Getting Started with MATLAB

Lab Exercises on December 5, 2021

Department of Electronics and Computer Engineering

Pulchowk Campus, Lalitpur

Ashlesh Pandey

PUL074BEX007

1 Objectives

- Familiarization and review of basic programming concepts in MATLAB.

2 Background Theory

2.1 Variables

Unlike many programming languages, MATLAB does not require prior definition of the variables, instead the variables can be simply written as,

```
variable name = expression;
```

For example:

```
a = sin(64) + 2;
```

If the user doesn't specify the name of the variable, MATLAB automatically creates the variable `ans`.

```
>> 3+2
ans=5
```

2.2 Vectors and Matrices

Vectors and matrices are used to combine separate scalar data into single, multidimensional signal.

```
>> x=[1:10]
>> x=[1 3 7 15]
>> y=[1:0.1:10]
>> z=[1:3;4:6;7:9]
>> [m,n]=size(z)
```

2.3 Arithmetic Operations

- Arithmetic operators: `+,-,*,/,\,^`
- Mathematical functions available: `abs`, `sqrt`, `log`, `sin`, `cos`

Mathematical Function	MATLAB Syntax
$f_1 = a_1 + b_1x + c_1x^2$	`f1 = a1 + b1*x + c1*x^2`
$f_2 = a_2 + b_2x + c_2x^2 + d_2x^3$	`f2 = a2 + b2*x + c2*x^2 + d2*x^3`
$g = e^{At}\left(C_1 \cos(Bt) + C_2 \sin(Bt)\right)$	`g = exp(A*t)*(C1*cos(B*t)+C2*sin(B*t))`
$u = 2xy^2 + \sin(x+y)$	`u = 2*x*y^2 + sin(x+y)`

2.4 Control Flow

Loops	FOR Loop	WHILE Loop	IF..ELSE..

Syntax	`for expression`
	`statements`
	`end`
		`while expression`
		`statements`
		`end`
			`if expression`
			`statements`
			`elseif expression`
			`statements`
			`else`
			`statements`
			`end`

A FOR loop allows a statement to be repeated a fixed, predetermined number of times. The problem is to fill the vector b with square roots of 1 to 1000. One way to do so, is by using a FOR loop. A file named tictoc.m is saved with the following content,

```
clear ; %To clear all previous variables, and to free memory.
tic ; %This function initializes an internal clock
```

```
for i = 1:1000
    b(i) = sqrt(i);
end
t=toc;
str=sprintf('The time required was: %f',t);
disp(str)
```

Listing 1: Matlab script to use tic toc with for loop

Command Window Observation

```
>> tictoc
The time required was: 0.000347
```

2.5 Different Tasks and Commands

2.5.1 User-Defined Functions

- **Commands:** `function [op1,op2,...]=cmd_name(ip1,ip2,...)`
- **Example:**

```
function y = fcn(x)
    y = sin(x.^2);
end
```

2.5.2 Polynomial Roots

- **Commands:** `roots(p)`
- **Example:**

```
>> p = [1 2 1];
>> r = roots(p)
r = -1 -1
```

2.5.3 2D Plotting

- **Commands:** `plot, subplot, figure, hold, stem, axis, title`
- **Example:**

```
>> t=[-2:0.01:2];
>> x=sin(t*10);
>> plot(t,x)
>> axis([-1 1 -1 1])
>> zoom
>> xlabel(`Time')
>> title(`My first plot')
>> specgram(x)
```

2.5.4 Dealing with Sound Files

- **Commands:** `wavread, wavwrite, auread, auwrite, sound(y,fsamp)`
- **Example:**

```
>> y=wavread('C:\sound.wav')
>> sound(y,44100);
```

2.5.5 Complex Numbers

- **Commands:** `j, real, imag, abs, angle`
- **Example:**

```
>> real(j)      >> imag(j)
>> abs(j)       >> angle(j)
```

2.5.6 Signal Processing and Image Processing

- **Commands:** `fft(), dft(), con(), dither(), gray2ind(), ind2gray(), ind2rgb(), imread(), imwrite()`
- **Example:**

```
>> A=imread('my_pic.jpg')
>> whos
>> imshow(A)
```

2.5.7 Transfer Function Representation and Frequency Response

- **Commands:** `tf2zp, zp2tf, freqs(), semilogx(), bode()`
- **Example:**

```
% Given H(s)=(2s+3)(s3+4s2+5)
>> num=[2 3];
>> den=[1 4 0 5];
>> [z,p,k]=tf2zp(num,den);
>> [num den]=zp2tf(z,p,k);
>> T=0:0.1:1;
>> y=step(num,den,t);
>> plot(t,y)
>> bode(num,den)
>> [mag,phase]=bode(num,den,w);
>> magdb=20*log10(mag);
>> semilogx(w,magdb)
>> semilogx(w,phase)
```

2.6 Getting Help from MATLAB

```
>> doc fft

>> help help

>> help cos

>> help fft

>> lookfor filter
```

3 Lab Exercises

Problem 1

Calculate $\left(1+\frac{2}{n^2}\right)^n$ **for n=3, 7**

```
function [y] = lab_1_1(n)
    y = (1+2/n^2)^n;
end
```

Listing 2: Matlab function for calculating given polynomial

Command Window Observation

```
>> lab_1_1(3)

ans = 1.8258
```

```
>> lab_1_1(7)

ans = 1.3232
```

Problem 2

Plot the function: $y = e^{-at}cos(\omega t)$**, for** $a = 2, \omega = 5$**, and** $t = 0 : 10$**.**

```
function y = exponential_cosine(a, w,t)
    y = exp(-a.*t).*cos(w.*t);
end
```

Listing 3: Matlab function to calculate the response for product of exponent and cosine

```
t=linspace(0,10,1000);
a=2;
w=5;
y=exponential_cosine(a,w,t);
l=tiledlayout(1,1);
title(l,{'Plot of y for a=2, \omega=5 and t=0:10', '(PUL074BEX007)'})
nexttile
plot(t,y,'Linewidth',1.5)
xlabel('t','interpreter','latex')
ylabel ('$y=e^{-at}\cos(\omega t)$','interpreter','latex')
print('lab_1_2','-depsc')
```

Listing 4: Matlab script to plot y for $a = 2$, $\omega = 5$, and $t = 0 : 10$

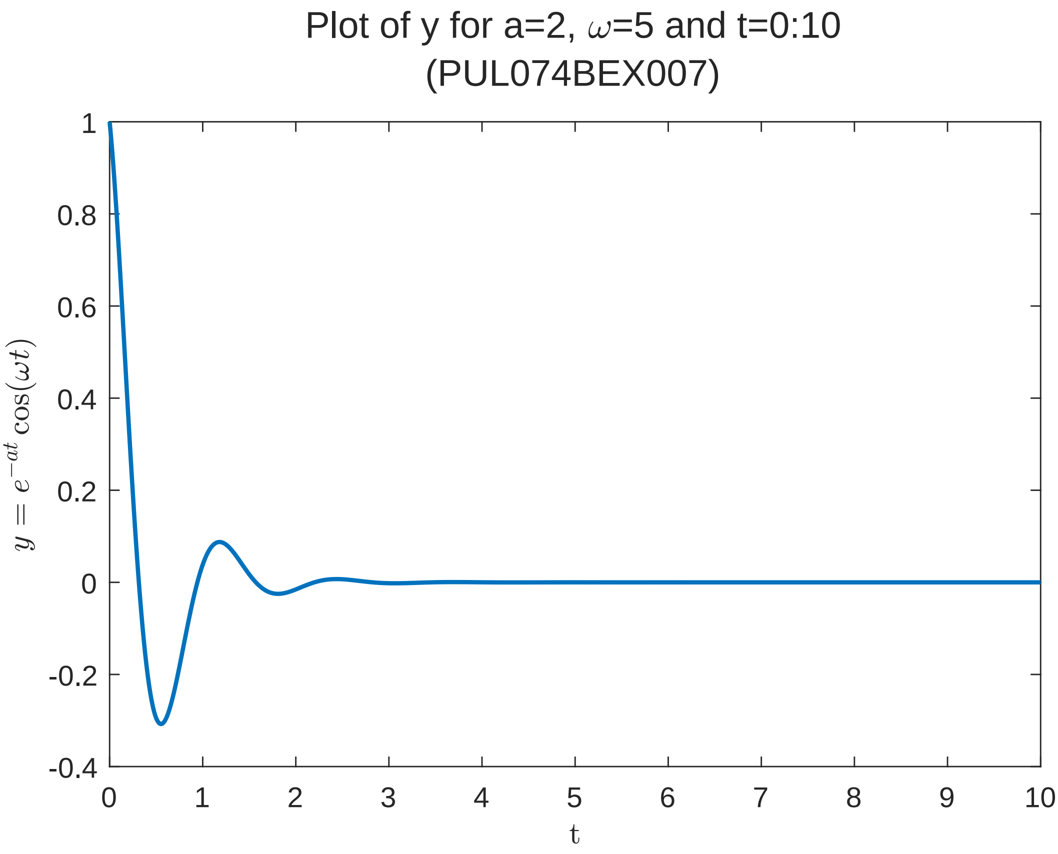

Figure 1: Plot for $y = e^{-at}cos(\omega t)$ for $a = 2$, $\omega = 5$, and $t = 0 : 10$

Problem 3

Try using the WHILE and the IF statements to calculate all the Fibonacci numbers so that the sum of two consecutive numbers is smaller than 10,000. How many are even? How many are odd? Try to plot them.

Hints

1. Matlab can increase the size of a vector as it is being created.

2. To determine whether a number n is even or odd you can use the function `rem(n,2)`. If `rem(n,2)` equals 0 then the number is even, otherwise it is odd.

```
function fibo_numbers = fibonacci_numbers(maxSum)
    f1=0;
    f2=1;
    fibo_numbers=[f1 f2];
    while (f1+f2) < maxSum
        next=f1+f2;
        f1=f2;
        f2=next;
        fibo_numbers(end+1)=next;
    end
end
```

Listing 5: Matlab function to return fibonacci numbers within a maximum sum

```
maxSum=10000;
fibo=fibonacci_numbers(maxSum);
l=tiledlayout(1,1);
str=sprintf('sum of two consecutive numbers less than %d',maxSum);
title(l,{'Fibonacci numbers with', str, '(PUL074BEX007)'})
nfibo=length(fibo);
fibo_even=[];
fibo_odd=[];
nexttile
hold on
xlim([0 nfibo])
for i = 1:nfibo
    if(rem(fibo(i),2)==0)
        fibo_even(end+1)=fibo(i);
        stem(i,fibo(i),'rs-','Linewidth',1.5)
    else
        fibo_odd(end+1)=fibo(i);
        stem(i,fibo(i),'bo-','Linewidth',1.5)
    end
end
xlabel('Index')
```

```
ylabel('Fibonacci Number')
legend('Even', 'Odd');
fprintf('Total fibonacci numbers: %d \n', nfibo);
fprintf('Even fibonacci numbers: %d \n', length(fibo_even));
fprintf('Odd fibonacci numbers: %d \n', length(fibo_odd));
print('lab_1_3','-depsc');
```

Listing 6: Matlab script plot even and odd fibonacci numbers and display their count

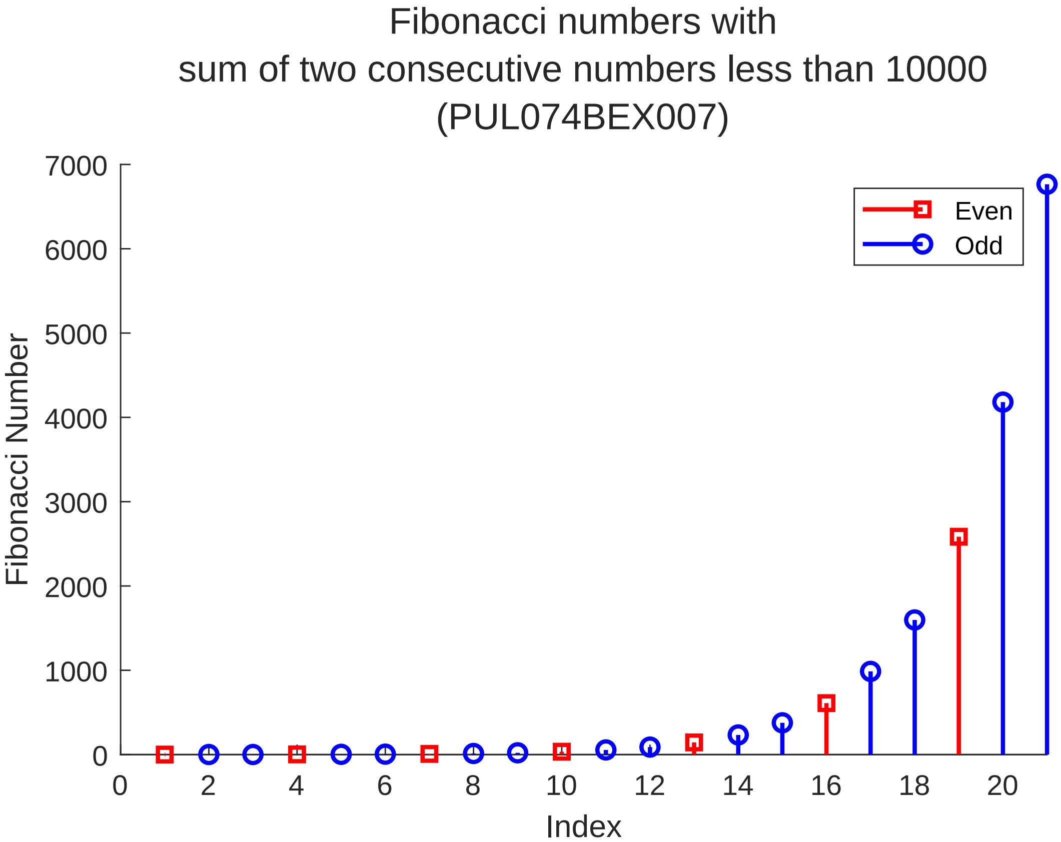

Figure 2: Plot for fibonacci numbers with sum of two consecutive numbers less than 10000

Command Window Observation

```
>> lab_1_3
Total fibonacci numbers: 21
Even fibonacci numbers: 7
Odd fibonacci numbers: 14
```

Problem 4

Given $f(x) = \dfrac{x^2+2x+3}{x+3}$**. Plot** $f(x)$ **for** $0 \leq x \leq 100$

```
x=linspace(0,100,1000);
f=(x.^2+2.*x+3)./(x+3);
l=tiledlayout(1,1);
title(l,{'Plot for f(x) for 0 \leq x \leq 100','(PUL074BEX007)'})
nexttile
plot(x,f,'Linewidth',1.5)
xlabel('$x$','interpreter','latex')
ylabel ('$f(x)=\frac{x^2+2x+3}{x+3}$','interpreter','latex')
print('lab_1_4','-depsc')
```

Listing 7: Matlab script to plot $f(x)$ for $0 \leq x \leq 100$

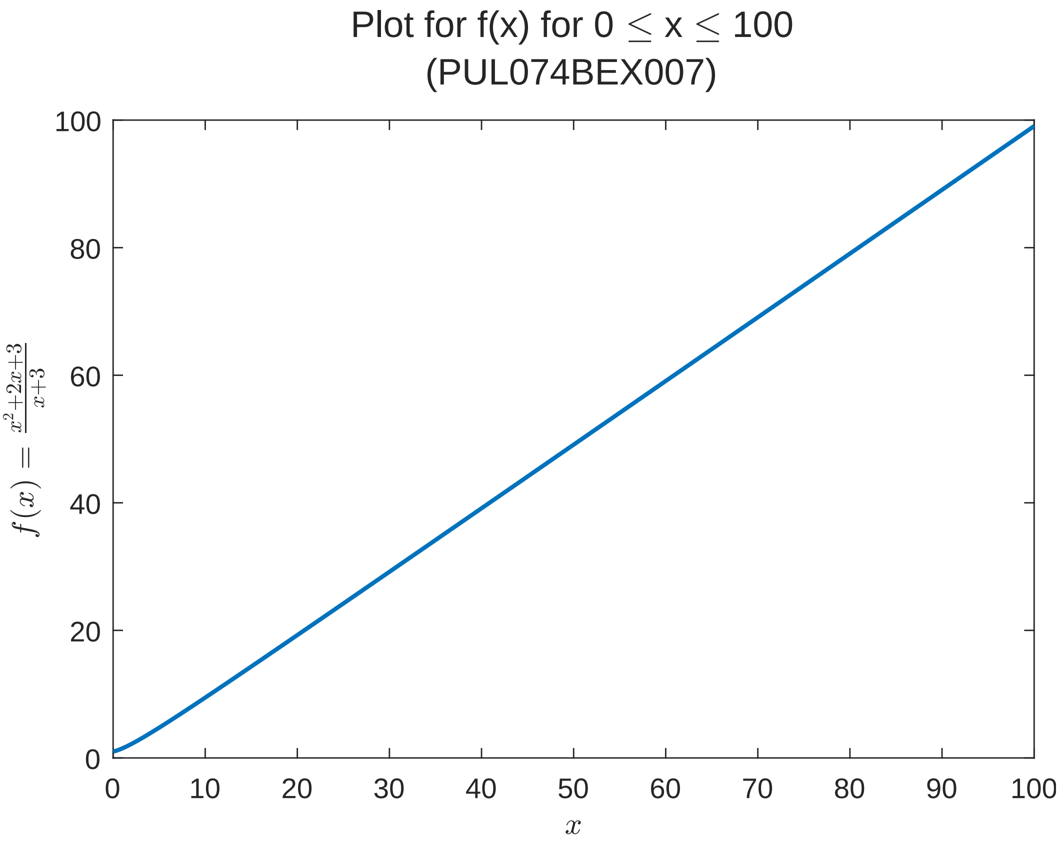

Figure 3: Plot for $f(x) = \dfrac{x^2+2x+3}{x+3}$ for $0 \leq x \leq 100$