1-\epsilon & 1-\epsilon \end{bmatrix} \Rightarrow \begin{array}{l} x_2 = 1 \\ x_1 = \dfrac{0-(-1)}{1} \end{array}$$

- Switch row 1 and row 2
- Partial pivoting for small pivot element

§2.7 Vector and matrix norms

- To measure the size of vectors and matrices
- Norm: $\mathbb{C}^n \to \mathbb{R}$ or $\mathbb{C}^{m\times n} \to \mathbb{R}$

1. $\|x\| \geq 0$ for all $x \in \mathbb{R}^n$
2. $\|x\| = 0$ if and only if $x = 0$
3. $\|\alpha x\| = |\alpha|\,\|x\|$ for any $x \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$
4. $\|x + y\| \leq \|x\| + \|y\|$ for all $x, y \in \mathbb{R}^n$ (the triangle inequality)

- The general vector norm of interest is the *p*-norm: $(p \geq 1)$

$$\|v\|_p = \left[\sum_{i=0}^{n} |v_i|^p\right]^{1/p}$$

- We are most interested in only three values of *p*: 1,2, or ∞

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2\right)^{\frac{1}{2}} = \sqrt{x^T x}$$

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

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

- Examples:

$$u = [1 \; -5 \; 3], \qquad \mathrm{v} = \begin{bmatrix} -2 \\ 2 \\ 0 \end{bmatrix}$$

- Shape of vector doesn't matter
- 1-norm uses p=1, so that

$$\|u\|_1 = |1| + |-5| + |3| = 9, \quad \|v\|_1 = |-2| + |2| + 0 = 4$$

- For the 2-norm

$$\|u\|_2 = (|1|^2 + |-5|^2 + |3|^2)^{1/2} = \sqrt{35},$$
$$\|v\|_2 = (|-2|^2 + |2|^2 + 0)^{1/2} = \sqrt{8} = 2\sqrt{2}$$

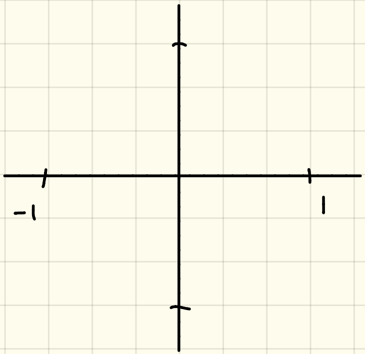
- Examples:

$$u = [1 \; -5 \; 3], \qquad \mathrm{v} = \begin{bmatrix} -2 \\ 2 \\ 0 \end{bmatrix}$$

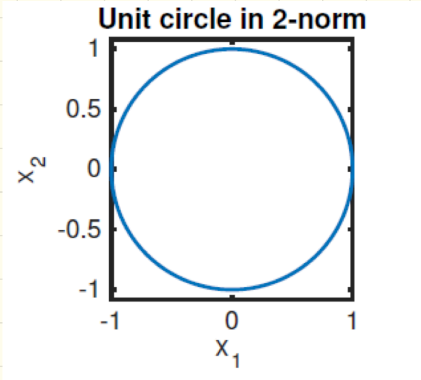
- ∞-norm now:

$$\|u\|_\infty = \max(|1|, |-5|, |3|) = 5$$

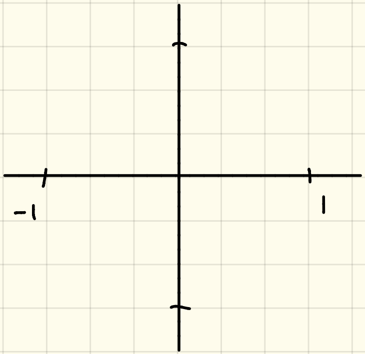
- Multiply by a scalar?
- Always ≥ 0?
- Only 0 if **0** vector?

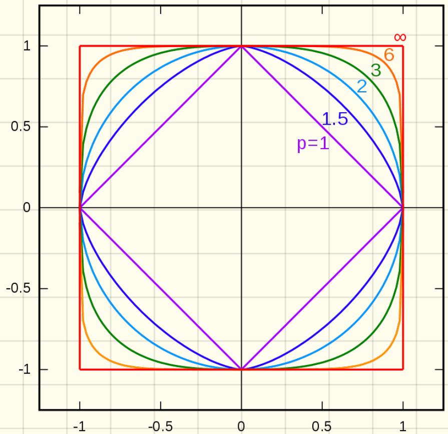


$\|x\|_1 = 1$



$\|x\|_2 = 1$



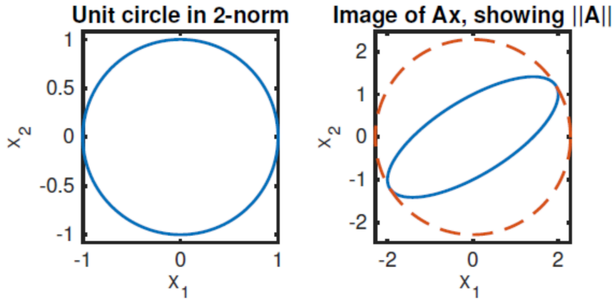


- $\|\boldsymbol{x}\|_\infty = \max(|x_1|, |x_2|) = 1$

For $\boldsymbol{x} = [x_1\ x_2]$ and $\|\boldsymbol{x}\|_2$, the induced matrix norm is $\|\boldsymbol{A}\|_2 = \max\limits_{\|\boldsymbol{x}\|_2=1}(\|\boldsymbol{A}\boldsymbol{x}\|_2) = \max\limits_{\|\boldsymbol{x}\|_2\neq 0}(\|\boldsymbol{A}\boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2)$

The magnitude of the biggest stretching is the norm of $\boldsymbol{A}$

```
subplot(1,2,2), plot(Ax(1,:),Ax(2,:)), axis equal
hold on, plot(twonorm*x(1,:),twonorm*x(2,:),'--')
title('Image of Ax, showing ||A||')
xlabel('x_1'), ylabel('x_2')
```



- The ∞-norm is the maximum row sum of the matrix A
- The 1-norm is the maximum column sum of A
- Mnemonic: think of the "direction" of 1 or ∞
- Example: $\boldsymbol{A} = \begin{bmatrix} 1 & 3 \\ -5 & 8 \end{bmatrix}$
- $\|\boldsymbol{A}\|_1 = \max(1+5, 3+8) = 11$
- $\|\boldsymbol{A}\|_\infty = \max(1+3, 5+8) = 13$

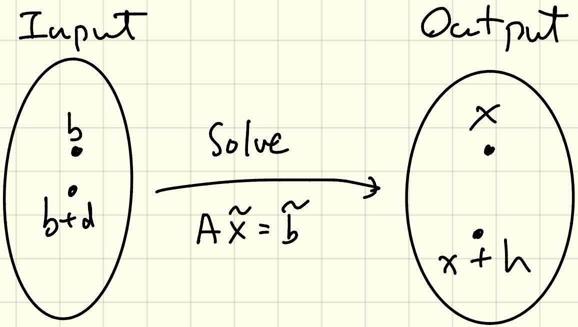
- *For any $n \times n$ matrix* $\mathbf{A}$ *and induced matrix norm,*

$$\|\mathbf{A}\mathbf{x}\| \le \|\mathbf{A}\| \cdot \|\mathbf{x}\|, \quad \text{for any } \mathbf{x} \in \mathbb{R}^n,$$
$$\|\mathbf{A}\mathbf{B}\| \le \|\mathbf{A}\| \cdot \|\mathbf{B}\|, \quad \text{for any } \mathbf{B} \in \mathbb{R}^{n\times n},$$
$$\|\mathbf{A}^k\| \le \|\mathbf{A}\|^k, \quad \text{for any integer } k \ge 0.$$

- $Ax \Rightarrow$ rotation & scaling, see singular value decomposition (SVD)

§ 2.8 Conditioning of Linear Systems

- Consider perturbation on b only



$$Ax = b$$

$$A(x+h) = b + d$$

perturbation

- To bound $\|h\|$ in terms of $\|d\|$

$$A(x+h) = b + d$$
$$\Rightarrow \cancel{Ax} + Ah = \cancel{b} + d$$
$$\Rightarrow Ah = d$$
$$\Rightarrow h = A^{-1} d$$
$$\Rightarrow \|h\| \le \|A^{-1}\| \cdot \|d\|$$

$$\frac{\|h\| / \|x\|}{\|d\| / \|b\|}$$

relative change in output

relative change in input (rhs)

$$= \frac{\|h\| \cdot \|b\|}{\|d\| \cdot \|x\|}$$

$$\le \frac{(\|A^{-1}\| \cdot \cancel{\|d\|})(\|A\| \cdot \cancel{\|x\|})}{\cancel{\|d\|} \cdot \cancel{\|x\|}} = \|A^{-1}\| \cdot \|A\|$$

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

- Matrix condition number $\kappa(A) = \|A^{-1}\| \cdot \|A\|$

- Can be shown that

$$\frac{\|\Delta x\|}{\|x\|} \le \kappa(A) \frac{\|\Delta b\|}{\|b\|}$$

$$\frac{\|\Delta x\|}{\|x\|} \le \kappa(A) \frac{\|\Delta A\|}{\|A\|}$$ [HW], Hint: ① $(A+\Delta E)(x+\Delta x) = b$

② $\Delta A \cdot \Delta x$ is small

- $\kappa(A) \geq 1$

$$1 = \|I\| = \|A^{-1} \cdot A\| \le \|A^{-1}\| \cdot \|A\| = \kappa(A)$$

- Go through Example 2.8.1 on page 81

Note: If $\kappa(A) > \varepsilon_{mach}^{-1}$, the matrix is numerical singular.

Residual and Backward Error

- Residual (vector): $r = b - Ax$

- Consider $r = b - A\tilde{x}$ → computed solution with perturbation in input (e.g. rhs)

$$= AA^{-1}b - A\tilde{x}$$
$$= A(A^{-1}b - \tilde{x})$$
$$= A(x - \tilde{x})$$

→ true solution w/o perturbation

$$\Rightarrow \|x - \tilde{x}\| \leq \|A^{-1}\| \cdot \|r\|$$

relative change of $\|\Delta x\|$

$$\frac{\|\Delta x\|}{\|x\|} = \frac{\|x - \tilde{x}\|}{\|x\|} \leq \frac{\|A^{-1}\| \cdot \|A\| \cdot \|r\|}{\|A\| \cdot \|x\|} \leq \frac{\kappa(A) \cdot \|r\|}{\|b\|}$$

$$b = Ax \Rightarrow \|b\| \leq \|A\| \cdot \|x\|$$

Unknown! Known! ($r = b - Ax$)

$$\frac{\|\Delta x\|}{\|x\|} \le \boxed{\kappa(A)} \frac{\boxed{\|r\|}}{\|b\|}$$

- (small) residual, 會被放大 $\kappa(A)$ 倍
- For ill-conditioned A ($\kappa(A)$ is large), $\frac{\|\Delta x\|}{\|x\|}$ can be big!
- To solve $Ax = b$, we can expect small backward error
  NOT small error

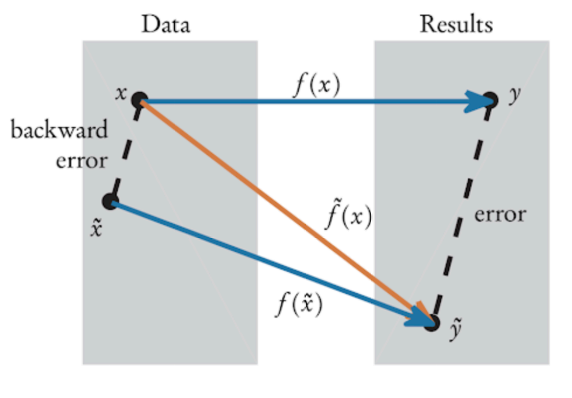


Figure 1.1. *Backward error.*

§2.9 Exploiting matrix structure

- Triangular, banded, sparse, symmetric, symmetric positive definite matrices
- <u>Banded matrices</u>

$$\begin{bmatrix} \times & \times & \times & & \\ \times & \times & \times & \times & \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}$$

Bandwidth : $b_u + b_\ell + 1 = 2 + 1 + 1$

$$\begin{cases} j - i > b_u \Rightarrow A_{ij} = 0 \\ i - j > b_\ell \Rightarrow A_{ij} = 0 \end{cases}$$

- learn these Matlab commands

diag(A), diag(A,1), diag(A,-1), diag([5 8 6 7], 2)

spy, A = gallery('tridiag', c, d, e)

triu(tril(rand(7), 1), -3)

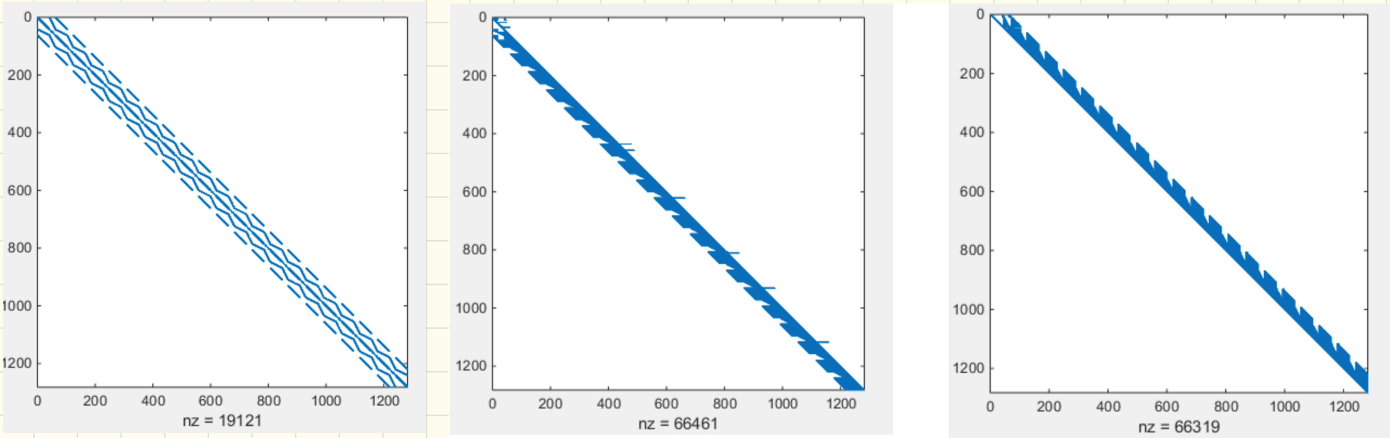
- Go through Examples 2.9.1

– Sparse matrices

– Go through Examples 2.9.2

– An example:

```
>> A=gallery('wathen',20,20);
>> spy(A)
>> [L,U]=lu(A);
>> spy(L)
>> spy(U)
```



Symmetric positive definite (SPD) matrices

- If $\boldsymbol{A}^T = \boldsymbol{A}$, then $\boldsymbol{A}$ is symmetric
- We can modify the LU factorization:
$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U} = \boldsymbol{L}\boldsymbol{I}\boldsymbol{U} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{D}^{-1}\boldsymbol{U}$$
- $\boldsymbol{D}$ is diagonal, with the elements that would have been in $\boldsymbol{U}$ from standard factorization
- Now it turns out that $\boldsymbol{D}^{-1}\boldsymbol{U} = \boldsymbol{L}^T$
- Then $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^T$

Example 2.9.4

- If $\boldsymbol{A}^T = \boldsymbol{A}$, and
- if $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} \geq 0$ for all $x \in R^n$ and $x \neq 0$,
- Then $\boldsymbol{A}$ is positive definite
- We can modify the LU factorization again to the Cholesky factorization:

$$\boldsymbol{A} = \boldsymbol{R}^T \boldsymbol{R}, \qquad \boldsymbol{R} = \boldsymbol{D}^{1/2} \boldsymbol{L}^T$$

- $\boldsymbol{R}$ has all positive entries on the diagonal
- The MATLAB function `chol` will compute the Cholesky factorization

- Start with `A=magic(5);`

```
B = A'*A

B =
        1055         865         695         770         840
         865        1105         815         670         770
         695         815        1205         815         695
         770         670         815        1105         865
         840         770         695         865        1055

R = chol(B)
R =
   32.4808   26.6311   21.3973   23.7063   25.8615
         0   19.8943   12.3234    1.9439    4.0856
         0         0   24.3985   11.6316    3.7415
         0         0         0   20.0982    9.9739
         0         0         0         0   16.0005

norm( R'*R - B )

ans =
     0
```