Mathematic Analysis with Matlab

Lecture 1: Overview and Introduction

Lecturer: *Dr.* Lian-Sheng Wang

Fall Semester 2021

Contact: lswang@xmu.edu.cn

Outline

General goal of this course

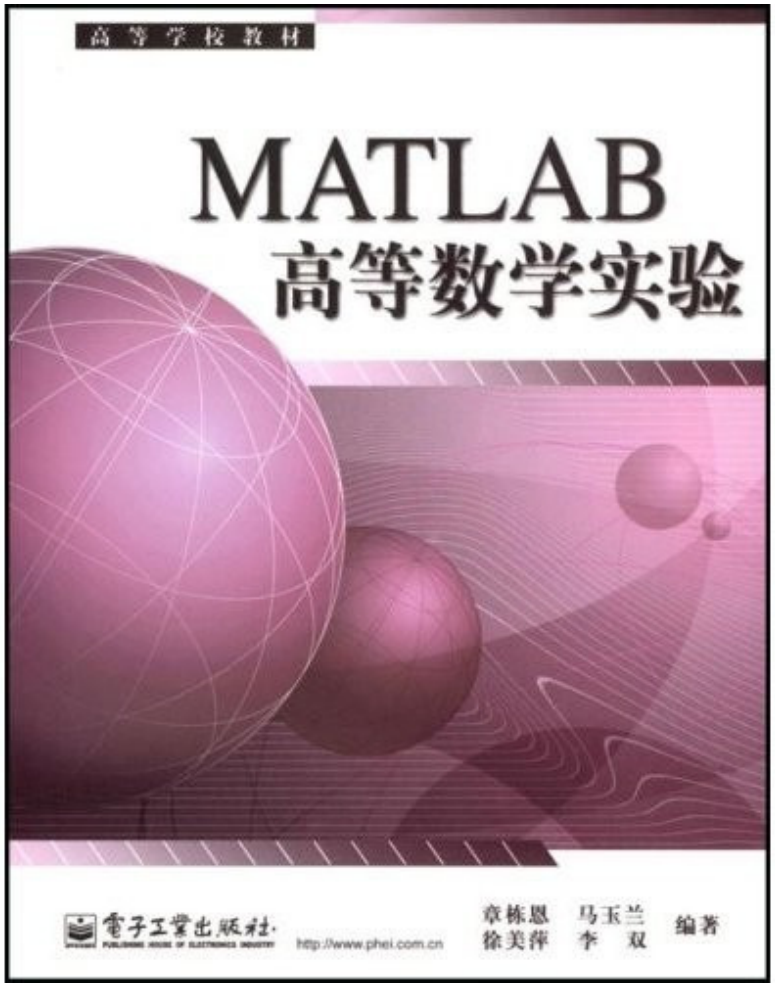
- *Six objectives* we want to reach
  1. Solve mathematical problems with Matlab
  2. A natural extension of your math courses: 7 subjects are new
  3. Show how to do modelling with your math
  4. Introduce you a powerful tool: convex optimization
  5. Visualization with Matlab
  6. Learn how to use **LaTex** editor

Data Processing is an emerging Indurstry

- All kinds of data around us
  1. Photos (visual)
  2. Web pages (textual)
  3. Sound and music (audual)
  4. Business data: price, logistics and index
  5. Communications
  6. Geographic

Textbook of this course (1)

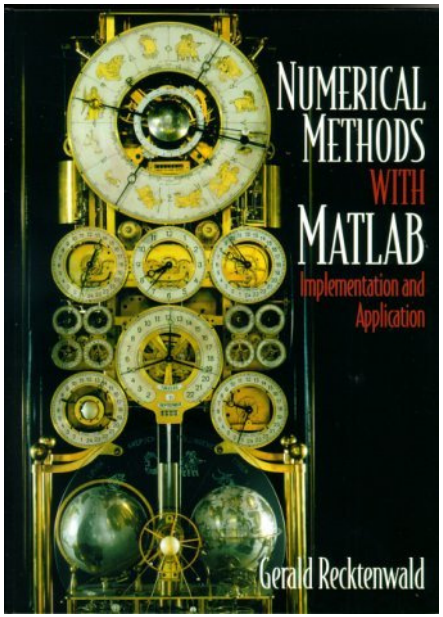
- You should have a textbook to refer to
- There are a lot of free materials online
- Google 'matlab tutorial' or specific functions



作者：章栋恩 马玉兰 徐美萍 李双
出版社：电子工业出版社

Textbook of this course (2)

- "Numerical Methods with MATLAB: Implementations and Applications"
- Homepage: http://web.cecs.pdx.edu/~gerry/nmm/course/
- Author: Gerald Recktenwald
- Publisher: Prentice Hall
- One can find exercises from the home page



Textbook of this course (3)

- "Introduction to Numerical Methods and Matlab Programming for Engineers"
- Author: GTodd Young and Martin J. Mohlenkamp
- The electronic version is downloadable for free

Exercises and Exams

- Two or Three exercises
- One or Two course projects
- Final score=20%×Asg.+20%×Cls.+50%×Prj.+10%×Atd.
- You are required to submit an 1-2 pages report along with your project
- No cheating, please!!!

Language in the Class

- English or Chinese?
- You might be uncomfortable at the beginning
- Me too:)
- Several advantages:
  - Computer science is defined in English
  - Get you guys used to English
- Relate terms in Chinese to their English names

TA in the Class

- Schedule
  - Mr. Jian-sen Guo will be with us across the semester
  - Mr. Jin-kai Ren will be with us across the semester
- Their duties
  - Feel free to raise your hand during the class if you have any problem
  - Help to organize and release problem sets
  - Mark your coursework

Content of 'Matlab based Mathematic Analysis' (1)

- Fundamentals about MATLAB
  - Basic operations in Matlab
  - Basic figure display in Matlab
  - Coding with Matlab
- Representation and visualization on a function
- Calculate 'limit' for function with Matlab
- Derivatives and its application
- Integral and 3D visualization
  - Indefinite integral
  - Definite integral
  - Numerical integral
  - 3D visualization for multiple variable functions
- Differential Calculus on multiple variable function

Content of 'Matlab based Mathematic Analysis' (2)

- Infinite series and Differential equations
  - Representation and Approximation of infinite series
  - Solving differential equations with Matlab
- Matrix and solving linear equations
- Eigenvalue decomposition and Matrix decomposition
  - Eigenvalue decomposition
  - Singular value decomposition (SVD)
  - Principle Content Analysis (PCA)
- Regression
  - Linear regression
  - Non-linear regression
- Optimization
  - Basics about Convex Optimization
  - Linear programming
  - Quadratic programming
  - Analytic hierarchy process

Outline

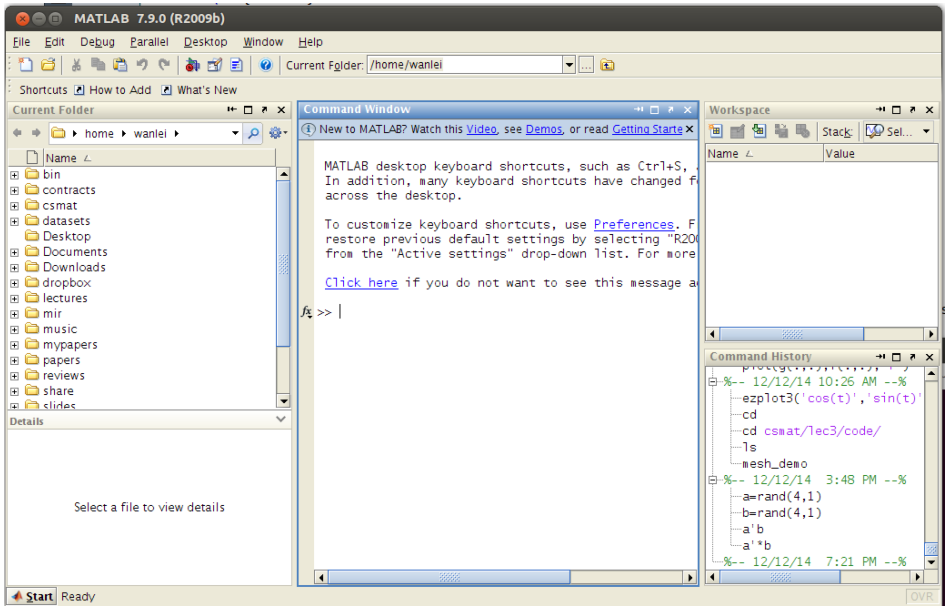
Brief history about Matlab

- Cleve Barry Moler from University of New Mexico developed the first version for his students in late 1970s
- Helped them to access to LINPACK and EISPACK
- Jack Little met with Cleve B. Moler in 1983
- Jack Little, Cleve Moler and Steve Bangert re-wrote this tool in C
- They co-founded MathWorks in 1984, and commercialized this tool

Integrated functional blocks in Matlab

- **Numerical computing**
- Parallel computing
- Computer vision
- Signal processing and communications
- Machine learning

Matlab interface



Windows	Function
Command Window	accept input command from user
Workspace	List variables defined by user, including ‘Name’, ‘Value’, ‘Class’ and ‘Bytes’
Current Directory	clear variables defined in workspace
Command History	display variables defined in workspace

Commonly used commands

Command	Function
clf	clear figure window
clc	clear command window
clear	clear variables defined in workspace
who	display variables defined in workspace
whos	display variables defined in workspace in detail
help command	display help info for 'command'
edit	open up matlab source code editor
format output_type	define output format (short|long|rat|compact)
quit	exit from Matlab

Global constants

Constant	explanation
ans	default variable to keep function output
pi	π
inf or Inf	∞
eps	2^{-52}
Flops	Num. of floating-point operations per second
NaN or nan	non numerical value

Try out basic commands

- Steps:
  1. a=cos(0.21*pi)
  2. b=exp(2);
  3. hi='hello, i am wanlei';
  4. sin(2*exp(1))
  5. who
  6. whos
  7. clear

Operations on array

1. Define an array
   - a=[1 2;3 4;5 6];
   - a=zeros(3,2);
2. Access elements in an array
   - a(1,1)
   - a(1:2,:);

Variable definition in comparison to C

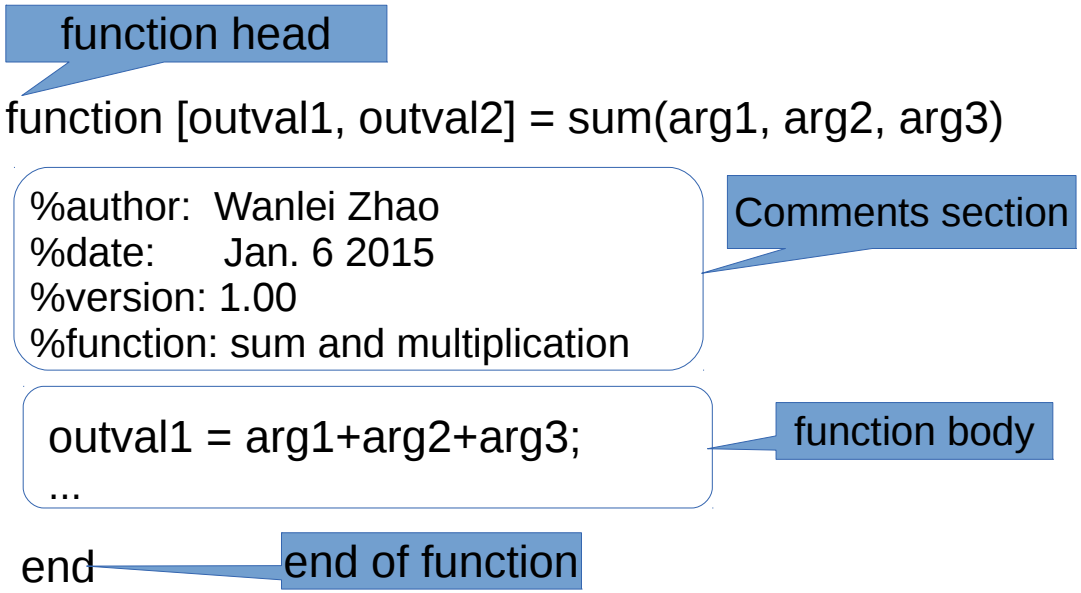
1. No explicit variable type
   - a=[1 2;3 4;5 6];
   - b='hello';
2. No allocate or deallocate
   - clear
3. One variable could be used for different purpose
   - a=[1 2;3 4;5 6];
   - a='hello'
4. Variables of different types are compatible
   - a='hello'
   - b=2
   - a+b

Definition for symbols and operations

1. Define symbols
   - syms x y z
2. Define equations
   - y=x+2;
   - z=x^2+2;
3. Plug values in
   - x=3;
   - eval(y)
   - eval(z)
   - try 'x=5' ..

Definition for script and function

- Like C language
- You can define functions
- Unlike C language, the script is NOT compiled into binary code
- Instead, it is interpreted line by line (like Python, perl etc.)



- Save to 'sum.m'[1]

[1] Save matlab script into file with the same name as function name

Outline

Drawing 2D figure: the commands

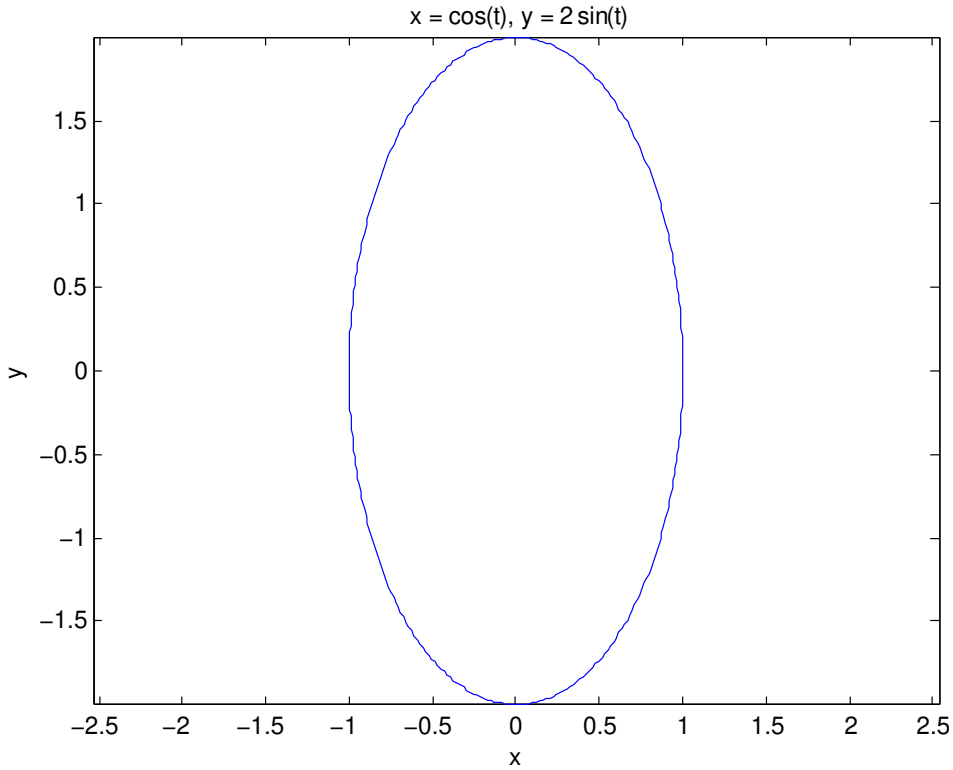
1. plot(x, y, 's')
2. ezplot('x(t)','y(t)',[t1,t2])

- Given:

$$\begin{aligned} x &= cos(t) \\ y &= 2 \cdot sin(t) \end{aligned} \tag{1}$$

- Input: ezplot('cos(t)','2*sin(t)',[0,2*pi])

Drawing 2D figure: the result



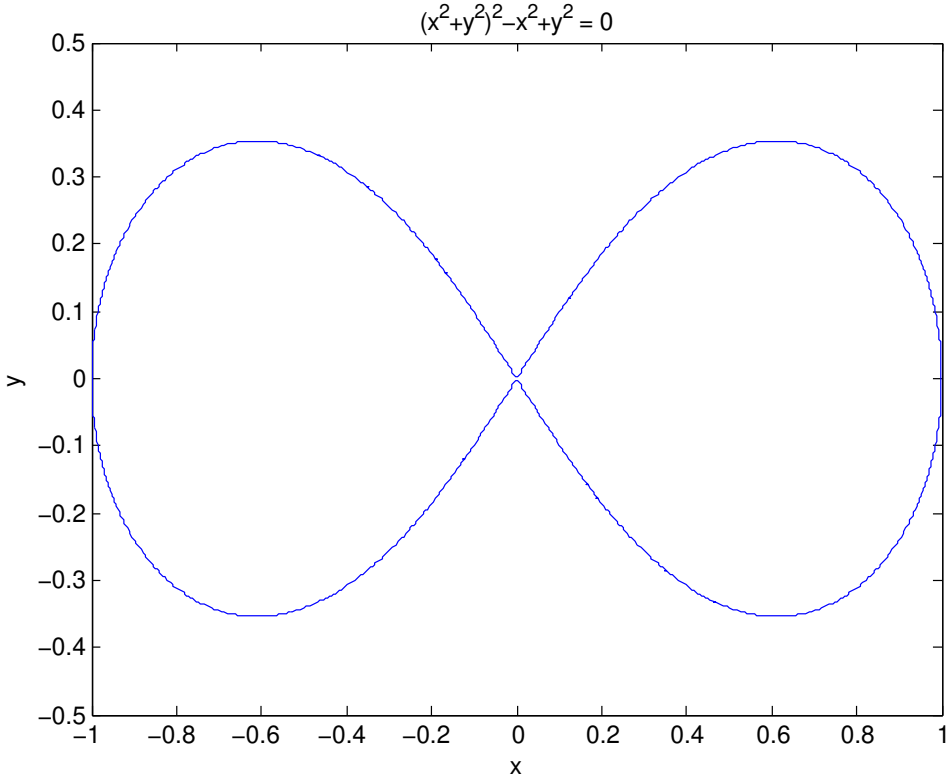
Drawing 2D figure: implicit function

- Given:

$$(x^2 + y^2)^2 = x^2 - y^2 \tag{2}$$

- Input: ezplot('(x^2+y^2)^2-x^2+y^2',[-1,1,-0.5,0.5])

Drawing 2D figure: the result



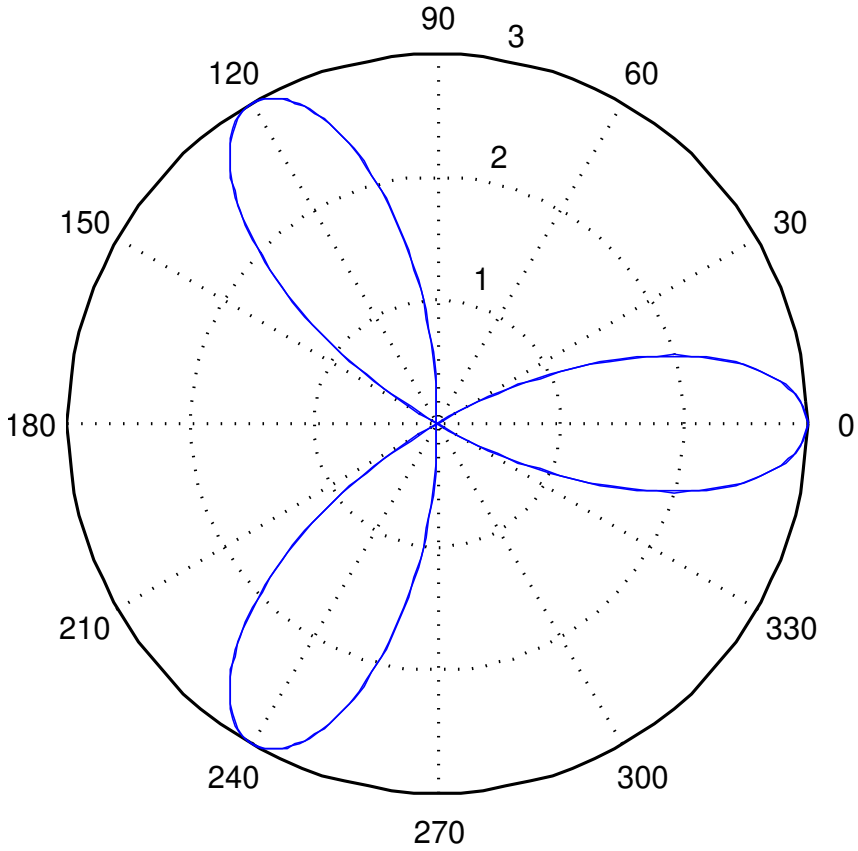
Drawing 2D figure: implicit function

- Given:

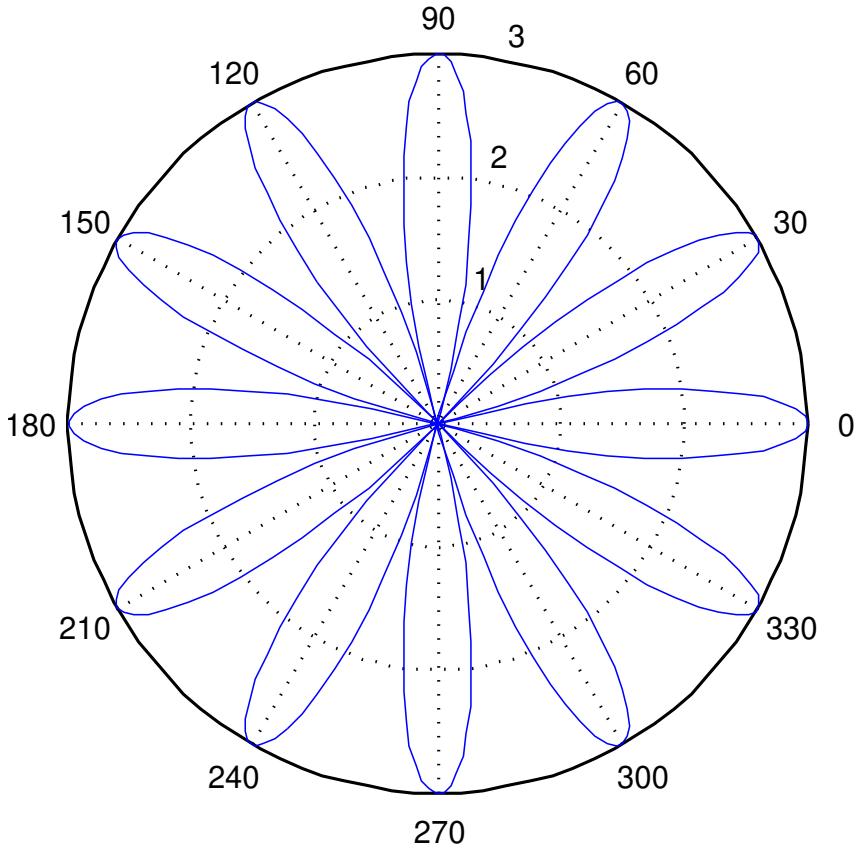
$$\rho = 3cos(3\theta) \tag{3}$$

- Input: ezpolar('3*cos(3*t)',[0,2*pi])

Drawing 2D figure: the result



r = 3 cos(3 t)



r = 3 cos(6 t)

Drawing 2D figure: implicit function

- Given:

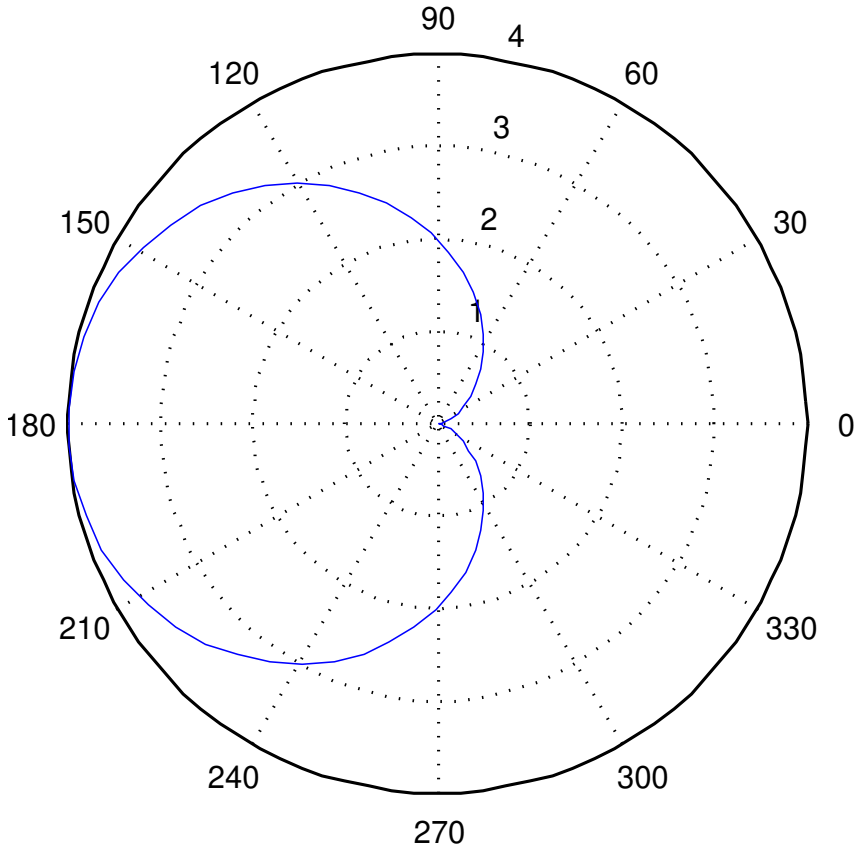
$$\rho = 2(1 - cos(\theta)) \tag{4}$$

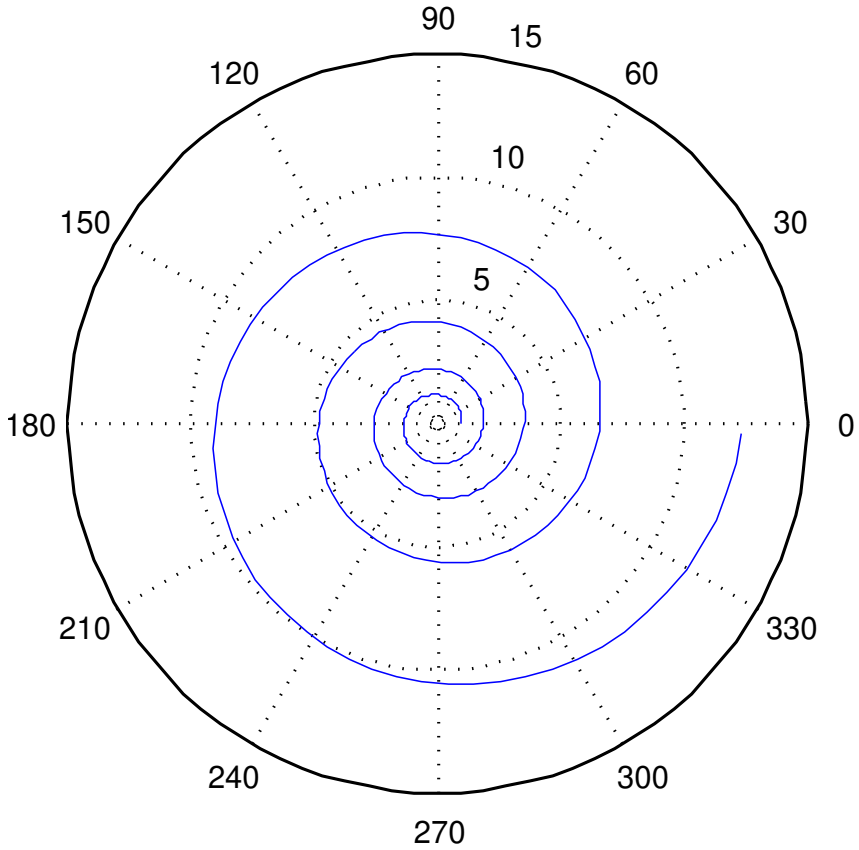
$$r = e^{\theta/10} \tag{5}$$

1. theta=0:0.1:2*pi;
2. rho=2*(1-cos(theta));
3. polar(theta,rho)

1. theta=0:0.1:8*pi;
2. rho=exp(0.1*theta);
3. polar(theta,rho)

Drawing 2D figure: the result





Drawing curve by segments

$$f(x) = \begin{cases} cos(x) & -4 \leq x \leq 0 \\ e^x & 0 < x \leq 4 \end{cases} \tag{6}$$

[**Hints**]

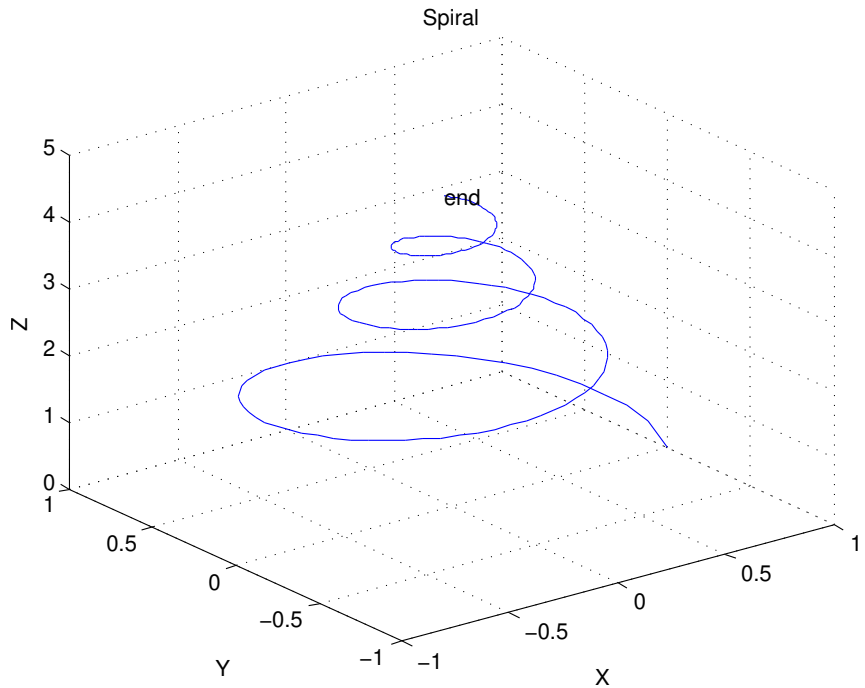
1. They are two different functions in different ranges
2. Define two functions, plot them respectively based on the range of x

Drawing 3D curve

$$\begin{cases} x = e^{(-0.1t)}cos(t), \\ y = e^{(-0.1t)}sin(t), \quad 0 < t < 20 \\ z = sqrt(t), \end{cases} \tag{7}$$

```
clear;
t = 0:0.1:20;
r = exp(-0.1*t)
x = r.*cos(t);
y = r.*sin(t);
z = sqrt(t);
plot3(x, y, z); grid on; hold on;
```

```
title('Spiral');
text(x(end), y(end), z(end), 'end')
xlabel('X'); ylabel('Y'); zlabel('Z');
```

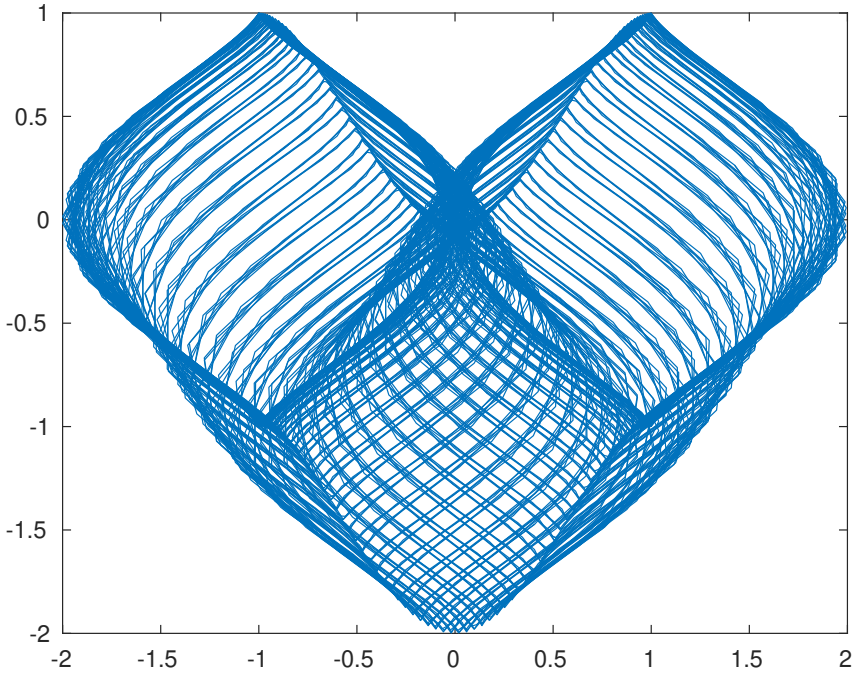


Exercise: drawing 2D curve (1)

$$\begin{cases} x = cos(at) - cos^3(bt), \\ y = sin(ct) - sin^4(dt), \quad -10 \leq t \leq 10 \end{cases} \tag{8}$$

- Write a script for that
- Try ‘a=1;’ ‘b=80;’ ‘c=80;’ ‘d=1;’

Exercise: drawing 2D curve (2)



Exercise 2 (1)

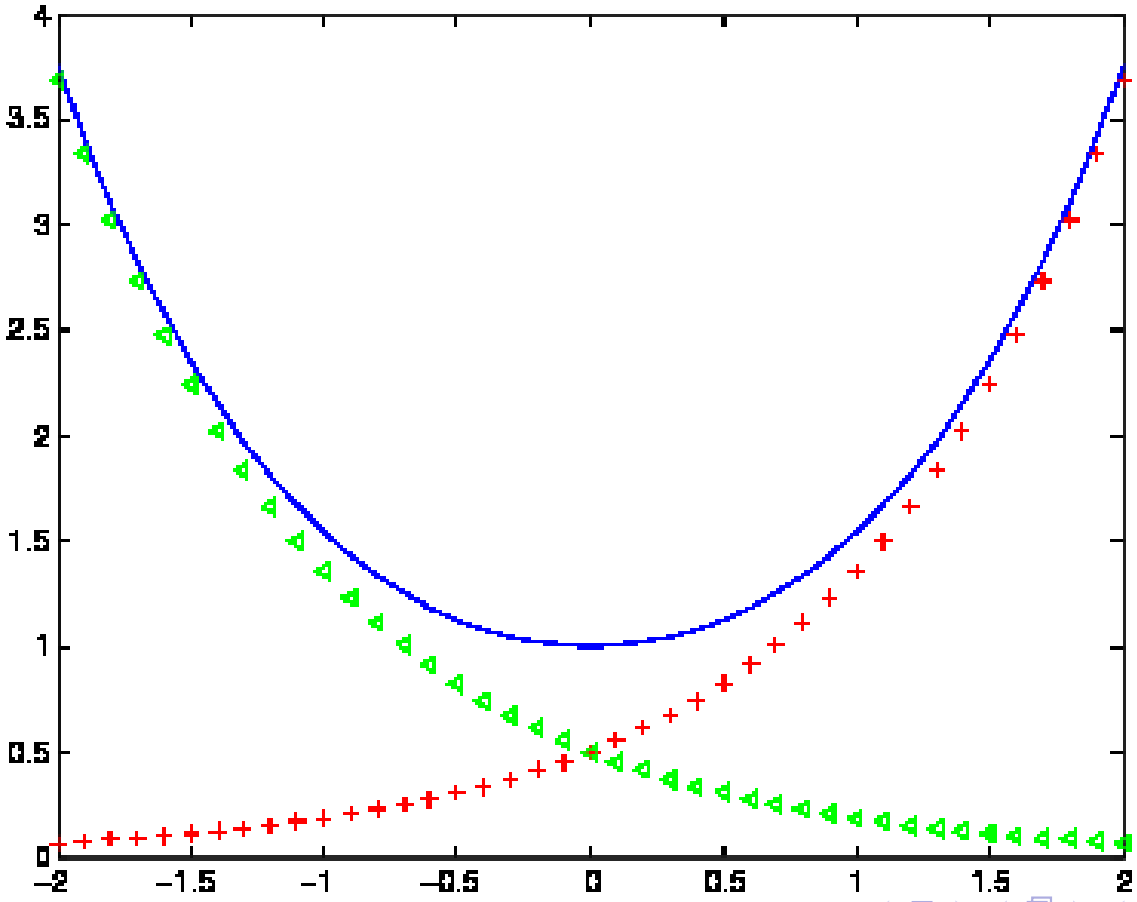
- Plot 'cosh(x)', along with $\frac{e^x}{2}$ and $\frac{e^{-x}}{2}$

$$cosh(x) = \frac{e^x + e^{-x}}{2} \tag{9}$$

Exercise 2 (2)

- Plot 'cosh(x)', along with $\frac{e^x}{2}$ and $\frac{e^{-x}}{2}$

$$cosh(x) = \frac{e^x + e^{-x}}{2} \tag{10}$$



Calling functions in comparison to C

1. No "#include ... " clause
2. But the functions must exist
3. Matlab is not completely case sensitive
   - Case sensitive to variables
   - Case sensitive to built-in functions
   - Case sensitive to user defined scripts and functions under UNIX/Linux
   - Case insesitive to user defined scripts or functions under Windows

Q & A

Thanks for your attention!