Solving Linear Systems of Equations

Prof. Christian Wallraven
wallraven@korea.ac.kr

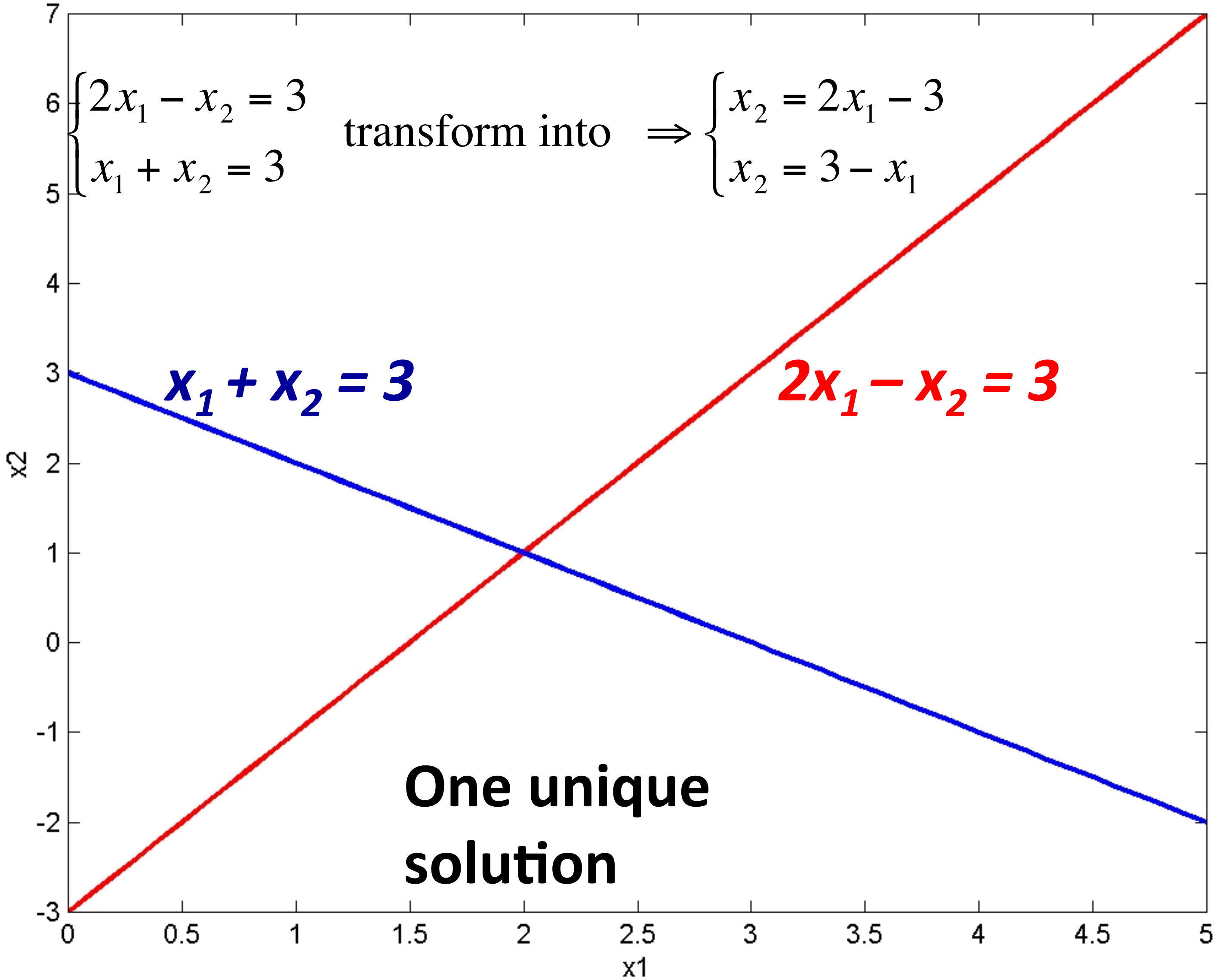
Small Matrices

CognitiveSystems

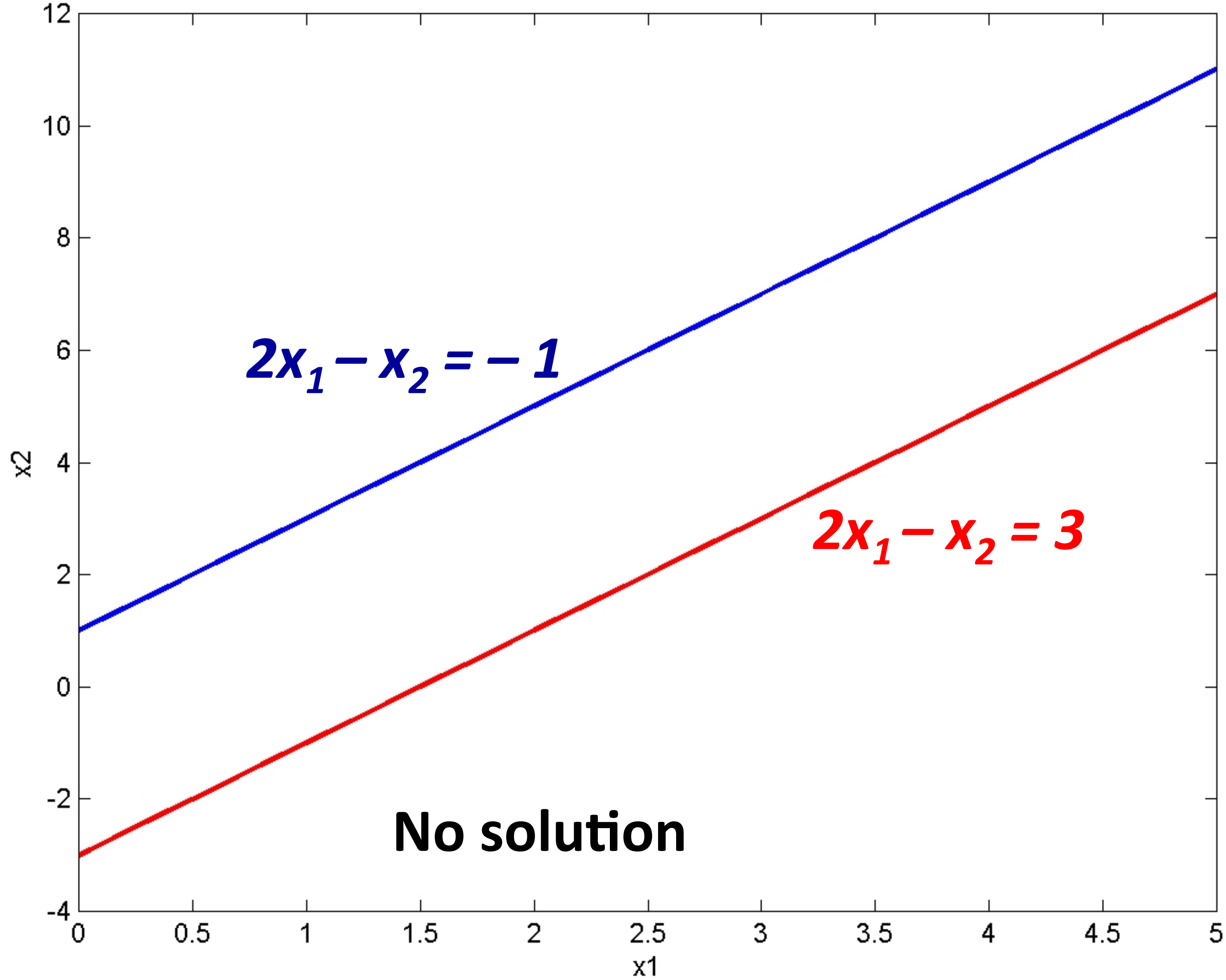
$$\begin{cases} a_{11}x_1 & + a_{12}x_2 & + \cdots & + a_{1n}x_n & = b_1 \\ a_{21}x_1 & + a_{22}x_2 & + \cdots & + a_{2n}x_n & = b_2 \\ \vdots & & & & \vdots \\ a_{n1}x_1 & + a_{n2}x_2 & + \cdots & + a_{nn}x_n & = b_n \end{cases}$$

- Such systems are called **linear systems of equations**
  - linear = linear in x!, or Ax = b
- Three solution methods
  - Graphical solution (only for small 2x2, or 3x3 systems)
  - Cramer's rule (only practical for small systems)
  - Gaussian Elimination (general method)

Graphical Method

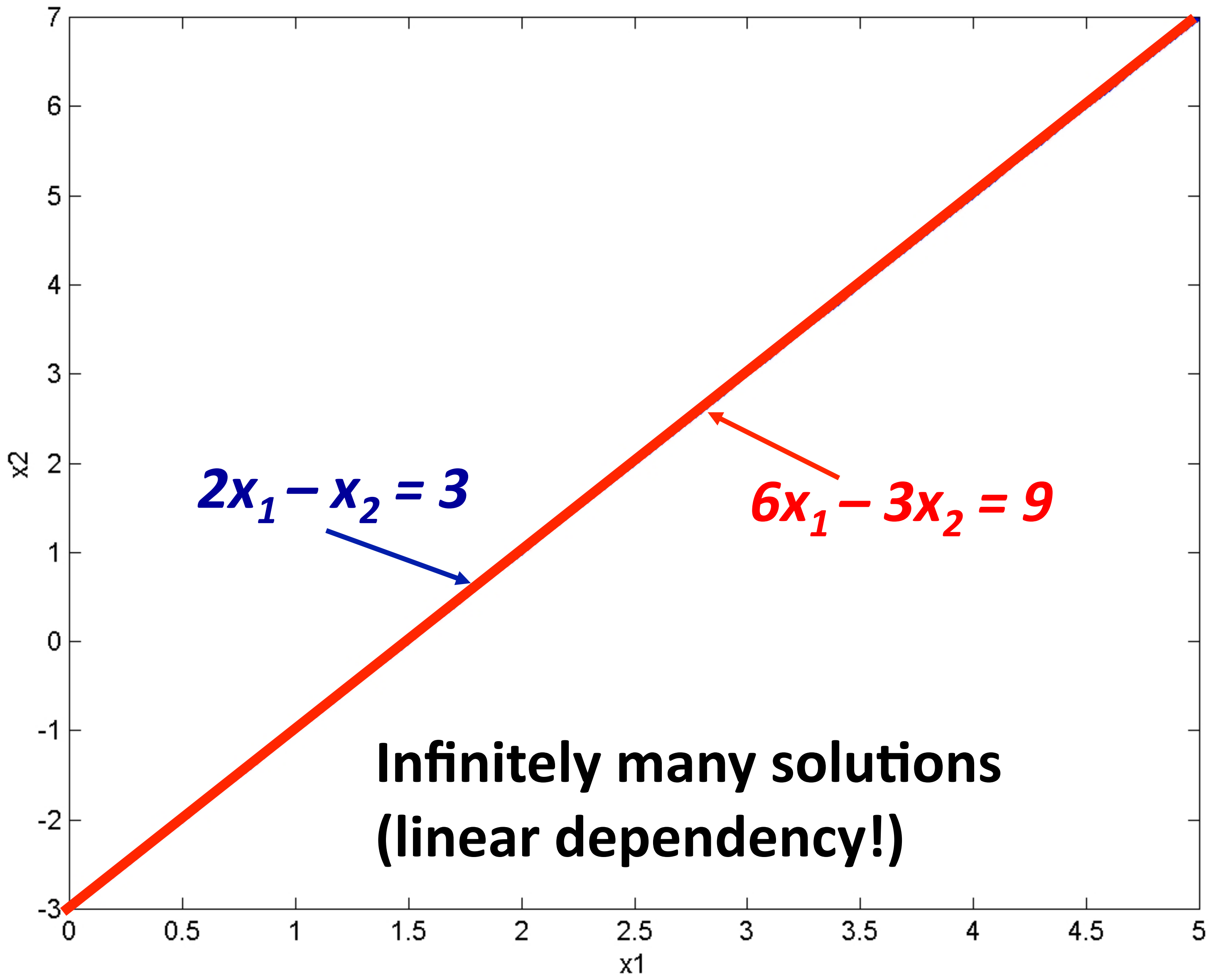


Graphical Method



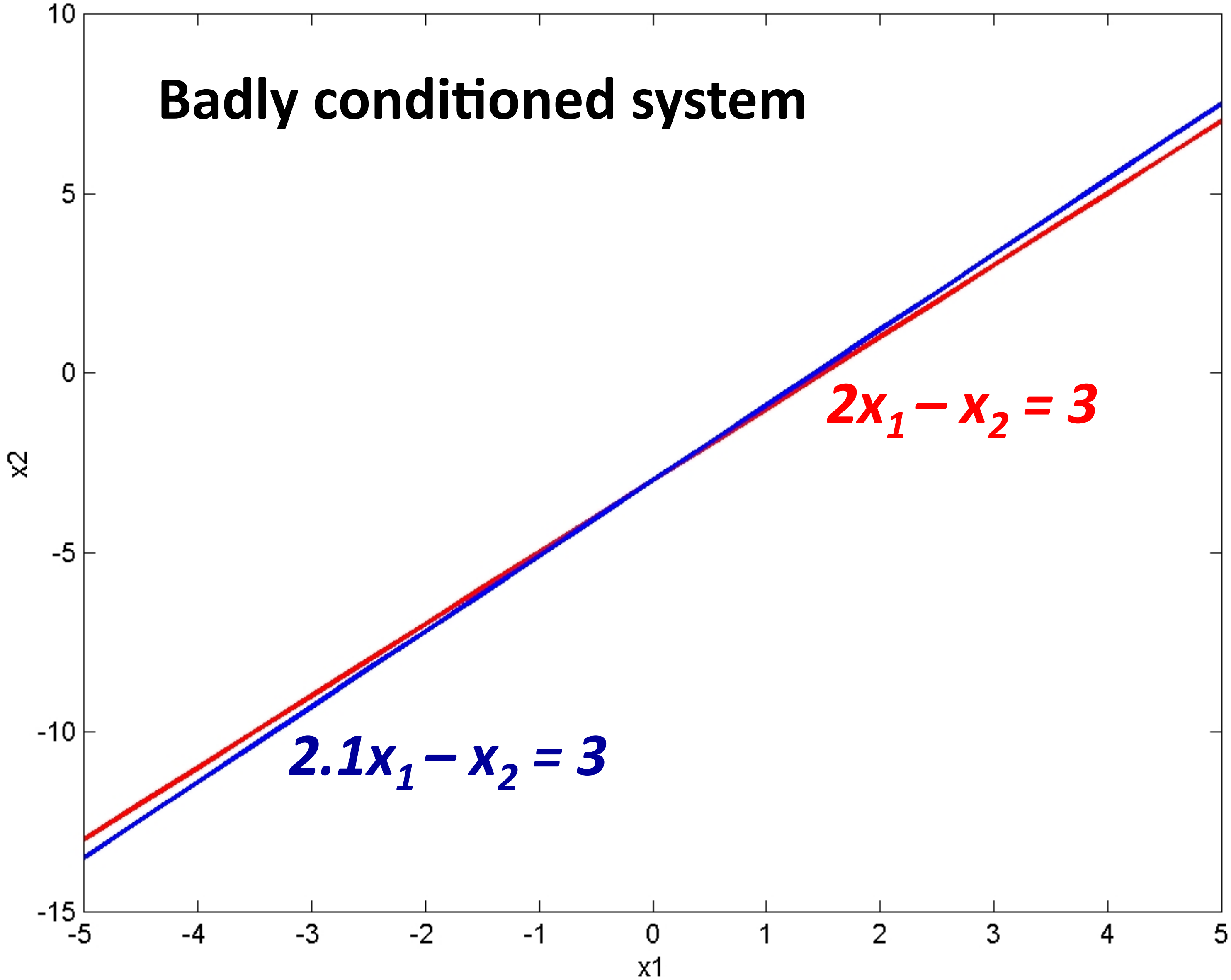
Graphical Method

CognitiveSystems



Graphical Method

CognitiveSystems



Cramer's Rule

CognitiveSystems

- Use the determinant D of the matrices
- 2 x 2 matrix

$$D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

- 3 x 3 matrix

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$= a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

Cramer's Rule

CognitiveSystems

- To find x_k for the following system

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &= b_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n
\end{aligned}
$$

- Make a new matrix, where k^{th} column of a′s is simply replaced with b′s (i.e., $a_{ik} \leftarrow b_i$)
- Then, the solutions x_k are given by: $x_k = \dfrac{D(new\ matrix)}{\mathrm{D}(\mathrm{a}_{\mathrm{ij}})}$

Example

- 3 x 3 matrix

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

$$x_1 = \frac{D_1}{D} = \frac{1}{D}\begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}$$

$$x_2 = \frac{D_2}{D} = \frac{1}{D}\begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{vmatrix}$$

$$x_3 = \frac{D_3}{D} = \frac{1}{D}\begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix}$$

Ill-Conditioned System

- What happens if the determinant D is very small or zero?
  - **D = det(A) ~ 0**
- Either there will be a **division by zero**
  - this happens, if the system is **linearly dependent**
- Or we will divide by a small number resulting **in numerical instabilities**

Simple Elimination Method

CognitiveSystems

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$$

Eliminate $x_2 \Rightarrow$ $$\begin{cases} a_{22}a_{11}x_1 + a_{22}a_{12}x_2 = a_{22}b_1 \\ a_{12}a_{21}x_1 + a_{12}a_{22}x_2 = a_{12}b_2 \end{cases}$$

Subtract to get

$$a_{22}a_{11}x_1 - a_{12}a_{21}x_1 = a_{22}b_1 - a_{12}b_2$$

$$x_1 = \frac{a_{22}b_1 - a_{12}b_2}{a_{22}a_{11} - a_{12}a_{21}} \Rightarrow x_2 = \frac{a_{11}b_2 - a_{21}b_1}{a_{11}a_{22} - a_{12}a_{21}}$$

Not very practical for large number (> 4) of equations

Gauss Elimination

CognitiveSystems

- Manipulate equations to eliminate one of the unknowns
- Develop algorithm to do this recursively
- At the end, we will get an **upper triangular matrix**

$$U = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ & a_{22} & a_{23} & \cdots & a_{2n} \\ & & a_{33} & \cdots & a_{3n} \\ & & & \ddots & \vdots \\ & & & & a_{nn} \end{bmatrix}$$

- From this, we can easily find solution by back substitution

Naive Gauss Elimination

- Direct method (no iteration required)
- Consists of the following steps
  - Forward elimination
  - Column-by-column elimination of the below-diagonal elements
  - Reduce to upper triangular matrix
  - Back-substitution

Naive Gauss Elimination

- Using

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n$$

- Multiply the first equation by a_{21} / a_{11} and subtract from second equation

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{n}x_n = b_1$$
$$\cancel{\left(a_{21} - \frac{a_{21}}{a_{11}}a_{11}\right)}x_1 + \left(a_{22} - \frac{a_{21}}{a_{11}}a_{12}\right)x_2 + \ldots + \left(a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}\right)x_n = b_2 - \frac{a_{21}}{a_{11}}b_1$$
$$\vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n$$

Gauss Elimination

- This will give:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a'_{22}x_2 + \ldots + a'_{2n}x_n &= b'_2 \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n &= b_n
\end{aligned}
$$

- Repeat this “forward elimination” for **every row** until:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= b_1 \\
a'_{22}x_2 + \ldots + a'_{2n}x_n &= b'_2 \\
&\vdots \\
a'_{n2}x_2 + \ldots + a'_{nn}x_n &= b'_n
\end{aligned}
$$

Gauss Elimination

CognitiveSystems

- First equation is **pivot equation**
- a_{11} is **pivot element**
- Now multiply **second** equation by a'_{32} / a'_{22} and subtract from **third** equation

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a'_{22}x_2 + a'_{23}x_3 + \ldots + a'_{2n}x_n &= b'_2 \\
\left(a'_{33} - \frac{a'_{32}}{a'_{22}}a'_{23}\right)x_3 + \ldots + \left(a'_{3n} - \frac{a'_{32}}{a'_{22}}a'_{2n}\right)x_n &= \left(b'_3 - \frac{a'_{32}}{a'_{22}}b'_2\right) \\
&\vdots
\end{aligned}
$$

Gauss Elimination

CognitiveSystems

- Repeat the elimination of a'_{i2} and get

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n &= b_1 \\
a'_{22}x_2 + a'_{23}x_3 + \ldots + a'_{2n}x_n &= b'_2 \\
a''_{33}x_3 + \ldots + a''_{3n}x_n &= b''_3 \\
&\vdots \\
a''_{n3}x_3 + \ldots + a''_{nn}x_n &= b''_n
\end{aligned}
$$

- Continue and get

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

Back Substitution

CognitiveSystems

- Now we can perform back substitution to get x_k
- By simple division

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

- Substitute this into $(n-1)^{th}$ equation

$$a_{n-1,n-1}^{(n-2)} x_{n-1} + a_{n-1,n}^{(n-2)} x_n = b_{n-1}^{(n-2)}$$

- Solve for x_{n-1}

Back Substitution

- Back substitution: start with x_n
- Repeat the process to solve for x_{n-2} , x_{n-3} , ... x_2, x_1

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

$$x_i = \frac{b_i^{(i-1)} - \sum_{j=i+1}^{n} a_{ij}^{(i-1)} x_j}{a_{ii}^{(i-1)}}$$

for i = n-1, n-2, ..., 1

$$a_{ii}^{(i-1)} \neq 0$$

Elimination of first column

CognitiveSystems

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ a_{21} & a_{22} & a_{23} & a_{24} & b_2 \\ a_{31} & a_{32} & a_{33} & a_{34} & b_3 \\ a_{41} & a_{42} & a_{43} & a_{44} & b_4 \end{array}\right]$$

$$f_{21} = a_{21} / a_{11}$$
$$f_{31} = a_{31} / a_{11}$$
$$f_{41} = a_{41} / a_{11}$$

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a'_{22} & a'_{23} & a'_{24} & b'_2 \\ 0 & a'_{32} & a'_{33} & a'_{34} & b'_3 \\ 0 & a'_{42} & a'_{43} & a'_{44} & b'_4 \end{array}\right]$$

$$(2) - f_{21} \times (1)$$
$$(3) - f_{31} \times (1)$$
$$(4) - f_{41} \times (1)$$

Elimination of second column

CognitiveSystems

$$\left[\begin{array}{cccc:c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a'_{22} & a'_{23} & a'_{24} & b'_2 \\ 0 & a'_{32} & a'_{33} & a'_{34} & b'_3 \\ 0 & a'_{42} & a'_{43} & a'_{44} & b'_4 \end{array}\right]$$

$$f_{32} = a'_{32} / a'_{22}$$

$$f_{42} = a'_{42} / a'_{22}$$

$$\left[\begin{array}{cccc:c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ \boxed{0} & a'_{22} & a'_{23} & a'_{24} & b'_2 \\ \boxed{0} & \boxed{0} & a''_{33} & a''_{34} & b''_3 \\ \boxed{0} & \boxed{0} & a''_{43} & a''_{44} & b''_4 \end{array}\right]$$

$$(3) - f_{32} \times (2)$$

$$(4) - f_{42} \times (2)$$

Elimination of third column

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a'_{22} & a'_{23} & a'_{24} & b'_2 \\ 0 & 0 & a''_{33} & a''_{34} & a''_3 \\ 0 & 0 & a''_{43} & a''_{44} & a''_4 \end{array}\right]$$

$$f_{43} = a''_{43} \,/\, a''_{33}$$

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a'_{22} & a'_{23} & a'_{24} & b'_2 \\ 0 & 0 & a''_{33} & a''_{34} & b''_3 \\ 0 & 0 & 0 & a'''_{44} & b'''_4 \end{array}\right]$$

Upper triangular matrix

$$(4) - f_{43} \times (3)$$

Back-Substitution

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & a_{14} & b_1 \\ 0 & a'_{22} & a'_{23} & a'_{24} & b'_2 \\ 0 & 0 & a''_{33} & a''_{34} & b''_3 \\ 0 & 0 & 0 & a'''_{44} & b'''_4 \end{array}\right]$$

Upper triangular matrix

$$x_4 = b'''_4 / a'''_{44}$$

$$x_3 = (b''_3 - a''_{34} x_4) / a''_{33}$$

$$x_2 = (b'_2 - a'_{23} x_3 - a'_{24} x_4) / a'_{22}$$

$$x_1 = (b_1 - a_{12} x_2 - a_{13} x_3 - a_{14} x_4) / a_{11}$$

$$\boxed{a_{11}, a'_{22}, a''_{33}, a'''_{44} \neq 0}$$

Example

$$\left[\begin{array}{cccc|c} 1 & 0 & 2 & 3 & 1 \\ -1 & 2 & 2 & -3 & -1 \\ 0 & 1 & 1 & 4 & 2 \\ 6 & 2 & 2 & 4 & 1 \end{array}\right] \qquad \begin{array}{l} \\ f_{21} = -1 \\ f_{31} = 0 \\ f_{41} = 6 \end{array}$$

$$\left[\begin{array}{cccc|c} 1 & 0 & 2 & 3 & 1 \\ 0 & 2 & 4 & 0 & 0 \\ 0 & 1 & 1 & 4 & 2 \\ 0 & 2 & -10 & -14 & -5 \end{array}\right] \qquad \begin{array}{l} \\ (2) - (1) \times f_{21} \\ (3) - (1) \times f_{31} \\ (4) - (1) \times f_{41} \end{array}$$

Forward Elimination

CognitiveSystems

$$\left[\begin{array}{cccc|c} 1 & 0 & 2 & 3 & 1 \\ 0 & 2 & 4 & 0 & 0 \\ 0 & 1 & 1 & 4 & 2 \\ 0 & 2 & -10 & -14 & -5 \end{array}\right] \qquad \begin{array}{l} \\ \\ f_{32} = 1/2 \\ f_{42} = 1 \end{array}$$

$$\left[\begin{array}{cccc|c} 1 & 0 & 2 & 3 & 1 \\ 0 & 2 & 4 & 0 & 0 \\ 0 & 0 & -1 & 4 & 2 \\ 0 & 0 & -14 & -14 & -5 \end{array}\right] \qquad \begin{array}{l} \\ \\ (3) - (2) \times f_{32} \\ (4) - (2) \times f_{42} \end{array}$$

Upper Triangular Matrix

$$\left[\begin{array}{cccc|c} 1 & 0 & 2 & 3 & 1 \\ 0 & 2 & 4 & 0 & 0 \\ 0 & 0 & -1 & 4 & 2 \\ 0 & 0 & -14 & -14 & -5 \end{array}\right] \qquad f_{43} = 14$$

$$\left[\begin{array}{cccc|c} 1 & 0 & 2 & 3 & 1 \\ 0 & 2 & 4 & 0 & 0 \\ 0 & 0 & -1 & 4 & 2 \\ 0 & 0 & 0 & -70 & -33 \end{array}\right] \qquad (4) - (3) \times f_{43}$$

Back-Substitution

$$\left[\begin{array}{cccc|c} 1 & 0 & 2 & 3 & 1 \\ 0 & 2 & 4 & 0 & 0 \\ 0 & 0 & -1 & 4 & 2 \\ 0 & 0 & 0 & -70 & -33 \end{array}\right]$$

$$x_4 = -33/-70 = 33/70$$

$$x_3 = 4x_4 - 2 = -4/35$$

$$x_2 = -2x_3 = 8/35$$

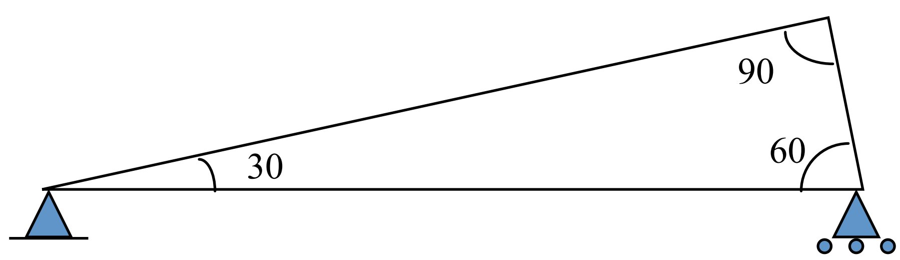
$$x_1 = 1 - 2x_3 - 3x_4 = -13/70$$

$$\vec{x} = \begin{bmatrix} -13/70 \\ 8/35 \\ -4/35 \\ 33/70 \end{bmatrix}$$

Practical application

CognitiveSystems

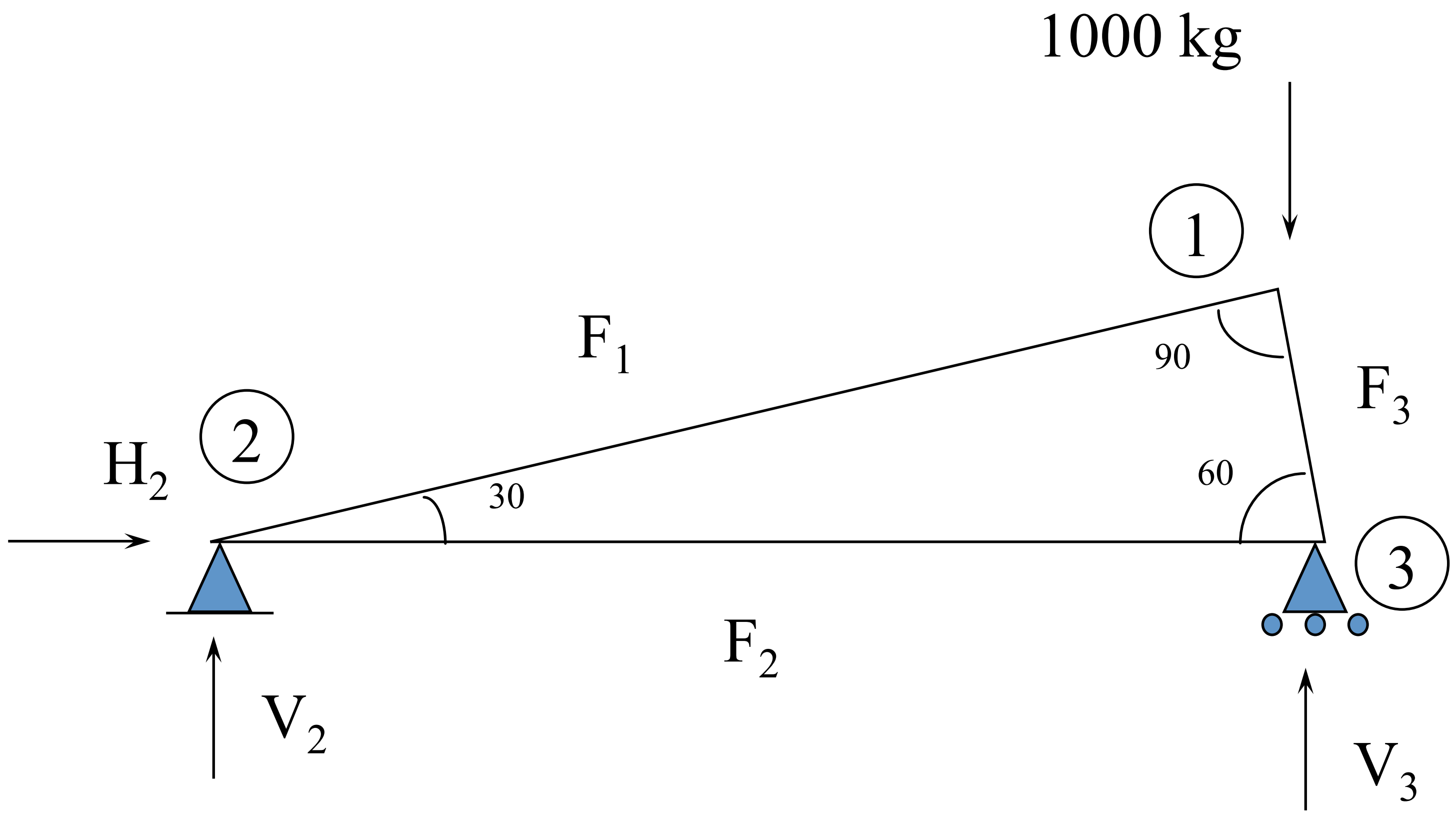
- Consider a problem in structural engineering
- Find the forces and reactions associated with a statically determinant truss



hinge: transmits both vertical and horizontal forces at the surface

roller: transmits vertical forces

Truss – force equilibrium

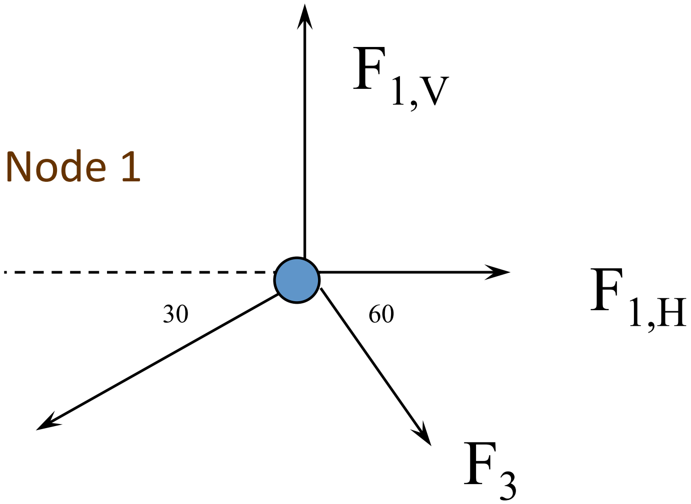


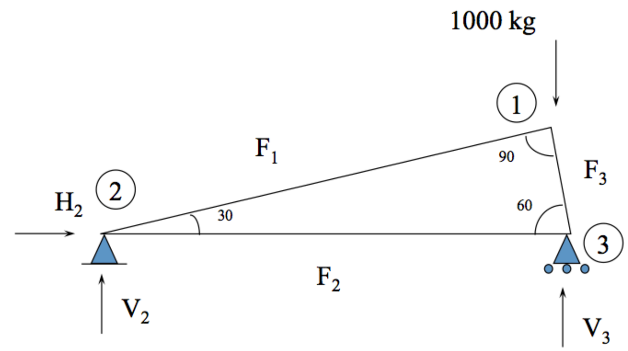
FREE BODY DIAGRAM

$$\sum F_H = 0$$
$$\sum F_v = 0$$

Truss – node 1

Node 1





$$\sum F_H = 0 = -F_1 \cos 30^\circ + F_3 \cos 60^\circ + F_{1,H}$$

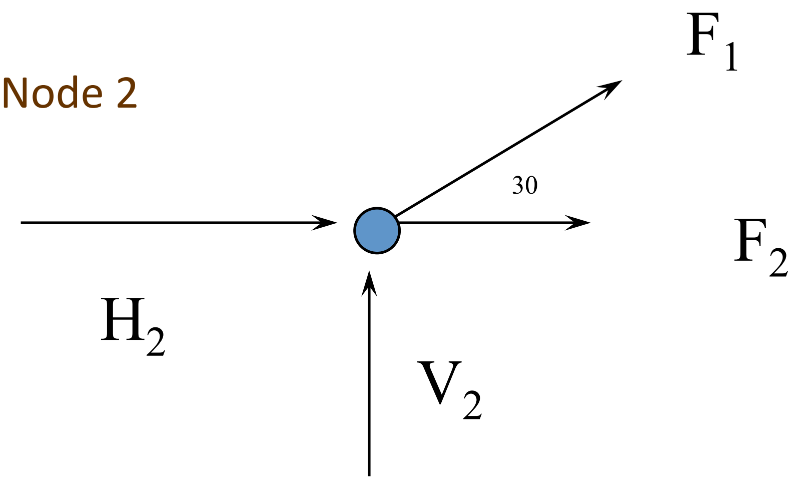
$$\sum F_V = 0 = -F_1 \sin 30^\circ - F_3 \sin 60^\circ + F_{1,V}$$

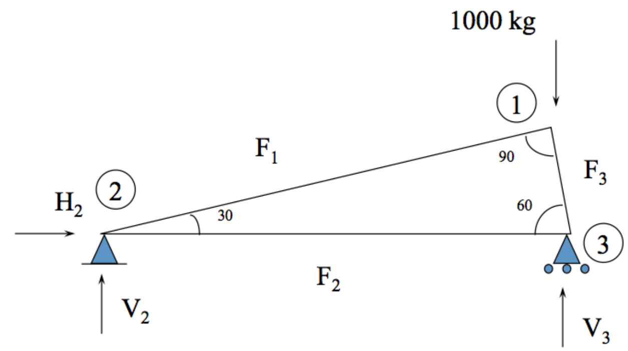
$$-F_1 \cos 30^\circ + F_3 \cos 60^\circ = 0$$

$$-F_1 \sin 30^\circ - F_3 \sin 60^\circ = -1000$$

Truss – node 2

Node 2



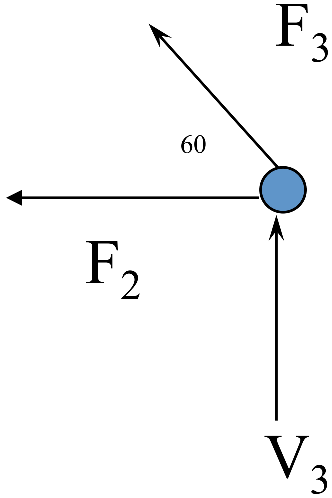


$$\sum F_H = 0 = H_2 + F_2 + F_1 \cos 30^\circ$$

$$\sum F_V = 0 = V_2 + F_1 \sin 30^\circ$$

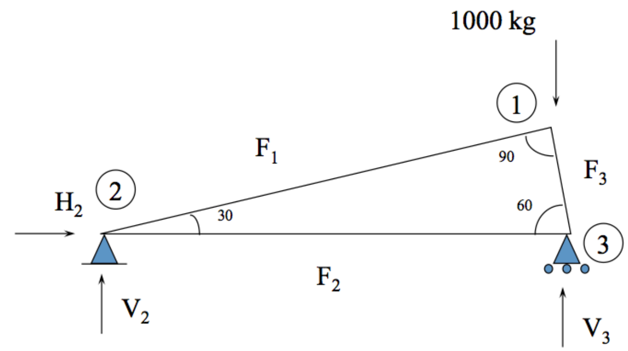
Truss – node 3

Node 3



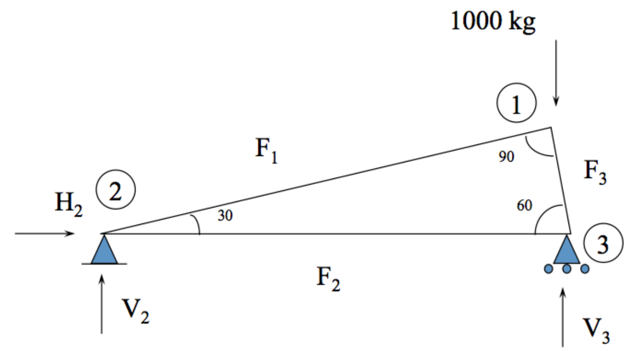
$$\sum F_H = 0 = -F_3 \cos 60^\circ - F_2$$

$$\sum F_V = 0 = F_3 \sin 60^\circ + V_3$$



Truss – all nodes combined

CognitiveSystems



$$-F_1\cos 30^\circ + F_3\cos 60^\circ = 0$$

$$-F_1\sin 30^\circ - F_3\sin 60^\circ = -1000$$

$$H_2 + F_2 + F_1\cos 30^\circ = 0$$

$$V_2 + F_1\sin 30^\circ = 0$$

$$-F_3\cos 60^\circ - F_2 = 0$$

$$F_3\sin 60^\circ + V_3 = 0$$

SIX EQUATIONS

SIX UNKNOWNS

Truss – matrix form

	F_1	F_2	F_3	H_2	V_2	V_3	
1	-cos30	0	cos60	0	0	0	0
2	-sin30	0	-sin60	0	0	0	-1000
3	cos30	1	0	1	0	0	0
4	sin30	0	0	0	1	0	0
5	0	-1	-cos60	0	0	0	0
6	0	0	sin60	0	0	1	0

Truss – matrix form

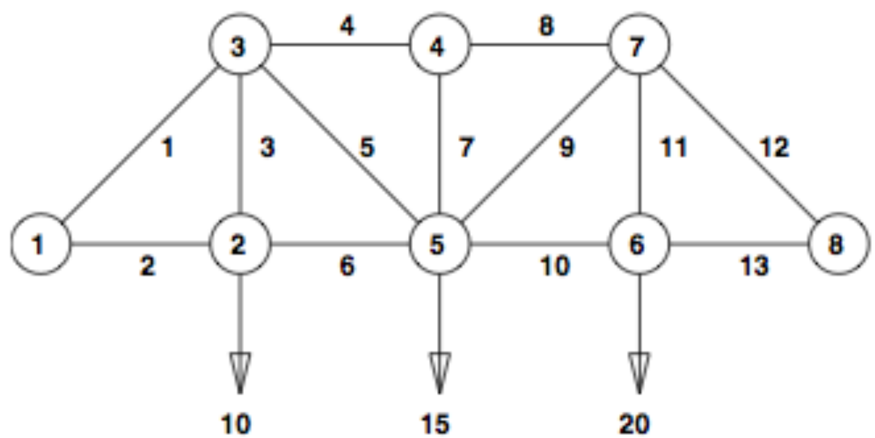
This is the basis for your matrices and the equation [A]{x}={c}

$$\begin{bmatrix} -0.866 & 0 & 0.5 & 0 & 0 & 0 \\ -0.5 & 0 & -0.866 & 0 & 0 & 0 \\ 0.866 & 1 & 0 & 1 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.866 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ H_2 \\ V_2 \\ V_3 \end{Bmatrix} = \begin{Bmatrix} 0 \\ -1000 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

What's the answer? → see homework assignment!

Statics – force equilibrium

- Shown here is a "truss" having 13 members (the numbered lines) connecting 8 joints (the numbered circles). The indicated loads, in tons, are applied at joints 2, 5, and 6, and we want to determine the **resulting force** on each member of the truss.

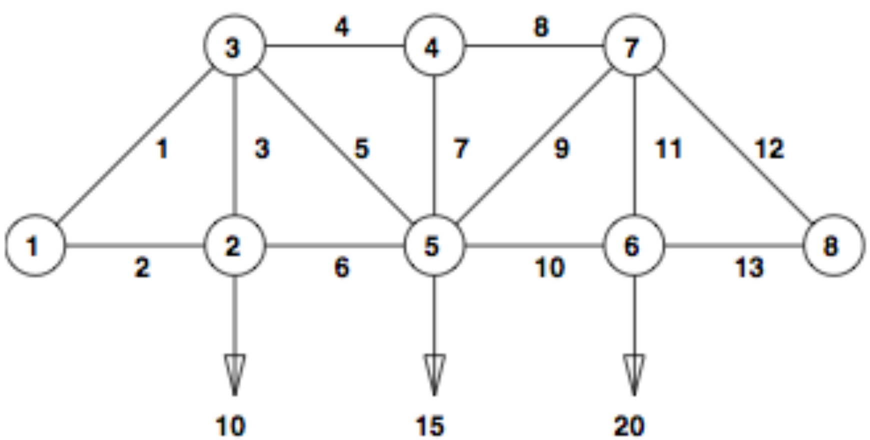


Statics – force equilibrium

- For the truss to be in static equilibrium, there must be **no net force, horizontally or vertically, at any joint**.
- Thus, we can determine the member forces by **equating the horizontal forces** to the left and right at each joint, and similarly **equating the vertical forces** upward and downward at each joint.
- For the eight joints, this would give 16 equations, which is more than the 13 unknown factors to be determined.
  - For the truss to be statically determinate, that is, for there to be a **unique** solution, we assume that joint 1 is rigidly fixed horizontally and vertically and that joint 8 is fixed vertically.

Force equations

- Equating the different forces yields the following system of equations
  - α = 1/sqrt(2) = sin(45deg) = cos(45)



$$
\begin{aligned}
\text{Joint 2:} \quad & f_2 = f_6, \\
& f_3 = 10; \\
\text{Joint 3:} \quad & \alpha f_1 = f_4 + \alpha f_5, \\
& \alpha f_1 + f_3 + \alpha f_5 = 0; \\
\text{Joint 4:} \quad & f_4 = f_8, \\
& f_7 = 0; \\
\text{Joint 5:} \quad & \alpha f_5 + f_6 = \alpha f_9 + f_{10}, \\
& \alpha f_5 + f_7 + \alpha f_9 = 15; \\
\text{Joint 6:} \quad & f_{10} = f_{13}, \\
& f_{11} = 20; \\
\text{Joint 7:} \quad & f_8 + \alpha f_9 = \alpha f_{12}, \\
& \alpha f_9 + f_{11} + \alpha f_{12} = 0; \\
\text{Joint 8:} \quad & f_{13} + \alpha f_{12} = 0.
\end{aligned}
$$

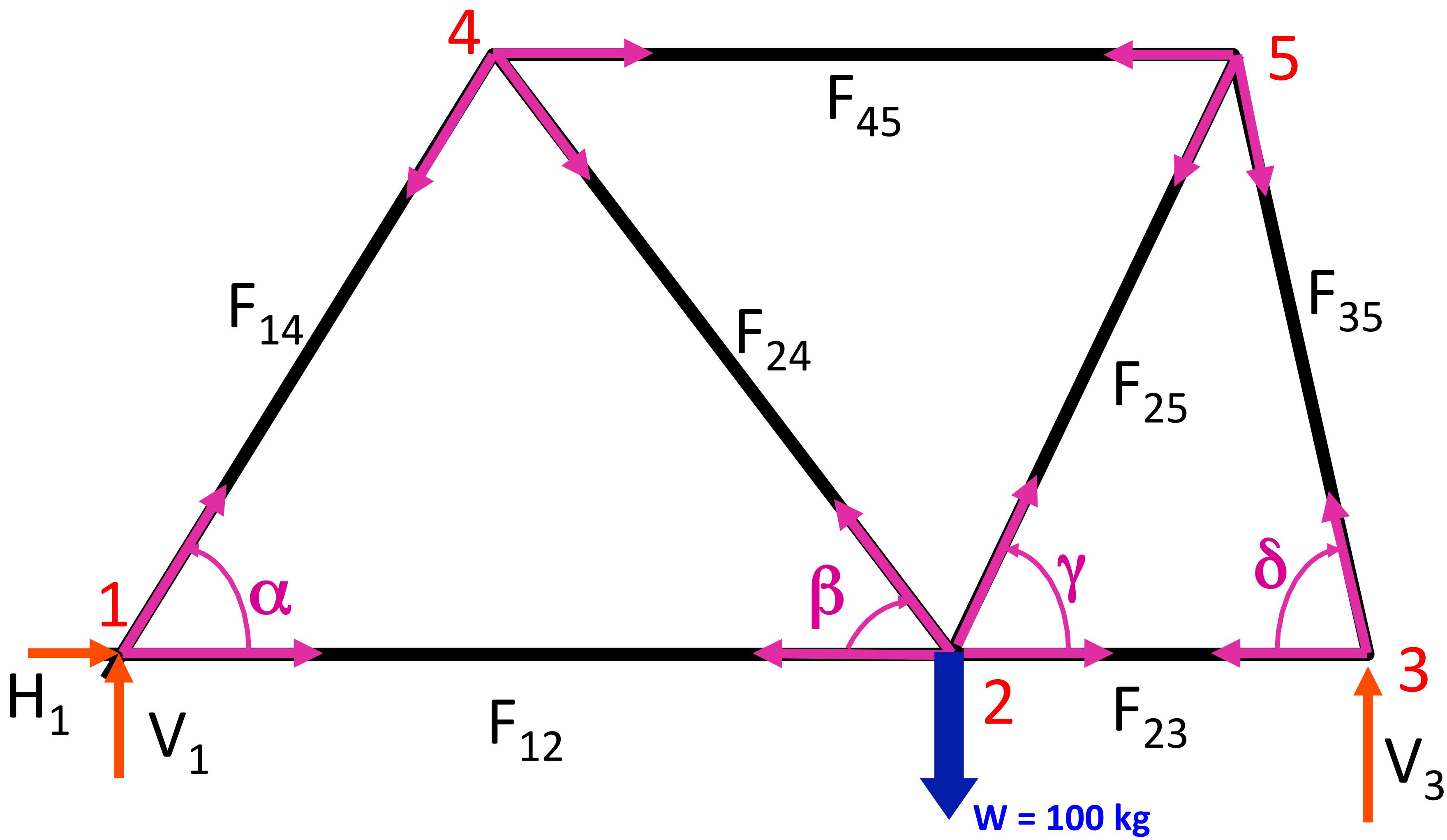
Solution in Matlab


```
% MYTRUSS   Solution to the truss problem.

n = 13;
A = zeros(n,n);
b = zeros(n,1);
alpha = 1/sqrt(2);

% Joint 2: f2 = f6
%          f3 = 10
A(1,2) = 1;
A(1,6) = -1;
A(2,3) = 1;
b(2) = 10;

% Joint 3: alpha f1 = f4 + alpha f5
%          alpha f1 + f3 + alpha f5 = 0
A(3,1) = alpha;
A(3,4) = -1;
A(3,5) = -alpha;
A(4,1) = alpha;
A(4,3) = 1;
A(4,5) = alpha;

% Joint 4: f4 = f8
%          f7 = 0
A(5,4) = 1;
A(5,8) = -1;
A(6,7) = 1;
b(6) = 0;

% Joint 5: alpha f5 + f6 = alpha f9 + f10
%          alpha f5 + f7 + alpha f9 = 15
A(7,5) = alpha;
A(7,6) = 1;
A(7,9) = -alpha;
A(7,10) = -1;
A(8,5) = alpha;
A(8,7) = 1;
A(8,9) = alpha;
b(8) = 15;

% Joint 6: f10 = f13
%          f11 = 20
A(9,10) = 1;
A(9,13) = -1;
A(10,11) = 1;
b(10) = 20;

% Joint 7: f8 + alpha f9 = alpha f12
%          alpha f9 + f11 + alpha f12 = 0
A(11,8) = 1;
A(11,9) = alpha;
A(11,12) = -alpha;
A(12,9) = alpha;
A(12,11) = 1;
A(12,12) = alpha;
b(12) = 0;

% Joint 8: f13 + alpha f12 = 0
A(13,13) = 1;
A(13,12) = alpha;
b(13) = 0;

x = A\b
```

Problem in statics – force equilibrium

CognitiveSystems

- You are tasked with designing a bridge that has 5 joints



Statics: Force Balance

CognitiveSystems

Node 1 $\begin{cases} \sum F_{y,1} = V_1 + F_{14}\sin\alpha = 0 \\ \sum F_{x,1} = H_1 + F_{12} + F_{14}\sin\alpha = 0 \end{cases}$

Node 2 $\begin{cases} \sum F_{y,2} = F_{24}\sin\beta + F_{25}\sin\gamma = 100 \\ \sum F_{x,2} = -F_{12} + F_{23} - F_{24}\cos\beta + F_{25}\cos\gamma = 0 \end{cases}$

Node 3 $\begin{cases} \sum F_{y,3} = V_3 + F_{35}\sin\delta = 0 \\ \sum F_{x,3} = -F_{23} - F_{35}\cos\delta = 0 \end{cases}$

Node 4 $\begin{cases} \sum F_{y,4} = -F_{14}\sin\alpha - F_{24}\sin\beta = 0 \\ \sum F_{x,4} = -F_{14}\cos\alpha + F_{24}\cos\beta + F_{45} = 0 \end{cases}$

Node 5 $\begin{cases} \sum F_{y,5} = -F_{25}\sin\gamma - F_{35}\sin\delta = 0 \\ \sum F_{x,5} = -F_{25}\cos\gamma + F_{35}\cos\delta - F_{45} = 0 \end{cases}$

Example: Forces in a Simple Truss

CognitiveSystems

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & \sin\alpha & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & \cos\alpha & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \sin\beta & \sin\gamma & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 1 & \cos\beta & \cos\gamma & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \sin\delta & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -\cos\delta & 0 \\
0 & 0 & 0 & 0 & -\sin\alpha & 0 & -\sin\beta & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -\cos\alpha & 0 & \cos\beta & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\sin\gamma & \sin\delta & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \cos\gamma & \cos\delta & -1
\end{bmatrix}
\begin{Bmatrix} V_1 \\ H_1 \\ V_3 \\ F_{12} \\ F_{14} \\ F_{23} \\ F_{24} \\ F_{25} \\ F_{35} \\ F_{45} \end{Bmatrix}
=
\begin{Bmatrix} 0 \\ 0 \\ 100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}
$$

Define Matrices A and b in script file

CognitiveSystems

```
function [A,b]=Truss(alpha,beta,gamma,delta)

A=zeros(10,10);
A(1,1)=1; A(1,5)=sin(alpha);
A(2,2)=1; A(2,4)=1; A(2,5)=cos(alpha);
A(3,7)=sin(beta); A(3,8)=sin(gamma);
A(4,4)=-1; A(4,6)=1; A(4,7)=-cos(beta);
A(4,8)=cos(gamma);
A(5,3)=1; A(5,9)=sin(gamma);
A(6,6)=-1; A(6,9)=-cos(delta);
A(7,5)=-sin(alpha); A(7,7)=-sin(beta);
A(8,5)=-cos(alpha); A(8,7)=cos(beta); A(8,10)=1;
A(9,8)=-sin(gamma); A(9,9)=-sin(delta);
A(10,8)=-cos(gamma); A(10,9)=cos(delta); A(10,10)=-1;

b=zeros(10,1); b(3,1)=100;
```

Gauss Elimination with Partial Pivoting

```
>> alpha=pi/6; beta=pi/3; gamma=pi/4; delta=pi/3;
>> [A,b] = Truss(alpha,beta,gamma,delta)
A =
    1.0000         0         0         0    0.5000         0         0         0         0         0
         0    1.0000         0    1.0000    0.8660         0         0         0         0         0
         0         0         0         0         0         0    0.8660    0.7071         0         0
         0         0         0   -1.0000         0    1.0000   -0.5000    0.7071         0         0
         0         0    1.0000         0         0         0         0         0    0.7071         0
         0         0         0         0         0   -1.0000         0         0   -0.5000         0
         0         0         0         0   -0.5000         0   -0.8660         0         0         0
         0         0         0         0   -0.8660         0    0.5000         0         0    1.0000
         0         0         0         0         0         0         0   -0.7071   -0.8660         0
         0         0         0         0         0         0         0   -0.7071    0.5000   -1.0000
b =
     0
     0
   100
     0
     0
     0
     0
     0
     0
     0
>> x = GaussPivot(A,b)
x =
   40.5827
         0
   48.5140
   70.2914
  -81.1655
   34.3046
   46.8609
   84.0287
  -68.6091
  -93.7218
```

Simple truss

MATLAB Script File: GaussNaive

CognitiveSystems

```
function x = GaussNaive(A,b)

% GaussNaive(A,b) :
% Solve Ax =b using Gaussian elimination without pivoting
% Input:
%     A = coefficient matrix
%     b = right-hand-side matrix
%
% Output:
%     x = solution matrix

% compute the matrix sizes
[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

% 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
```

```
>> format short
>> x = GaussNaive(A,b)
m =
     4
n =
     4
Aug =
     1     0     2     3     1
    -1     2     2    -3    -1
     0     1     1     4     2
     6     2     2     4     1
factor =
    -1
Aug =
     1     0     2     3     1
     0     2     4     0     0
     0     1     1     4     2
     6     2     2     4     1
factor =
     0
Aug =
     1     0     2     3     1
     0     2     4     0     0
     0     1     1     4     2
     6     2     2     4     1
factor =
     6
Aug =
     1     0     2     3     1
     0     2     4     0     0
     0     1     1     4     2
     0     2   -10   -14    -5
```

Aug = [A, b]

Eliminate first column

```
factor =
    0.5000
Aug =
     1     0     2     3     1
     0     2     4     0     0
     0     0    -1     4     2
     0     2   -10   -14    -5
factor =
     1
Aug =
     1     0     2     3     1
     0     2     4     0     0
     0     0    -1     4     2
     0     0   -14   -14    -5
```

Eliminate second column

```
factor =
    14
Aug =
     1     0     2     3     1
     0     2     4     0     0
     0     0    -1     4     2
     0     0     0   -70   -33
```

Eliminate third column

```
x =
         0
         0
         0
    0.4714
x =
         0
         0
   -0.1143
    0.4714
x =
         0
    0.2286
   -0.1143
    0.4714
x =
   -0.1857
    0.2286
   -0.1143
    0.4714
```

x_4 x_3 x_2 x_1

Back-substitution

Print all `factor` and `Aug`

(do not suppress output)

Algorithm for Gauss elimination

- **1. Forward elimination**
- **for each equation j, $j = 1$ to $n-1$**

  for all equations k greater than j

  (a) multiply equation j by a_{kj}/a_{jj}

  (b) subtract the result from equation k

- **This leads to an upper triangular matrix**
- **2. Back-Substitution**

  (a) determine x_n from

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

  (b) put x_n into (n-1)th equation, solve for x_{n-1}

  (c) repeat from (b), moving back to n-2, n-3, etc. until all equations are solved

Operation Count

- Important as matrix gets large
- For Gauss elimination
- Elimination routine uses on the order of $O(n^3/3)$ operations
- Back-substitution uses $O(n^2/2)$

Operation Count

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

Total operation counts for elimination stage = $2n^3/3 + O(n^2)$

Total operation counts for back substitution stage = $n^2 + O(n)$

Operation Count

CognitiveSystems

- Number of flops (floating-point operations) for Naive Gauss elimination

n	Elimination	Back Sbustitution	Total Flops	$\frac{2n^3}{3}$	Percentage Due to Elimination
10	705	100	805	667	87.58%
100	671550	10000	681550	666667	98.53%
1000	6.67×10^8	1000000	6.68×10^6	6.67×10^8	99.85%

- Computation time increase rapidly with n
- Most of the effort is incurred in the elimination step
- Improve efficiency by reducing the elimination effort

Partial Pivoting

CognitiveSystems

Problems with Gauss elimination

- division by zero
- round off errors
- ill conditioned systems

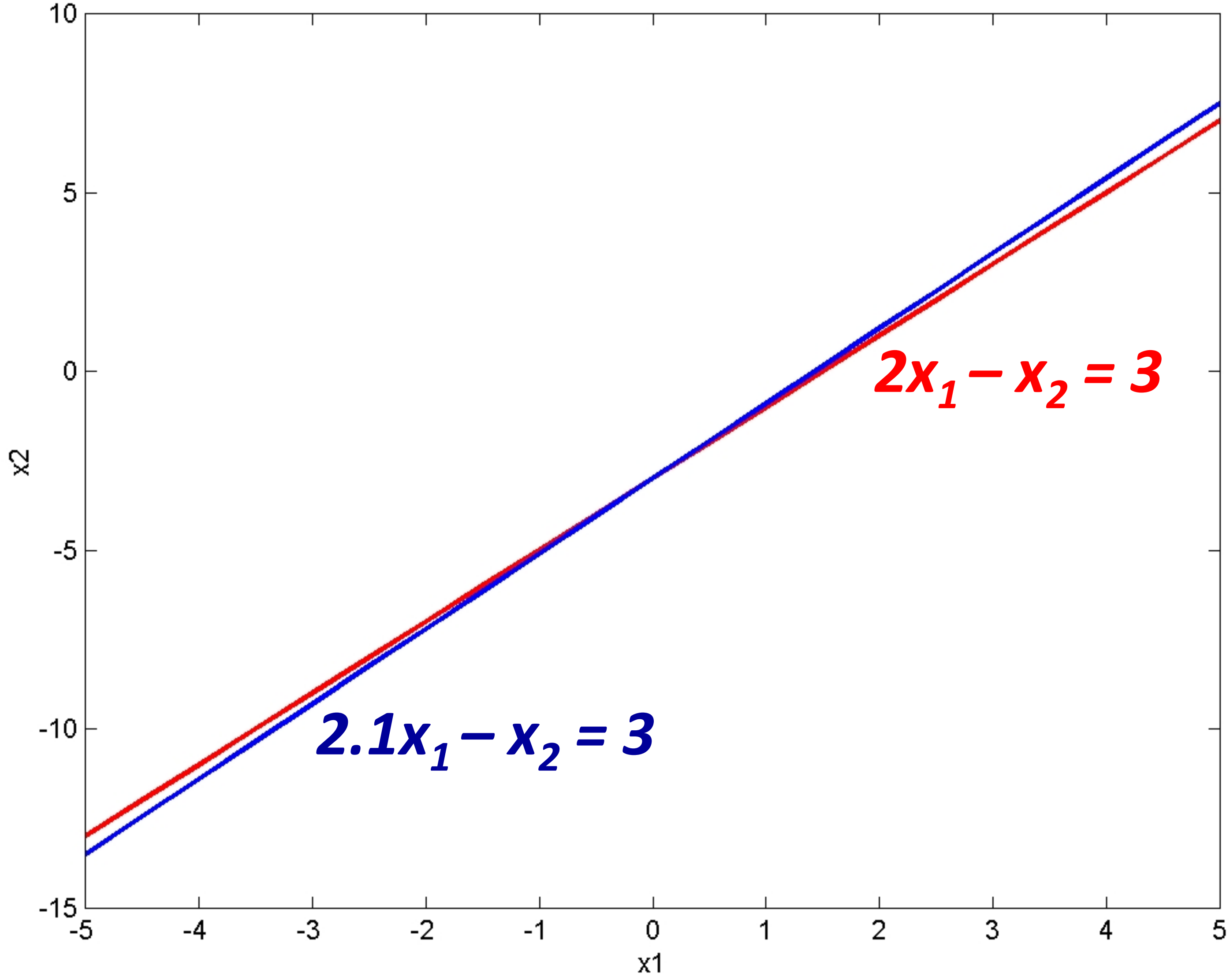
Use “Pivoting” to avoid this

- Find the row with largest absolute coefficient below the pivot element
- Switch rows (“partial pivoting”)
- complete pivoting switch columns also (rarely used)

Round-off Errors

- A lot of chopping with more than $n^3/3$ operations
- More important - error is propagated
- For large systems (more than 100 equations), round-off error can be very important (machine dependent)
- Ill conditioned systems - small changes in coefficients lead to large changes in solution
- Round-off errors are especially important for ill-conditioned systems

Ill-conditioned System



Ill-Conditioned System

CognitiveSystems

- Consider

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases} \Rightarrow \begin{cases} x_2 = -\dfrac{a_{11}}{a_{12}}x_1 + \dfrac{b_1}{a_{12}} \\ x_2 = -\dfrac{a_{21}}{a_{22}}x_1 + \dfrac{b_2}{a_{22}} \end{cases}$$

- Since slopes are almost equal

$$\boxed{\frac{a_{11}}{a_{12}} \cong \frac{a_{21}}{a_{22}}}$$

$$D = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \approx 0$$

Divided by small number

Determinant

CognitiveSystems

- **Calculate determinant using Gauss elimination**

$$U = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ & a'_{22} & a'_{23} & \dots & a'_{2n} \\ & & a''_{33} & \dots & a''_{3n} \\ & & & & \vdots \\ & & & & a_{nn}^{(n-1)} \end{bmatrix}$$

$$\boxed{\det(A) = \det(U) = a_{11} a'_{22} a''_{33} \dots a_{nn}^{(n-1)}}$$

Gauss Elimination with Partial Pivoting

- <u>Forward elimination</u>
- for each equation *j*, $j = 1$ to $n-1$
  - first scale each equation *k* greater than *j*
  - then pivot (switch rows)
  - Now perform the elimination

    (a) multiply equation *j* by a_{kj} / a_{jj}

    (b) subtract the result from equation

Partial (Row) Pivoting

CognitiveSystems

$$\begin{aligned} x_1 \quad\quad\quad & + 2x_3 + 3x_4 = 1 \\ -x_1 + 2x_2 & + 2x_3 - 3x_4 = -1 \\ x_2 & + x_3 + 4x_4 = 2 \\ 6x_1 + 2x_2 & + 2x_3 + 4x_4 = 1 \end{aligned}$$

$$\left[A \,\vdots\, b\right] = \left[\begin{array}{cccc|c} 1 & 0 & 2 & 3 & 1 \\ -1 & 2 & 2 & -3 & -1 \\ 0 & 1 & 1 & 4 & 2 \\ 6 & 2 & 2 & 4 & 1 \end{array}\right]$$

Forward Elimination

CognitiveSystems

$$\left[\begin{array}{cccc|c} 6 & 2 & 2 & 4 & 1 \\ -1 & 2 & 2 & -3 & -1 \\ 0 & 1 & 1 & 4 & 2 \\ 1 & 0 & 2 & 3 & 1 \end{array}\right]$$

Interchange rows 1 & 4

$$f_{21} = -1/6$$

$$f_{31} = 0$$

$$f_{41} = 1/6$$

$$\left[\begin{array}{cccc|c} 6 & 2 & 2 & 4 & 1 \\ \boxed{0} & 7/3 & 7/3 & -7/3 & -5/6 \\ \boxed{0} & 1 & 1 & 4 & 2 \\ \boxed{0} & -1/3 & 5/3 & 7/3 & 5/6 \end{array}\right]$$

$$(2) - (1) \times f_{21}$$

$$(3) - (1) \times f_{31}$$

$$(4) - (1) \times f_{41}$$

Forward Elimination

CognitiveSystems

$$\left[\begin{array}{cccc|c} 6 & 2 & 2 & 4 & 1 \\ 0 & 7/3 & 7/3 & -7/3 & -5/6 \\ 0 & 1 & 1 & 4 & 2 \\ 0 & -1/3 & 5/3 & 7/3 & 5/6 \end{array}\right]$$

No interchange required

$$f_{32} = 3/7$$

$$f_{42} = 1/7$$

$$\left[\begin{array}{cccc|c} 6 & 2 & 2 & 4 & 1 \\ 0 & 7/3 & 7/3 & -7/3 & -5/6 \\ 0 & 0 & 0 & 5 & 33/14 \\ 0 & 0 & 2 & 2 & 5/7 \end{array}\right]$$

$$(3) - (2) \times f_{32}$$

$$(4) - (2) \times f_{42}$$

Back-Substitution

CognitiveSystems

$$\left[\begin{array}{cccc|c} 6 & 2 & 2 & 4 & 1 \\ 0 & 7/3 & 7/3 & -7/3 & -5/6 \\ 0 & 0 & 2 & 2 & 5/7 \\ 0 & 0 & 0 & 5 & 33/14 \end{array}\right]$$

Interchange rows 3 & 4

$$f_{43} = 0$$

$$x_4 = (33/14)/5 = 33/70$$

$$x_3 = (5/7 - 2\,x_4)/2 = -4/35$$

$$x_2 = (-5/6 + 7/3x_4 - 7/3x_3)/(7/3) = 8/35$$

$$x_1 = (1 - 4\,x_4 - 2\,x_3 - 2\,x_2)/6 = -13/70$$

$$\vec{x} = \begin{bmatrix} -13/70 \\ 8/35 \\ -4/35 \\ 33/70 \end{bmatrix}$$

MATLAB M-File: GaussPivot

```
function x = GaussPivot(A,b)
% GaussPivot(A,b) :
% Solve Ax =b using Gaussian elimination with pivoting
% Input:
%    A = coefficient matrix
%    b = right-hand-side matrix
%
% Output:
%    x = solution matrix

% compute the matrix sizes
[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
        %pivot the rows
        Aug([k,ipr],:) = Aug([ipr,k],:);
    end
  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
```

Partial Pivoting

Partial Pivoting (switch rows)

largest element in {x}

`[big,i] = max(x)`

index of the largest element

```
>> format short
>> x=GaussPivot0(A,b)
Aug =
     1     0     2     3     1
    -1     2     2    -3    -1
     0     1     1     4     2
     6     2     2     4     1
big =
     6
i =
     4
ipr =
     4
Aug =
     6     2     2     4     1
    -1     2     2    -3    -1
     0     1     1     4     2
     1     0     2     3     1
factor =
   -0.1667
Aug =
    6.0000    2.0000    2.0000    4.0000    1.0000
         0    2.3333    2.3333   -2.3333   -0.8333
         0    1.0000    1.0000    4.0000    2.0000
    1.0000         0    2.0000    3.0000    1.0000
factor =
     0
Aug =
    6.0000    2.0000    2.0000    4.0000    1.0000
         0    2.3333    2.3333   -2.3333   -0.8333
         0    1.0000    1.0000    4.0000    2.0000
    1.0000         0    2.0000    3.0000    1.0000
factor =
    0.1667
Aug =
    6.0000    2.0000    2.0000    4.0000    1.0000
         0    2.3333    2.3333   -2.3333   -0.8333
         0    1.0000    1.0000    4.0000    2.0000
         0   -0.3333    1.6667    2.3333    0.8333
```

Aug = [A b]

Find the first pivot element and its index

Interchange rows 1 and 4

Eliminate first column

No need to interchange

```
big =
     2.3333
i =
      1
ipr =
      2
factor =
     0.4286
Aug =
     6.0000     2.0000     2.0000     4.0000     1.0000
          0     2.3333     2.3333    -2.3333    -0.8333
          0          0          0     5.0000     2.3571
          0    -0.3333     1.6667     2.3333     0.8333
```

Second pivot element and index

No need to interchange

```
factor =
    -0.1429
Aug =
     6.0000     2.0000     2.0000     4.0000     1.0000
          0     2.3333     2.3333    -2.3333    -0.8333
          0          0          0     5.0000     2.3571
          0          0     2.0000     2.0000     0.7143
```

Eliminate second column

```
big =
      2
i =
      2
ipr =
      4
Aug =
     6.0000     2.0000     2.0000     4.0000     1.0000
          0     2.3333     2.3333    -2.3333    -0.8333
          0          0     2.0000     2.0000     0.7143
          0          0          0     5.0000     2.3571
```

Third pivot element and index

Interchange rows 3 and 4

```
factor =
      0
Aug =
     6.0000     2.0000     2.0000     4.0000     1.0000
          0     2.3333     2.3333    -2.3333    -0.8333
          0          0     2.0000     2.0000     0.7143
          0          0          0     5.0000     2.3571
```

Eliminate third column

Back substitution

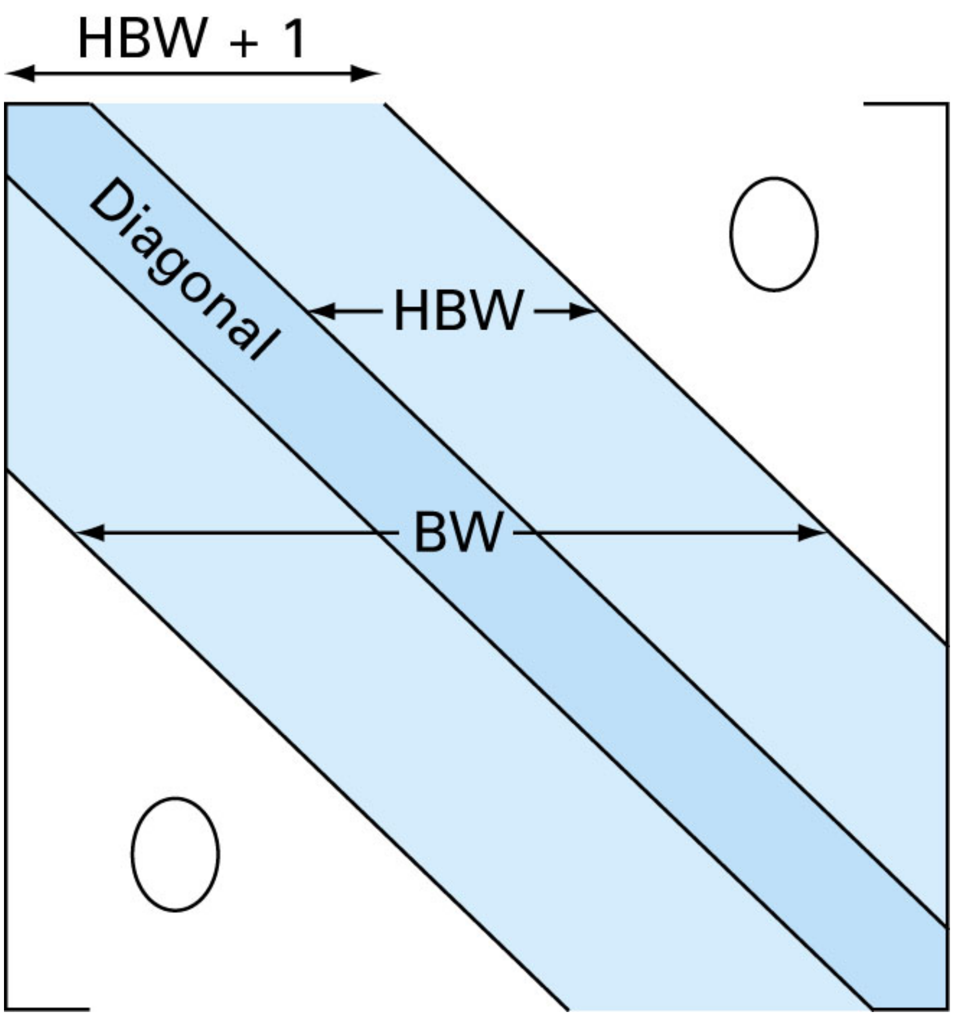
```
x =
          0
          0
          0
     0.4714
x =
          0
          0
    -0.1143
     0.4714
x =
          0
     0.2286
    -0.1143
     0.4714
x =

    -0.1857
     0.2286
    -0.1143
     0.4714
```

Save factors f_{ij} for

***LU* Decomposition**

Banded Matrix

CognitiveSystems



HBW: Half Band Width

CognitiveSystems

•Banded Matrix

$a_{i,j} = 0$
if j > i + HB
or j< i - HB

HB: Half Bandwidth
B: Bandwidth

B = 2*HB + 1

In this example
HB = 1 & B = 3

$$\begin{vmatrix} a & b & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ c & d & e & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & f & g & h & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & i & j & k & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & l & m & n & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & k & z & p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & q & r & s & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t & u & v \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & w & x \end{vmatrix}$$

Tridiagonal Matrix

CognitiveSystems

- Only three nonzero elements in each equation (*$3n$* instead of n^2 elements)
- Subdiagonal, diagonal, superdiagonal

- Row Scaling (not implemented in textbook)

  -- scale the diagonal element to $a_{ii} = 1$

- Solve by Gauss elimination

Tridiagonal Matrix

CognitiveSystems

$$
\begin{bmatrix}
f_1 & g_1 & & & & & \\
e_2 & f_2 & g_2 & & & & \\
 & e_3 & f_3 & g_3 & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & \boxed{e_i \quad f_i \quad g_i} & & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & e_{n-1} & f_{n-1} & g_{n-1} \\
 & & & & & e_n & f_n
\end{bmatrix}
\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_i \\ \vdots \\ x_{n-1} \\ x_n \end{Bmatrix}
=
\begin{Bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_i \\ \vdots \\ r_{n-1} \\ r_n \end{Bmatrix}
$$

- Special case of banded matrix with bandwidth = 3
- Save storage, 3 × n instead of n × n

Tridiagonal Matrix

CognitiveSystems

- **Forward elimination**

$$\begin{cases} f_k = f_k - \dfrac{e_k}{f_{k-1}} g_{k-1} \\ r_k = r_k - \dfrac{e_k}{f_{k-1}} r_{k-1} \end{cases} \quad k = 2,3,\ldots,n$$

Use factor = e_k / f_{k-1}

to eliminate subdiagonal element

Apply the same matrix operations to right hand side

- **Back substitution**

$$x_n = \frac{r_n}{f_n}$$

$$x_k = \frac{r_k - g_k x_{k+1}}{f_k} \qquad k = n-1, n-2, \ldots, 3, 2, 1$$

Hand Calculations: Tridiagonal Matrix

$$\begin{bmatrix} 1 & -2 & 0 & 0 \\ -2 & 5 & -1 & 0 \\ 0 & -1 & 2 & -0.5 \\ 0 & 0 & -0.5 & 1.25 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} -3 \\ 5 \\ 2 \\ 3.5 \end{Bmatrix}$$

(a) Forward elimination

$$\begin{cases} f_2 = f_2 - \dfrac{e_2}{f_1} g_1 = 5 - \dfrac{-2}{1}(-2) = 1 \\[2ex] r_2 = r_2 - \dfrac{e_2}{f_1} r_1 = 5 - \dfrac{-2}{1}(-3) = -1 \end{cases}$$

$$\begin{cases} f_3 = f_3 - \dfrac{e_3}{f_2} g_2 = 2 - \dfrac{-1}{1}(-1) = 1 \\[2ex] r_3 = r_2 - \dfrac{e_2}{f_1} r_1 = 2 - \dfrac{-1}{1}(-1) = 1 \end{cases}$$

$$\begin{cases} f_4 = f_4 - \dfrac{e_4}{f_3} g_3 = 1.25 - \dfrac{-0.5}{1}(-0.5) = 1 \\[2ex] r_4 = r_4 - \dfrac{e_4}{f_3} r_3 = 3.5 - \dfrac{-0.5}{1}(1) = 4 \end{cases}$$

(b) Back substitution

$$x_4 = \frac{r_4}{f_4} = \frac{4}{1} = 4$$

$$x_3 = \frac{r_3 - g_3 x_4}{f_3} = \frac{1 - (-0.5)(4)}{1} = 3$$

$$x_2 = \frac{r_2 - g_2 x_3}{f_2} = \frac{-1 - (-1)(3)}{1} = 2$$

$$x_1 = \frac{r_1 - g_1 x_2}{f_1} = \frac{-3 - (-2)(2)}{1} = 1$$

MATLAB M-file: Tridiag

CognitiveSystems

```
function x = Tridiag(e, f, g, r)
% Tridiag(e,f,g,r):
%     Tridiagonal system solver
% Input:
%   e = subdiagonal vector
%   f = diagonal vector
%   g = superdiagonal vector
%   r = right hand side vector
% Output:
%   x = solution vector

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
```

Example: Tridiagonal matrix

$$
\begin{bmatrix}
1 & -2 & & & & \\
-2 & 6 & 4 & & & \\
 & 4 & 9 & -0.5 & & \\
 & & -0.5 & 3.25 & 1.5 & \\
 & & & 1.5 & 1.75 & -3 \\
 & & & & -3 & 13
\end{bmatrix}
\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{Bmatrix}
=
\begin{Bmatrix} -3 \\ 22 \\ 35.5 \\ = 7.75 \\ 4 \\ -33 \end{Bmatrix}
$$

```
function [e,f,g,r] = example

e=[ 0 -2    4  -0.5  1.5   -3];
f=[ 1  6    9  3.25 1.75   13];
g=[-2  4 -0.5   1.5   -3    0];
r=[-3 22 35.5 -7.75    4  -33];
```

```
» [e,f,g,r] = example

e =
         0   -2.0000    4.0000   -0.5000    1.5000   -3.0000
f =
    1.0000    6.0000    9.0000    3.2500    1.7500   13.0000
g =
   -2.0000    4.0000   -0.5000    1.5000   -3.0000         0
r =
   -3.0000   22.0000   35.5000   -7.7500    4.0000  -33.0000

» x = Tridiag (e, f, g, r)
x =      1      2      3     -1     -2     -3
```

Note: `e(1) = 0 and g(n) = 0`