UNIVERSIDAD
ANA G. MÉNDEZ
UAGM
Recinto de Gurabo
Escuela de Ingeniería
Ing. José Domingo Pérez

COMP 411 – Numerical Methods
Final Proyect – Dec/12/2022

Christian A. Piñero Torres (S01248423)
Prof. Yahya Masalmah
COMP 411 – CRN: 10154

Introduction

In this project we will be applying all the knowledge that we get from the Numerical Methods Course. Numerical Methods is solving a mathematical problem with an approximate computer method. In this project we will be implementing some of these methods. Some of the topics that we learned early on the course was hot to calculate the percentage of error, and we will use it to find the error percentage between Simpsons rule and the trapezoidal rule. We implement the Newton method, which is an algorithm use to find the approximation of the zeros or roots of and real function, but also, we can find the minimum or maximum of a function, by finding the zeros of the first derivate. The root finding procedure that uses several roots to find a better approximate of a root is the Secant method. This type of method is a recursive method. One of the last methods that we use in this project is the Lagrange method, which consists in find the local maxima and minima of a function.

Universidad Ana G. Méndez -Recinto de Gurabo
Electrical and Computer Engineering Program
Numerical Methods with Programming
COMP 411
Final Project
Due: Monday, December 12, 2022, at 11:59PM

Source Code

Problem 1: Write separate MATLAB functions for Newton's, the Secant method, the modified secant method. The functions should be written so that they can be called in MATLAB by typing.

1. [X, NumIters]=Newton(@f, @df, x0, TOL, MaxIters)
2. [X, NumIters]=Secant(@f, x0, x1, TOL, MaxIters)
3. [[X, NumIters]=ModifiedSecant(@f, δ, x0,TOL, MaxIters)

1. Newtons methods

```
clc
f= @(x) x^3-x-1;
df= @(x) 3*x^2-1;
x0= 1.5;
TOL= 0.001;
MaxIters=50;

Newton(f,df,x0,TOL,MaxIters);
function [ X, NumIters] = Newton( f,df,x0,TOL,MaxIters)
     for i = 1: MaxIters
          f0=vpa(subs(f,x0)); %Calculating the value of
     function at x0
     f0_der=vpa(subs(df,x0));
     X=x0-(f0/f0_der);
     NumIters=i-1;
     err = abs(X-x0);
     if err <TOL
          break;
     end
   end

fprintf('Root is : %.2f with accuracy: %.2f\n',X,err);
fprintf('Number of iterations:%d\n',NumIters); end
```

2. Secant Method

```
% Matlab code of secant method
% f is input function,x0 is first point of guess
interval,
%x1 is second point of guess interval,TOL is given
tolerance
function [X,Numitr]= secant(f,x0,x1,ToL,Maxitr)
x(1)=x0;
x(2)=x1;
Numitr=0;
   for i=3:Maxitr
      x(i) = x(i-1) - (f(x(i-1)))*((x(i-1) - x(i-
2))/(f(x(i-1)) - f(x(i-2))));
      Numitr=Numitr+1;
      if abs((x(i)-x(i-1))/x(i))*100<ToL
            X=x(i);
            Numitr;
            break
     end
end
```

3. **Modified Secant**

```
% Matlab code of Modified secant method
% f is input function,x0 is starting point
%TOL is given tolerance,delta is small perturbation factor
function [X,Numitr]= Modified_secant(f,x0,ToL,delta,Maxitr)
x(1)=x0;
Numitr=0;
for i=2:Maxitr
    x(i) = x(i-1) - (f(x(i-1))*delta*x(i-1))/(f(x(i-
    1)+delta*x(i-1)) - f(x(i-1)));
    Numitr=Numitr+1;
    if abs((x(i)-x(i-1))/x(i))*100<ToL
      X=x(i);
      Numitr;
      break
  end
end
```

Application Problems

Problem 2: Figure 1 shows a circuit with a resistor, an inductor, and a capacitor in parallel. Kirchhoff's rules can be used to express the impedance of the system as:

$$\frac{1}{Z} = \sqrt{\frac{1}{R^2} + \left(wC - \frac{1}{wL}\right)^2}$$

Where Z =impedance (Ω) and w=the angular frequency. Find the w that result in an impedance of 75 Ω using the implemented methods in problem 1.

1. Newton x0=1
2. Secant method with x0=1, x1=100
3. Modified Secant method with x0=1

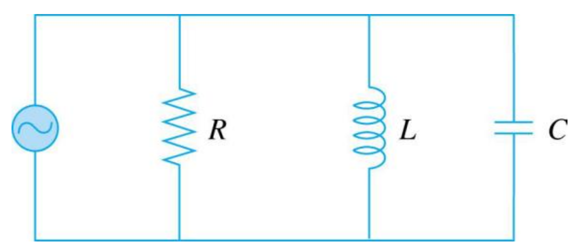


Figure 1. RLC Circuit.

Use the following parameters: R=225 Ω, $C = 0.6x10^{-6}\,F$ and $L = 0.5H$. Determine how many iterations of each method are necessary to determine the answer to $\varepsilon_s = 0.1\%$.
If any difficulties arise, please use the graphical method to explain them.

a) Newton $x0 = 1$

$$X_{n+1} = X_n - \frac{F(X_n)}{F'(X_n)}$$

$F(X_n) \rightarrow$ Funcion dada con el valor de X

$F'(X_n) \rightarrow$ Derivada de la Funcion

$$\left(wC - \frac{1}{wL}\right)^2 = \frac{1}{Z^2} - \frac{1}{R^2}$$

$$wC - \frac{1}{wL} = \sqrt{\frac{1}{Z^2} - \frac{1}{R^2}}$$

$$wC - \frac{1}{wL} = \sqrt{\frac{1}{(75)^2} - \frac{1}{(226)^2}}$$

$$\frac{w^2CL - 1}{wL} = 0.0125$$

$$w^2CL - 0.0125wL - 1 = 0$$

$$w^2(0.6 \times 10^6)(0.5) - 0.0125w(0.5) - 1 = 0$$

$$(3 \times 10^{-7})w^2 - 0.0625w - 1 = 0$$

* Implementamos Newton's methods

$$F(w) = (3 \times 10^{-7})w^2 - 0.0625w - 1 = 0$$

$$F'(w) = 2(3 \times 10^{-7})w - 0.0625 = 0$$

$$w_1 = w_0 - \frac{F(w)}{F'(w)}$$

$$w_1 = 1 - \frac{(3 \times 10^{-7})w^2 - 0.0625w - 1}{2(3 \times 10^{-7})w - 0.0625} = 2.08 \times 10^6\ Hz$$

b) Secant method's, $w_0 = 1$; $w_1 = 100$, Similar

$$x_{n+1} = x_n - \frac{F(x_n)(x_{n-1} - x_n)}{F(x_{n-1}) - F(x_n)}$$

c) Modified Secant method, $w_0 = 1$, Similar

$$x_{n+1} = x_n - \frac{\delta x_n \, F(x_n)}{F(x_n + \delta x_n) - F(x_n)} \qquad \delta = 0.01$$

$w = 2.08 \times 10^6$ Hz

Problem 3: Given the following data points,

x	1.6	2	2.5	3.2	4	4.5
y	2	8	14	15	8	2

1. Construct the divided difference table for this table
2. Find the Newton's interpolating polynomial using the divided difference technique (1st , 2nd order) find f(2.8).
3. Repeat the previous step using Lagrange interpolation
4. Repeat 1 and 2 by running the given code for each technique in the handouts
5. Write a MATLAB code that plot the polynomials found in 2 and 3 on the same scale.

1. Tabla ; $X = X_i$ y $Y = F(X_i)$

X_i	$F(x_i)$	1st order	2nd order	3rd order	4th order	5th order
1.6	2					
		$\frac{8-2}{2-1.6} = 10$				
2	8		$\frac{12-10}{2.5-1.6} = 2.22$			
		$\frac{14-8}{2.5-2} = 12$		$\frac{8.81-2.22}{3.2-1.6} = 6.9$		
2.5	14		$\frac{1.43-12}{3.2-2} = 8.81$		$\frac{1.01-(-6.9)}{4-1.6} = 3.3$	
		$\frac{15-14}{3.2-2.5} = 1.43$		$\frac{-6.79-(8.81)}{4-2} = 1.01$		$\frac{0.6-3.3}{4.5-1.6} = -1$
3.2	15		$\frac{-8.75-1.43}{4-25} = -6.79$		$\frac{2.15-1.01}{4.5-2} = 0.6$	
		$\frac{8-15}{4-3.2} = -8.75$		$\frac{-2.5-(-6.79)}{4.5-2.5} = 2.15$		
4	8		$\frac{-12-(-8.75)}{4.5-3.2} = 2.5$			
		$\frac{2-8}{4.5-4} = -12$				
4.5	2					

2. Newton's Interpolating

$$F(x) = F(x_0) + (x - x_0)F(x_0, x_1) + (x - x_0)(x - x_1)F(x_0, x_1, x_2)$$

$$F(2.8) = F(1.6) + (2.8 - 1.6)(2.22) + (2.8 - 1.6)(2.8 - 2)(-6.9)$$

$$F(2.8) = 2 + 2.664 + (-6.624)$$

$$F(2.8) = -1.96$$

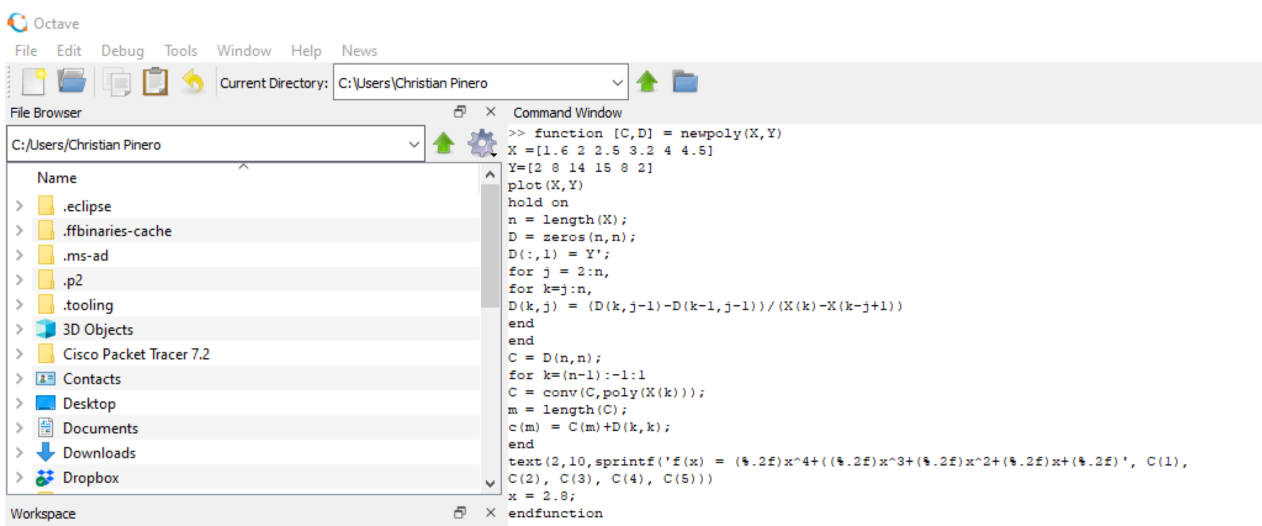
3. La Grange:

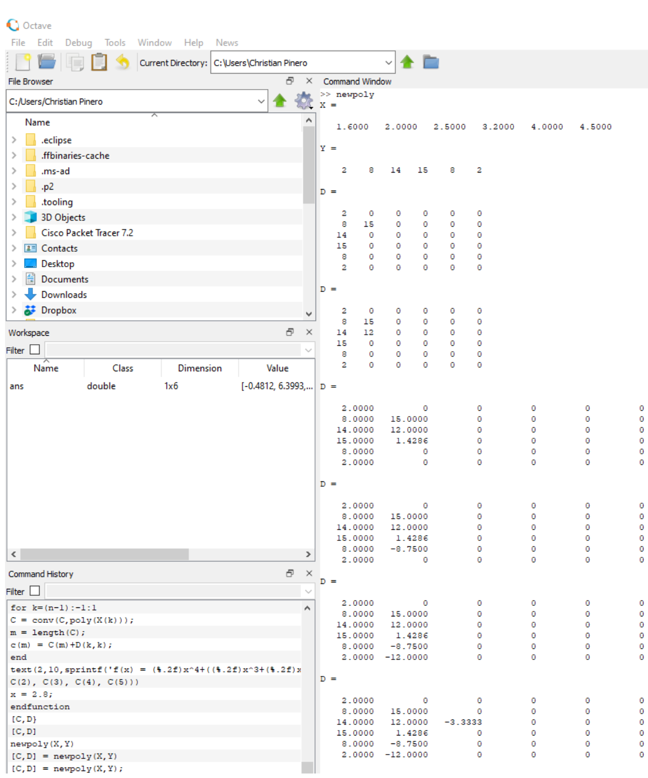
$$F(x) = \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)(x-x_5)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)(x_0-x_5)} F(x_0) + \ldots + \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_5-x_0)(x_5-x_1)(x_5-x_2)(x_5-x_3)(x_5-x_4)} F(x_5)$$

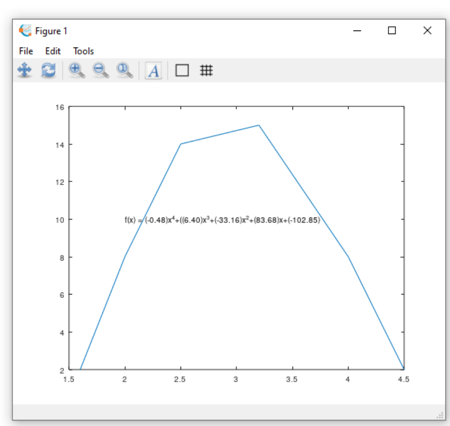
$$F(2.8) = \frac{(2.8-2)(2.8-2.5)(2.8-3.2)(2.8-4)(2.8-4.5)}{(1.6-2)(1.6-2.5)(1.6-3.2)(1.6-4)(1.6-4.5)} \times 2 + \frac{(2.8-1.6)(2.8-2.5)(2.8-3.2)(2.8-4)(2.8-4.5)}{(2-1.6)(2-2.5)(2-3.2)(2-4)(2-4.5)} \times 8$$

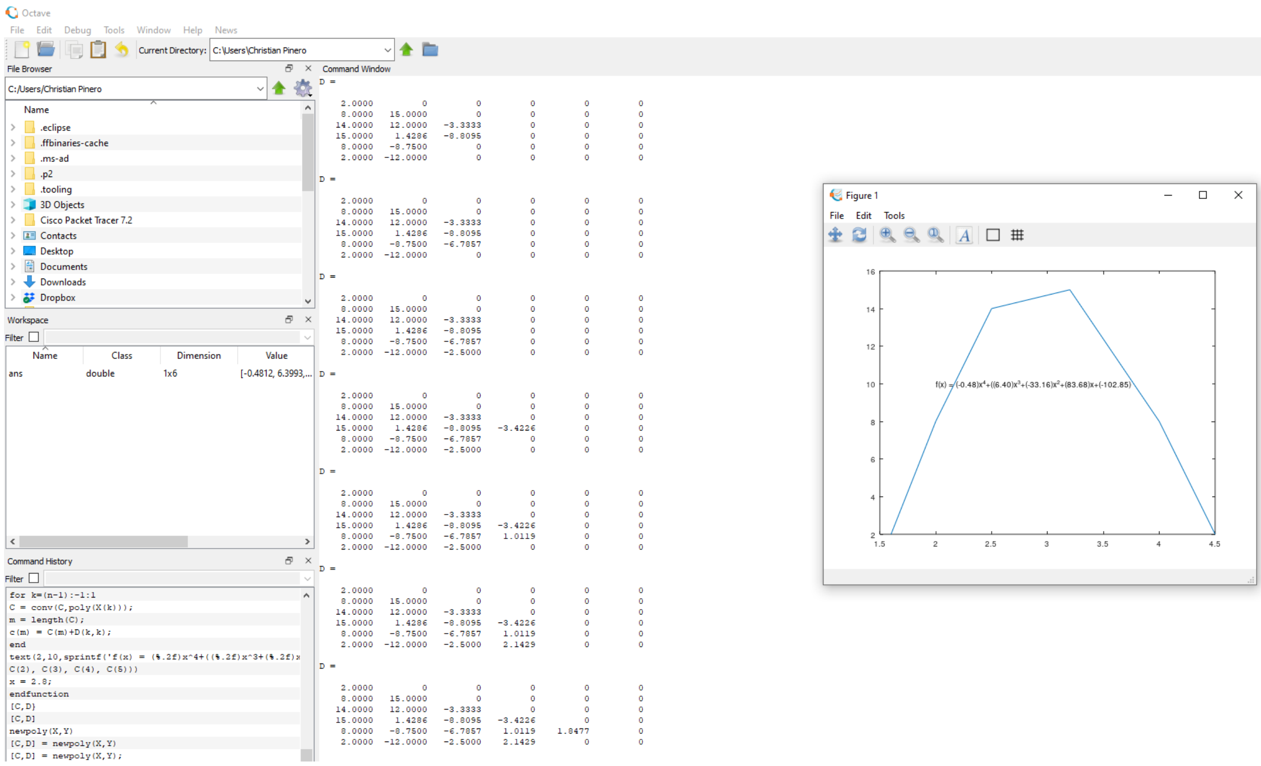
$$+ \frac{(2.8-1.6)(2.8-2)(2.8-3.2)(2.8-4)(2.8-4.5)}{(2.5-1.6)(2.6-2)(2.5-3.2)(2.5-4)(2.5-4.5)} \times 14 + \frac{(2.8-1.6)(2.8-2)(2.8-4)(2.8-4.5)}{(3.2-1.6)(4.5-2)(4.5-2.5)(4.5-3.2)(4.5-4)} \times 2$$

$$F(2.8) = 15.60$$

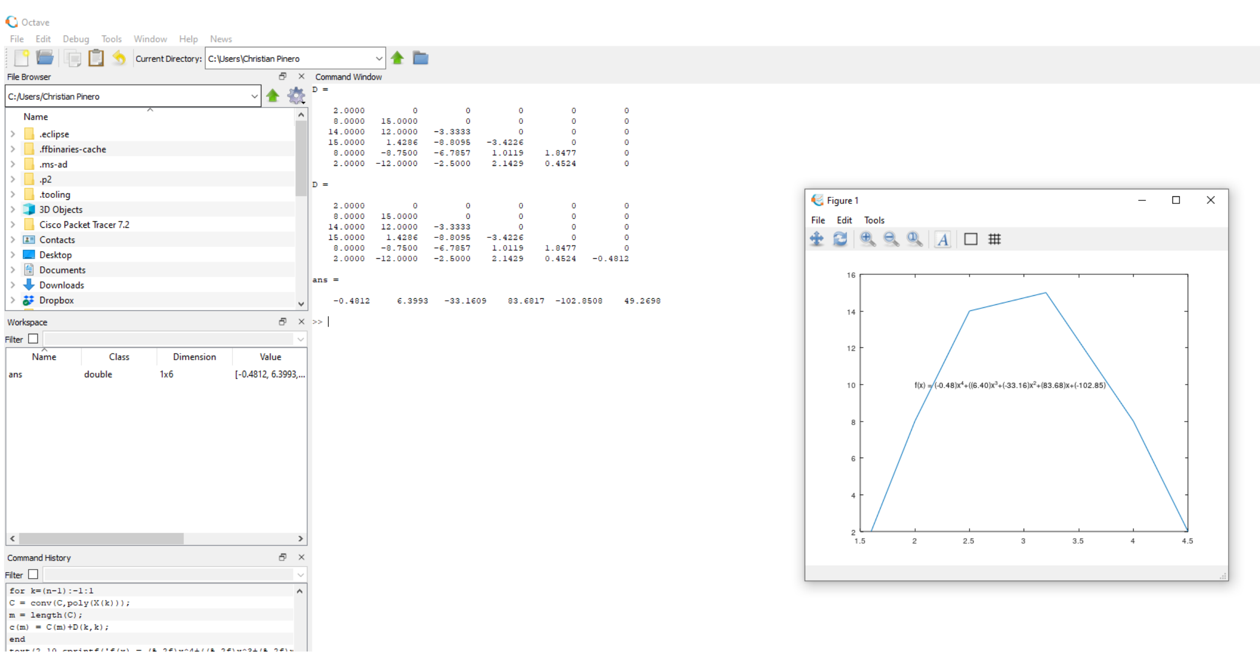






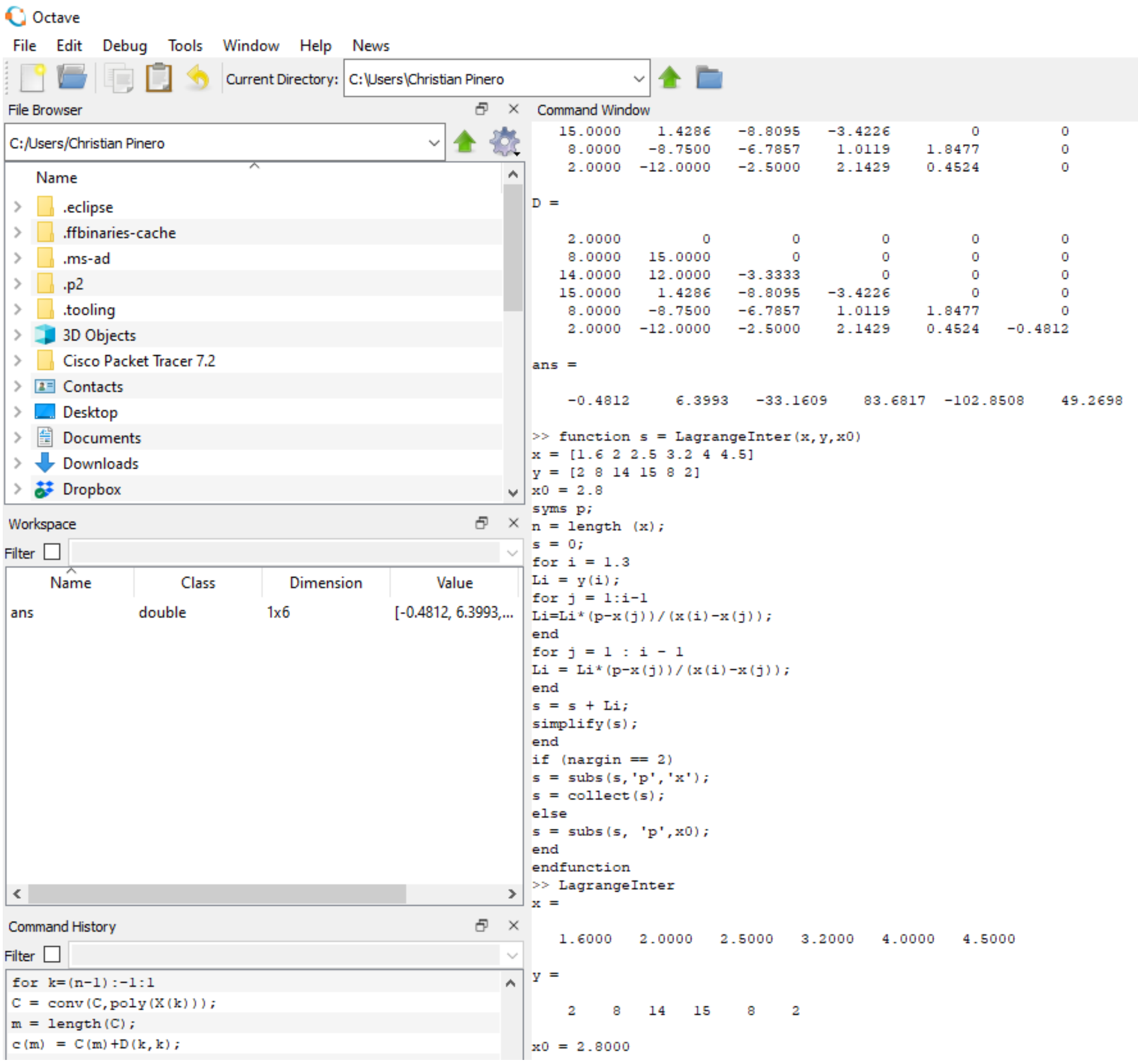


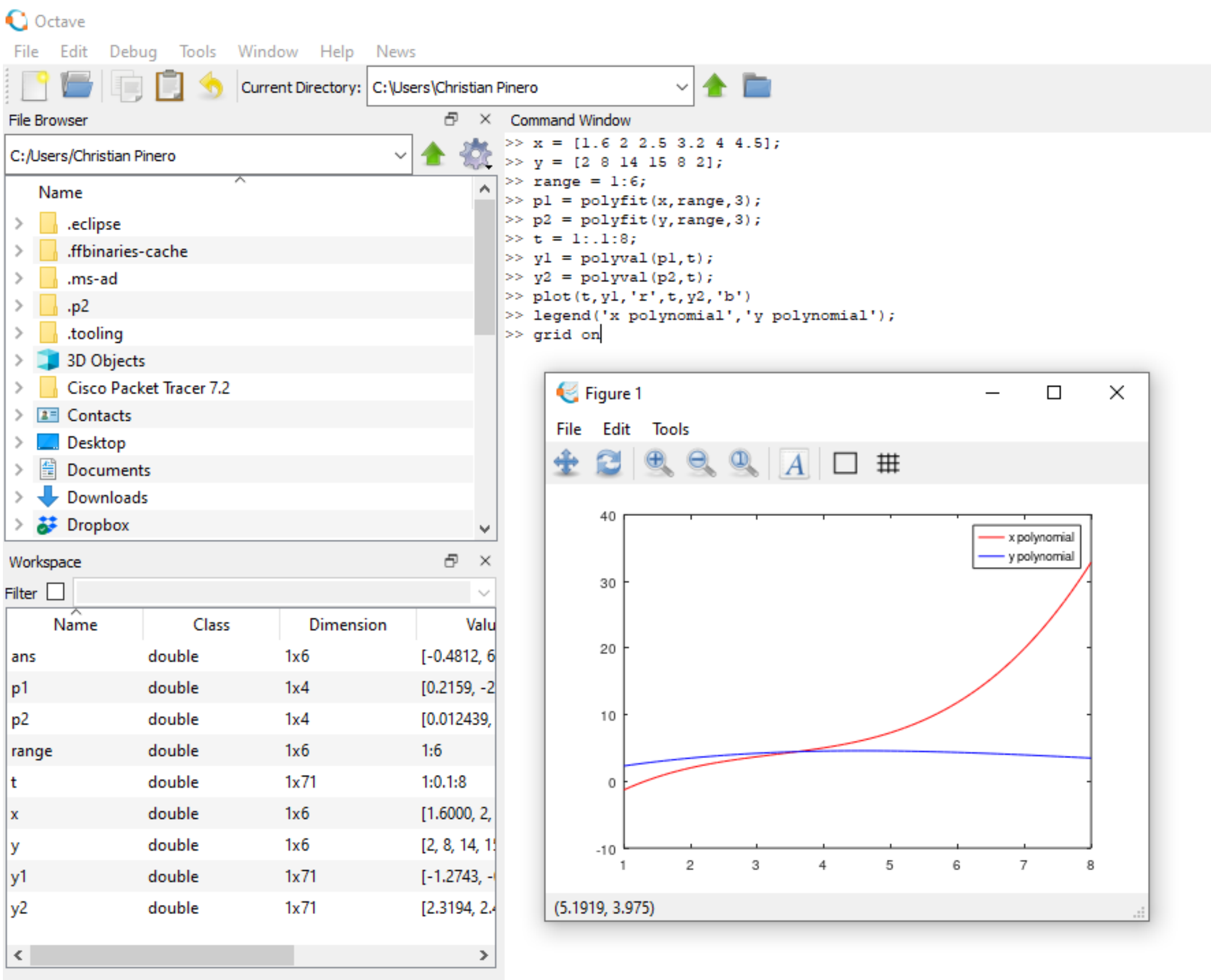
Octave
File Edit Debug Tools Window Help News
Current Directory: C:\Users\Christian Pinero
File Browser
C:/Users/Christian Pinero
Name
.eclipse
.ffbinaries-cache
.ms-ad
.p2
.tooling
3D Objects
Cisco Packet Tracer 7.2
Contacts
Desktop
Documents
Downloads
Dropbox
Workspace
Filter
Name Class Dimension Value
ans double 1x6 [-0.4812, 6.3993,...
Command History
for k=(n-1):-1:1
C = conv(C,poly(X(k)));
m = length(C);
c(m) = C(m)+D(k,k);
end
text(2,10,sprintf('f(x) = (%.2f)x^4+((%.2f)x^3+(%.2f)x
C(2), C(3), C(4), C(5)))
x = 2.8;
endfunction
[C,D}
[C,D]
newpoly(X,Y)
[C,D] = newpoly(X,Y)
[C,D] = newpoly(X,Y);
Command Window
D =
2.0000 0 0 0 0 0
8.0000 15.0000 0 0 0 0
14.0000 12.0000 -3.3333 0 0 0
15.0000 1.4286 -8.8095 -3.4226 0 0
8.0000 -8.7500 -6.7857 1.0119 1.8477 0
2.0000 -12.0000 -2.5000 2.1429 0 0
Figure 1
File Edit Tools
f(x) = (-0.48)x⁴+((6.40)x³+(-33.16)x²+(83.68)x+(-102.85)



Octave
File Edit Debug Tools Window Help News
Current Directory: C:\Users\Christian Pinero
Command Window
D =
2.0000 0 0 0 0 0
8.0000 15.0000 0 0 0 0
14.0000 12.0000 -3.3333 0 0 0
15.0000 1.4286 -8.8095 -3.4226 0 0
8.0000 -8.7500 -6.7857 1.0119 1.8477 0
2.0000 -12.0000 -2.5000 2.1429 0.4524 0
D =
2.0000 0 0 0 0 0
8.0000 15.0000 0 0 0 0
14.0000 12.0000 -3.3333 0 0 0
15.0000 1.4286 -8.8095 -3.4226 0 0
8.0000 -8.7500 -6.7857 1.0119 1.8477 0
2.0000 -12.0000 -2.5000 2.1429 0.4524 -0.4812
ans =
-0.4812 6.3993 -33.1609 83.6817 -102.8508 49.2698
>>
Workspace
Name Class Dimension Value
ans double 1x6 [-0.4812, 6.3993,...
Command History
for k=(n-1):-1:1
C = conv(C,poly(X(k)));
m = length(C);
c(m) = C(m)+D(k,k);
end
Figure 1
File Edit Tools
f(x) = (-0.48)x⁴+((6.40)x³+(-33.16)x²+(83.68)x+(-102.85)

Lagrange


Octave
File Edit Debug Tools Window Help News
Current Directory: C:\Users\Christian Pinero
File Browser
C:/Users/Christian Pinero
Name
.eclipse
.ffbinaries-cache
.ms-ad
.p2
.tooling
3D Objects
Cisco Packet Tracer 7.2
Contacts
Desktop
Documents
Downloads
Dropbox
Workspace
Filter
Name Class Dimension Value
ans double 1x6 [-0.4812, 6.3993,...
Command History
Filter
for k=(n-1):-1:1
C = conv(C,poly(X(k)));
m = length(C);
c(m) = C(m)+D(k,k);
Command Window
15.0000 1.4286 -8.8095 -3.4226 0 0
8.0000 -8.7500 -6.7857 1.0119 1.8477 0
2.0000 -12.0000 -2.5000 2.1429 0.4524 0
D =
2.0000 0 0 0 0 0
8.0000 15.0000 0 0 0 0
14.0000 12.0000 -3.3333 0 0 0
15.0000 1.4286 -8.8095 -3.4226 0 0
8.0000 -8.7500 -6.7857 1.0119 1.8477 0
2.0000 -12.0000 -2.5000 2.1429 0.4524 -0.4812
ans =
-0.4812 6.3993 -33.1609 83.6817 -102.8508 49.2698
>> function s = LagrangeInter(x,y,x0)
x = [1.6 2 2.5 3.2 4 4.5]
y = [2 8 14 15 8 2]
x0 = 2.8
syms p;
n = length (x);
s = 0;
for i = 1.3
Li = y(i);
for j = 1:i-1
Li=Li*(p-x(j))/(x(i)-x(j));
end
for j = 1 : i - 1
Li = Li*(p-x(j))/(x(i)-x(j));
end
s = s + Li;
simplify(s);
end
if (nargin == 2)
s = subs(s,'p','x');
s = collect(s);
else
s = subs(s, 'p',x0);
end
endfunction
>> LagrangeInter
x =
1.6000 2.0000 2.5000 3.2000 4.0000 4.5000
y =
2 8 14 15 8 2
x0 = 2.8000

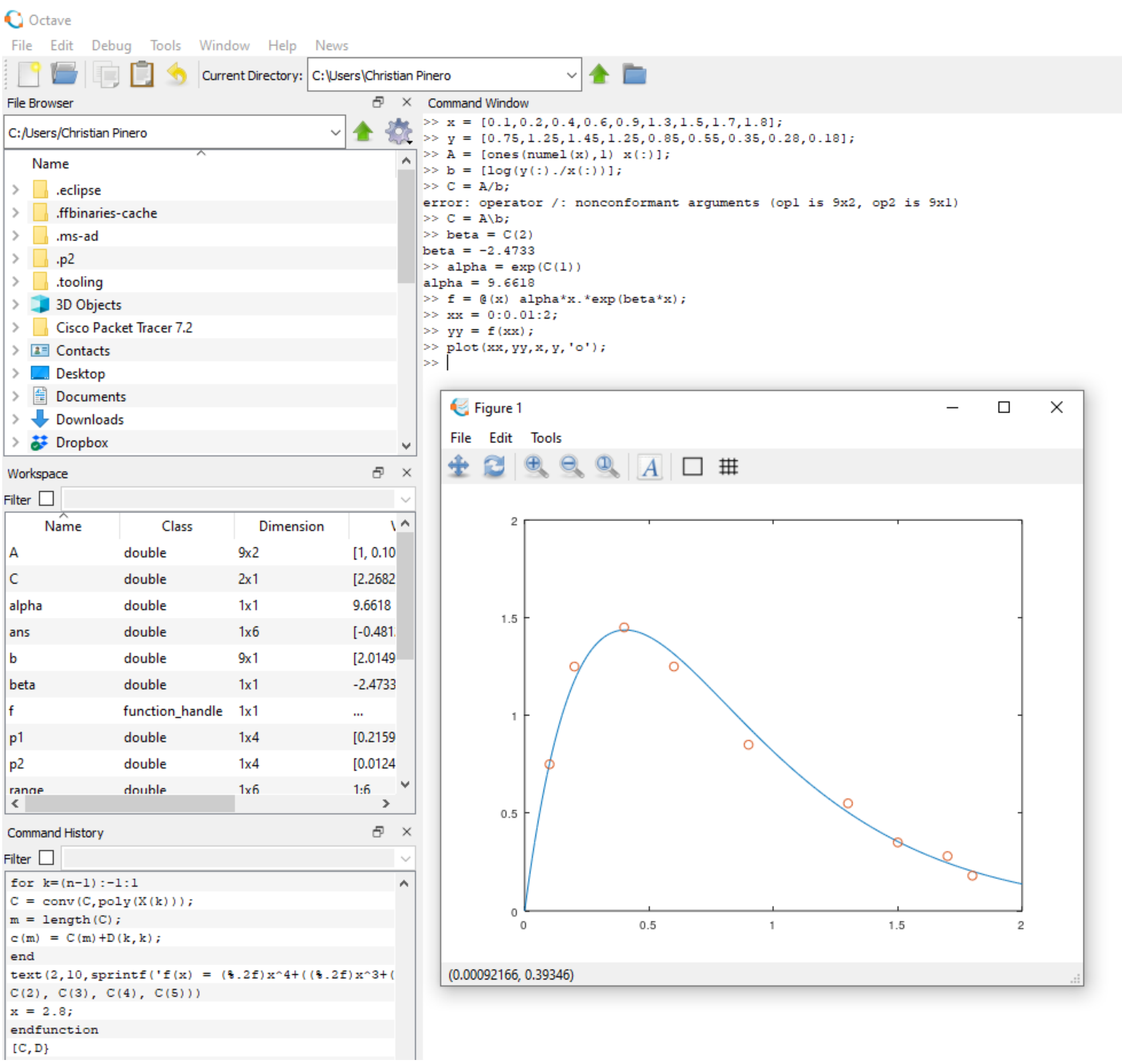

Octave
File Edit Debug Tools Window Help News
Current Directory: C:\Users\Christian Pinero
File Browser
C:/Users/Christian Pinero
Name
.eclipse
.ffbinaries-cache
.ms-ad
.p2
.tooling
3D Objects
Cisco Packet Tracer 7.2
Contacts
Desktop
Documents
Downloads
Dropbox
Workspace
Filter
Name Class Dimension Valu
ans double 1x6 [-0.4812, 6
p1 double 1x4 [0.2159, -2
p2 double 1x4 [0.012439,
range double 1x6 1:6
t double 1x71 1:0.1:8
x double 1x6 [1.6000, 2,
y double 1x6 [2, 8, 14, 1
y1 double 1x71 [-1.2743, -
y2 double 1x71 [2.3194, 2.
Command Window
>> x = [1.6 2 2.5 3.2 4 4.5];
>> y = [2 8 14 15 8 2];
>> range = 1:6;
>> p1 = polyfit(x,range,3);
>> p2 = polyfit(y,range,3);
>> t = 1:.1:8;
>> y1 = polyval(p1,t);
>> y2 = polyval(p2,t);
>> plot(t,y1,'r',t,y2,'b')
>> legend('x polynomial','y polynomial');
>> grid on
Figure 1
File Edit Tools
40
30
20
10
0
-10
1 2 3 4 5 6 7 8
x polynomial
y polynomial
(5.1919, 3.975)

Problem 4: Use nonlinear regression to estimate α and β of the following model

$$y = \alpha_4 x e^{\beta_4 x}$$

Based on the following data. Develop a plot of your fitting model along with the data.

X	0.1	0.2	0.4	0.6	0.9	1.3	1.5	1.7	1.8
Y	0.75	1.25	1.45	1.25	0.85	0.55	0.35	0.28	0.18



Problem 5: The function $f(x) = 2e^{-1.5x}$ can be used to generate the following table of unequally spaced data:

x	0	0.05	0.15	0.25	0.35	0.475	0.6
$f(x)$	2	1.8555	1.5970	1.3746	1.1831	0.9808	0.8131

Evaluate the integral from a 5 0 to b 5 0.6 using (a) analytical means, (b) the trapezoidal rule, and (c) a combination of the trapezoidal and Simpson's rules; employ Simpson's rules wherever possible to obtain the highest accuracy. For (b) and (c), compute the percent relative error (ε_t)

a) Analytical Means:

$F(x) = 2e^{-1.5x}$

$$\int_a^b f(x)\,dx = \int_0^{0.6} (2e^{-1.5x})\,dx$$

$$= \left[\left(\frac{2}{-1.5}\right) e^{-1.5x} \right]_0^{0.6}$$

$$= \left[-1.33333\, e^{-1.5x} \right]_0^{0.6}$$

$$= -1.33333 \left[e^{-1.5\times 0.6} - e^{0} \right]$$

$$= 0.79125$$

b) Trapezoidal Rule ; $h = 0.1$

$$\int_a^b e^{-1.5x} dx = (0.05-0) \times \frac{2+1.8555}{2} + (0.15-0.05) \times \frac{1.85555+1.5970}{2} + (0.25-0.15) \times \frac{1.5970+1.3746}{2} + (0.35-0.25) \times \frac{1.3746+1.1831}{2}$$

$$+ (0.475-0.35) \times \frac{1.1831+0.9808}{2} + (0.6-0.475) \times \frac{0.8131+0.9808}{2}$$

$$\int_0^{0.6} e^{-1.5x} dx = 0.4175925 + 0.375247$$

$$= 0.7928395$$

c) Simpson Rule

$$\int_0^{0.6} 2e^{-1.5x} dx = (0.05-0) \times \frac{2+1.8555}{2} + (0.35-0.05) \times \frac{1.8555+3(1.5970+1.3746)+1.1831}{8} + (0.6-0.35)\,\frac{1.1831+4(0.9808)+0.8131}{6}$$

$$= 0.71899$$

d) Trapezoidal Rule error

$$\text{Error} = \frac{0.79125 - 0.7928395}{0.79125} \times 100$$

$$= -0.20\,\%$$

Simpson Rule Error

$$\text{Error} = \frac{0.79125 - 0.71899}{0.79125} \times 100$$

$$= 9.13\,\%$$

Problem 6: The logistic model is used to simulate population as in

$$\frac{dp}{dt} = k_{gm}\left(1 - \frac{p}{p_{max}}\right)p$$

Where p=population, k_{gm}=the maximum growth rate under unlimited conditions, and p_{max}= the carrying capacity. Simulate the world's population from 1950 to 2000 using one of the numerical methods discussed in the course. Employ the following initial conditions and parameter values for your simulation: p_0 (in 1950) =2555 million people, k_{gm}=0.026/yr, and p_{max}= 12,000 million people. Have the function generate output corresponding to the dates for the following measured population data. Develop a plot of your simulation along with the data.

t	1950	1960	1970	1980	1990	2000
p	2555	3040	3708	4454	5276	6079

$$K_{gm} = \left(1 - \frac{P}{P_{max}}\right)P$$

$$\int \frac{dp}{\left(1-\frac{P}{P_{max}}\right)P} = \int K_{gm}\, dt$$

$$\frac{1}{P\left(1-\frac{P}{P_{max}}\right)} = \frac{1}{P\left(\frac{P_{max}-P}{P_{max}}\right)}$$

$$= \frac{P_{max}}{P(P_{max}-P)}$$

$$= \frac{P_{max}}{P} + \frac{1}{P_{max}-P}$$

$$= \frac{1}{P} + \frac{1}{P_{max}-P}$$

$$\int \frac{dp}{P} + \int \frac{dp}{P_{max}-P} = \int K dt$$

$$\int \frac{dp}{P} + \int \frac{dp}{1-P} = \int K_{gm}\, dt$$

$$\ln |P| - \ln |1-P| = K_{gm}t + C$$

$$\ln \left|\frac{1-P}{P}\right| = -K_{gm}t - C$$

$$\left|\frac{1-P}{P}\right| = e^{-kt-c}$$

$$\frac{1-P}{P} = Ae^{-kt}$$

$$A(\pm e^{-t})$$

$$P = \frac{1}{1+Ae^{K_{gm}t}} \quad ; \quad A = \frac{1-P_0}{P_0}$$

$P_0 = 1950 \rightarrow 2555$; $K_{gm} = 0.026$; $P_{max} = 12000$

$$P = \frac{1}{1-Ae^{-K_{gm}t}}$$

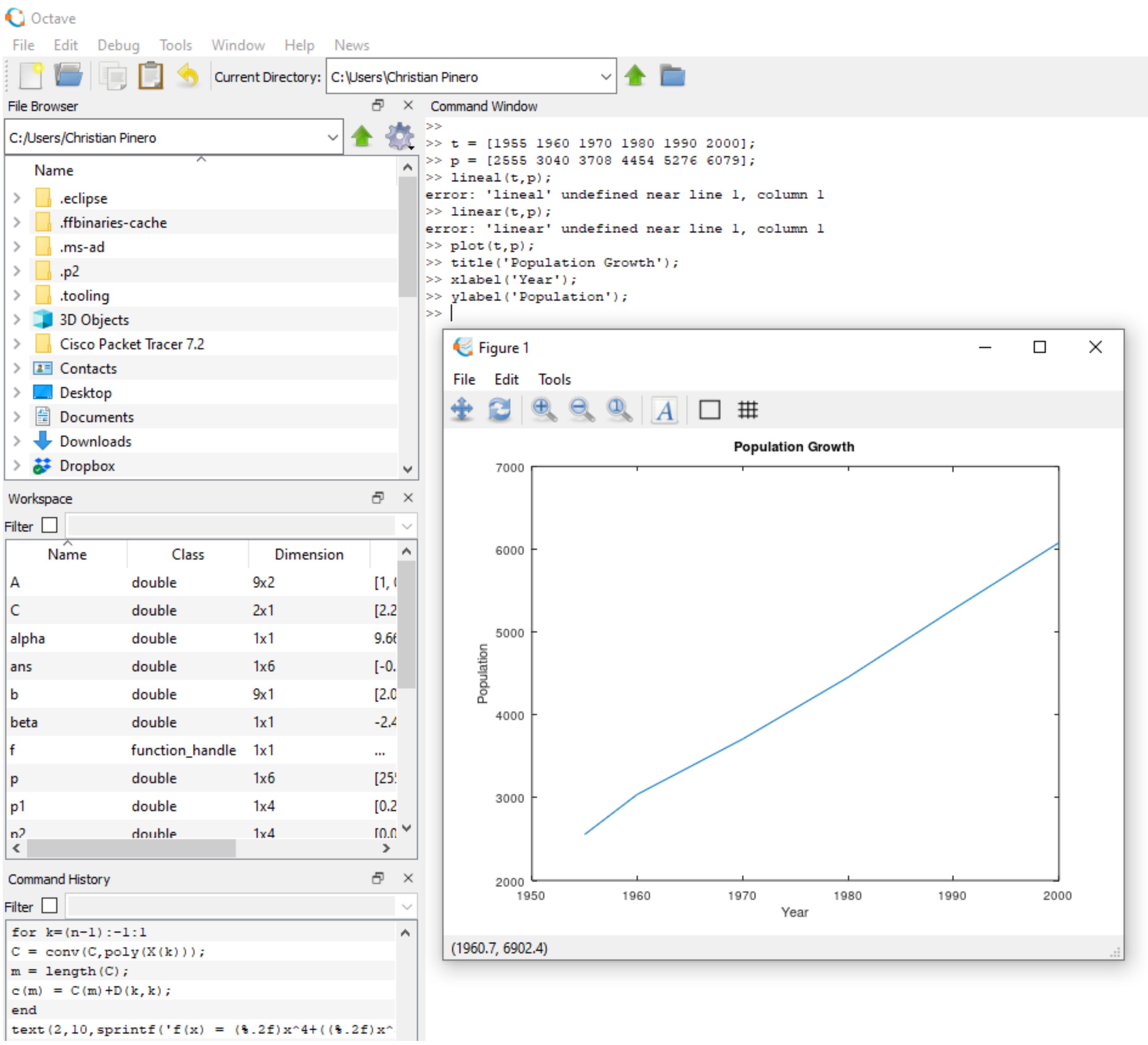
$$= \frac{1-P_0}{P_0} = -0.999$$

$$P = \frac{1}{1-(-0.999)e^{-0.026}}$$

$$P = 0.5067$$

$$\frac{dp}{dt} = 0.026\left(1 - \frac{0.5067}{12000}\right)0.5067$$

$$= 0.013$$


Octave
File Edit Debug Tools Window Help News
Current Directory: C:\Users\Christian Pinero
File Browser
C:/Users/Christian Pinero
Name
.eclipse
.ffbinaries-cache
.ms-ad
.p2
.tooling
3D Objects
Cisco Packet Tracer 7.2
Contacts
Desktop
Documents
Downloads
Dropbox
Workspace
Filter
Name Class Dimension
A double 9x2
C double 2x1
alpha double 1x1
ans double 1x6
b double 9x1
beta double 1x1
f function_handle 1x1
p double 1x6
p1 double 1x4
Command History
Filter
for k=(n-1):-1:1
C = conv(C,poly(X(k)));
m = length(C);
c(m) = C(m)+D(k,k);
end
text(2,10,sprintf('f(x) = (%.2f)x^4+((%.2f)x^
Command Window
>>
>> t = [1955 1960 1970 1980 1990 2000];
>> p = [2555 3040 3708 4454 5276 6079];
>> lineal(t,p);
error: 'lineal' undefined near line 1, column 1
>> linear(t,p);
error: 'linear' undefined near line 1, column 1
>> plot(t,p);
>> title('Population Growth');
>> xlabel('Year');
>> ylabel('Population');
>>
Figure 1
File Edit Tools
Population Growth
Population
Year
7000
6000
5000
4000
3000
2000
1950
1960
1970
1980
1990
2000
(1960.7, 6902.4)

Conclusion

In this class and project, we learn on how us, as future engineers, can provide real solutions where problems can't not be implemented as coding in a programming language. By learning how to calculate and resolve this type of problems we also can correct if the results are correct or not. In some way, we can use it as a verification step of what we are doing is correct. Also, by using this type of methods, we can compare the methods on what method will converge fast than the others or if they converge or not. Numerical Methods have become an important way of doing calculations that we see in the everyday life and validating if they are correct or not, but also make a great success on the results that we get from them.