Lecture 1

Introduction to MATLAB

Ye Ding (丁 烨)

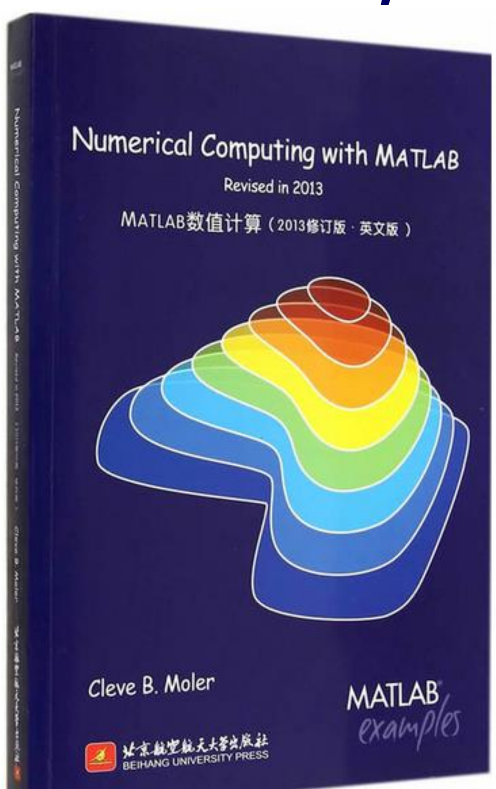
Email: y.ding@sjtu.edu.cn

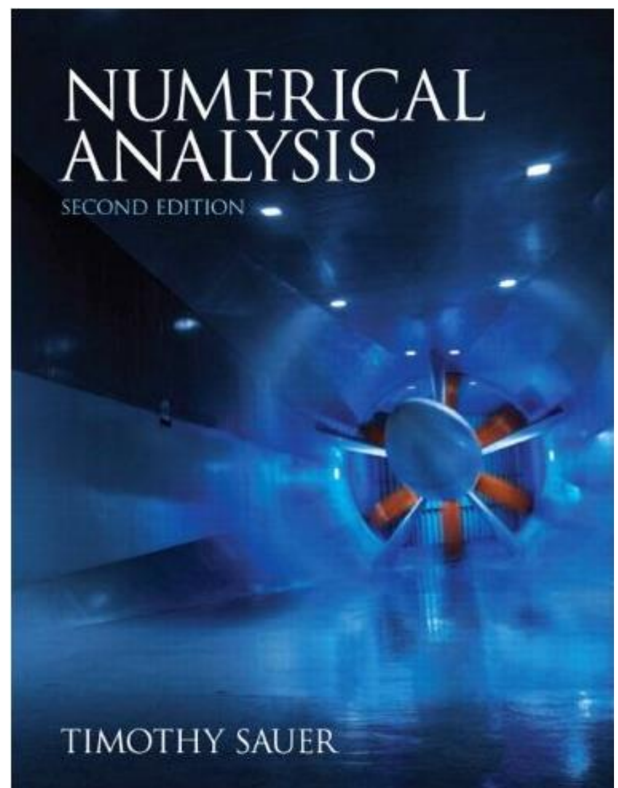
School of Mechanical Engineering

Shanghai Jiao Tong University

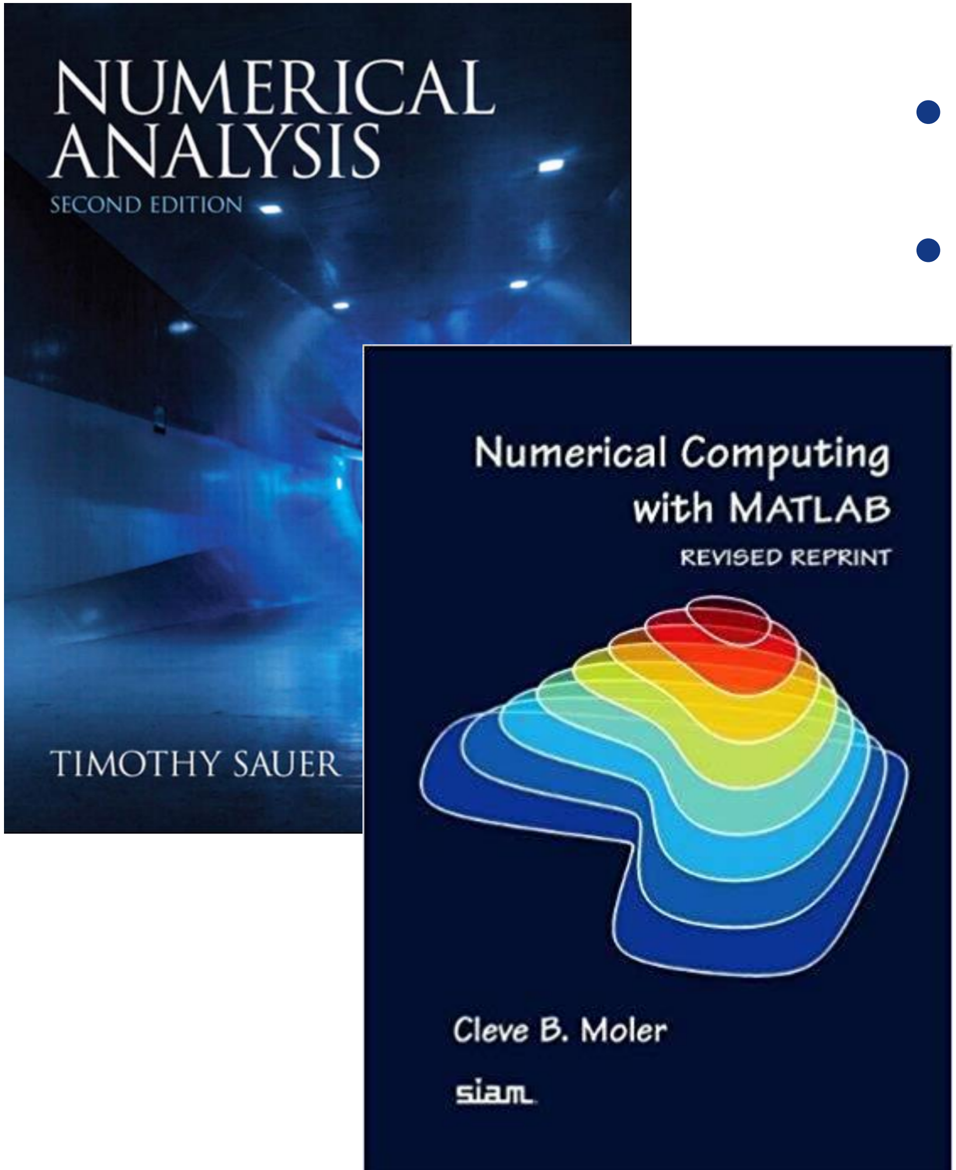
References

- **Cleve Moler, *Numerical Computing with MATLAB*. Society for Industrial and Applied Mathematics, 2004.**
- **Timothy Sauer, *Numerical analysis* (2nd ed.), Pearson Education, 2012.**





电子教材使用注意事项



- 建议阅读纸质版
- 对于疫情期间无法获得纸质版的同学，可以下载电子版用于本人使用（由于版权问题，请大家务必不要扩散）

About this Course

1. Where to download the PPT slides?

Canvas系统：

https://oc.sjtu.edu.cn/courses/29735

2. Teaching Assistant

- **张炜佳** 787861909@sjtu.edu.cn
- **陈永学** cyxsjtu@sjtu.edu.cn

3. QQ

702026916，“Matlab2021课程群”

About this Course

4. 成绩构成

- 期末考试：40%
- 课程设计（project）：25%
- 平时作业：25%
- 课堂测试：10%

Project (Teamwork)

1. 课题来源

- 机器人、信号处理相关的推荐主题
- 科创项目

2. 人员分组

- 8组（3人/组 ≤ 人数 ≤ 4人/组）；自由组合

3. 课程设计讨论

- 第9、12 、16周，周五，6-8节；地点：教室

Lectures

1. Basic of MATLAB
2. Programming with MATLAB

□ Algorithms

3. Solving Nonlinear Equations
4. Linear Equations
5. Interpolation
6. Least Squares
7. Differentiation and Integration
8. Ordinary Differential Equations
9. Boundary Value Problems
10. Partial Differential Equations
11. Optimization
12. Fourier Transform
13. Eigenvalues

Variables and Operators

Variables in MATLAB

A variable is created simply by directly allocating a value to it.

```
>> x = 3
x =
    3
```

Once the variable is declared, we can use it

```
>> x^3
ans =
   27
```

Variables and Operators

Variables in MATLAB

A variable is created simply by directly allocating a value to it.

```
>> x = 2 + 3*i
x =
   2.0000 + 3.0000i
```

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

Variables and Operators

Variables in MATLAB

Once the variable is declared, we can use it

```
>> y = x + 2
y =
     5
```

```
>> y = x/0
y =
   Inf
```

Variables and Operators

Functions

MATLAB contains a vast collection of built-in functions from elementary functions like sine and cosine, to more sophisticated functions like matrix inversions, matrix eigenvalues, and fast Fourier transforms.

cos(x)	cosine of x
sin(x)	sine of x
tan(x)	tangent of x
sqrt(x)	square root of x
abs(x)	absolute of x
exp(x)	exponential of x
log(x)	log to the base e of x
asin(x)	inverse sine function of x

Variables and Operators

Variables in MATLAB

Once the variable is declared, we can use it

```
>> y = 5
y =
    5
```

```
>> z = exp(5)
z =
    1.484131591025766e+02
```

Variables and Operators

- **Variables in MATLAB**

Once the variable is declared, we can use it

```
>> x = pi
x =
   3.141592653589793
```

```
>> y = sin(pi)
y =
    1.224646799147353e-16
```

Variables and Operators

Vectors in MATLAB

A row vector variable of *n* elements can be defined in MATLAB:

V = [v1, v2, v3,..., vn]

V = [v1 v2 v3... vn]

```
>> x = [1,2,5,7,8]
x =
     1     2     5     7     8
```

Variables and Operators

- **Vectors in MATLAB**

A column vector variable of *n* elements can be defined in MATLAB:

V = [v1; v2; v3;...; vn]

V = [v1 v2 v3... vn]’

```
>> x = [1,2,5]'
x =
    1
    2
    5
```

Variables and Operators

Vectors in MATLAB

Defining a vector variable without explicitly bracketing all its elements together:

```
>> x = 1:5
x =
    1    2    3    4    5
```

```
>> x = 1:2:10
x =
    1    3    5    7    9
```

Variables and Operators

Vectors in MATLAB

Defining a vector variable without explicitly bracketing all its elements together:

```
>> x = linspace(0,1,4)
x =
        0    0.3333    0.6667    1.0000
```

Variables and Operators

Vectors in MATLAB

Selecting an element of a vector or a subset of elements:

```
>> x = [3,7,9,10,17]
x =
     3     7     9    10    17

>> x(3)
ans =
     9
```

Variables and Operators

Vectors in MATLAB

Selecting an element of a vector or a subset of elements:

```
>> x = [3,7,9,10,17]
x =
    3    7    9    10    17

>> x(1:3)
ans =
    3    7    9
```

Variables and Operators

- **Matrices in MATLAB**
- MATLAB treats all variables as matrices. They are assigned to expressions by using an equal sign and their names are case-sensitive. Here, we will give some basic methods with matrices through examples.

  - How to define or form a matrix
  - The most useful matrix functions
  - Algorithm of matrix

Variables and Operators

- **Matrices and matrix computation**

  - **How to define or form a matrix**

1) Input the data directly

```
>> A=[4 -2 5; 6 1 7; -1 0 6]
A=    4   -2    5
      6    1    7
     -1    0    6
```

Variables and Operators

Matrices and matrix computation

➢ How to define or form a matrix

2) Using the built-in functions

Like ones, zeros, rand, eyes.

```
>> ones(2)
ans=   1   1
       1   1
>> zeros(2,4)
ans=   0   0   0   0
       0   0   0   0
```

Variables and Operators

Matrices and matrix computation

Selecting an element of a matrix or a subset of elements:

```
>> A = magic(3)
A =
     8     1     6
     3     5     7
     4     9     2
```

```
>> A(2,3)
ans =
     7
```

$$\begin{array}{cc} \begin{array}{cccc} & \text{Column 2} & & \end{array} & \\ \begin{bmatrix} 1 & 5 & 0 & 2 \\ 5 & 4 & 6 & 6 \\ 3 & 3 & 0 & 5 \\ 9 & 2 & 8 & 7 \end{bmatrix} & \begin{array}{l} \leftarrow \text{row 1} \\ \\ \\ \leftarrow \text{row 4} \end{array} \\ \begin{array}{cccc} & & \text{Column 3} & \end{array} & \end{array}$$

Variables and Operators

Matrices and matrix computation

Selecting an element of a matrix or a subset of elements:

```
>> A = magic(3)
A =
    8     1     6
    3     5     7
    4     9     2
```

```
>> A(6)
ans =
    9
```

Variables and Operators

Matrices and matrix computation

Selecting an element of a matrix or a subset of elements:

```
>> A = magic(3)
A =
    8     1     6
    3     5     7
    4     9     2
```

```
>> A(:,2)
ans =
    1
    5
    9
```

```
>> A(1,:)
ans =
    8     1     6
```

```
>> A(:)
ans = ?
```

Variables and Operators

- **Matrices and matrix computation**

Selecting an element of a matrix or a subset of elements:

```
>> A = magic(3)
A =
     8     1     6
     3     5     7
     4     9     2
```

```
>> B = A(2:3,2:3)
B =
     5     7
     9     2
```

Variables and Operators

- **Matrices and matrix computation**

Selecting an element of a matrix or a subset of elements:

```
>> A = magic(3)
A =
    8   1   6
    3   5   7
    4   9   2
```

```
>> A(:,2) = []
A =
    8   6
    3   7
    4   2
```

Variables and Operators

Matrices and matrix computation

- Algorithm of matrix

```
>> x=[1 5 8];  y=[2 4 3];    x+y
ans=   3    9    11
>> x.*y
 ans=  2    20    24
>>A=[2 5; 0 6];  B=[1 3;4 2];   A*B
ans=   22    16
       24    12
>> A^2
ans=   4    40
       0    36
>> A.^2
ans =  4    25
       0    36
```

Variables and Operators

Functions

diag(v) *Creates a diagonal matrix with the vector v in the diagonal.*

diag(A) *Extracts the diagonal of the matrix as a column vector.*

eye(n) *Creates the identity matrix of order n.*

eye(m, n) *Create an m × n matrix with ones on the main diagonal and zeros elsewhere.*

zeros(m, n) *Creates the zero matrix of order m × n.*

ones(m, n) *Creates the matrix of order m × n with all its elements equal to 1.*

rand(m, n) *Creates a uniform random matrix of order m × n.*

Variables and Operators

Functions

reshape(A, m, n) *Returns an m × n matrix formed by taking consecutive entries of A by columns.*

size (A) *Returns the order (size) of the matrix A.*

length (v) *Returns the length of the vector v.*

tril (A) *Returns the lower triangular part of the matrix A.*

triu (A) *Returns the upper triangular part of the matrix A.*

A' *Returns the transpose of the matrix A.*

inv (A) *Returns the inverse of the matrix A.*

Variables and Operators

Functions

expm (Z) *Matrix exponential function by default*

logm (Z) *Logarithmic matrix function*

sqrtm (Z) *Matrix square root function*

```
>> A = [1 1 0; 0 0 2; 0 0 -1];
>> expm(A)
>> exp(A)
```

Variables and Operators

- **Function**

```
>> A=[4 -2 5; 6 1 7; -1 0 6];
>> [U,D]=eig(A);
U= 0.1900 + 0.0000i  0.2773 + 0.4102i  0.2773 - 0.4102i
   0.8900 + 0.0000i  0.8629 + 0.0000i  0.8629 + -0.0000i
   0.4146 + 0.0000i  -0.0245 + 0.0980i  -0.0245 - 0.0980i
D= 5.5417 + 0.0000i  0.0000 + 0.0000i  0.0000 + 0.0000i
   0.0000 + 0.0000i  2.7291 + 3.6474i  0.0000 + 0.0000i
   0.0000 + 0.0000i  0.0000 + 0.0000i  2.7291 - 3.6474i
```

Variables and Operators

Function: Matrix Calculation

```
>> A = magic(3);
>> rank_of_A = rank(A);          % Rank of matrix
>> det_of_A = det(A);            % Matrix determinant
>> inv_of_A = inv(A);            % Matrix inverse
>> trace_of_A = trace(A);        % Sum of diagonal elements
>> cond_of_A = cond(A);          % Condition number
>> b = [1;2;3];
>> x = A\b;                      % Solution of  A*x = b
>> y = b'/A;                     % Solution of  y*A = b'
```

Plotting in MATLAB

Plotting and graphics in MATLAB

Plotting functions is very easy in MATLAB. There are several built-in functions for 2D or 3D plotting.

- **2D plot**

  plot bar area pie hist

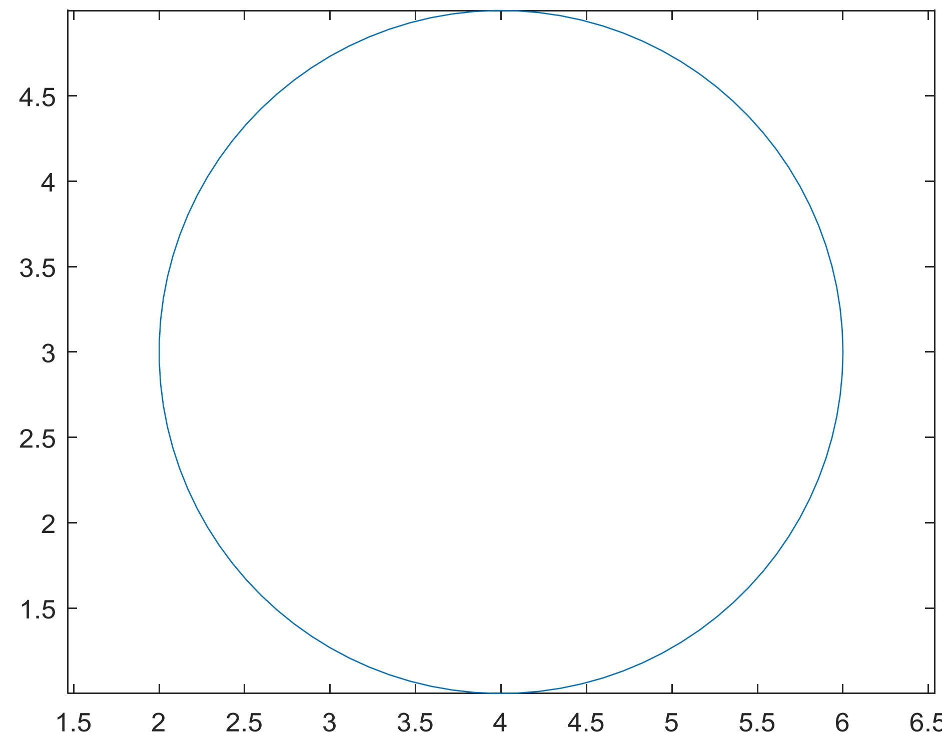
- **3D plot**

  plot3 surf mesh contour pcolor

Plotting in MATLAB

- **Plots**
  - **2D plot**

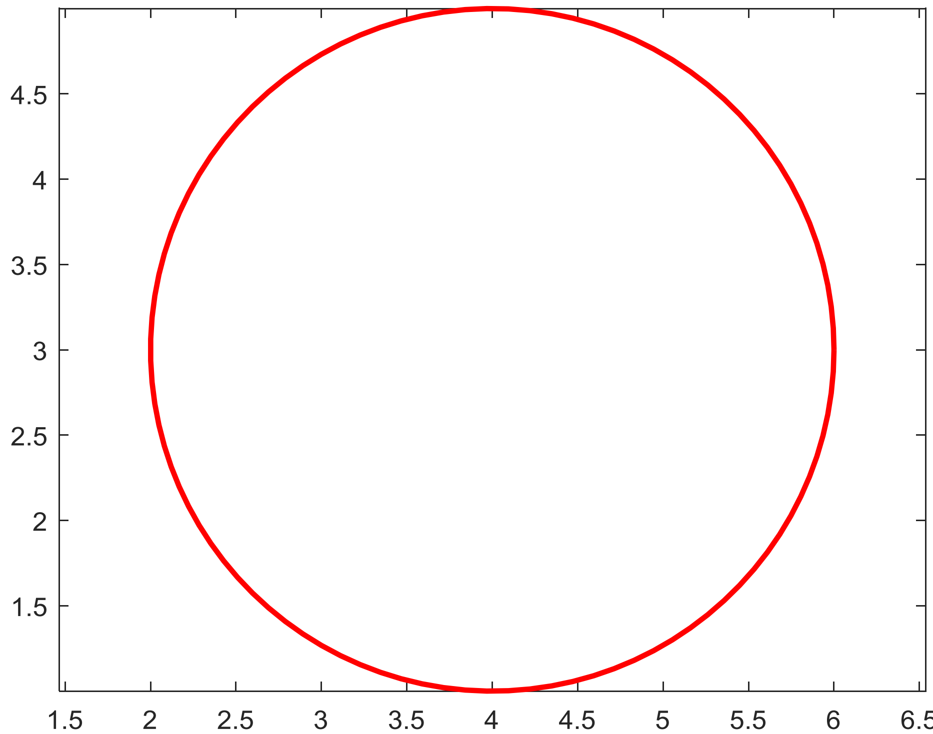
```
r = 2;
xc = 4;
yc = 3;
theta = linspace(0,2*pi);
x = r*cos(theta) + xc;
y = r*sin(theta) + yc;
plot(x,y)
axis equal
```



Plotting in MATLAB

- **Plots**
  - **2D plot**

```
r = 2;
xc = 4;
yc = 3;
theta = linspace(0,2*pi);
x = r*cos(theta) + xc;
y = r*sin(theta) + yc;
plot(x,y,'-r','LineWidth',2)
axis equal
```

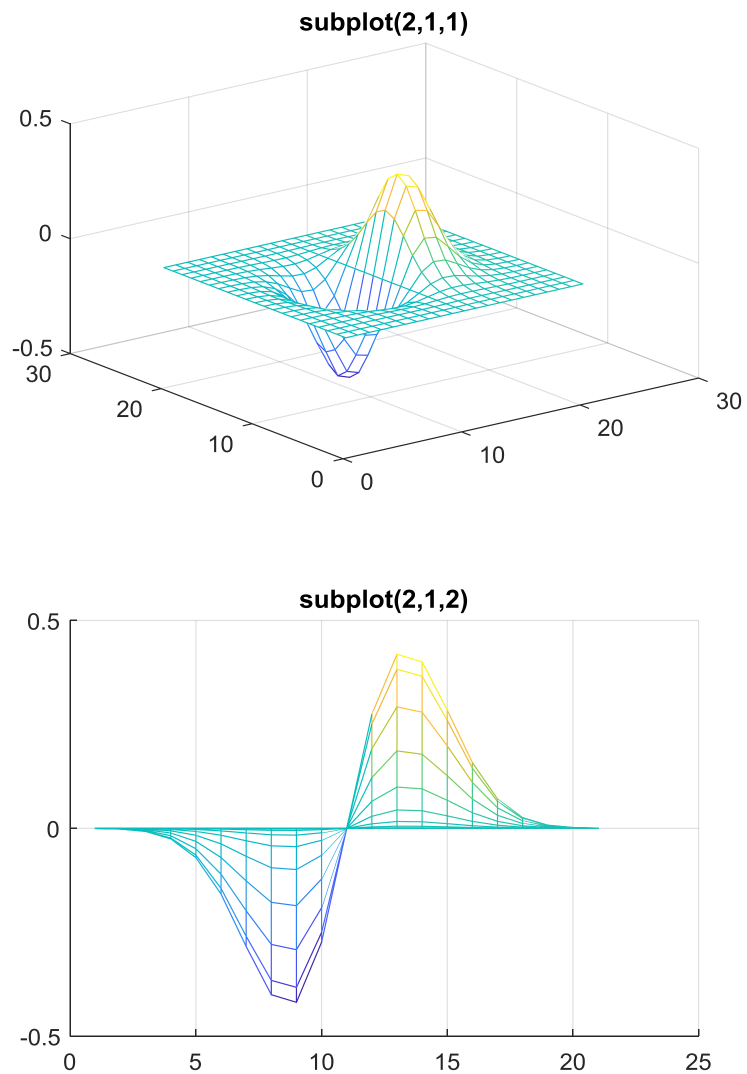


Plotting in MATLAB

Plots

- **3D plot**

```
[X,Y] = meshgrid(-3:0.3:3);
Z = X .* exp(-X.^2 - Y.^2);
subplot(2,1,1)
mesh(Z)
title('subplot(2,1,1)')
subplot(2,1,2)
mesh(Z)
view(0,0)
title('subplot(2,1,2)')
```



Variables and Operators

- **Character variables in MATLAB**

A character variable (chain) is simply a character string enclosed in single quotes

```
>> c = 'string'
c =
string

>> upper(c)
ans =
STRING
```

Variables and Operators

Output format

It is possible to make the printed output from a MATLAB function look good by using the ***disp*** function.

disp(x) displays the array x, without printing the array name. If x is a string, the text is displayed.

```
>> iter=9;
   disp(['Newton method converges after', num2str(iter),'iterations']);
ans= Newton method converges after 9 iterations
>> disp('The input matrix A is')
ans=  The input matrix A is
```

Variables and Operators

Multidimensional Arrays

A three-dimensional array, uses three subscripts:

- **The first references array dimension 1, the row.**
- **The second references dimension 2, the column.**
- **The third references dimension 3, the page.**

```
>> A(:,:,1) = [5 7 8; 0 1 9; 4 3 6];
>> A(:,:,2) = [1 0 4; 3 5 6; 9 8 7];
```

```
A(:,:,1) =                A(:,:,2) =
     5    7    8               1    0    4
     0    1    9               3    5    6
     4    3    6               9    8    7
```

Variables and Operators

- **cell →To create cell array**

```
>> C = cell(2,2);
>> C{1,1} = magic(3);
>> C{1,2} = 100;
>> C{2,1} = 'SJTU';
   C =
   [3x3 double]    [100]
   'SJTU'          []
>> C{1,1}(2,3)
   ans =  7
```

Variables and Operators

- **cell →To create cell array**

D = cellfun('f ',C)
Applies the function f (isempty, islogical, isreal, length, ndims, or prodofsize) to each element of the array C.

celldisp (C)
Displays the contents of the array C.

Variables and Operators

- **Create a structure array**

```
>> Student.name = 'Jack';
>> Student.score = 80;
>> Student.tests = [60,70,70];
   Student =
   name: 'Jack'
   score: 80
   tests: [60 70 70]
s = struct('field1', values1, 'field2', values2, ...)
>> Student = struct('name', 'Jack', 'score', 80,'test',[60,70,70])
```

Thank You !