جـــامـــعـــة دمـــشـــق
كـليـة الهـنـدسـة المـدنيـة

تـصـمـيـم بـمـعـونـة الـحـاسـب
Computer Aided Design

Lecture 3

Control Structures & Figures

25/11/2025

Wael Darwich

Damascus University

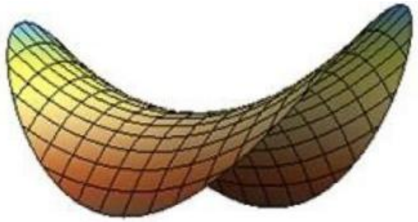
Civil Engineering Faculty

Last Lecture

- Matrix Calculator Continued…
  - Linear Equations
  - Symbolic Math Toolbox
- Control Structures
  1. Conditions
  2. Loops

Content

- Loops Continued…
- Examples
- Notes:
    1. Debugging
    2. Performance
    3. Numerical Tolerance
- Figures
    - 2D plot
    - Functions
    - Examples, Beams and Trusses
    - 3D plot



$$z = \frac{x^2}{a^2} - \frac{y^2}{b^2}, \quad \frac{x^2}{a^2} + \frac{y^2}{b^2} < 1$$

Endless Loops!

```
counter = 1;
while true
    disp(counter)
    counter = counter + 1;
    if counter > 3
        break
    end
end
```

```
1
2
3
```

Nested Loops

Outer

Inner

```
for x = 1:3
  for y = 4:5
     disp([x y])
  end
end
```

x: 3 iterations

y: 2 iterations

3×2 = 6 iterations

```
1  4
1  5
2  4
2  5
3  4
3  5
```

When working with nested loops, the outer loop changes only after the inner loop is completely done

Ends

- if / end : same number
- for, while / end : same number
- (1+2×(3+4)) = (1+2×(3+4))

Not (1+2×(3+4))

A bracket will always close the last opened one

Same for if/for/while & end

Nested Loops

Nested loops

```
for x = 1:3
  for y = 4:5
      disp([x y])
  end
end
```

Allowed

Unnested loops

```
for x = 1:3
  for y = 4:5
      disp([x y])
end
  end
```

Allowed as **nested**

Whitespace is ignored

Loops Comparison

- `for`
- `while`

For Syntax

```
for counter = start:step:end
   code
end
```

- Using counter (different value for each iteration)
- Predefined number of steps, will not change

While Syntax

```
code1
while condition
   code2
end
```

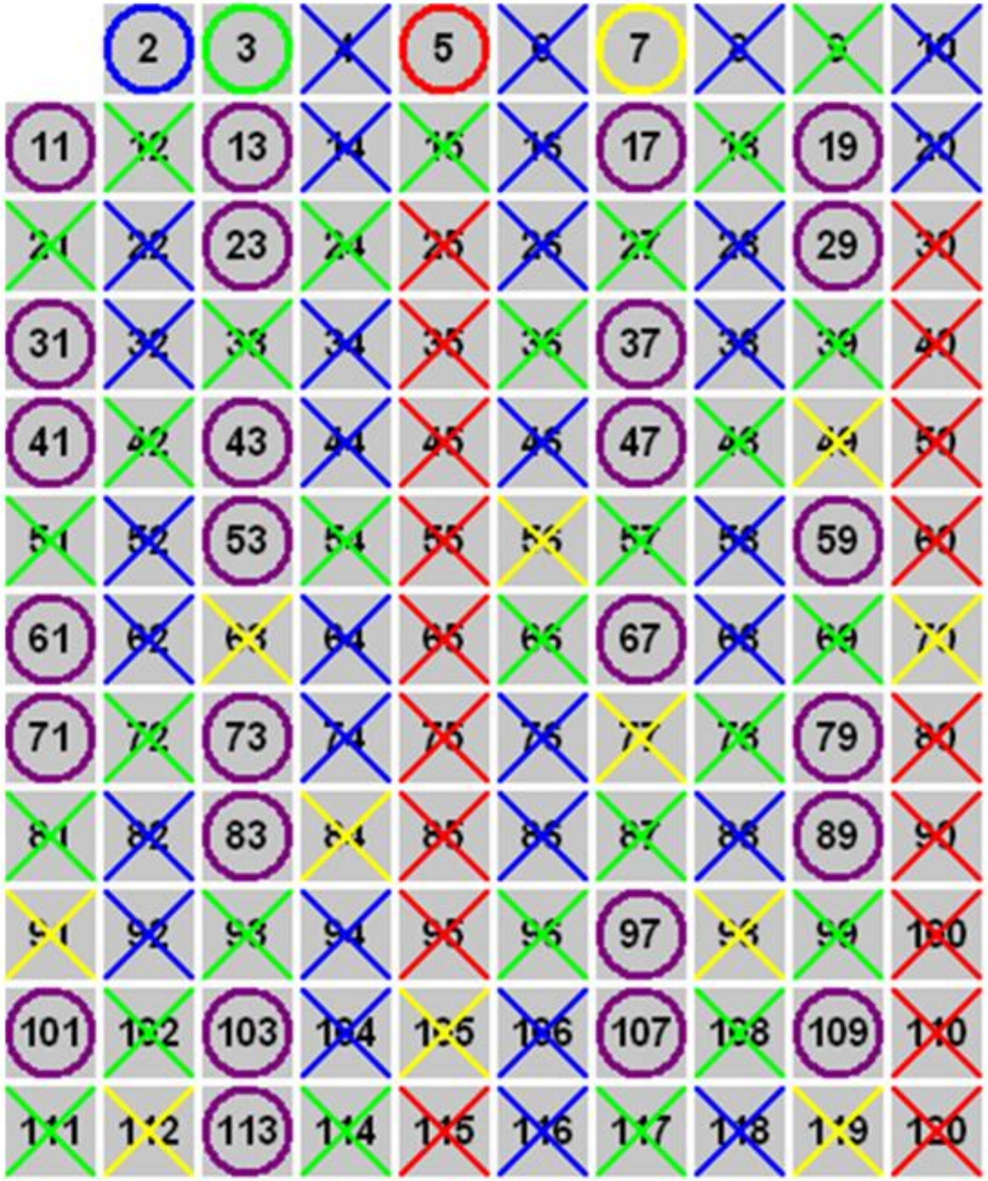
- No counter
- Undefined number of steps, could be 0
- `code1` : Needs to satisfy `condition` to get into the loop
- `code2` : Needs to break `condition` to get out of the loop

Prime Numbers Example

To find all prime numbers less than or equal to integer n (Eratosthenes' method)

1. Create a list of consecutive integers from `2` to `n`: (`2, 3, ..., n`).
2. Initially, let `p` equal `2`, the first prime number.
3. Strike from the list all multiples of `p` less than or equal to `n`. (`2p, 3p,` etc.)
4. Find the first number remaining on the list after `p` (this number is the next prime); replace `p` with this number.
5. Repeat steps `3` and `4` until p^2 is greater than `n`.
6. All the remaining numbers in the list are prime.

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120



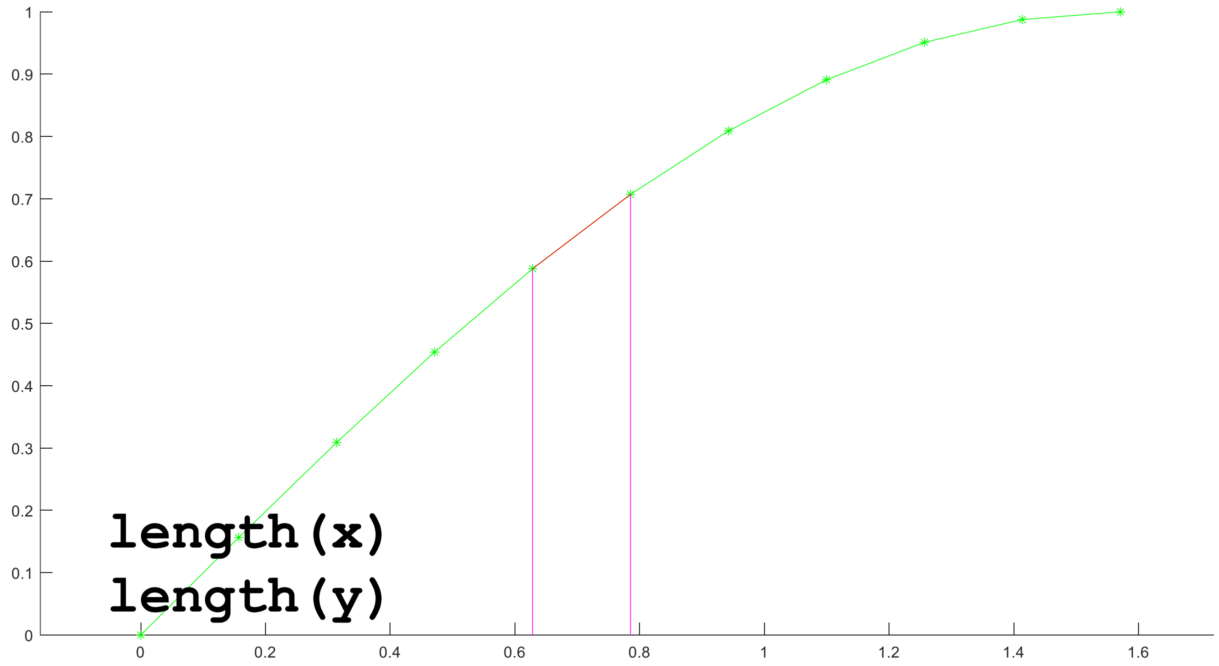
Prime numbers

2	3	5	7
11	13	17	19
23	29	31	37
41	43	47	53
59	61	67	71
73	79	83	89
97	101	103	107
109	113		

Numerical Integration

- Calculate area under function `y=sin(x)` in the range `[0 π/2]`

```
x1 = 0; x2 = pi/2;
range = x2 - x1;
strips = 10;
dx = range/strips;
x = x1 : dx : x2; % 11 point
y = sin(x);
A = 0;
for i = 1:strips
   dA = dx * 0.5*(y(i)+y(i+1));
   A = A + dA;
end
disp(A)
```



length(x)
length(y)

```
for i = 1:strips
dA = dx * 0.5*(y(i)+y(i+1));
```

Symbolic Math Toolbox

```
syms x
int(sin(x))                     -cos(x)

int(sin(x), 0, pi/2)           1
```

Other Examples

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

```
function y = sin_ex(x)
y = 0;
for n = 0 : 10
    y = y + (-1)^n * x^(2*n+1) / factorial(2*n+1);
end
```

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

$$= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

```
function y = cos_ex(x)
y = 0;
for n = 0 : 10
    y = y + (-1)^n * x^(2*n) / factorial(2*n);
end
```

Other Examples

$$\sinh x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots$$

$$\cosh x = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots$$

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

$$\ln(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n} = -x - \frac{x^2}{2} - \frac{x^3}{3} - \cdots$$

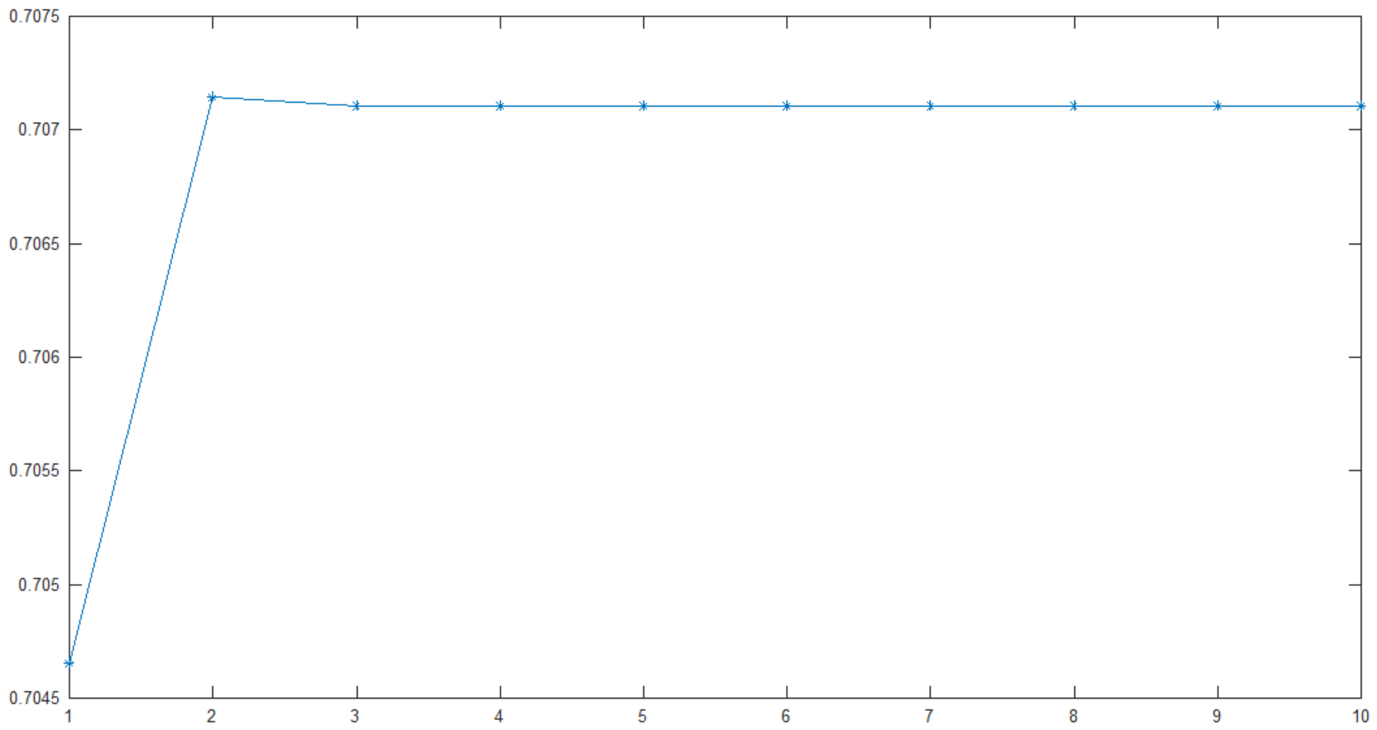
$$\ln(1+x) = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n} = x - \frac{x^2}{2} + \frac{x^3}{3} - \cdots$$

$$\sqrt{1+x} = \sum_{n=0}^{\infty} \frac{(-1)^n (2n)!}{(1-2n)(n!)^2 (4^n)} x^n = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3 - \frac{5}{128}x^4 + \cdots$$

Number of Steps

- Engineers can determine the required number of steps
  - Accuracy
  - Efficiency

Max n	sin(45°)
1	0.704652651209168
2	0.707143045779360
3	0.707106469575178
4	0.707106782936867
5	0.707106781179619
6	0.707106781186568
7	0.707106781186547
8	0.707106781186547
9	0.707106781186547
10	0.707106781186547



Other Examples

- FindMax
- Least Common Multiple LCM
- Greatest Common Divisor GCD

Download:

- https://damascusu

- https://cad.el.sy/

جامعة دمشق / كلية الهندسة المدنية

محاضرات تصميم بمعونة الحاسب / الفصل الأول / 2026-2025

المحاضرة رقم 1 : 2025/11/11 المدرج الرابع

Files:

- Presentation
- PDF
- MATLAB:
  - script1 lec1
  - script2 lec1
  - script3 lec1
  - cosinerule lec1
  - matrices lec1.

المحاضرة رقم 2 : 2025/11/18 المدرج الأول

Files:

- Presentation
- PDF
- MATLAB:
  - examples lec2
  - linear eq lec2
  - root ex
  - root lec2
  - ebill ex
  - loops lec2

Important Notes

- Clear your command window `clc`
- check for missed or redundant end
  - Select all code (Ctrl+A) and indent (Ctrl+I) to
- Debug using break points and F5, F10, F11
- Save .m files and .mat workspace

```
function tariff = ebill_ex(kwh)
% Calculate electricity bill
if kwh < 1
    tariff = 0;
elseif kwh <= 600
    tariff = kwh * 2;
elseif kwh <= 1000
    tariff = 600 * 2 + (kwh - 600) * 6 ;
elseif kwh <= 1500
    tariff = 600 * 2 + 400 * 6 + (kwh - 1000) * 20;
```

Performance

- Write script to check if a point `P(x, y)` is inside a circle:

  centre $c(x_c, y_c) = (0, 0)$ and radius $r = 1$

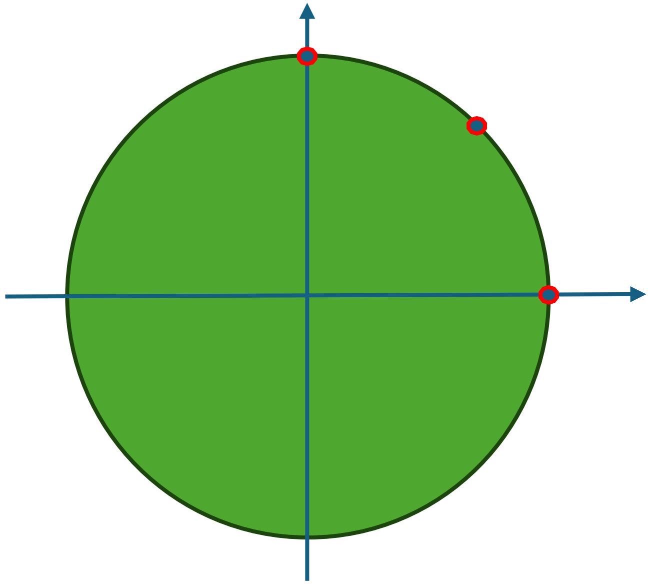
```
function isinside = inside(x, y)
xc = 0; yc = 0;
r = 1;
dist = sqrt((x-xc)^2 + (y-yc)^2);
if dist <= r
    isinside = true;
else
    isinside = false
end
```

```
function isinside = inside(x, y)
xc = 0; yc = 0;
r = 1;
dist2 = (x-xc)^2 + (y-yc)^2;
if dist2 <= r^2
    isinside = true;
                   false;
```

- `sqrt` vs `^2`
- Multiple points with the same circle

Numerical Tolerance

```
>> inside(1, 0)
ans =
     1
>> inside(0, 1)
ans =
     1
>>
```



Numerical Tolerance

```
>> inside(2^.5/2,2^.5/2)
ans =
     1
>> inside(0.707106781186548, 0.707106781186548)
ans =
     0
>>
```

```
>> 2^.5/2

ans =

   0.707106781186548
```

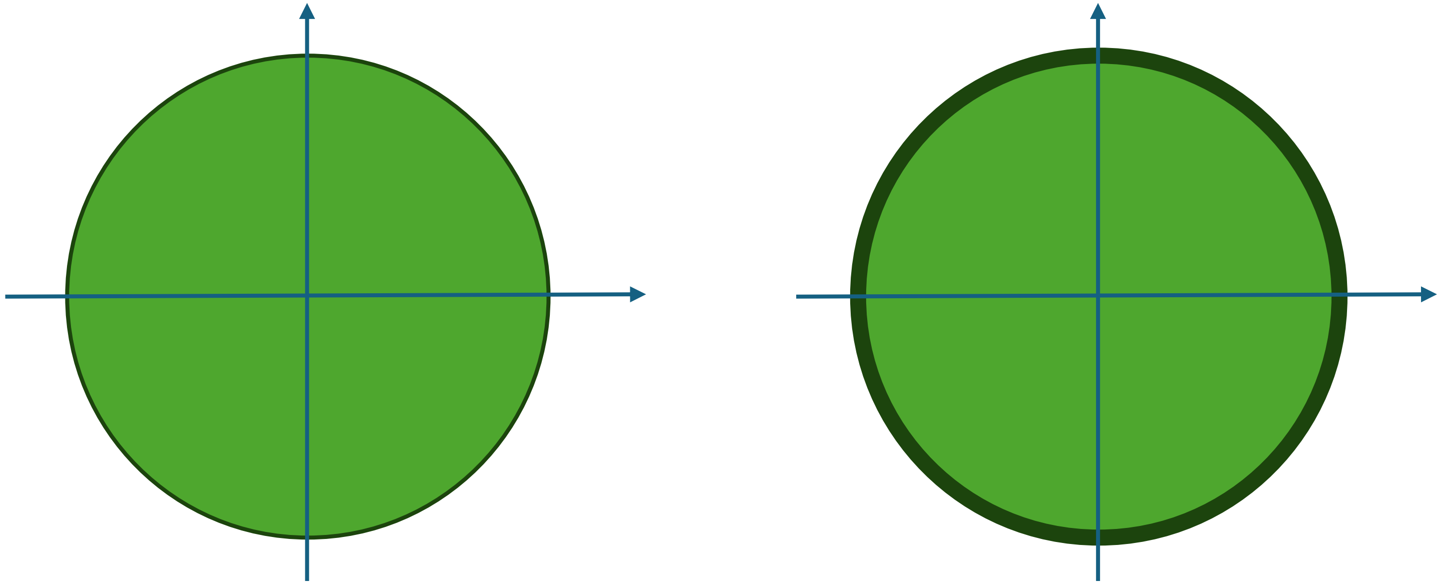
Change inside

```
function isinside = inside(x, y)
xc = 0; yc = 0;
r2 = 1;
dist2 = (x-xc)^2 + (y-yc)^2;
if dist2 <= r2
    isinside = true;
else
    isinside = false;
end
```

```
function isinside = inside(x, y)
xc = 0; yc = 0;
r2 = 1;
dist2 = (x-xc)^2 + (y-yc)^2;
if dist2 <= (r2 + 1E-6)
    isinside = true;
else
    isinside = false;
end
```

Numerical Tolerance

```
>> inside(2^.5/2,2^.5/2)
ans =
     1
>> inside(0.707106781186548, 0.707106781186548)
ans =
     1
>>
```



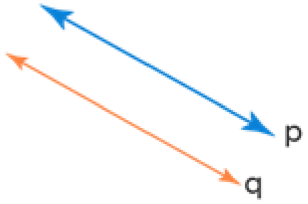
Numerical Precision

- sin (0)
- sin (pi)
- sin (2 * pi)

```
>> sin(0)

ans =

     0

>> sin(pi)

ans =

     1.224646799147353e-16

>> sin(2*pi)

ans =

    -2.449293598294706e-16
```

Numerical Precision

- Parallel lines
  - Line slope
  - Intersection
- Lunar calendar



Figures

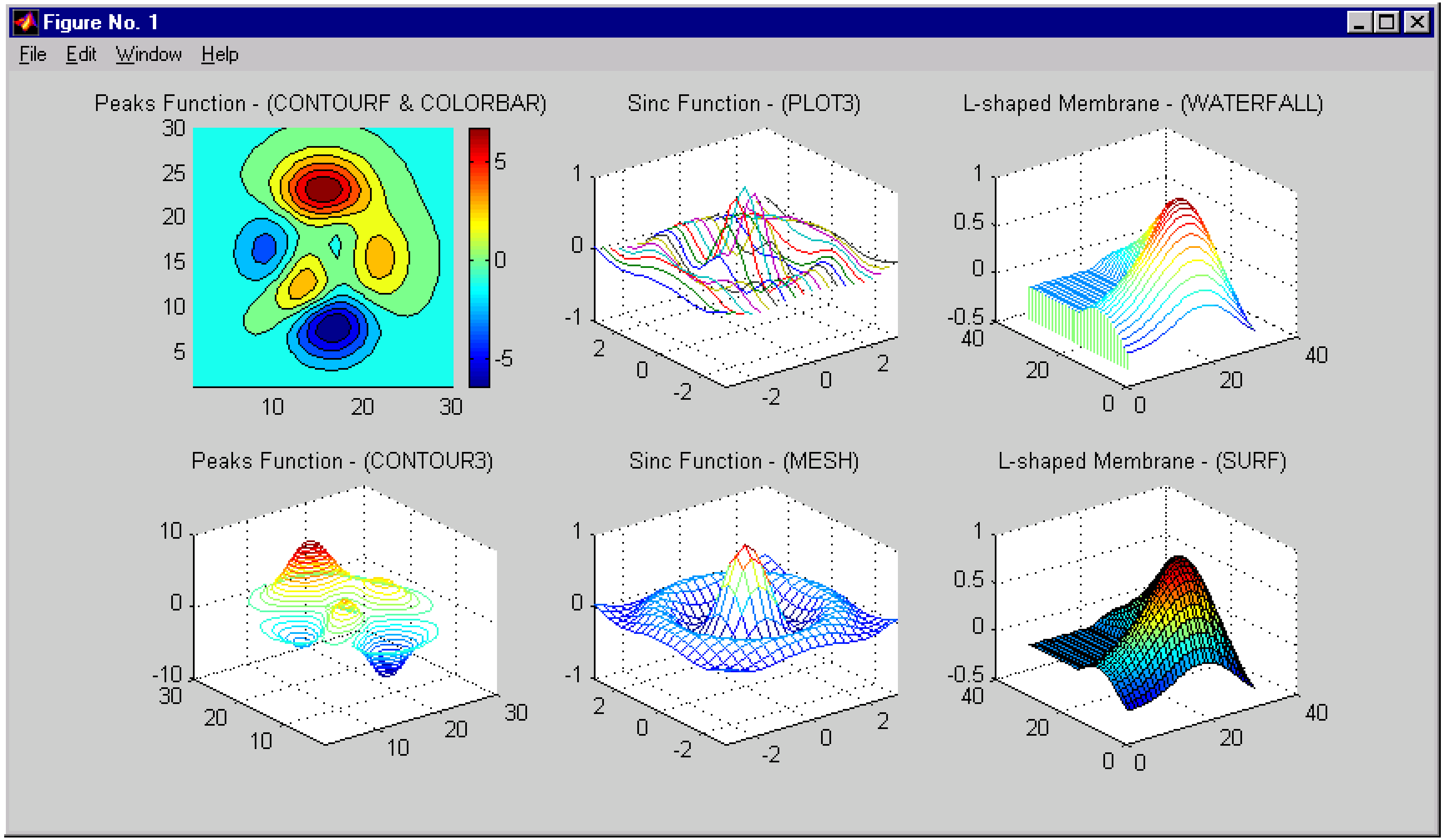
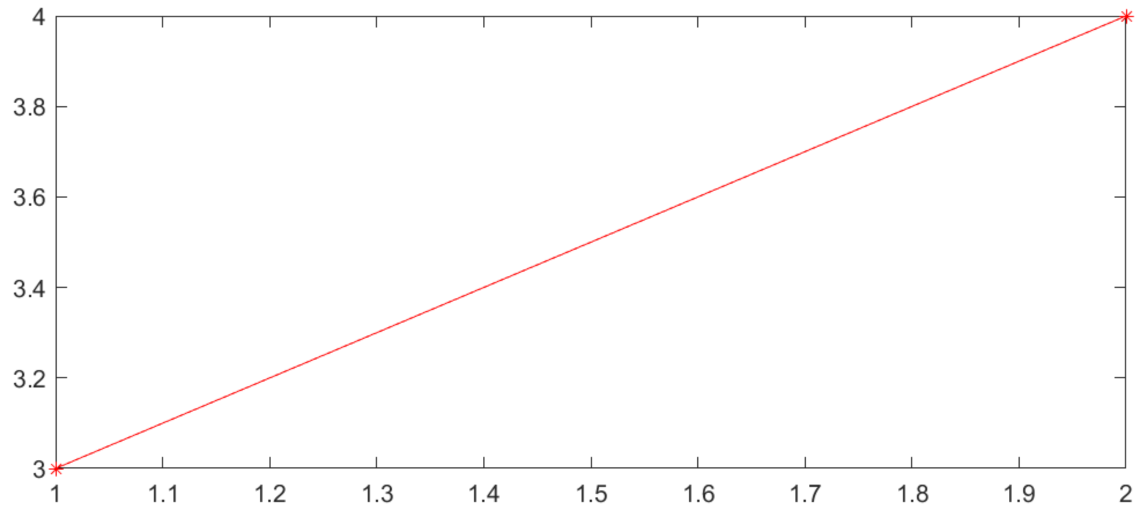


Figure No. 1
File Edit Window Help
Peaks Function - (CONTOURF & COLORBAR)
Sinc Function - (PLOT3)
L-shaped Membrane - (WATERFALL)
Peaks Function - (CONTOUR3)
Sinc Function - (MESH)
L-shaped Membrane - (SURF)

Plot

```
x = [1 2];
y = [3 4];


plot(x, y)
plot(x, y, '-*r')
```



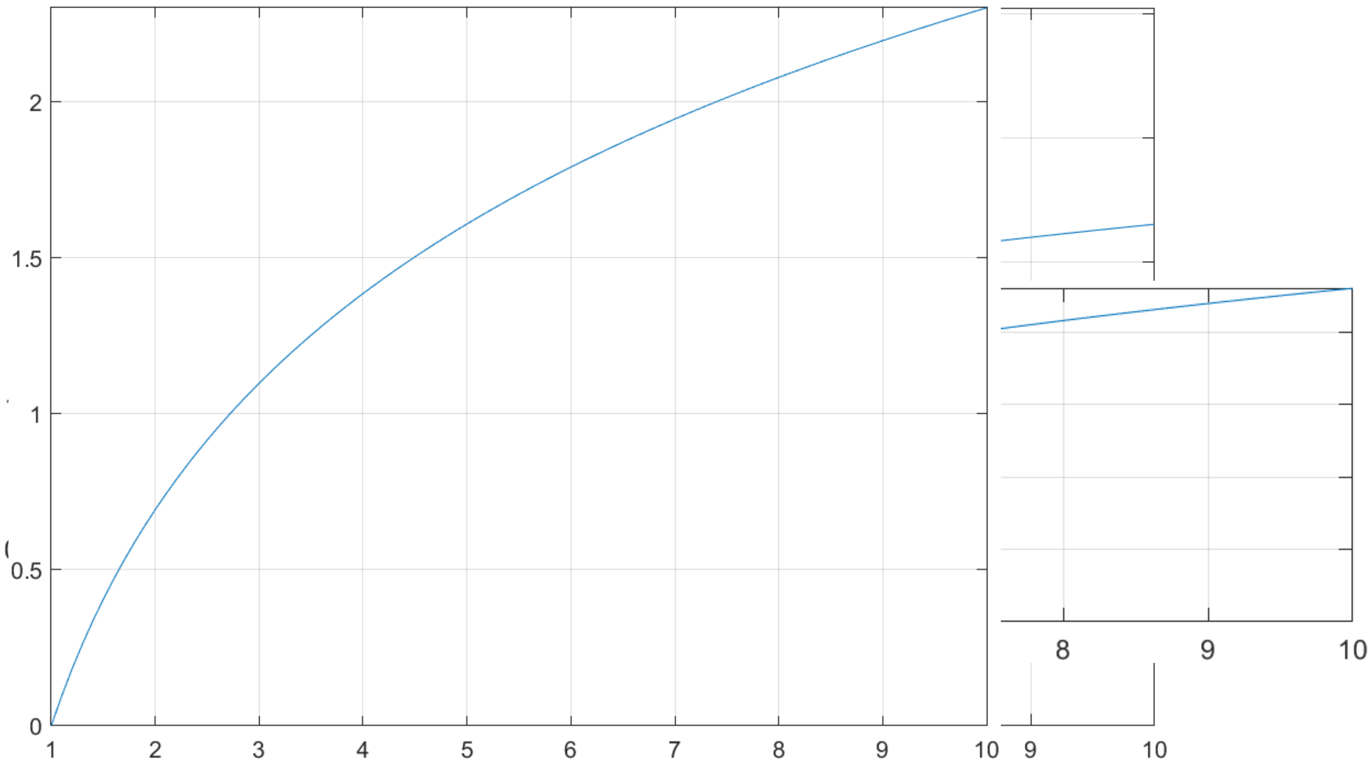
Plot

- `plot(X,Y)` plots vector Y versus vector X.
- `plot(X,Y,S)` where S is a character string:

`b`	blue	`.`	point	`-`	solid
`g`	green	`o`	circle	`:`	dotted
`r`	red	`x`	x-mark	`-.`	dashdot
`c`	cyan	`+`	plus	`--`	dashed
`m`	magenta	`*`	star	(none)	no line
`y`	yellow	`s`	square		
`k`	black	`d`	diamond		
`w`	white	`v`	triangle (down)		
		`^`	triangle (up)		
		`<`	triangle (left)		
		`>`	triangle (right)		
		`p`	pentagram		
		`h`	hexagram		

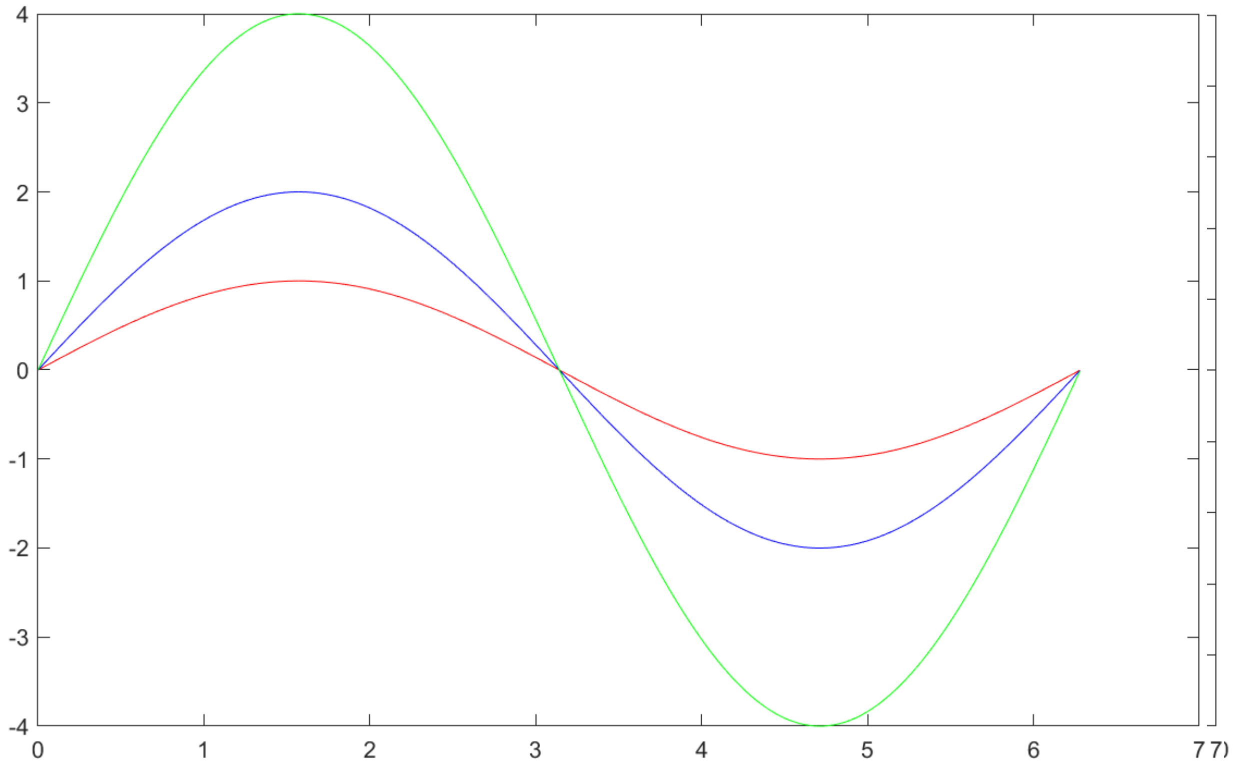
Plot

```
x=1:10;
y=log(x);
plot(x,y, '*:')

x=1:0.1:10;
y=log(x);
plot(x,y, '-')

grid
axis equal
axis([1 10 0 2.5])   % x1 x2 y1 y2
axis([1 Inf 0 Inf])
daspect([3 1 1])     % Data aspect ratio
```



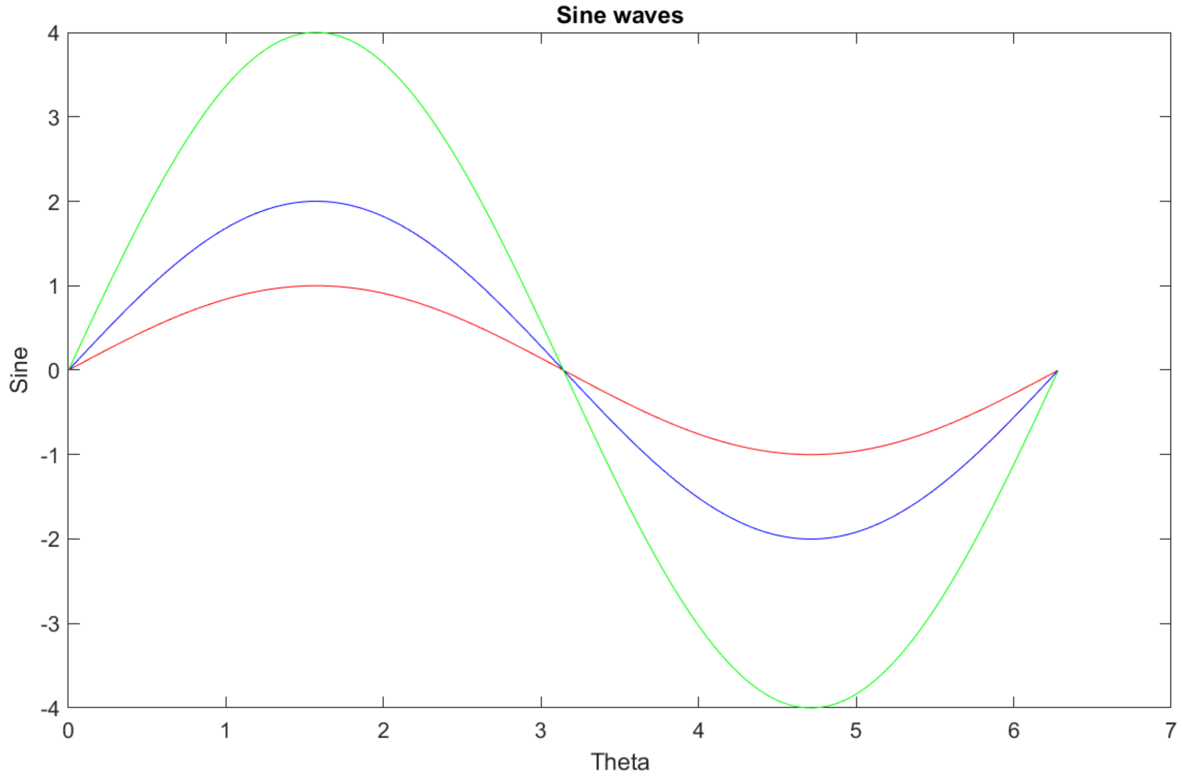
Plot

```
x = 0:0.1:2*pi; % length 63
y = sin(x);
plot(x,y)

plot(y)
% plots Y columns vs. index

x = 0:.01:2*pi;
y1 = sin(x);
y2 = sin(2*x);
y3 = sin(4*x);
plot(x, y1, 'r', x, y2 , 'b', x, y3, 'g')
plot(x, y1, 'r', x, 2*y1 , 'b', x, 4*y1, 'g')
```



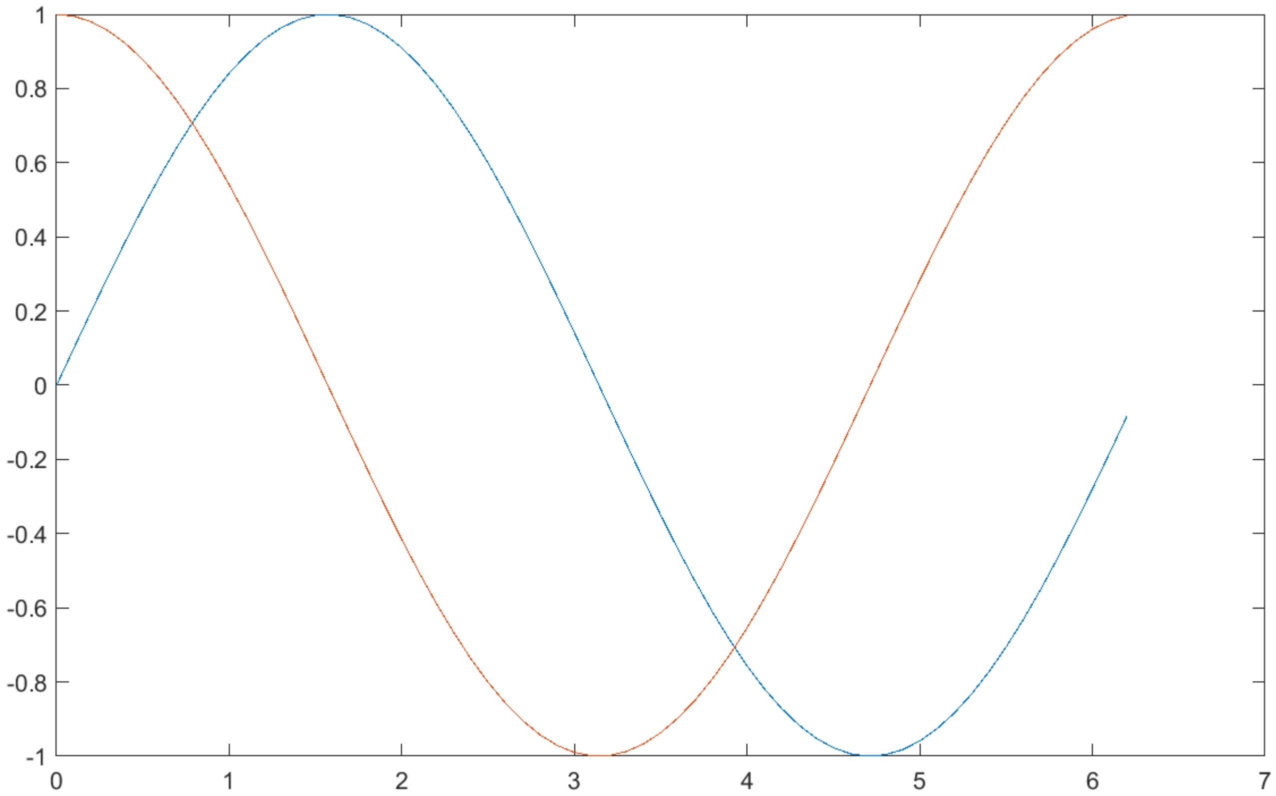
Plot

- `title('Sine waves')`
- `xlabel('Theta')`
- `ylabel('Sine’)`
- `clf`
- `hold on`
- `hold off`



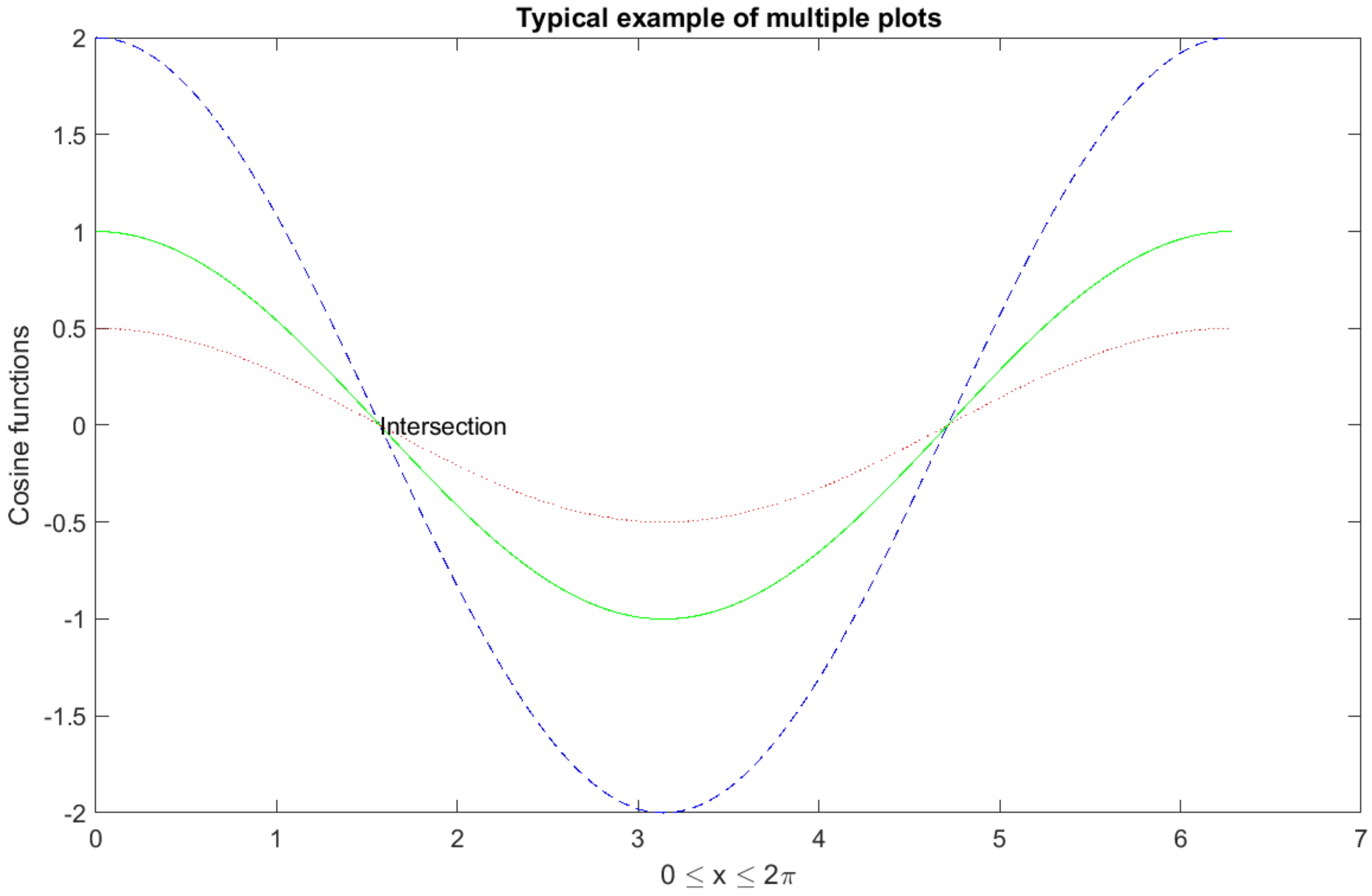
Plot

```
x=0:0.1:2*pi;
plot(x, sin(x))
hold on
plot(x, cos(x))
hold off
```



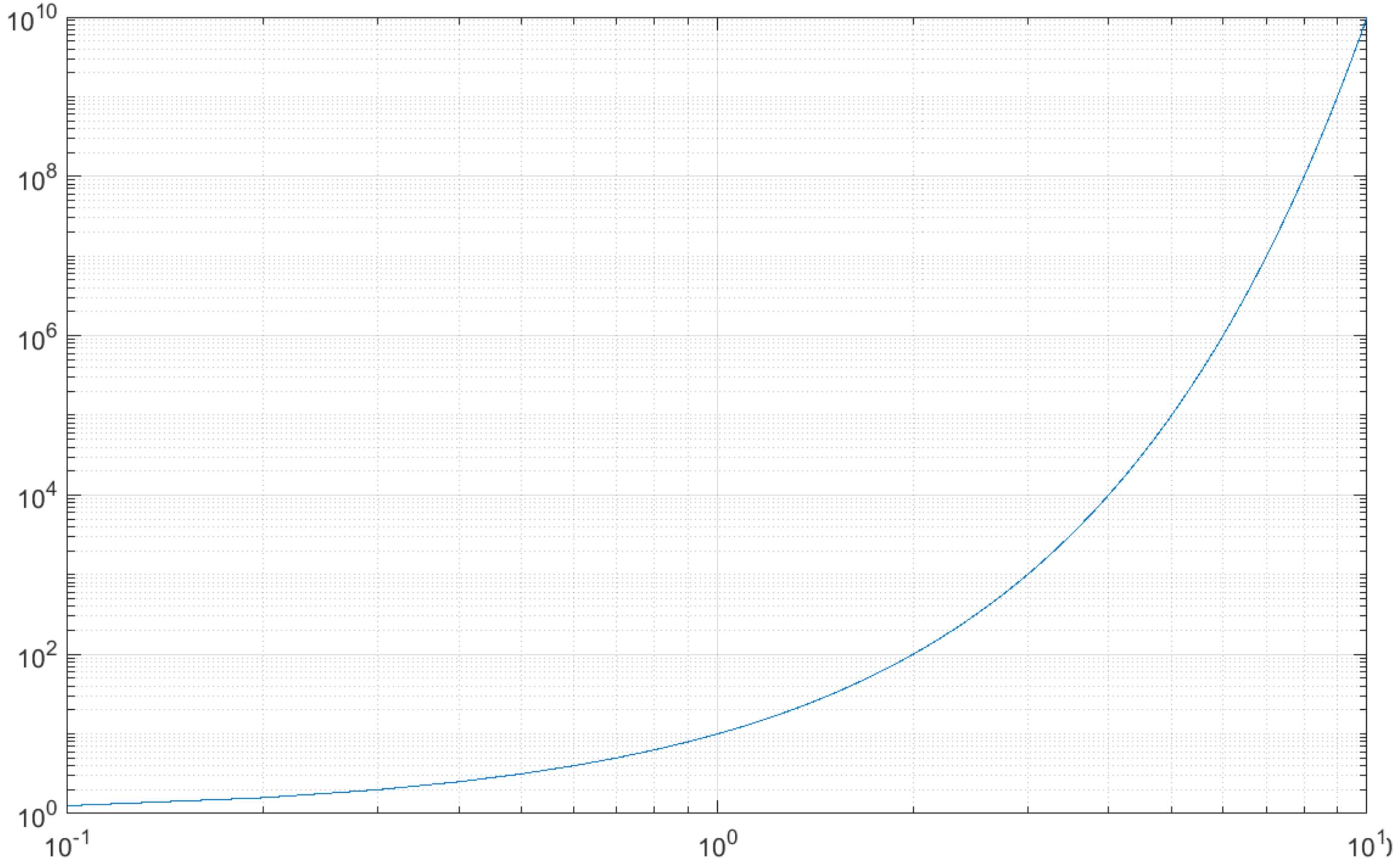
Plot

```
x = 0:pi/100:2*pi;
y1 = 2*cos(x);
y2 = cos(x);
y3 = 0.5*cos(x);
plot(x,y1,'--b',x,y2,'-g',x,y3,':r')
% Use LaTeX
xlabel('0 \leq x \leq 2\pi')
ylabel('Cosine functions')
title('Typical example')
text(pi/2, 0, 'Intersection')
```



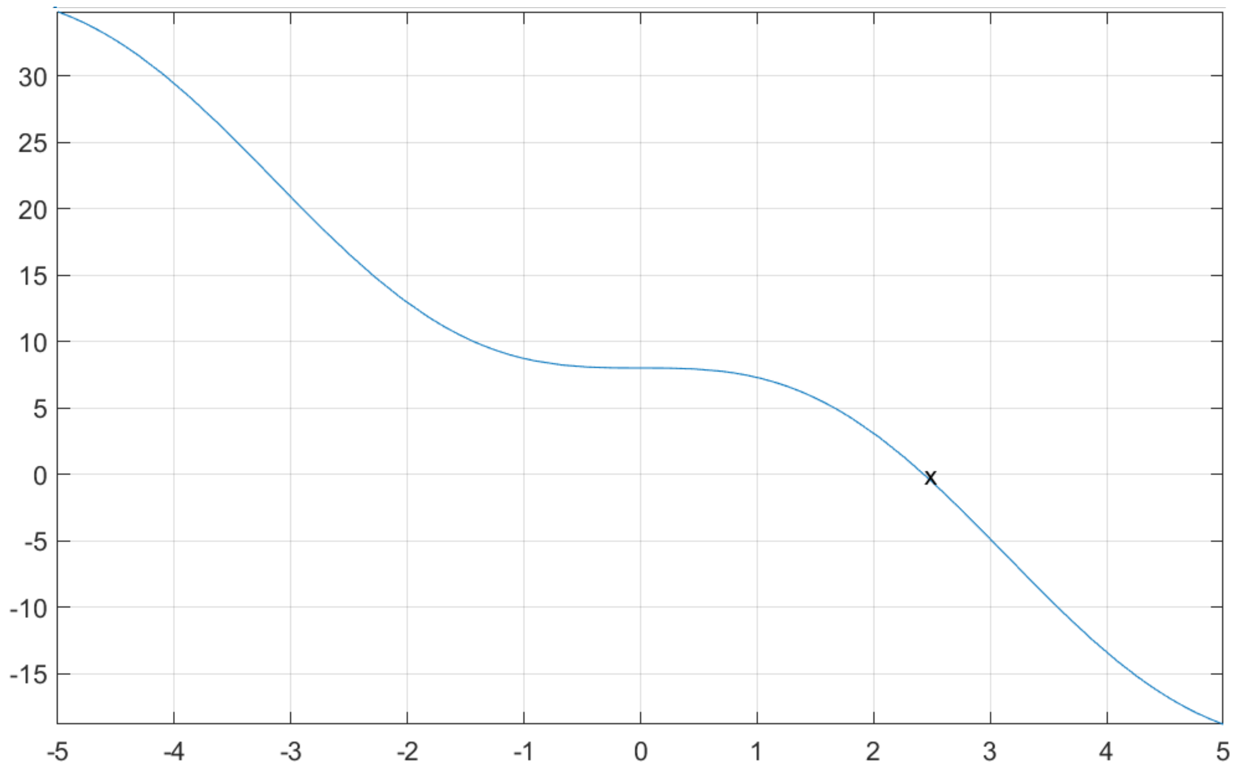
Logarithmic Scale

```
x=0:0.1:10;
y=10.^x;
plot(x,y),          grid on
semilogx(x,y),      grid on
semilogy(x,y),      grid on
loglog(x,y),        grid on
```



Fplot

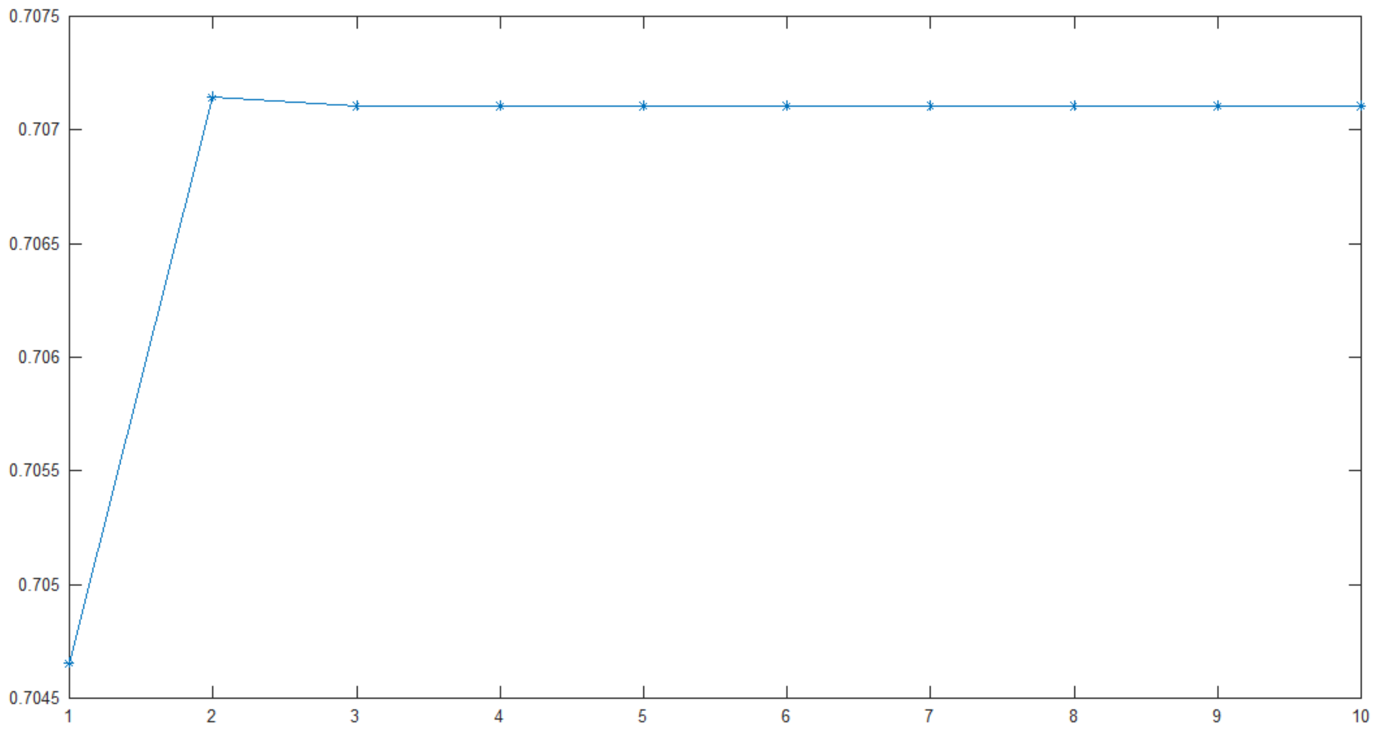
```
fplot(@sin)
fplot(@sin, [0 pi])
fplot(@(x) x.^2 + 5.*x + 6,[-6.5 1.5])
fplot(@(x) 8 - 4.5 * (x-sin(x)))
fzero(@(x) 8 - 4.5 * (x-sin(x)), 0)
ans =
    2.4305
```



Example 1

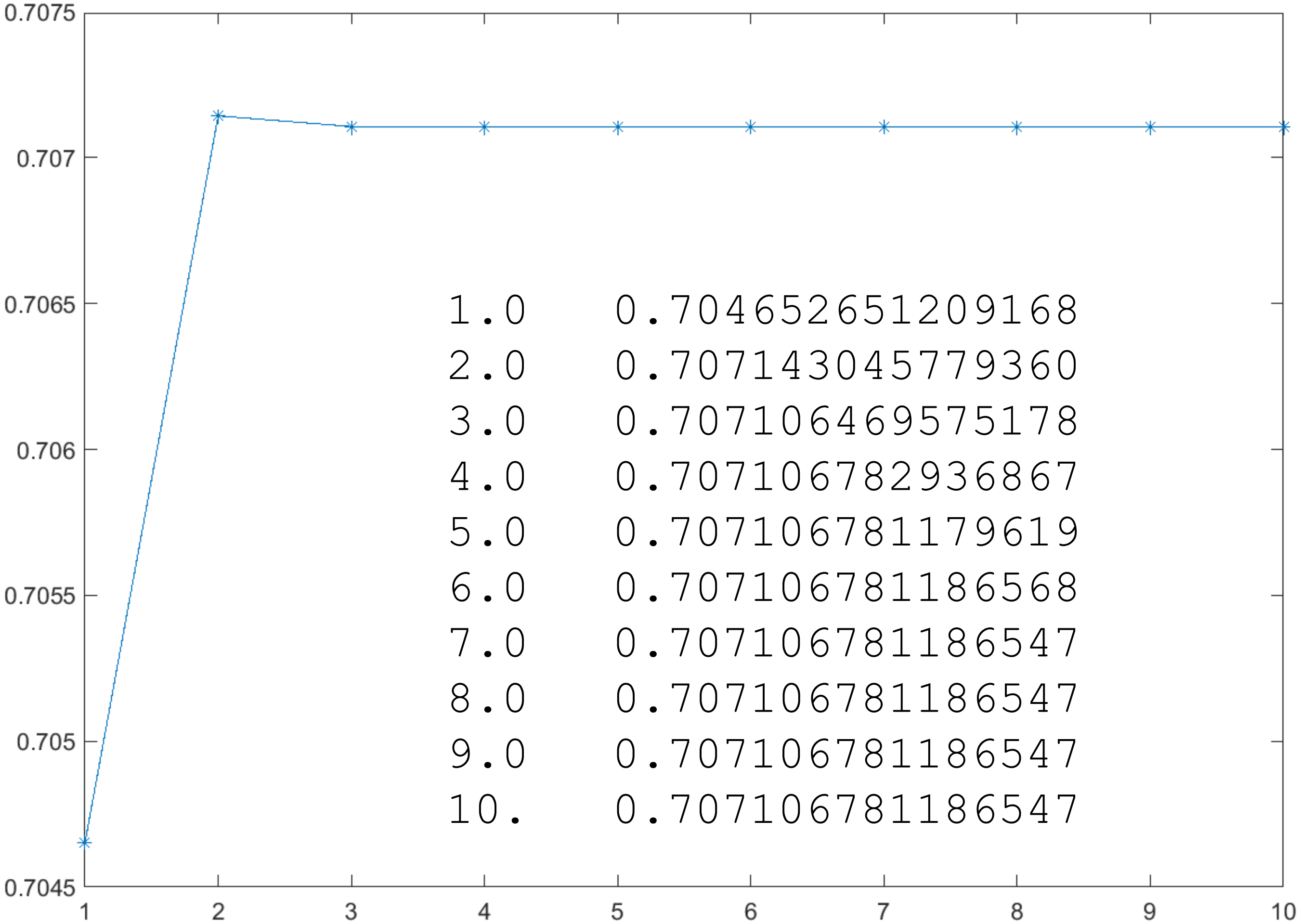
- Engineers can determine the required number of steps
  - Accuracy
  - Efficiency

Max n	sin(45°)
1	0.704652651209168
2	0.707143045779360
3	0.707106469575178
4	0.707106782936867
5	0.707106781179619
6	0.707106781186568
7	0.707106781186547
8	0.707106781186547
9	0.707106781186547
10	0.707106781186547

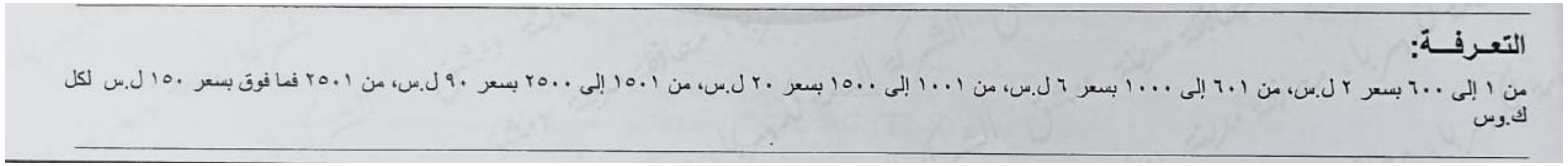


Number of Steps

```
theta = deg2rad(45);
x = 1:10;
y = zeros(size(x));

for n = x
    y(n) = sin_ex(theta, n);
end
disp([x; y]');
plot(x, y, '*-')

function result = sin_ex(theta, maxn)
    result = 0;
    for n = 0 : maxn
        result = result + (-1)^n * theta ^(2*n+1) / factorial(2*n+1);
    end
end
```



Example 2



التعـرفــة:

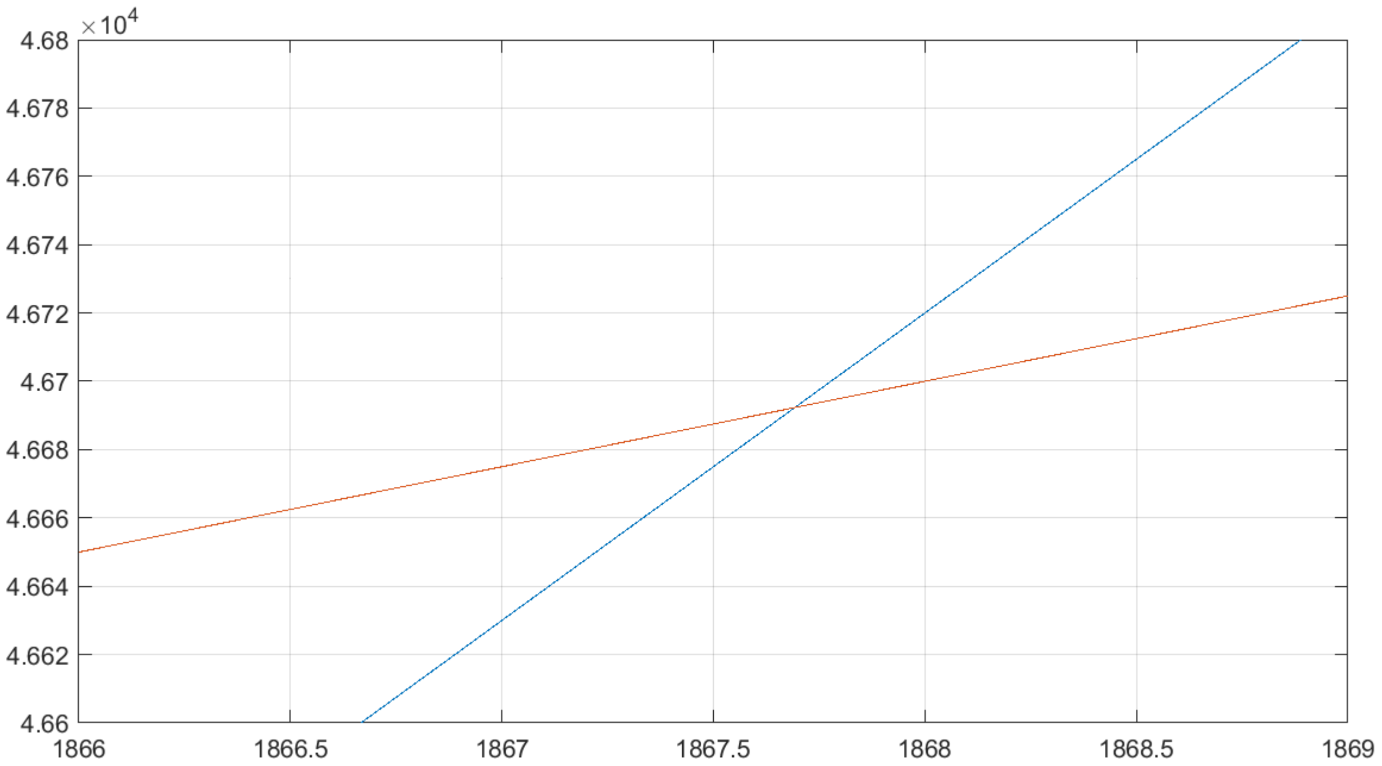
من ١ إلى ٦٠٠ بسعر ٢ ل.س، من ٦٠١ إلى ١٠٠٠ بسعر ٦ ل.س، من ١٠٠١ إلى ١٥٠٠ بسعر ٢٠ ل.س، من ١٥٠١ إلى ٢٥٠٠ بسعر ٩٠ ل.س، من ٢٥٠١ فما فوق بسعر ١٥٠ ل.س لكل ك.وس

```
function tariff = ebill_ex(kwh)

if kwh < 1
  tariff = 0;
elseif kwh <= 600
 tariff = kwh * 2;
elseif kwh <= 1000
 tariff = (kwh - 600) * 6 + 600 * 2;
elseif kwh <= 1500
 tariff = (kwh - 1000) * 20 + 600 * 2 + 400 * 6;
elseif kwh <= 2500
 tariff = (kwh - 1500) * 90 + 600 * 2 + 400 * 6 + 500 * 20;
else
 tariff = (kwh - 2500) * 150 + 600 * 2 + 400 * 6 + 500 * 20 + 1000 * 90;
end
```

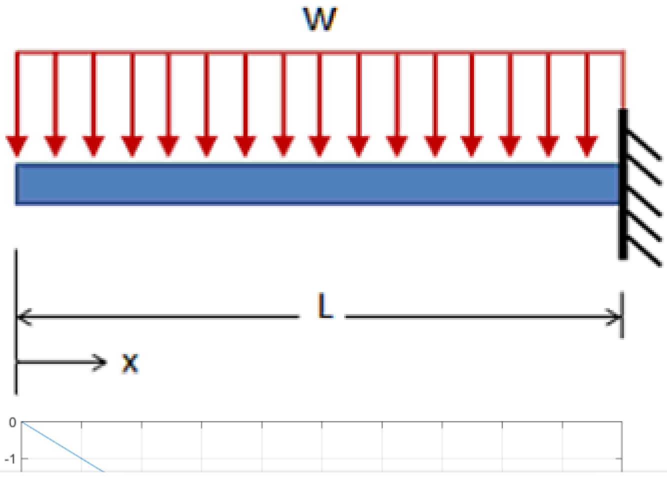
Compare to
fixed rate of 25
per kwh!

Tariff

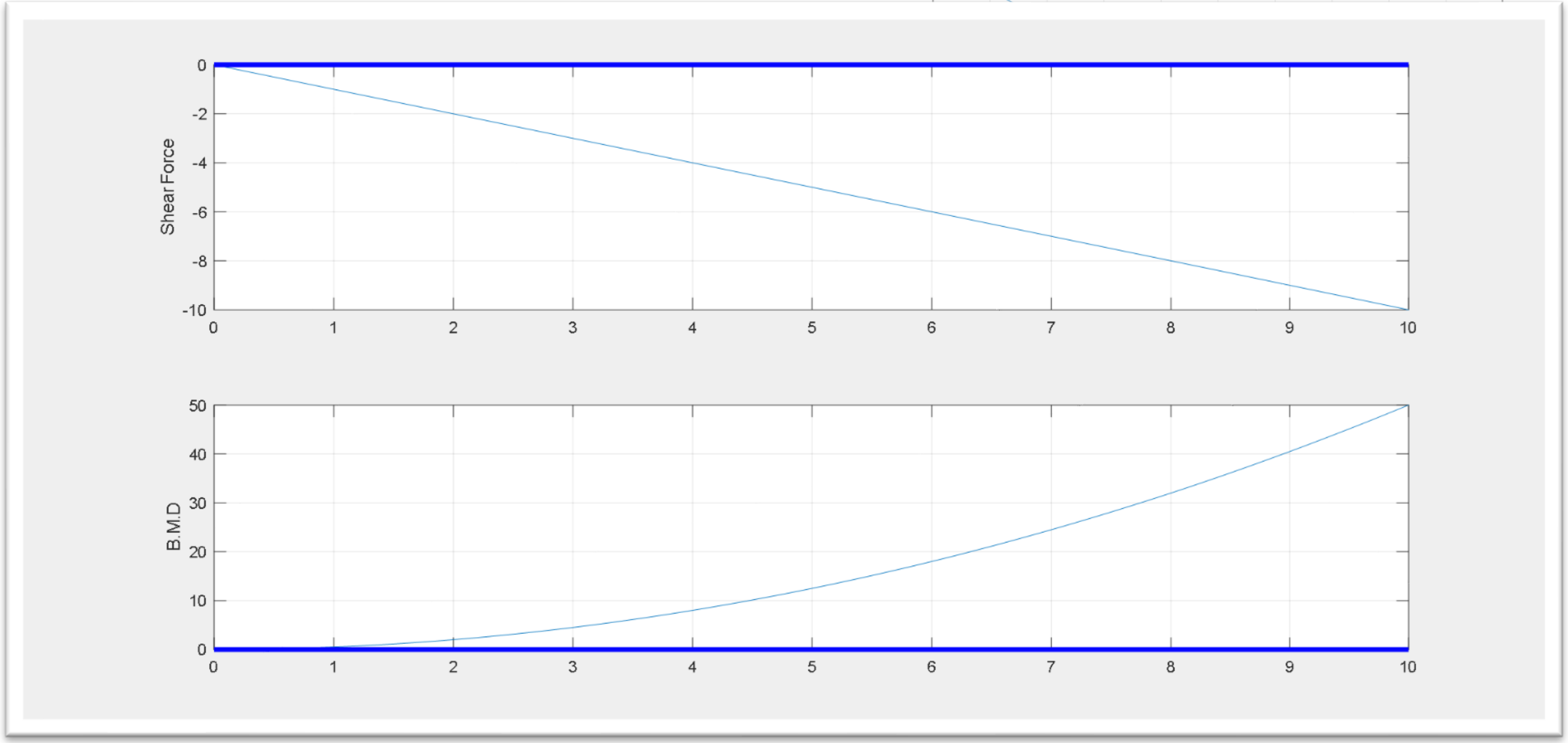
```
x = 1:2600;
y = zeros(size(x));

for kws = x
    y(kws) = ebill_ex(kws);
end

plot(x, y)

y2 = 25 * x;
hold on
plot(x, y2)
axis([1866 1869 46600 46800])
```



Example 3

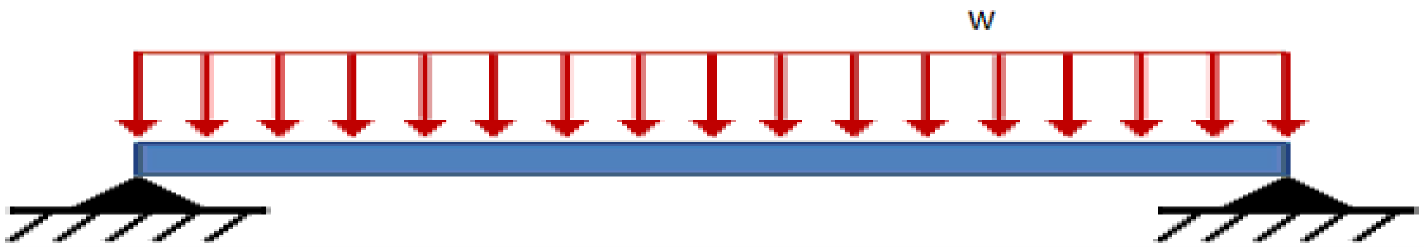


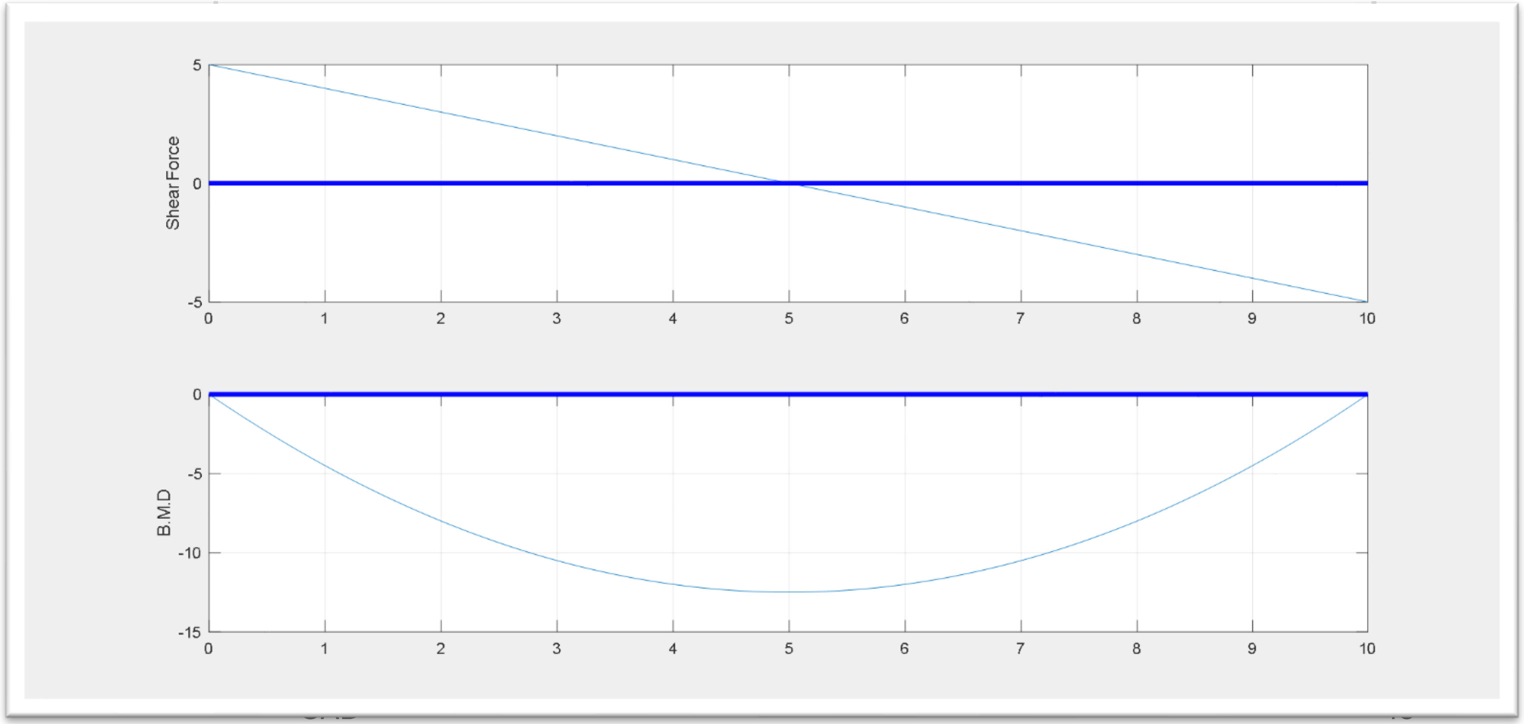

```
L = input('Length L='); % e.g. 10
w = input('w=');        % e.g. 1
x = 0:0.1:L;
v = -w*x;
m = w*x.*x/2;
subplot(2,1,1);
plot(x, v), grid
ylabel('Shear Force');
subplot(2,1,2);
plot(x, m), grid
ylabel('B.M.D');
```



Example 4

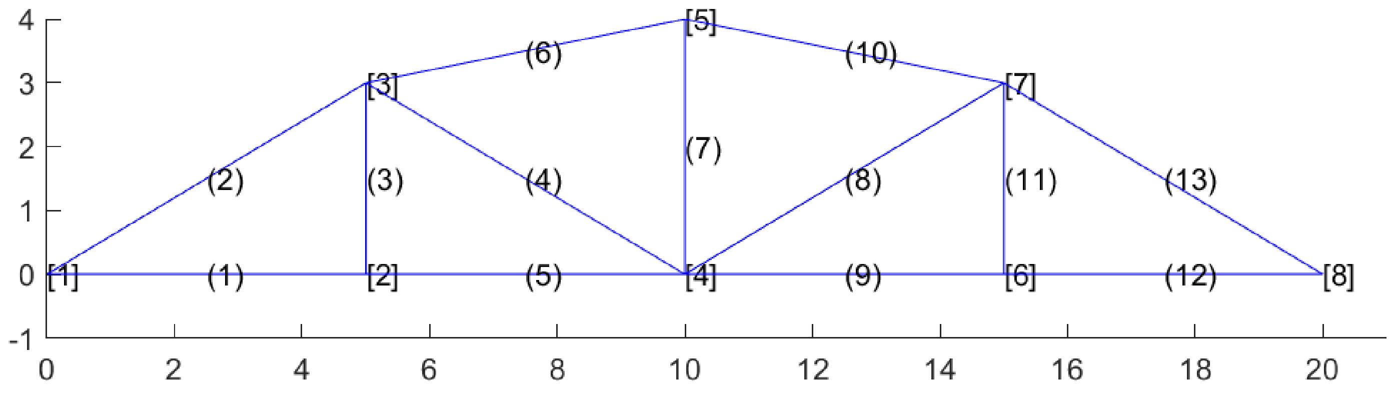
```
L = input('Length L='); % e.g. 10
w = input('w=');        % e.g. 1
x = 0:0.1:L;
R = w*L/2; % Support reaction
v = R-w*x;
m = -R.*x+w*x.*x/2;
subplot(2,1,1);
plot(x, v), grid
ylabel('Shear Force');
subplot(2,1,2);
plot(x, m), grid
ylabel('B.M.D');
```





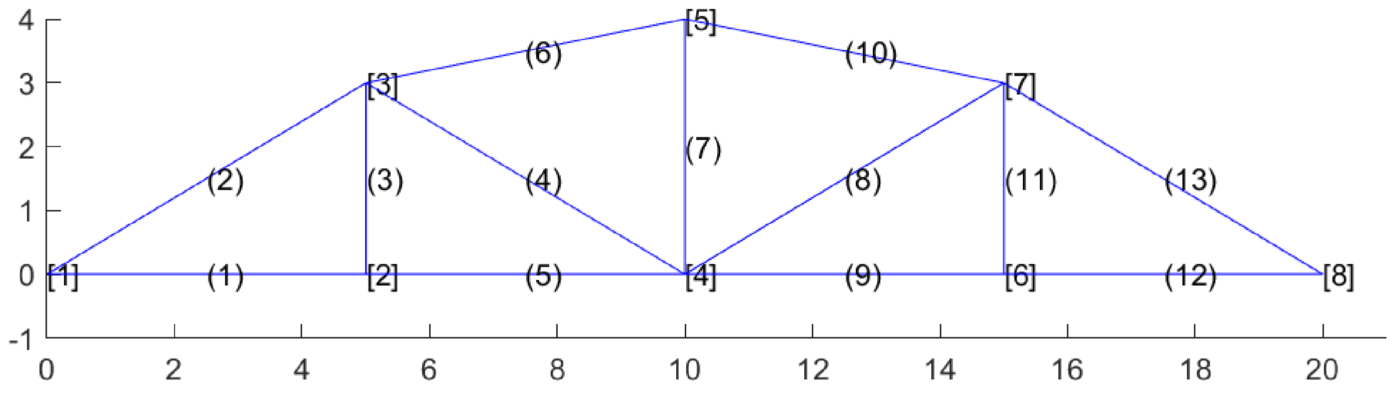
Truss

```
x0 = 0; y0 = 0; dx = 5; y1 = 3; y2 = 4;
% node x y
nodes = [
x0 + 0 * dx     y0;
x0 + 1 * dx     y0;
x0 + 1 * dx     y1;
x0 + 2 * dx     y0;
x0 + 2 * dx     y2;
x0 + 3 * dx     y0;
x0 + 3 * dx     y1;
x0 + 4 * dx     y0;
];
```



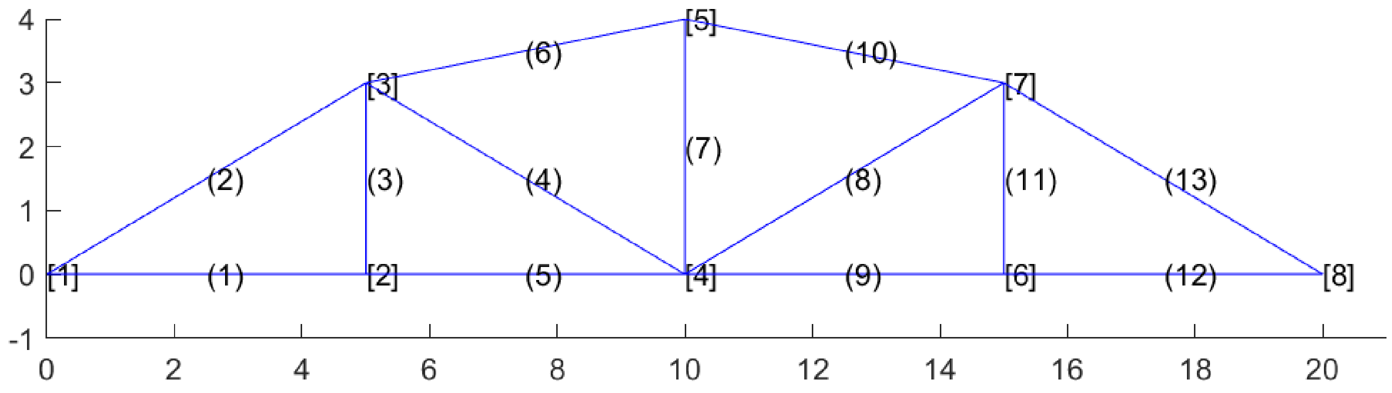
Truss

```
clf

axis equal

hold on



for n = 1:length(nodes)

    % x = nodes(n, 1)

    text(nodes(n, 1), nodes(n, 2), "["+num2str(n)+"]")
end
```



Truss

```
elements = [
1 2;
1 3;
2 3;
3 4;
2 4;
3 5;
4 5;
4 7;
4 6;
5 7;
6 7;
6 8;
7 8
];
```



Truss

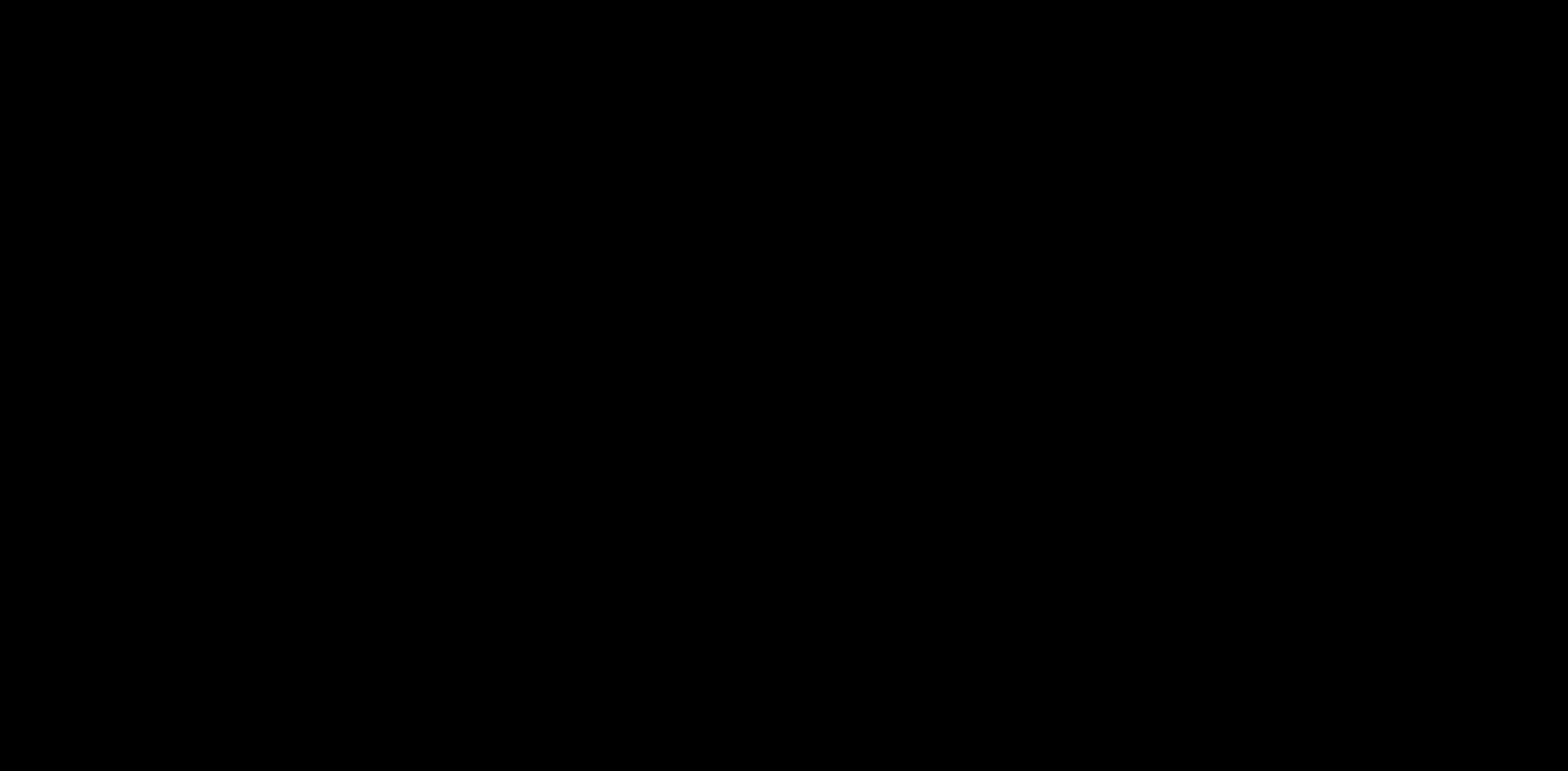
```
for el = 1:length(elements)
    n1 = elements(el, 1);
    n2 = elements(el, 2);
    plot([nodes(n1, 1) nodes(n2, 1)], [nodes(n1, 2) nodes(n2, 2)], 'b')
    x = (nodes(n1, 1) + nodes(n2, 1))/2;
    y = (nodes(n1, 2) + nodes(n2, 2))/2;
    text(x, y, "("+num2str(el)+")")
end
```

Truss

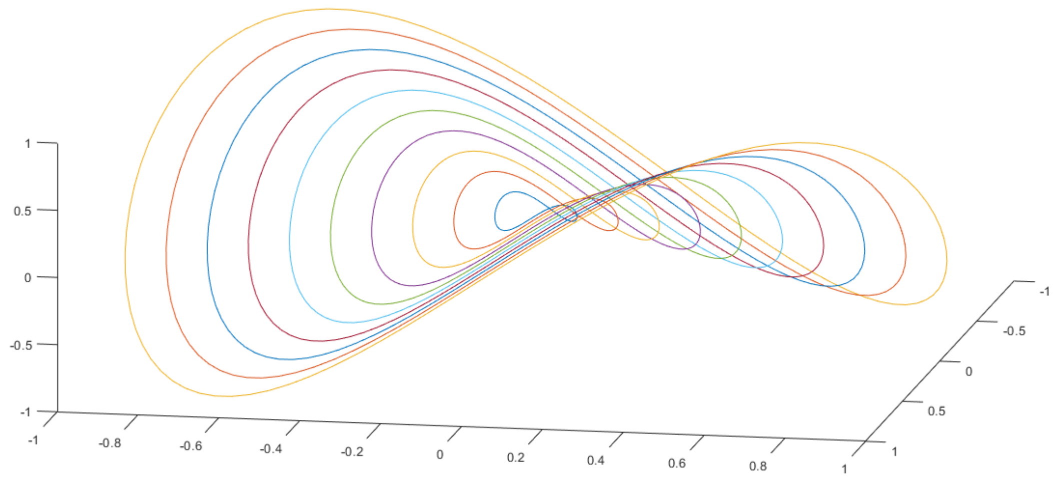
```
displacement = [
0 0;
0.0012 -0.0079;
0.0040 -0.0073;
0.0023 -0.0090;
0.0024 -0.0073;
0.0036 -0.0085;
0.0007 -0.0077;
0.0048 0; ];
```

Deformed Truss

```
max_scale = 100;
delay = 0.01;
for scale = 1:max_scale
    clf
    axis equal
    grid, hold on
    coord = nodes + scale .* displacement;
    for el = 1:length(elements)
        n1 = elements(el, 1);
        n2 = elements(el, 2);
        plot([coord(n1, 1) coord(n2, 1)], [coord(n1, 2) coord(n2, 2)], 'b')
    end
    axis([0 21 -2 y2])
    drawnow
    pause(delay)
end
```

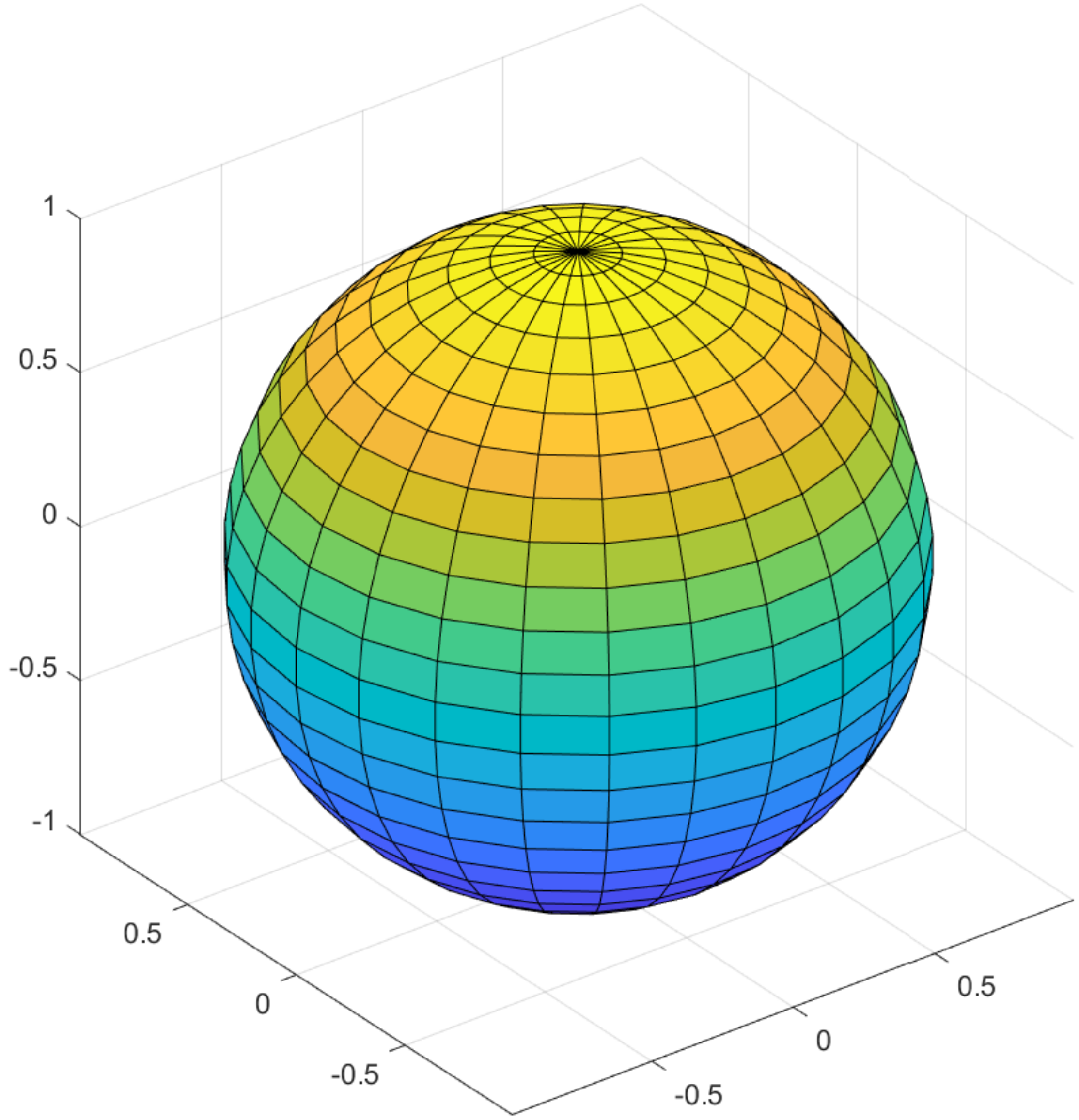


3D Graphics



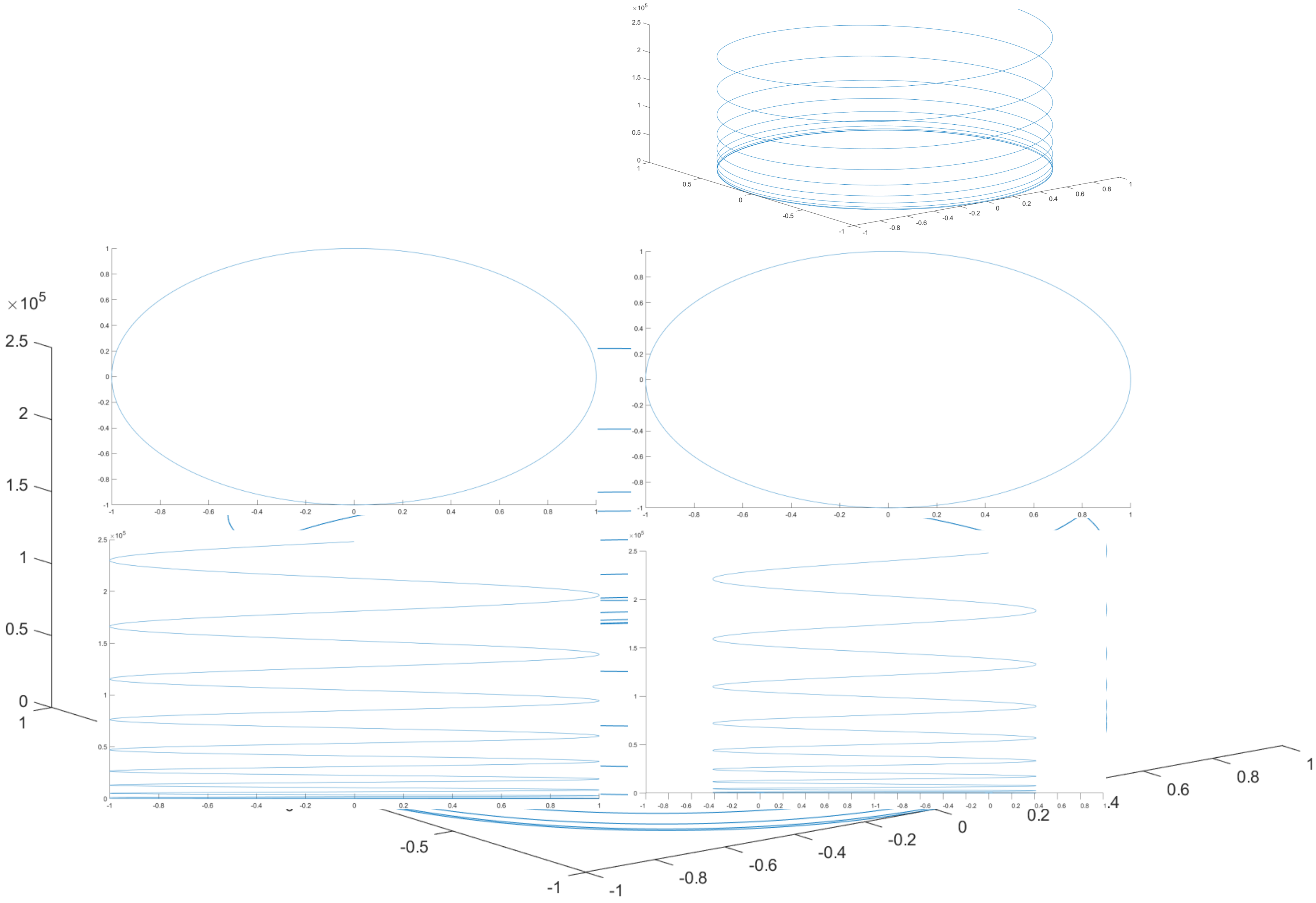
3D Graphics

```
sphere(25)

axis equal
```

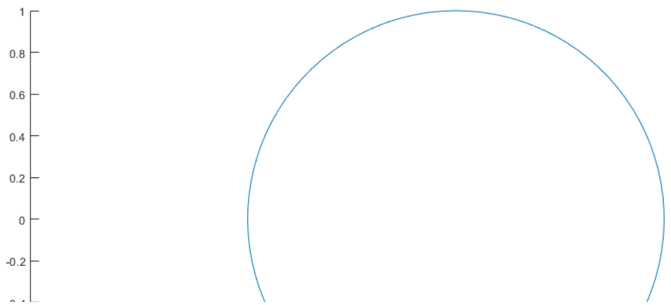


Plot3

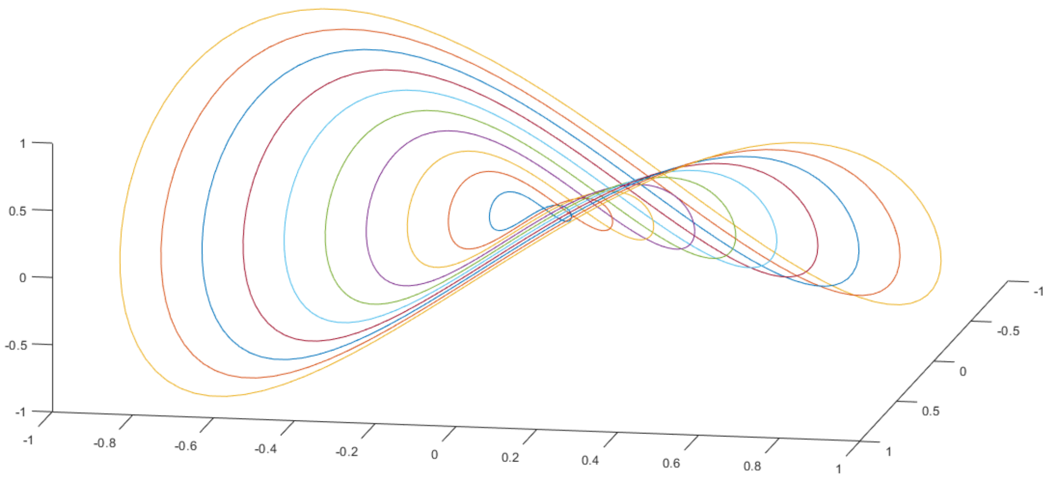
```
t=.01:.01:20*pi;
x=cos(t);
y=sin(t);
z=t.^3;

plot3(x,y,z)
view(2)
view(3)
view(45, 0)
view(90, 0)
view(0, 90)
```

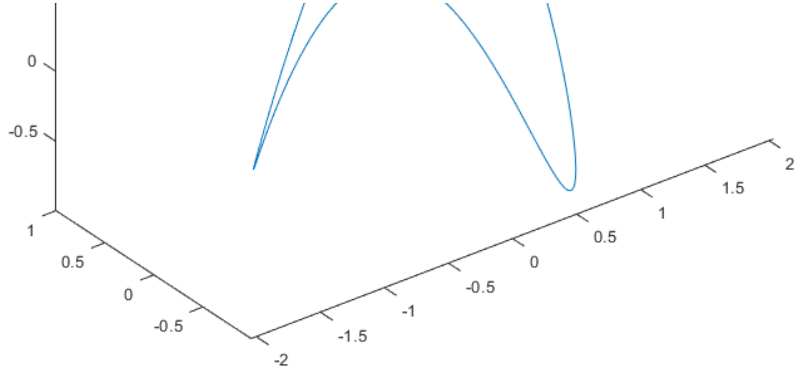


Plot3



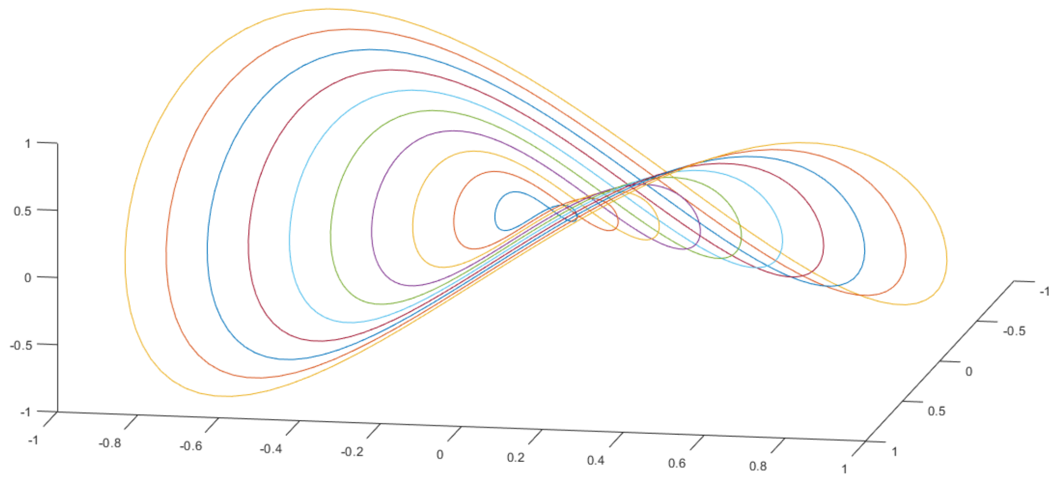

```
theta = linspace(0,2*pi,100);

    x = 1*sin(theta);
    y = 1*cos(theta);
    z = 1*sin(2*theta);
    plot3(x,y,z);

view(2)
axis equal
view(3)
```





Plot3

```
clf
hold on
theta = linspace(0,2*pi,100);
for r = 0:0.1:1
    x = r*sin(theta);
    y = r*cos(theta);
    z = r*sin(2*theta);
    plot3(x,y,z);
end
view(2)
axis equal
view(3)
daspect([1 1 3])
```

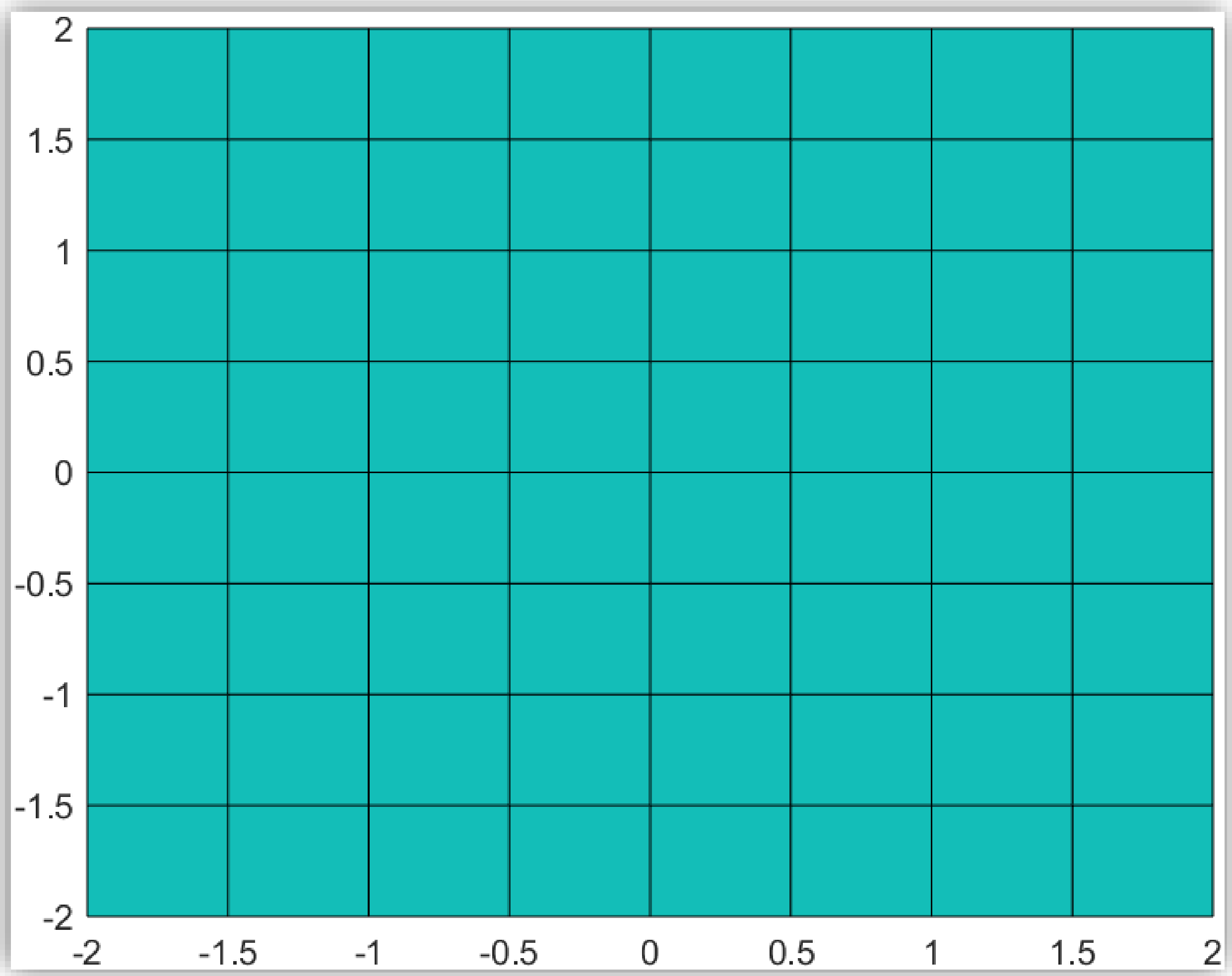


3D Graphics

```
[x, y]=meshgrid(-2:0.5:2);

x =

   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
   -2.0000   -1.5000   -1.0000   -0.5000         0    0.5000    1.0000    1.5000    2.0000
```

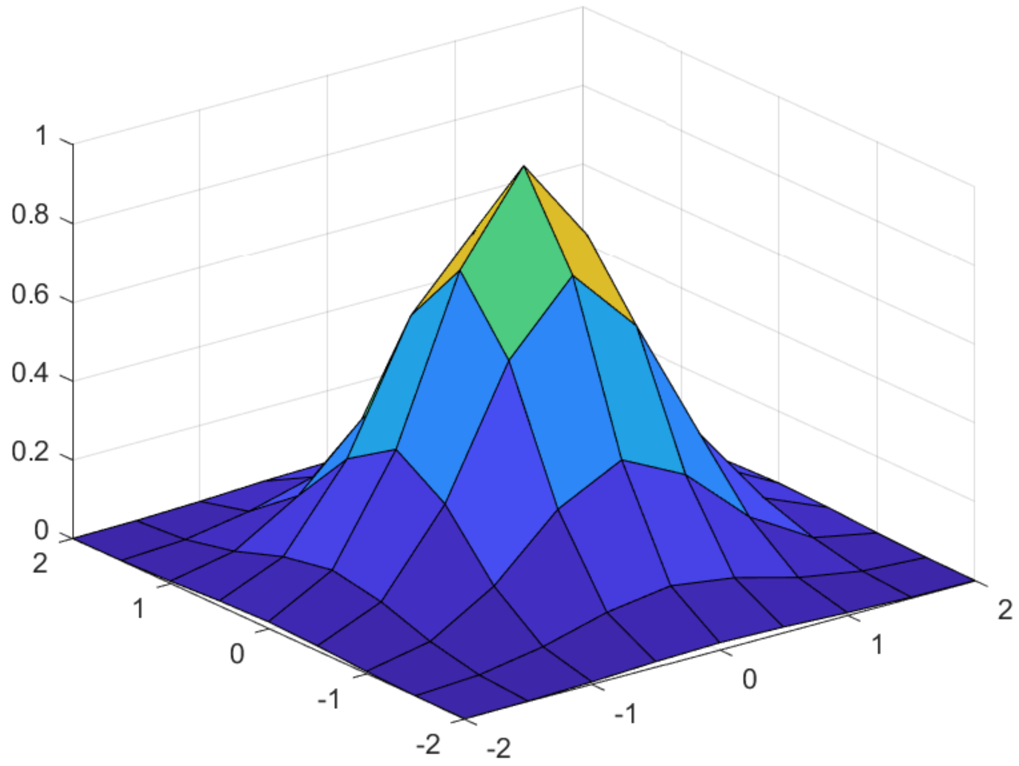
3D Graphics

```
[x, y]=meshgrid(-2:0.5:2);

y =

   -2.0000   -2.0000   -2.0000   -2.0000
   -1.5000   -1.5000   -1.5000   -1.5000
   -1.0000   -1.0000   -1.0000   -1.0000
   -0.5000   -0.5000   -0.5000   -0.5000
         0         0         0         0
    0.5000    0.5000    0.5000    0.5000
    1.0000    1.0000    1.0000    1.0000
    1.5000    1.5000    1.5000    1.5000
    2.0000    2.0000    2.0000    2.0000
```



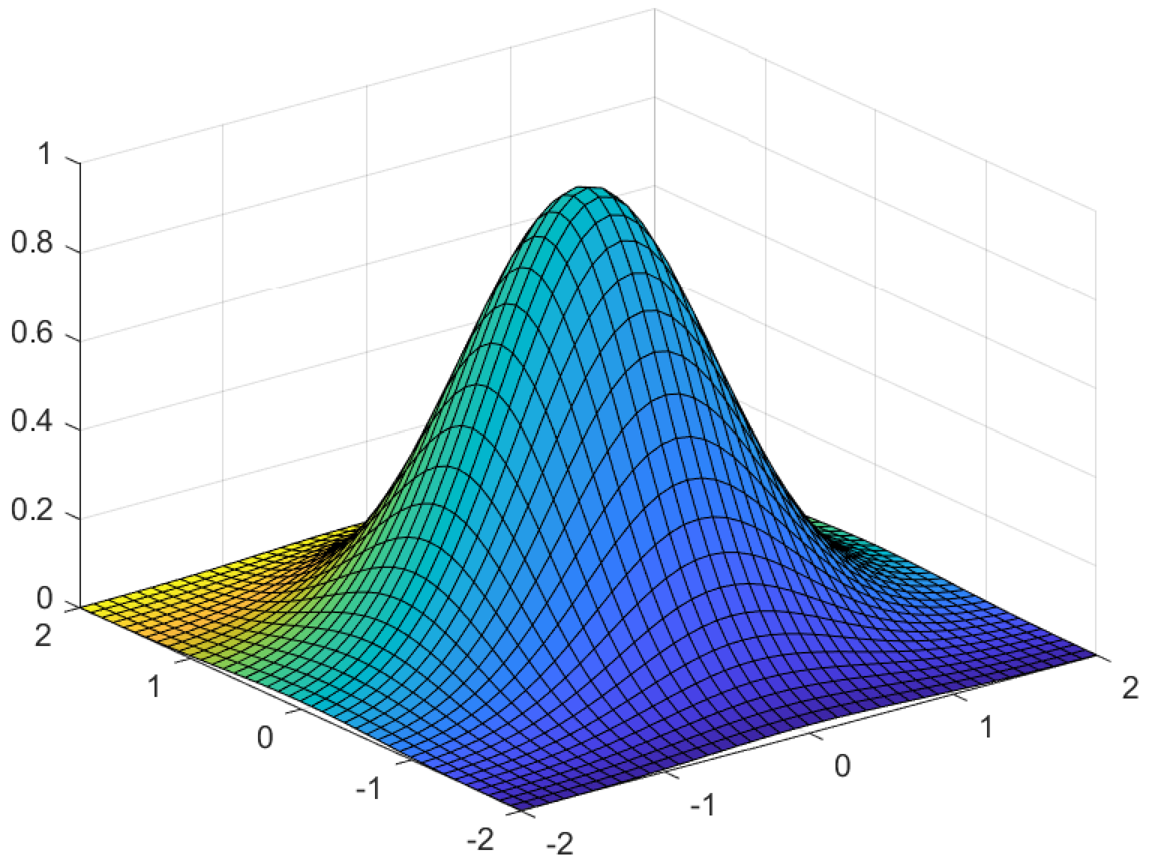
3D Graphics

```
[x, y]=meshgrid(-2:0.5:2);
z=exp(-x.^2 - y.^2);
surf(x, y, z)
```



3D Graphics

```
[x, y]=meshgrid(-2:0.1:2);
z=exp(-x.^2 - y.^2);
surf(x, y, z)
contour(x, y, z)
meshc(x, y, z)
view(0, -45)
surf(x, y, z, x)
surf(x, y, z, y)
```



3D Truss

- Nodes `[x y z]`
- `plot3`