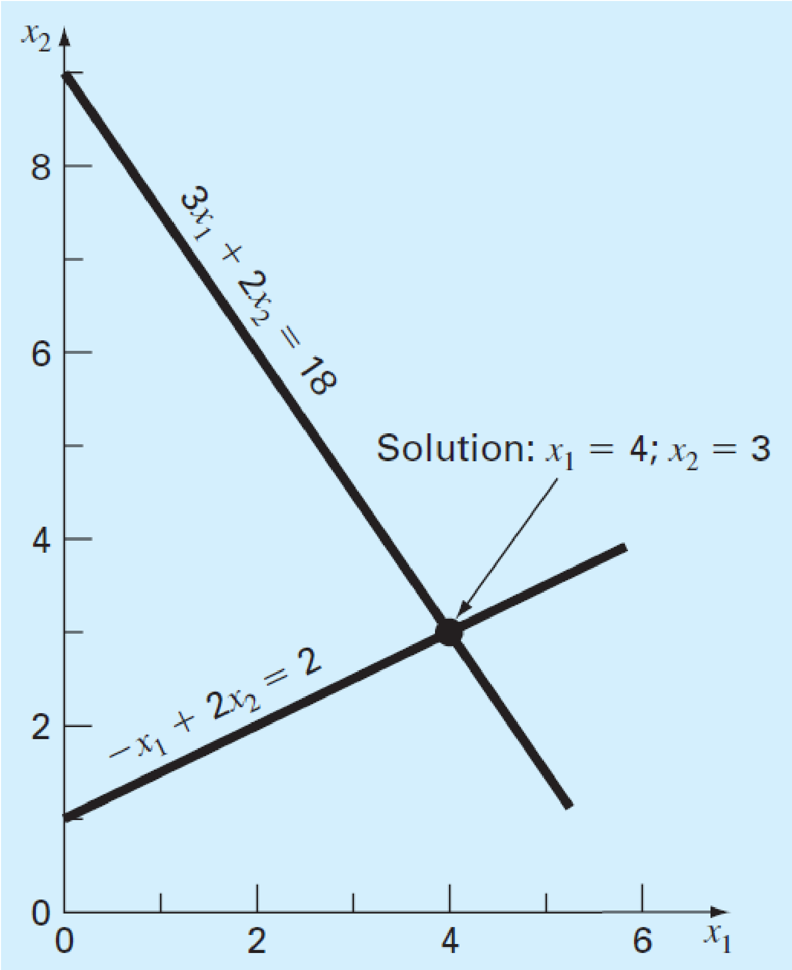
Gauss Elimination

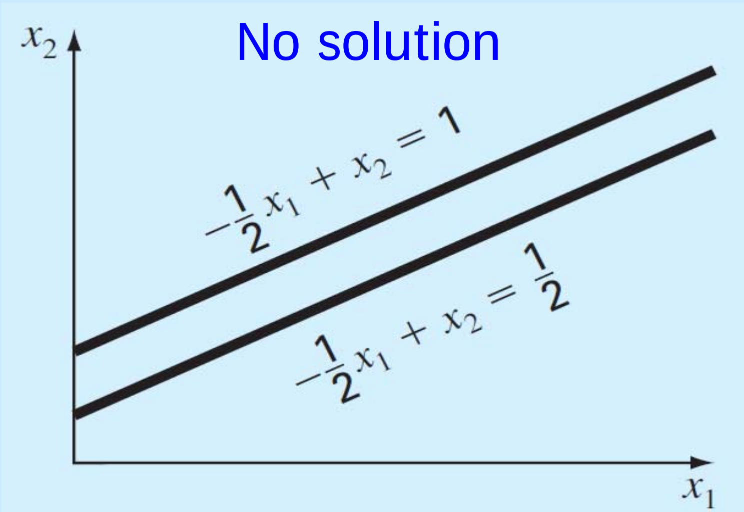
Hsiao-Lung Chan
Dept Electrical Engineering
Chang Gung University, Taiwan
chanhl@mail.cgu.edu.tw

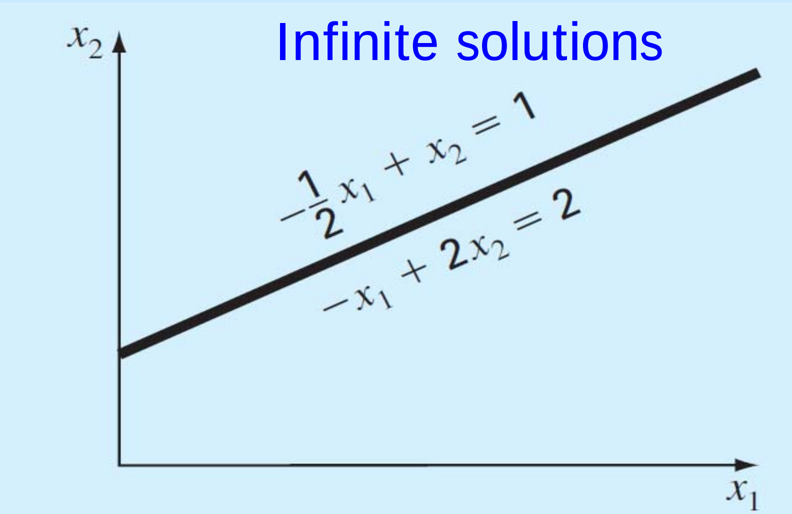
Solving small numbers of equations by graphical method

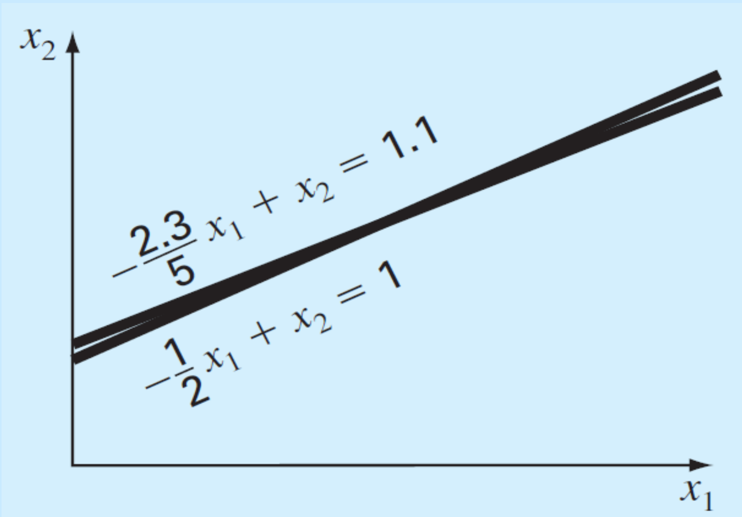
- The location of the intercept provides a solution



Singular and ill-conditioned systems







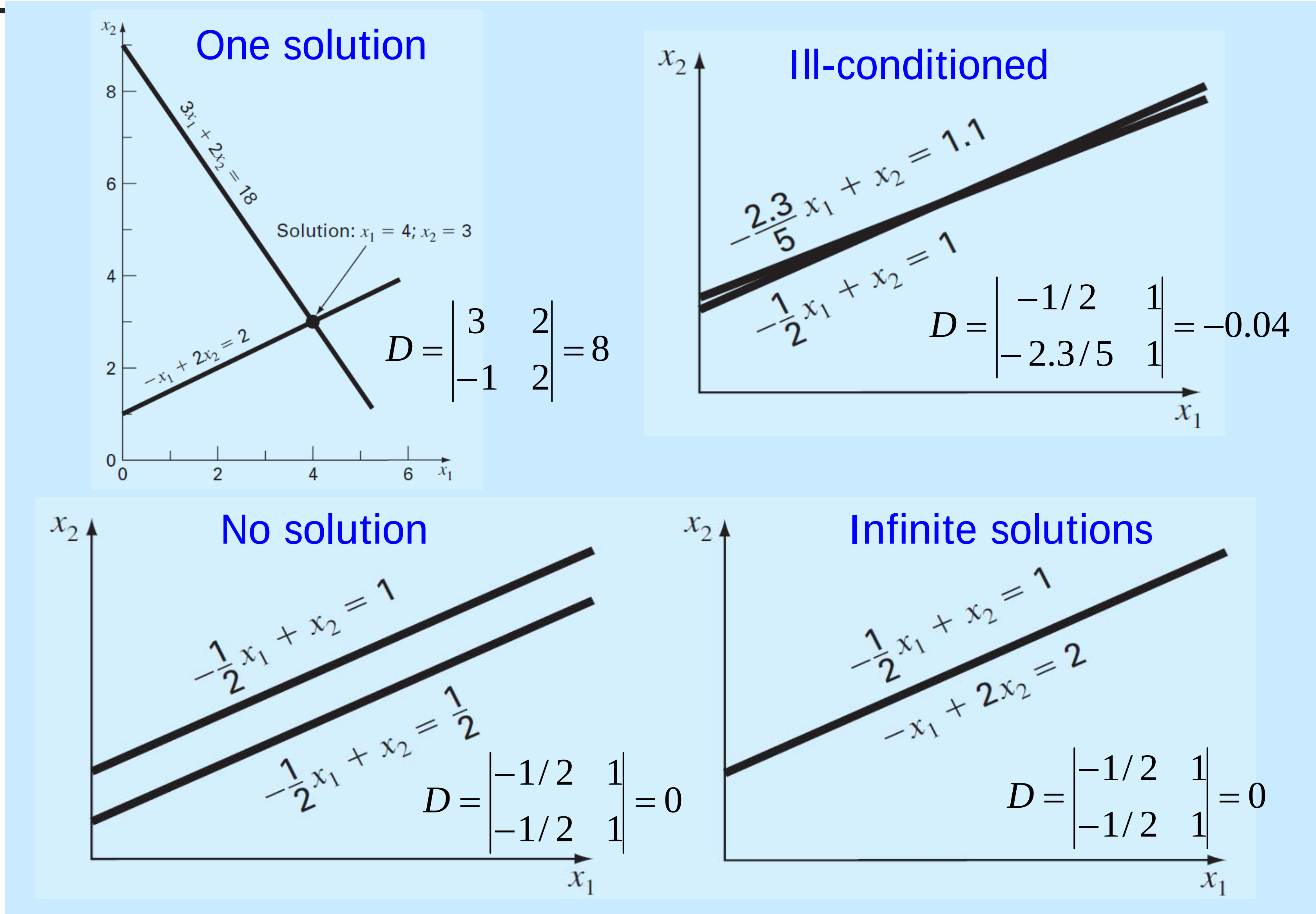
Ill-conditioned system where the slopes are so close that the point of intersection is difficult to detect visually

Determinants

D=|A|

1×1	$\left\|a_{11}\right\| = a_{11}$
2×2	$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$
3×3	$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$

Determinants in linear equations



Cramer's Rule

- Each unknown in a system of linear algebraic equations may be expressed as a fraction of two determinants

$$x_1 = \frac{\begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}}{D}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

```
A=[0.3 0.52 1; 0.5 1 1.9; 0.1 0.3 0.5];
D=det(A);
A(:,1)=[-0.01; 0.67; -0.44];
x1=det(A)/D;
```

Naive Gauss elimination

- A sequential process of removing unknowns from equations
- ‘Naive’ means the process does not check for division-by-zero

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

$$\downarrow$$

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ & a'_{22} & a'_{23} & b'_2 \\ & & a''_{33} & b''_3 \end{array}\right]$$

(*a*) Forward elimination

$$\downarrow$$

$$x_3 = b''_3/a''_{33}$$

$$x_2 = (b'_2 - a'_{23}x_3)/a'_{22}$$

$$x_1 = (b_1 - a_{13}x_3 - a_{12}x_2)/a_{11}$$

(*b*) Back substitution

Forward elimination of unknown

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2$$

$$\vdots \qquad \vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n$$

multiplied by a_{21}/a_{11}

$$a_{21}x_1 + \frac{a_{21}}{a_{11}}a_{12}x_2 + \frac{a_{21}}{a_{11}}a_{13}x_3 + \cdots + \frac{a_{21}}{a_{11}}a_{1n}x_n = \frac{a_{21}}{a_{11}}b_1$$

subtracted from 2nd equation

$$\left(a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right)x_2 + \cdots + \left(a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}\right)x_n = b_2 - \frac{a_{21}}{a_{11}}b_1$$

$$a'_{22}x_2 + \cdots + a'_{2n}x_n = b'_2$$

continuing n-2 eliminations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\ a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n &= b'_2 \\ a''_{33}x_3 + \cdots + a''_{3n}x_n &= b''_3 \\ \ddots \qquad & \vdots \\ a_{nn}^{(n-1)}x_n &= b_n^{(n-1)} \end{aligned}$$

backward substitution

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1 \\
a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n &= b'_2 \\
a''_{33}x_3 + \cdots + a''_{3n}x_n &= b''_3 \\
\ddots \qquad\qquad & \ \vdots \\
a_{nn}^{(n-1)}x_n &= b_n^{(n-1)}
\end{aligned}
$$

$$
x_n = \frac{b_n^{(n-1)}}{a_{nn}^{(n-1)}}
$$

$$
x_i = \frac{b_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}} \qquad \text{for } i = n-1, n-2, \ldots, 1
$$

Example of Gauss elimination

Forward elimination

$$\begin{aligned} 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 \\ 0.1x_1 + 7x_2 - 0.3x_3 &= -19.3 \\ 0.3x_1 - 0.2x_2 + 10x_3 &= 71.4 \end{aligned}$$

f_{21}=0.1/3

f_{31}=0.3/3

$$\begin{aligned} 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 \\ 7.00333x_2 - 0.293333x_3 &= -19.5617 \\ -0.190000x_2 + 10.0200x_3 &= 70.6150 \end{aligned}$$

f_{32}=-0.19/7.00333

$$\begin{aligned} 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 \\ 7.00333x_2 - 0.293333x_3 &= -19.5617 \\ 10.0120x_3 &= 70.0843 \end{aligned}$$

Backward substitution

$$x_3 = \frac{70.0843}{10.0120} = 7.00003 \qquad x_2 = \frac{-19.5617 + 0.293333(7.00003)}{7.00333} = -2.50000$$

$$x_1 = \frac{7.85 + 0.1(-2.50000) + 0.2(7.00003)}{3} = 3.00000$$

M-file to implement naive Gauss elimination

```
function x = GaussNaive(A,b)
[m,n] = size(A);
if m~=n, error('Matrix A must be square'); end
nb = n+1;
Aug = [A b];
% forward elimination
for k=1:n-1
   for i=k+1:n
      factor=Aug(i,k)/Aug(k,k);
      Aug(i,k:nb)=Aug(i,k:nb)-factor*Aug(k,k:nb);
   end
end
```

GaussNaive M-file (cont.)

```
% back substitution
x = zeros(n,1);
x(n) = Aug(n,nb)/Aug(n,n);
for i = n-1:-1:1
   x(i) = (Aug(i,nb)-Aug(i,i+1:n)*x(i+1:n))/Aug(i,i);
end
```

Program efficiency

- Execution time depends on the amount of floating-point operations (flops).

$$\sum_{i=1}^{m} 1 = 1 + 1 + 1 + \cdots + 1 = m \qquad \sum_{i=k}^{m} 1 = m - k + 1$$

$$\sum_{i=1}^{m} i = 1 + 2 + 3 + \cdots + m = \frac{m(m+1)}{2} = \frac{m^2}{2} + O(m)$$

$$\sum_{i=1}^{m} i^2 = 1^2 + 2^2 + 3^2 + \cdots + m^2 = \frac{m(m+1)(2m+1)}{6} = \frac{m^3}{3} + O(m^2)$$

Program efficiency of Gauss elimination

- Forward elimination

```
for k=1:n-1
  for i=k+1:n
    factor=Aug(i,k)/Aug(k,k);
    Aug(i,k:nb)=Aug(i,k:nb)-factor*Aug(k,k:nb);
  end
end
```

Outer Loop k	Inner Loop i	Addition/Subtraction Flops	Multiplication/Division Flops
1	$2, n$	$(n-1)(n)$	$(n-1)(n+1)$
2	$3, n$	$(n-2)(n-1)$	$(n-2)(n)$
⋮	⋮		
k	$k+1, n$	$(n-k)(n+1-k)$	$(n-k)(n+2-k)$
⋮	⋮		
$n-1$	n, n	$(1)(2)$	$(1)(3)$

Program efficiency of Gauss elimination (cont.)

- Total addition/subtraction flops of forward elimination

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

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

$$\Rightarrow\ [n^3+O(n^2)]-[n^3+O(n^2)]+\left[\frac{1}{3}n^3+O(n^2)\right]=\frac{n^3}{3}+O(n^2)$$

- Similar analysis for multiplication/division flops

$$[n^3+O(n^2)]-[n^3+O(n)]+\left[\frac{1}{3}n^3+O(n^2)\right]=\frac{n^3}{3}+O(n^2)$$

- Total $\dfrac{2n^3}{3}+O(n^2)$

Program efficiency of Gauss elimination (cont.)

$$\underbrace{\frac{2n^3}{3} + O(n^2)}_{\substack{\text{Forward} \\ \text{elimination}}} + \underbrace{n^2 + O(n)}_{\substack{\text{Back} \\ \text{substitution}}} \xrightarrow{\text{as } n \text{ increases}} \frac{2n^3}{3} + O(n^2)$$

TABLE 9.1 Number of flops for naive Gauss elimination.

n	Elimination	Back Substitution	Total Flops	$2n^3/3$	Percent Due to Elimination
10	705	100	805	667	87.58%
100	671550	10000	681550	666667	98.53%
1000	6.67×10^8	1×10^6	6.68×10^8	6.67×10^8	99.85%

Problems arise with naive Gauss elimination

- If a coefficient along the diagonal is 0 (division by 0) or close to 0 (round-off error)

$$\begin{aligned} 2x_2 + 3x_3 &= 8 \\ 4x_1 + 6x_2 + 7x_3 &= -3 \\ 2x_1 - 3x_2 + 6x_3 &= 5 \end{aligned}$$

$$\begin{aligned} 0.0003x_1 + 3.0000x_2 &= 2.0001 \\ 1.0000x_1 + 1.0000x_2 &= 1.0000 \end{aligned}$$

↓ Gauss elimination

$$-9999x_2 = -6666$$

$$x_1 = \frac{2.0001 - 3(2/3)}{0.0003}$$

Significant Figures	x_2	x_1	Absolute Value of Percent Relative Error for x_1
3	0.667	−3.33	1099
4	0.6667	0.0000	100
5	0.66667	0.30000	10
6	0.666667	0.330000	1
7	0.6666667	0.3330000	0.1

Pivoting

- **Partial pivoting :** Determine the coefficient with the largest absolute value in the column below the pivot element. The rows can then be switched so that the largest element is the pivot element.
- **Complete pivoting :** If the rows to the right of the pivot element are also checked and columns switched.

Example by partial pivoting

$$0.0003x_1 + 3.0000x_2 = 2.0001$$
$$1.0000x_1 + 1.0000x_2 = 1.0000$$

↓ Partial pivoting

$$1.0000x_1 + 1.0000x_2 = 1.0000$$
$$0.0003x_1 + 3.0000x_2 = 2.0001$$

↓ Gauss elimination

$$x_2 = 2/3$$
$$x_1 = \frac{1 - (2/3)}{1}$$

Significant Figures	x_2	x_1	Absolute Value of Percent Relative Error for x_1
3	0.667	0.333	0.1
4	0.6667	0.3333	0.01
5	0.66667	0.33333	0.001
6	0.666667	0.333333	0.0001
7	0.6666667	0.3333333	0.0000

M-file to implement partial pivoting

```
function x = GaussPivot(A,b)
[m,n]=size(A);
if m~=n, error('Matrix A must be square'); end
nb=n+1;
Aug=[A b];

% forward elimination
for k = 1:n-1
  % partial pivoting
  [big,i]=max(abs(Aug(k:n,k)));
  ipr=i+k-1;
  if ipr~=k  % rows exchange
    Aug([k,ipr],:)=Aug([ipr,k],:);
  end
```

Partial pivoting M-file (cont.)

```
   for i = k+1:n
      factor=Aug(i,k)/Aug(k,k);
      Aug(i,k:nb)=Aug(i,k:nb)-factor*Aug(k,k:nb);
    end
end

% back substitution
x=zeros(n,1);
x(n)=Aug(n,nb)/Aug(n,n);
for i = n-1:-1:1
   x(i)=(Aug(i,nb)-Aug(i,i+1:n)*x(i+1:n))/Aug(i,i);
end
```

Tridiagonal systems

- A banded system with a bandwidth of 3

$$\begin{bmatrix} f_1 & g_1 & & & & & & \\ e_2 & f_2 & g_2 & & & & & \\ & e_3 & f_3 & g_3 & & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & \\ & & & & & e_{n-1} & f_{n-1} & g_{n-1} \\ & & & & & & e_n & f_n \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \end{Bmatrix} = \begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ \cdot \\ r_{n-1} \\ r_n \end{Bmatrix}$$

- Solved using the same method as Gauss elimination but with much less effort

M-file for tridiagonal system solver

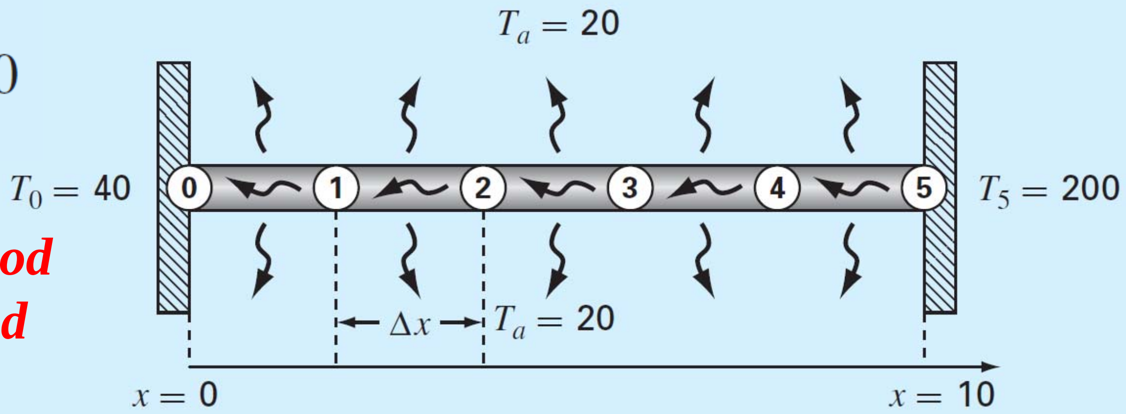
```
function x = Tridiag(e,f,g,r)
% e, f, g = subdiagonal, diagonal &  superdiagonal vectors
n=length(f);
% forward elimination
for k=2:n
  factor = e(k)/f(k-1);
  f(k) = f(k) - factor*g(k-1);
  r(k) = r(k) - factor*r(k-1);
end
% back substitution
x(n) = r(n)/f(n);
for k = n-1:-1:1
   x(k) = (r(k)-g(k)*x(k+1))/f(k);
end
```

A heated rod

- Steady-state, differential equation based on heat conservation

$$\frac{d^2T}{dx^2} + h'(T_a - T) = 0$$

Heat flows through the rod as well as between the rod and the surrounding air



- Finite-difference approximation

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + h'(T_a - T_i) = 0$$

A heated rod (cont.)

Given $h' = 0.01$, $T_a=20$, $T(0)=40$, $T(10) = 200$, get a solution

$$\frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} + h'(T_a - T_i) = 0$$

Using $\Delta x=2$

$$-T_{i-1} + 2.04T_i - T_{i+1} = 0.8$$

$$\begin{aligned} -T_0 + 2.04T_1 - T_2 &= 0.8 \\ -T_1 + 2.04T_2 - T_3 &= 0.8 \\ -T_2 + 2.04T_3 - T_4 &= 0.8 \\ -T_3 + 2.04T_4 - T_5 &= 0.8 \end{aligned}$$

$T_0=40$
$T_5=200$

$$\begin{bmatrix} 2.04 & -1 & 0 & 0 \\ -1 & 2.04 & -1 & 0 \\ 0 & -1 & 2.04 & -1 \\ 0 & 0 & -1 & 2.04 \end{bmatrix} \begin{Bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{Bmatrix} = \begin{Bmatrix} 40.8 \\ 0.8 \\ 0.8 \\ 200.8 \end{Bmatrix}$$

A heated rod (cont.)

```
% Using left division
A=[2.04 -1 0 0; -1 2.04 -1 0; 0 -1 2.04 -1; 0 0 -1 2.04];
b=[40.8 0.8 0.8 200.8]';
T=(A\b)';

% Using tridiagonal system solver
e=[0 -1 -1 -1];
f=[2.04 2.04 2.04 2.04];
g=[-1 -1 -1 0];
r=[40.8 0.8 0.8 200.8];
T=Tridiag(e,f,g,r);
```

Reference

- Steven C. Chapra "Applied Numerical Methods with MATLAB", 3rd ed., McGraw Hill, 2012.