

MATLAB: Introduction

Optimization Techniques

2019-1st semester

CONTENTS

- Basics
 - ✓ 변수 선언
 - ✓ 문법 사용
 - ✓ 행렬 연산
- Graph
 - ✓ 작성절차
 - ✓ 2D
 - ✓ 3D
- Symbolic Variable
- Graphical Solution

BASICS

- Basics
 - ✓ 변수 선언
 - ✓ 문법 사용
 - ✓ 행렬 연산
- Graph
 - ✓ 작성절차
 - ✓ 2D
 - ✓ 3D
- Symbolic Variable
- Graphical Solution

변수 선언

- 스칼라

- 변수에 바로 입력

```
>> a=5
```

```
a =
```

```
5
```

- 행렬

- 행은 세미콜론(:)이나 줄바꿈으로 구분
- 열은 쉼표(,)나 공백으로 구분

```
>> A=[1 2; 3 4]
```

```
>> A=[1,2;3,4]
```

```
>> A=[1 2
      3 4]
```

```
A =
```

```
1 2
```

```
3 4
```

- 문자열

- 작은따옴표(')로 둘러싸인 문자들은 문자열로 인식

```
>> a='c'
```

```
a =
```

```
d
```

- 벡터

- 행렬과 같은 방식으로 사용

```
>> A=[1, 2, 3, 4]
```

```
>> A=[1 2 3 4]
```

```
A =
```

```
1 2 3 4
```

```
>>A=[1; 2]
```

```
>>A=[1
      2]
```

```
A =
```

```
1
```

```
2
```

BASICS

- Basics
 - ✓ 변수 선언
 - ✓ **문법 사용**
 - ✓ 행렬 연산
- Graph
 - ✓ 작성절차
 - ✓ 2D
 - ✓ 3D
- Symbolic Variable
- Graphical Solution

문법 사용: IF 문

- MATLAB

```

if 수식 1
    문장 1;
elseif 수식 2
    문장 2;
else
    문장 3;
end

```

- C언어

```

if 수식 1{
    문장 1;}
else if 수식 2{
    문장 2;}
else{
    문장 3;}

```

- MATLAB 예제

```

a=3;
b=5;

if (a==b)
    s='a is same with b';
elseif (a<b)
    s='a is smaller than b';
else
    s='a is lager than b';
end

>>s
s =

a is smaller than b

```

문법 사용: SWITCH 문

– MATLAB

```
switch (수식)
  case 수식1,
    문장1;
  case 수식2,
    문장2;
  otherwise
    문장3;
end
```

– C언어

```
switch (수식) {
  case value1 : 문장1;
  case value2 : 문장2;
  default : 문장3;
}
```

– MATLAB 예제

```
a=9;

switch(a)
  case 8,
    x=8+a;
  case 9,
    x=9+a;
  otherwise,
    x=0;
end
```

```
>> x
x =
    18
```

문법 사용: WHILE 문

– MATLAB

```
while (수식)
    문장
end
```

– C언어

```
while (수식) {
    문장
}
```

– MATLAB 예제

```
s=0;
n=1;

while(n<=100)
    s=s+n;
    n=n+1;
end
```

```
>> s
s =
    5050
```


문법 사용: FOR 문

- MATLAB

```
for k=초기 값:증감 값:최
종 값,
    문장
end
```

- C언어

```
for (초기 값; 최종 값; 증
감 값){
    문장
}
```

- MATLAB 예제

```
n=0;
for k=1 : 2 : 100,
    n=n+k;
end

>> n
n =
    2500
```

BASICS

- Basics
 - ✓ 변수 선언
 - ✓ 문법 사용
 - ✓ **행렬 연산**
- Graph
 - ✓ 작성절차
 - ✓ 2D
 - ✓ 3D
- Symbolic Variable
- Graphical Solution

행렬 연산: 사칙 연산

- 더하기와 빼기

```
>> A=[1 2;3 4];
>> B=[5 6;7 8];
>> A+B
ans =
     6     8
    10    12
```

- 곱하기

```
>> A=[1 2;3 4];
>> B=[5 6;7 8];
>> A*B
ans =
    19    22
    43    50
```

- 나누기

- $X=A^{-1}B$ 를 $A\backslash B$ 로 표현 가능

```
>> A=[1 2;3 4];
>> B=[5;6];
>> X=A\B
ans =
    -4
     4.5
```

- $X=BA^{-1}$ 를 $B./A$ 로 표현 가능

```
>> A=[1 2;3 4];
>> B=[5;6];
>> X=A./B
ans =
    -4
     4.5
```

행렬 연산: 배열 연산

- 연산자에 '.' 를 붙이면 각 요소가 일대 일로 계산됨

```
>> A=[1 2;3 4];
>> A^2
ans =
    7    10
   15    22
```

```
>> A=[1 2;3 4];
>> A.^2
ans =
    1     4
    9    16
```

```
>> A=[1 2;3 4];
>> B=[5 6;7 8];
>> A*B
ans =
   19    22
   43    50
```

```
>> A=[1 2;3 4];
>> B=[5 6;7 8];
>> A.*B
ans =
    5    12
   21    32
```

행렬 연산: 기본 행렬

- 빈 행렬(empty matrix)

```
>> a=[]
ans =
     []
```

- 모든 요소가 1인 행렬
(matrix of ones)

```
>> ones(2)      >> ones(1,2)
ans =           ans =
     1     1         1     1
     1     1         1     1
```

- 영 행렬(zero matrix)

```
>> zeros(2)      >> zeros(1,2)
ans =            ans =
     0     0         0     0
     0     0         0     0
```

- 단위 행렬(identity matrix)

```
>> eye(2)        >> eye(1,2)
ans =            ans =
     1     0         1     0
     0     1         0     0
```

행렬 연산: 명령어 (1)

– sum(요소의 합)

```
>> a=[1 2;3 4];
>> sum(a)
ans =
     4     6
```

– inv(역행렬 계산)

```
>> a=[1 2;3 4];
>> inv(a)
ans =
    -2.0000    1.0000
     1.5000   -0.5000
```

– det(determinant 계산)

```
>> a=[1 2;3 4];
>> det(a)
ans =
    -2
```

– eig(eigenvalue 계산)

```
>> a=[1 2;3 4];
>> eig(a)
ans =
    -0.3723
     5.3723
```

행렬 연산: 명령어 (2)

- a'(전치행렬)

```
>> a=[1 2;3 4];  
>> a'  
ans =  
     1     3  
     2     4
```

- a(2,2): 2행 2열을 출력함

```
>> a=[1 2;3 4];  
>> a(2,2)  
ans =  
     4
```

- a(2,:): 2행 모두를 출력

```
>> a=[1 2;3 4];  
>> a(2,:)  
ans =  
     3     4
```

행렬 연산: 분해와 조립

- 분해

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \longrightarrow B = \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$$

$$\gg B = A(2:3, 2:3)$$

$$B = \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$$

- 조립

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 0 & 0 \\ 7 & 0 & 0 \end{bmatrix} \longleftarrow B = \begin{bmatrix} 5 & 6 \\ 8 & 9 \end{bmatrix}$$

$$\gg A(2:3, 2:3) = B$$

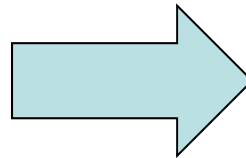
$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

행렬 연산: 3차원 행렬

>> A(:, :, 1)=[1,2;3,4];

>> A(:, :, 2)=[5,6;7,8];

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

GRAPH

- Basics
 - ✓ 변수 선언
 - ✓ 문법 사용
 - ✓ 행렬 연산
- Graph
 - ✓ **작성절차**
 - ✓ 2D
 - ✓ 3D
- Symbolic Variable
- Graphical Solution

작성절차

- Plot 할 데이터 준비
- Single or multi graphic 결정
- Plot 함수 호출

- Line 또는 market의 속성 설정
- 축의 범위 설정
- 그래픽 객체에 labeling
- 파일로 출력

GRAPH

- Basics
 - ✓ 변수 선언
 - ✓ 문법 사용
 - ✓ 행렬 연산
- Graph
 - ✓ 작성절차
 - ✓ **2D**
 - ✓ 3D
- Symbolic Variable
- Graphical Solution

2D: PLOT

- 작성 예제

```

% 출력할 data
x=0:0.01:10;
y=sin(x).*cos(x+pi/2);

% plot함수 호출
plot(x,y)

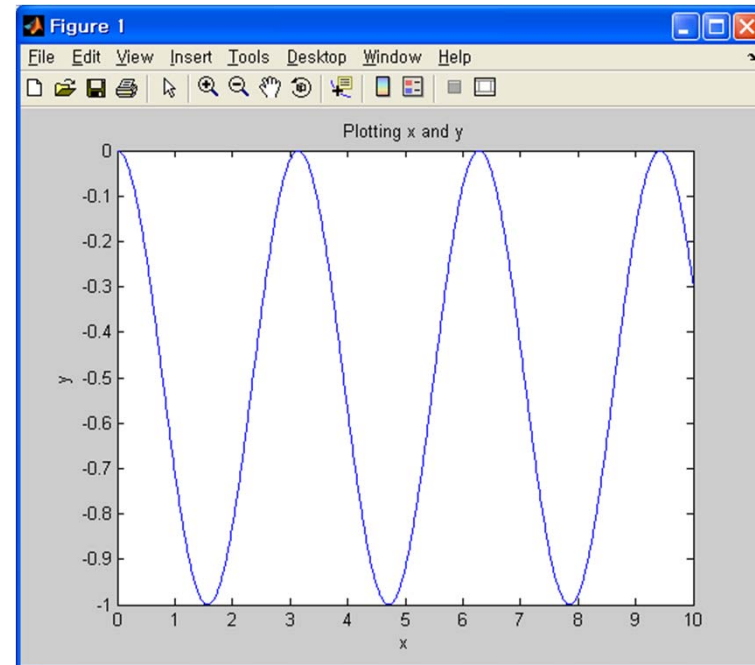
% 속성을 미리 입력해서 사용 가능
% plot(x,y,'b--')

% line 속성
set(plot(x,y),'LineStyle','--')

% labeling
xlabel('x');
ylabel('y');
title('Plotting x and y');

% ex1.bmp파일로 출력
print -dbitmap ex1.bmp

```

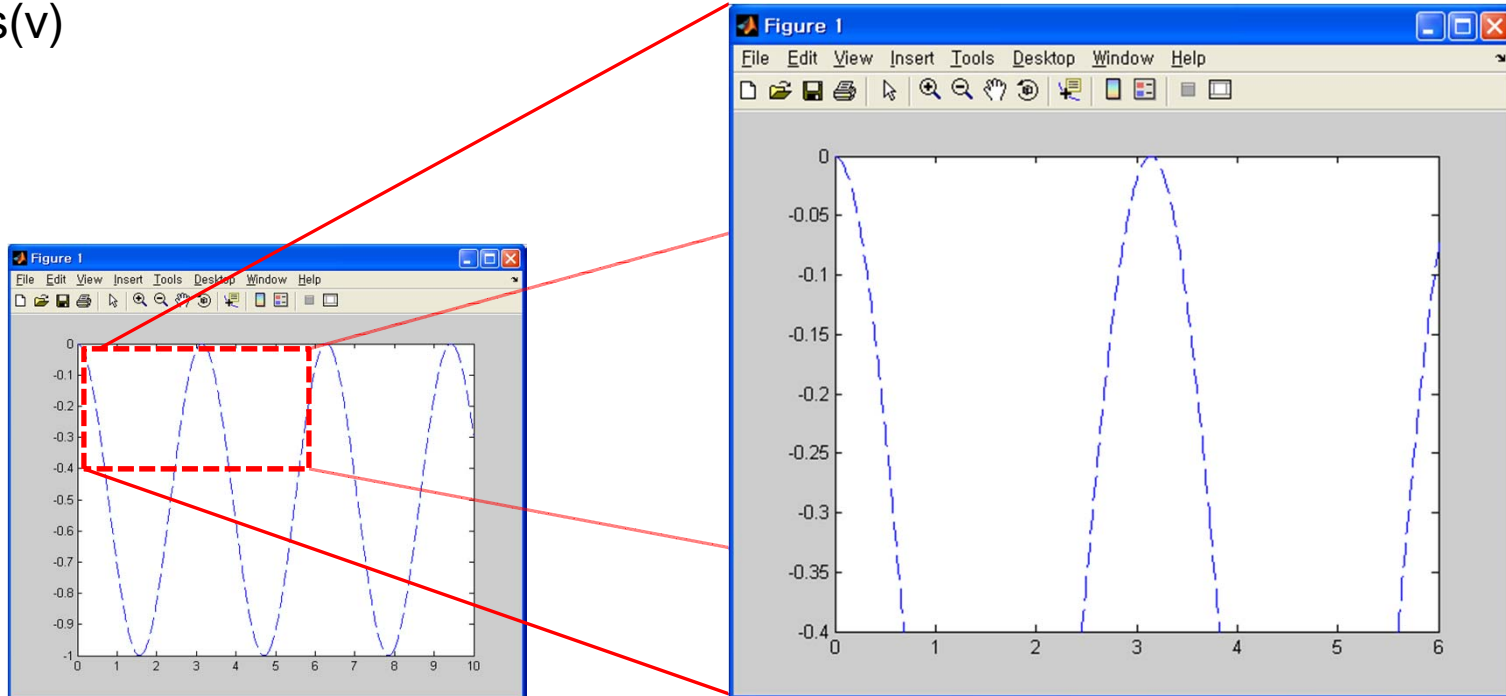


2D: AXIS

– Graph의 범위 변경

% x축은 0 ~ 6 y축은 -0.4 ~ 0

```
v=[0,6,-0.4,0];
axis(v)
```



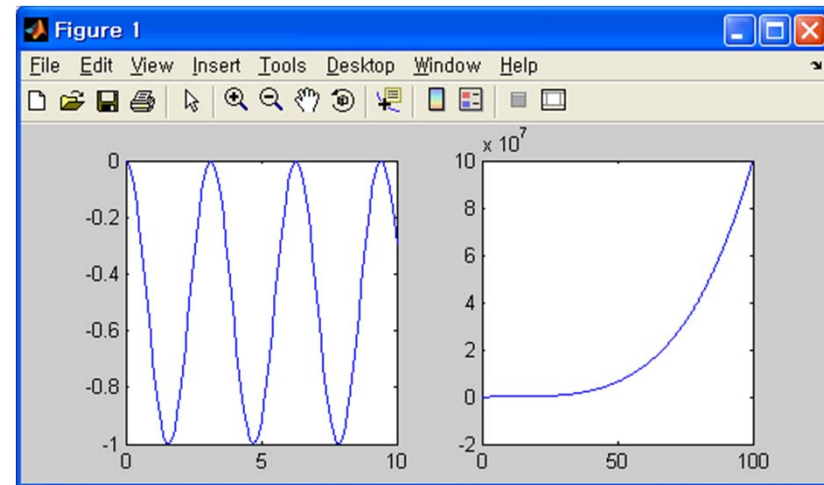
2D: MULTI PLOT

– Subplot(mni)

- $m \times n$ 으로 나누어진 영역 중 i 번째 영역에 그래프를 출력

```
x1=0:0.1:10;
f1=sin(x1).*cos(x1+pi/2);
subplot(121)
plot(x1,f1)
```

```
x2=0:0.1:100;
f2=x2.^4-0.4*x2.^2-30*x2+5;
subplot(122)
plot(x2,f2)
```



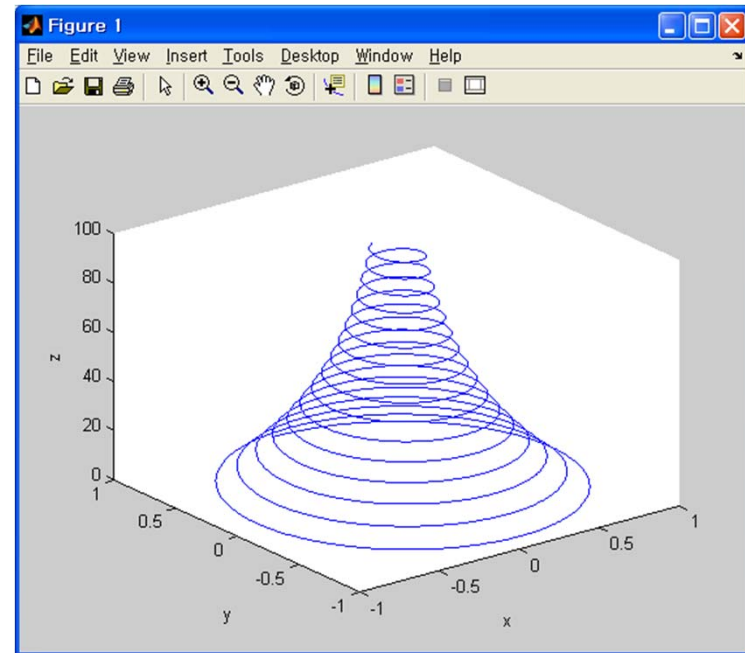
GRAPH

- Basics
 - ✓ 변수 선언
 - ✓ 문법 사용
 - ✓ 행렬 연산
- Graph
 - ✓ 작성절차
 - ✓ 2D
 - ✓ **3D**
- Symbolic Variable
- Graphical Solution

3D: PLOT3

- 2d plot을 3d로 확장

```
t=0:0.01:100;  
x=exp(-0.02*t).*sin(t);  
y=exp(-0.02*t).*cos(t);  
  
plot3(x,y,t);
```



3D: CONTOUR

- 등위면을 plot

```
x=0:0.1:10;
```

```
y=0:0.1:10;
```

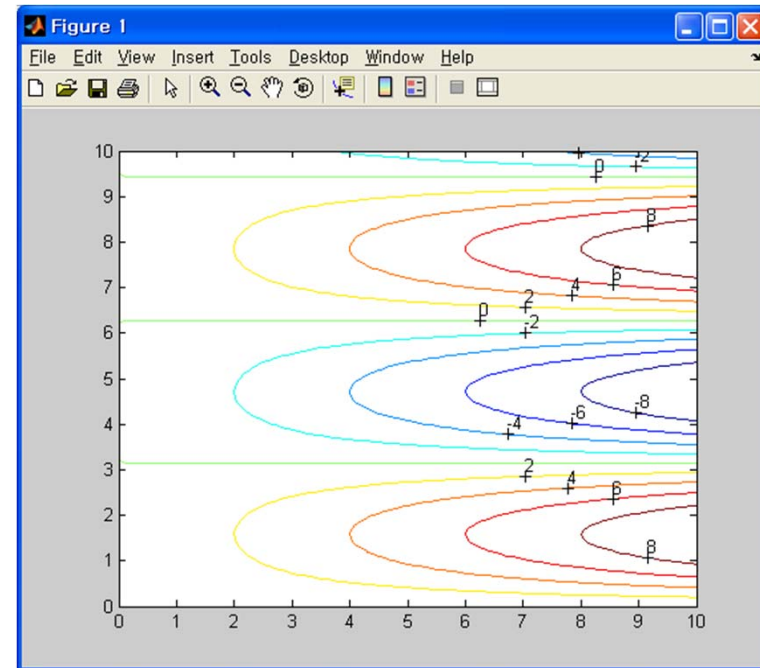
```
% x,y 벡터를 X,Y 행렬로 변환  
[X,Y]=meshgrid(x,y);
```

```
Z=X.*sin(Y);
```

```
c=contour(X,Y,Z);
```

```
% contour3은 3차원 contour
```

```
% 윤곽선에 높이 라벨을 추가  
clabel(c);
```

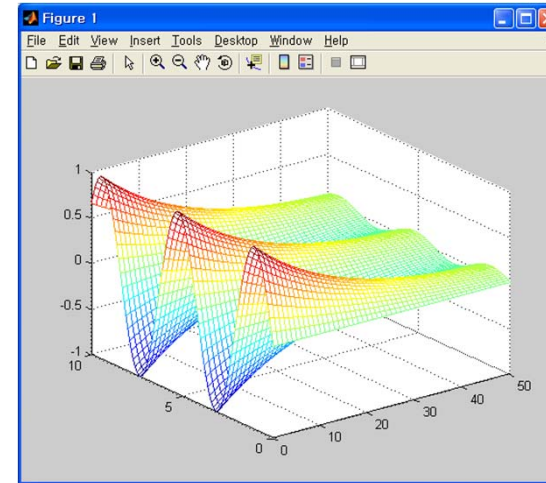


3D: MESH & SURF

- 격자로 이루어진 그래프

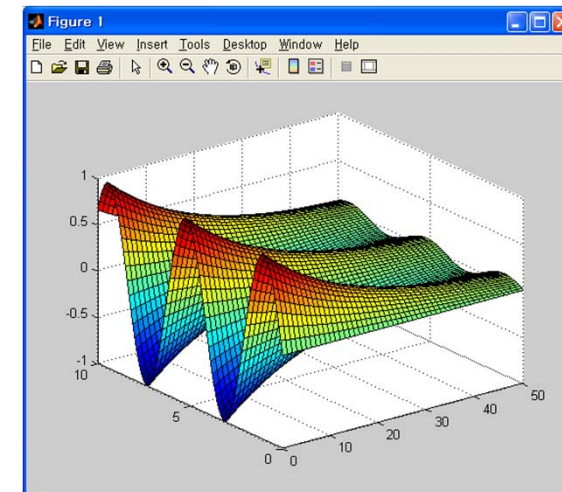
```
x=0:1:50;
y=0:0.1:10;
[X,Y]=meshgrid(x,y);
Z=exp(-0.05*X).*sin(1.5*Y);
```

```
mesh(X,Y,Z)
```



- 면으로 이루어진 그래프

```
surf(X,Y,Z)
```



SYMBOLIC VARIABLE

- Basics
 - ✓ 변수 선언
 - ✓ 문법 사용
 - ✓ 행렬 연산
- Graph
 - ✓ 작성절차
 - ✓ 2D
 - ✓ 3D
- **Symbolic Variable**
- Graphical Solution

SYMBOLIC: 변수 선언

- 수치 데이터가 아닌 기호의 연산이 가능

```
>> a=sym('x')
a =
    x
```

```
>> b=sym('x^2+sin(y)')
b =
    x^2+sin(y)
```

```
>> c=a+b
c =
    x+x^2+sin(y)
```

```
>> syms x y
>> z=x+y*i
>> z^2
ans =
    (x+i*y)^2
```

```
>> expand(ans)
ans =
    x^2+2*i*x*y-y^2
```

SYMBOLIC: 연산 (1)

- 변수 선언 외에도 MATLAB 함수를 사용한 연산 가능

```
% symbolic 객체로 이루어진 행렬
>> syms x y
>> A=[cos(x),-sin(x);sin(x),cos(x)];
>> B=[y,0;0,-y];
```

```
>> A
A =
[ cos(x), -sin(x)]
[ sin(x), cos(x)]
```

```
>> B
B =
[ y, 0]
[ 0, -y]
```

```
>> A+B
ans =
[ cos(x)+y, -sin(x)]
[ sin(x), cos(x)-y]
```

```
>> A-B
ans =
[ cos(x)-y, -sin(x)]
[ sin(x), cos(x)+y]
```

```
>> eig(A)
ans =
cos(x)+(cos(x)^2-1)^(1/2)
cos(x)-(cos(x)^2-1)^(1/2)
```

SYMBOLIC: 연산 (2)

- 방정식의 해 구하기

- fzero - 1변수 방정식의 해

```
>> x0=2; %초기값
>> x=fzero('fun',x0)
x =
    1.5708
```

```
function f=fun(x)
f=(sin(x)^2+cos(x)-1)/(12+2*sin(x))^4;
```

- fsolve - 연립 방정식의 해

```
>> x0=[-5;-5]; %초기값
>> x=fsolve('fun2',x0)
x =
    0.5671
    0.5671
```

```
function f=fun2(x)
f=[2*x(1)-x(2)-exp(-x(1));-x(1)+2*x(2)-exp(-x(2))];
```

- solve

```
>> syms x
>> f=x^2-x+3*sqrt(2*x^2-3*x+2)-x/2-7;

>> x=solve(f)
x =
    -1/2
     2
```

% symbolic객체 x를 수치 데이터로 변환

```
>> double(x)
ans =
    -0.5000
     2.0000
```

```
>> syms t q r
solve('t*q-t/r*q^2-2*r*q=0','t')
```

```
ans =

-(2*r^2)/(q - r)
```

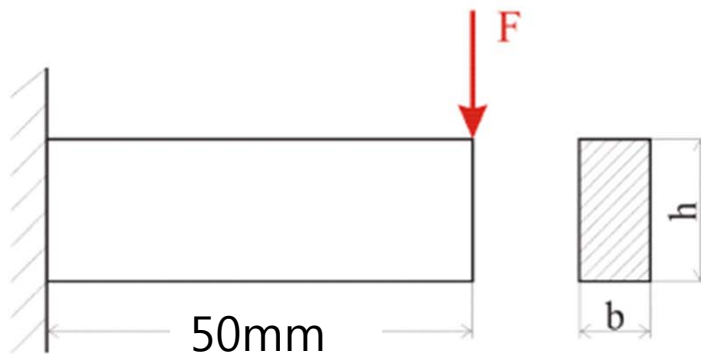
GRAPHICAL SOLUTION

- Basics
 - ✓ 변수 선언
 - ✓ 문법 사용
 - ✓ 행렬 연산
- Graph
 - ✓ 작성절차
 - ✓ 2D
 - ✓ 3D
- Symbolic Variable
- **Graphical Solution**

EXAMPLE

A **cantilever beam** loaded with force $F=24000$ N. Where the cross-section parameters: Width $b_{[20,40]}$ and height $h_{[30,90]}$ can vary on their range to minimize the beam weight, subject to these constraint:

- 1) Max normal stress can not exceed the σ_{\max} value,
- 2) Max shear stress can not exceed the τ_{\max} and
- 3) Height h should not be larger than twice the width b .



PROBLEM FORMULATION

Mathematically this problem can be stated as:

Objective: $\min \text{Weight}(b,h)$

Design Variables: $b_L < b < b_U, \quad 20 < b < 40$ [mm]

$h_L < h < h_U, \quad 30 < h < 90$ [mm]

Design Constraints: $\sigma(b,h) = 6F/(bh^2) \leq \sigma_{\max}, \quad \text{with} \quad \sigma_{\max} = 70 \text{ MPa}$

$\tau(b,h) = F/(bh) \leq \tau_{\max}, \quad \text{with} \quad \tau_{\max} = 15 \text{ MPa}$

$h \geq 2*b$

MATLAB CODE

- 정식화 내용을 m-file에 아래와 같이 표현

```
[b,h]=meshgrid(0:0.5:50,0:0.5:100);
```

```
f=b.*h;
```

```
g1=(6*24000*50)./(b.*(h.^2))-70;  
g2=24000./(b.*h)-15;
```

```
g3=2*b-h;
```

```
g4=b-40;  
g5=-b+20;  
g6=h-90;  
g7=-h+30;
```

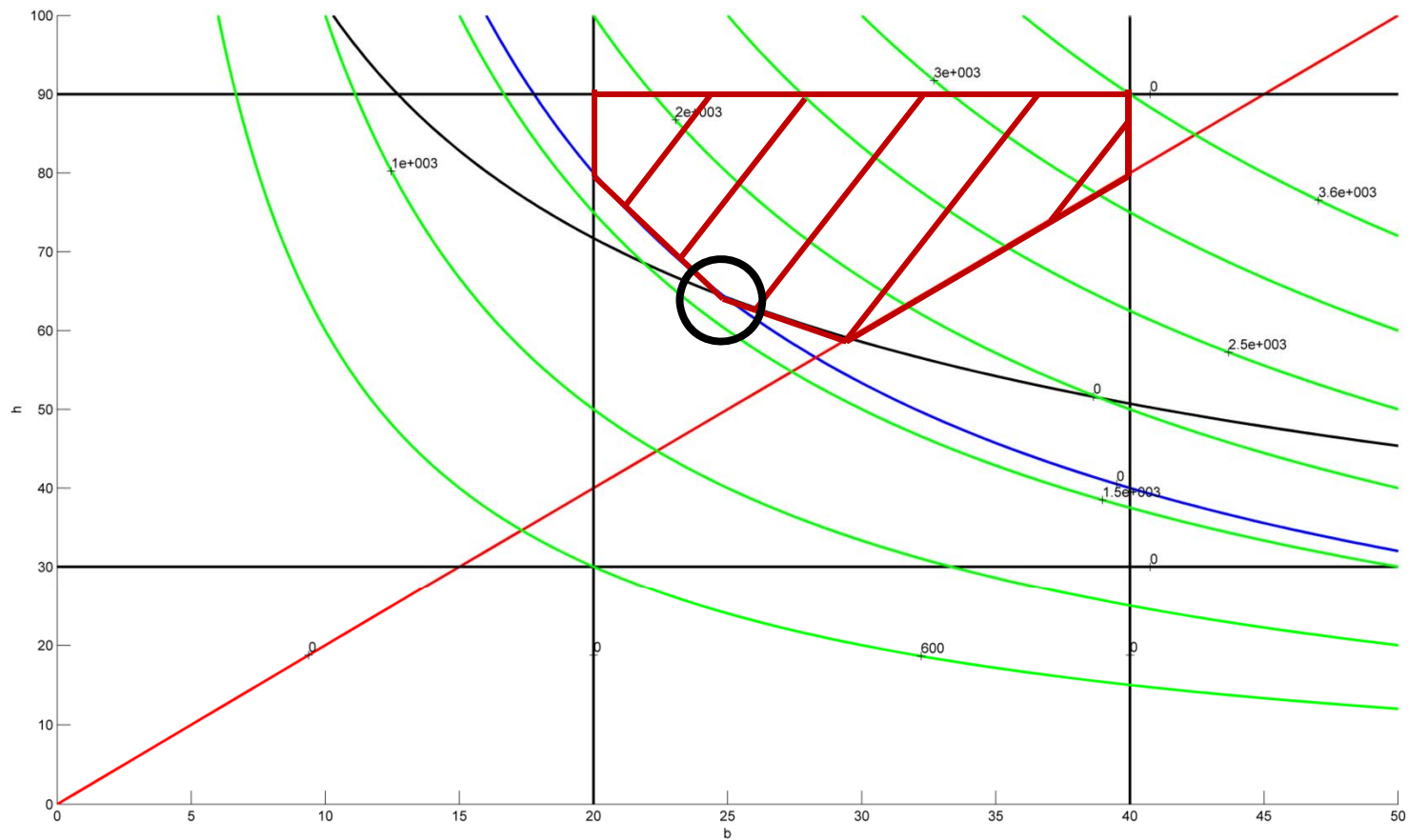
PLOT: CONTOUR (1)

- 작성할 그래프를 Contour 함수로 표현

```
cla reset
axis auto
xlabel('b'), ylabel('h')
hold on
cv1=[0 1];
const1=contour(b,h,g1,[0 0],'k','linewidth',2);
clabel(const1)
const2=contour(b,h,g2,[0 0],'b','linewidth',2);
clabel(const2)
const3=contour(b,h,g3,[0 0],'r','linewidth',2);
clabel(const3)
const4=contour(b,h,g4,[0 0],'k','linewidth',2);
clabel(const4)
const5=contour(b,h,g5,[0 0],'k','linewidth',2);
clabel(const5)
const6=contour(b,h,g6,[0 0],'k','linewidth',2);
clabel(const6)
const7=contour(b,h,g7,[0 0],'k','linewidth',2);
clabel(const7)
const8=contour(b,h,f,[600 1000 1500 2000 2500 3000 3600],'g','linewidth',2);
clabel(const8)
```

PLOT: CONTOUR (2)

– Feasible region



FSOLVE

- fsolve 함수를 이용하여 교점을 찾음

```
% function을 다음과 같이 먼저 정의
function f=fopt(x)
f=[(6*24000*50)./(x(1).*(x(2).^2))-70;24000./(x(1).*x(2))-15];
```

```
% 초기값은 25,60으로 설정
>>x0=[25,60];
>>x=fsolve('fopt',x0)
```

<pre>x = 24.8889 64.2857</pre>	→	<pre>목적함수 1600</pre>
-----------------------------------	---	----------------------