浙江工业大学

MATLAB & Communication Simulations

Lab Report1

Name 凌智城

Teacher 张昱

Class 通信工程 1803 班

Student ID 201806061211

Department 信息工程学院

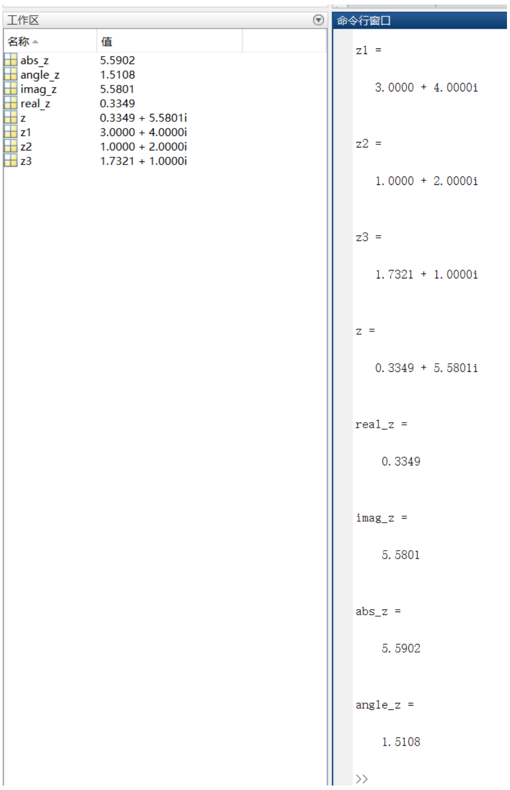
Date： Mar. 27th,2021

I. Task_1

A. Code

```
clc;                    %Clear the contents of the command window
clear;                  %Clear the contents of the workspace
close;                  %Close current figure
z1 = 3 + 4i;
z2 = 1 + 2i;
z3 = 2*exp(pi*1i/6);    %element in array X returns x with exponent e
z = z1 * z2 / z3;
real_z = real(z);       %return the real part of z
imag_z = imag(z);       %return the image part of z
abs_z = abs(z);         %return the amplitude of z
angle_z = angle(z);     %return the angle of z
```

B. Figure



fugure1: lab1_1

C. Discussion

The first Matlab code of this course. We can use clc to clear the contents of the command window, use clear to clear the contents of the workspace ,use close to close current figure. In the complex field, we can use real(z), imag(z), abs(z), abgle(z) and conj(z) to perform related calculations.

II. Task_2

A. Code

```
clc;
clear;
close;
z1 = 3 + 4i;
z2 = 1 + 2i;
z3 = 2*exp(pi*1i/6);
z = z1 * z2 / z3;
real_z = real(z);
imag_z = imag(z);
abs_z = abs(z);
angle_z = angle(z);
a = [z1 z2]';          % []'vector transpose
b = [z2 z3];           % [] vector
c = b * a;             % * vector multiplication
c1 = abs(c);
c2 = angle(c);
c(1,:) = 1;            % set the first row to '1'
c(:,2) = 1;            % set the second column to '1'
```

B. Figure

工作区

名称	值
a	[3.0000 - 4.0000i;1.0000 - 2.0000i]
abs_z	5.5902
angle_z	1.5108
b	[1.0000 + 2.0000i,1.7321 + 1.0000i]
c	[1,1]
c1	14.7394
c2	-0.0315
imag_z	5.5801
real_z	0.3349
z	0.3349 + 5.5801i
z1	3.0000 + 4.0000i
z2	1.0000 + 2.0000i
z3	1.7321 + 1.0000i

figure2: lab1_2

C. Discussion

Performing vector operations, we should pay attention to distinguish between ‘*’ and ‘.*’ and so on. Also, we can use A([a:b],[c:d]) to reset the a to b row and c to d column matrix A.

III. Task_3

A. Code

```
clc;
clear;
close;
x = 0:pi/50:2*pi;     % from 0 to 2*pi,step is pi/50
y1 = sin(x);
y2 = cos(x);
y3 = log(x);
```

```
figure(1);                    % Create a new picture

subplot(2,2,1)
% subplot(2,2,1) means this figure has 2 rows and 2 columns in total,
% and this subfigure is the first one of those subfigures.
plot(x,y1)
title('sin(x)')              % set the title of this subfigure
xlabel('x')                  % set the x-axis label of this subfigure
ylabel('sin(x)')             % set the x-axis label of this subfigure
legend('sin(x)')             % set the legend of this subfigure
grid on                      % set gridlines

subplot(2,2,2)
plot(x,y2)
title('cos(x)')
xlabel('x')
ylabel('cos(x)')
legend('cos(x)')
grid on

subplot(2,2,3)
plot(x,y3)
title('log(x)')
xlabel('x')
ylabel('log(x)')
legend('log(x)')
grid on

subplot(2,2,4)
```

```
plot(x,y1,'bh')
xlabel('x')
ylabel('sin(x)')

hold on                          % Continue to draw on this subfigure
plot(x,y2,'gP')
xlabel('x')
ylabel('cos(x)')

hold on
plot(x,y3,'r>')
xlabel('x')
ylabel('log(x)')

title('sin&&cos&&log')
legend('sin(x)','cos(x)','log(x)')
grid on
```

B. Figure

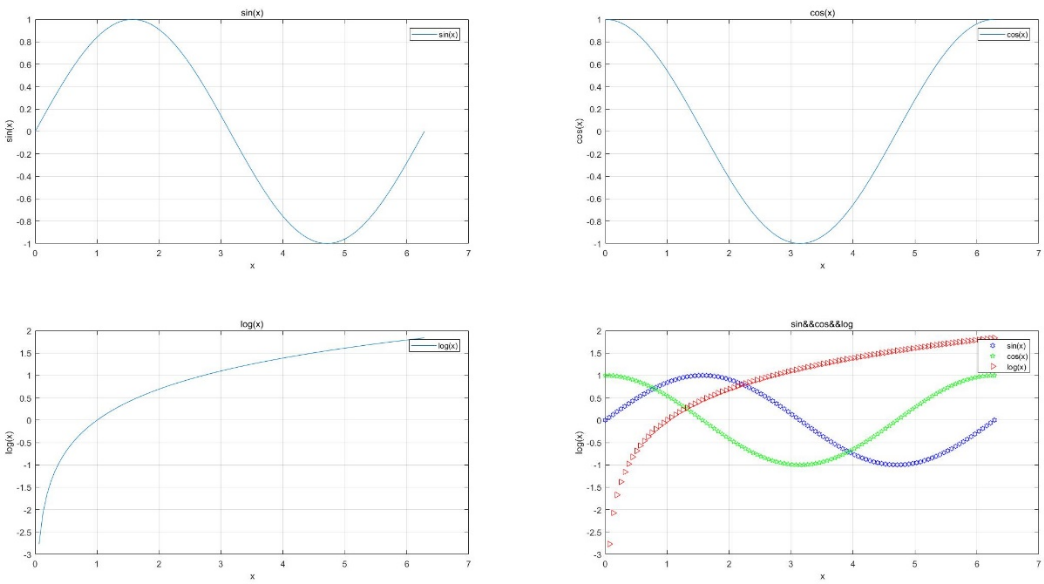


figure3: lab1_3

C. Discussion

While we plot a figure, we should not only draw the original picture, but also annotate the title, legend and axis. In order to better distinguish different curves in the same figure, we can use different linearities and colors.

IV. Task_4

A. Code

```
clc;
clear;
close;
%% main code
n = 15;                    %Set the order of the Fibonacci sequence
fibonacci_for(n);          %Call functions
fibonacci_while(n);        %Call functions

%% function fibonacci_for(n)
function fibo = fibonacci_for(n)
fibo = zeros(1,n);
fibo(1) = 1;               % Initialize the first two items of the sequence
fibo(2) = 1;
for k = 3:n                % Loop from the third item
    fibo(k) = fibo(k-1) + fibo(k-2);
end
figure;
plot(fibo,'bx')
for ii = 1:1:n             % Label each point
    text(ii,fibo(ii),['(' num2str(ii) ',' num2str(fibo(ii)) ')'])
end
```

```
title('fibonacci-for(n)')
xlabel('n')
ylabel('fibo')
legend('fibonacci-for(n)')
grid on

%% function fibonacci_while(n)
function fibo = fibonacci_while(n)
fibo = zeros(n, 1);
fibo(1) = 1;                % Initialize the first two items of the sequence
fibo(2) = 1;
k = 3;                      % Loop from the third item
while k <= n
    fibo(k) = fibo(k-1) + fibo(k-2);
    k = k + 1;
end
figure;
plot(fibo,'rd')
for ii = 1:1:n              % Label each point
    text(ii,fibo(ii),['(' num2str(ii) ',' num2str(fibo(ii)) ')'])
end
title('fibonacci-while(n)')
xlabel('n')
ylabel('fibo')
legend('fibonacci-while(n)')
grid on
```

B. Figure

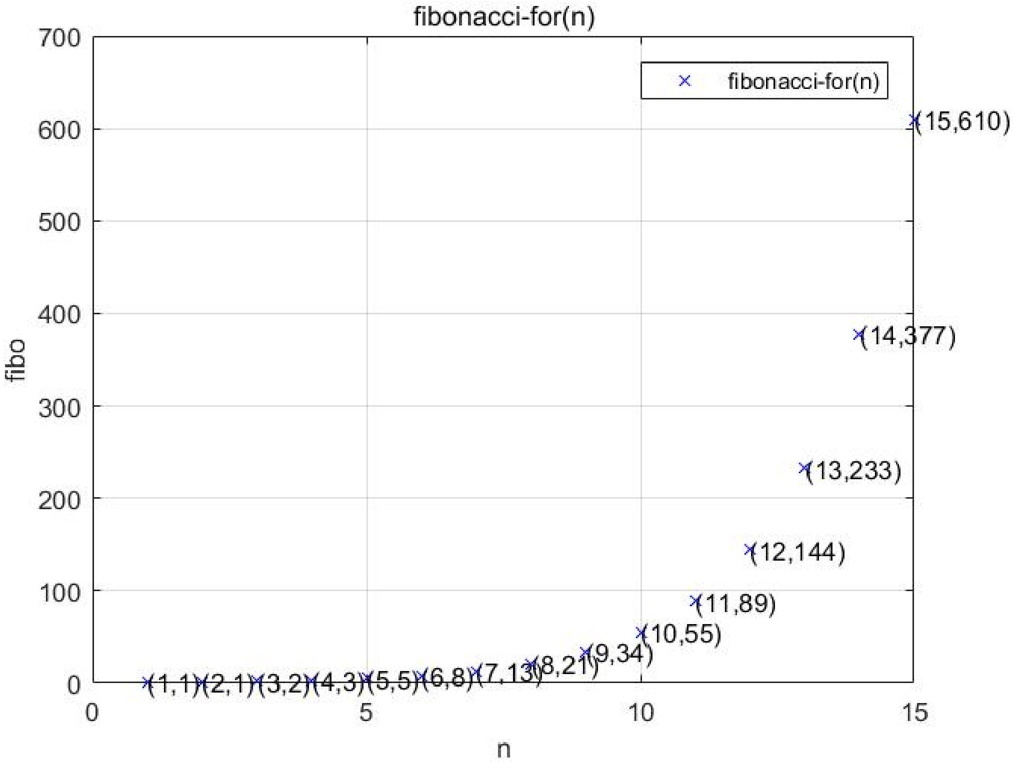


figure4: Fibonacci_for

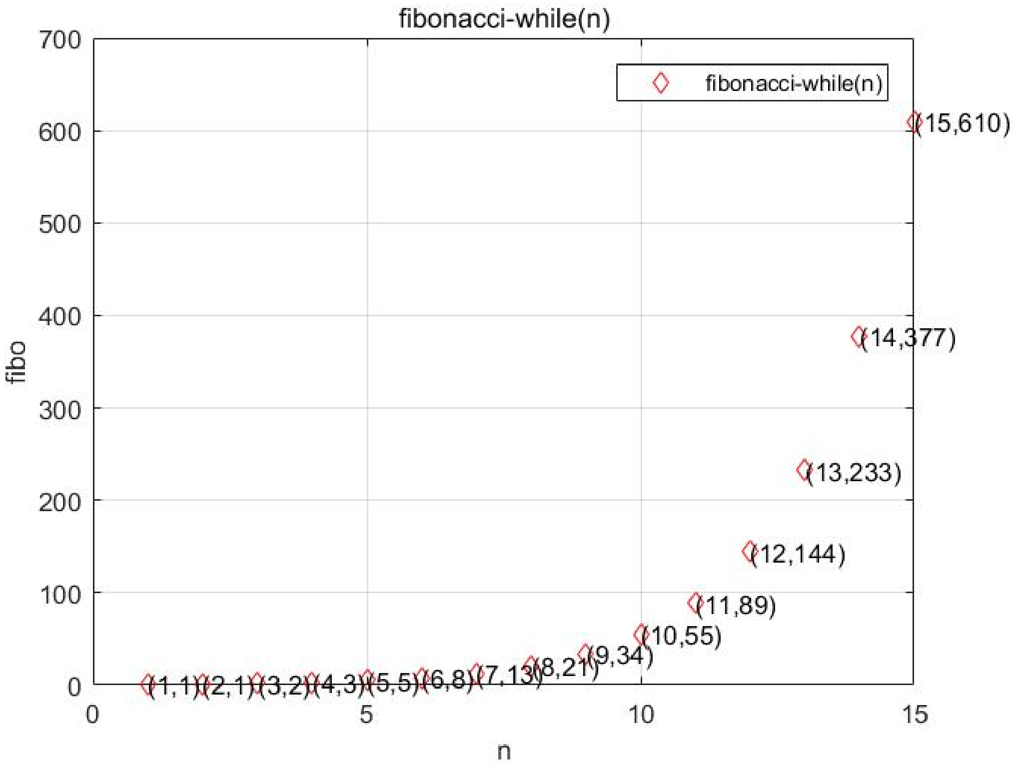


figure5: Fibonacci_while

C. Discussion

In this task, we can let a function call itself, of course we can also not use this method, using for loop or while loop instead. The most important thing is initialization of the sequence, then we could use iteration or loop.