Math 481 #

Samuel Roiz (4/20) math 481A

---

1. Implementing Lagrange's Interpolating Polynomial

Code:

File name... langrange.m

```
function poly2= lagrange(X,f,x)
poly2=0;
for i=1:length(X)
p2=1;
for j=1:length(X)
if(i==j)
p2=p2*f(i);
else
p2=p2*(x-X(j))/(X(i)-X(j));
end
end
poly2=poly2+p2;
end
end
```

File name... executeforlagrange

```
clc
clear all
figure
s=;
for n=3:6
X=linspace(1,2.9,n);
Y=1./X;
x=linspace(1,3,100);
p=zeros(1,100);
for i=1:100
p(i)=lagrange(X,Y,x(i));
end
```

```
plot(x,p,'Color',rand(3,1));
hold on;
sn-2=sprintf('n=
end
plot(x,1./x,'Color',rand(3,1));
legend(s,'actual');
```

Outcome Figure Example:

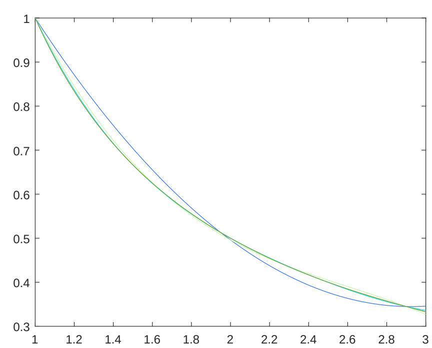

Figure 1: Lagrange Example

green is n = 4, blue is n = 3, and red is actual.

2. Cubic Spine Program

Code:
File name... cubic$_{s}pline.m$
$n = input('Enternfor(n+1)nodes, n :');$
$x = zeros(1, n+1);$
$a = zeros(1, n+1);$

```
for i = 0:n
fprintf('Enter x(
x(i+1) = input(' ');
a(i+1) = input(' ');
end


m = n - 1;
h = zeros(1,m+1);
for i = 0:m
h(i+1) = x(i+2) - x(i+1);
end
xa = zeros(1,m+1);
for i = 1:m
xa(i+1) = 3.0*(a(i+2)*h(i)-a(i+1)*(x(i+2)-x(i))+a(i)*h(i+1))/(h(i+1)*h(i));
end
xl = zeros(1,n+1);
xu = zeros(1,n+1);
xz = zeros(1,n+1);
xl(1) = 1;
xu(1) = 0;
xz(1) = 0;
for i = 1:m
xl(i+1) = 2*(x(i+2)-x(i))-h(i)*xu(i);
xu(i+1) = h(i+1)/xl(i+1);
xz(i+1) = (xa(i+1)-h(i)*xz(i))/xl(i+1);
end
xl(n+1) = 1;
xz(n+1) = 0;
b = zeros(1,n+1);
```

```
c = zeros(1,n+1);
d = zeros(1,n+1);
c(n+1) = xz(n+1);
for i = 0:m
j = m-i;
c(j+1) = xz(j+1)-xu(j+1)*c(j+2);
b(j+1) = (a(j+2)-a(j+1))/h(j+1) - h(j+1) * (c(j+2) + 2.0 * c(j+1)) / 3.0;
d(j+1) = (c(j+2) - c(j+1)) / (3.0 * h(j+1));
end
fprintf('numbers x(0), ..., x(n) are:');
for i = 0:n
fprintf('
end
fprintf('coefficients of the spline on the subintervals are:');
fprintf(' a(i) b(i) c(i) d(i)');
for i = 0:m
fprintf('
end
```

Outcome of Program:

cubic$_spline_retry$

$Enternfor(n+1)nodes, n:$ 3

$Enterx(0)andf(x(0))onseparatelines:$

0

1

$Enterx(1)andf(x(1))onseparatelines:$

1

2.72

$Enterx(2)andf(x(2))onseparatelines:$

2

7.39

$Enterx(3)andf(x(3))onseparatelines:$

3

20.09

$Thenumbersx(0), ..., x(n)are:$

0.00001.00002.00003.0000

$The\ coefficients\ of\ the\ spline\ on\ the\ subintervals\ are:$

$a(i)$	$b(i)$	$c(i)$	$d(i)$
1.00000000	1.46866667	0.00000000	0.25133333
2.72000000	2.22266667	0.75400000	1.69333333
7.39000000	8.81066667	5.83400000	-1.94466667

Here is another code that is used:
clear all
close all

ff=@(x) log(x+3);
x1=linspace(0,10,100); y1=ff(x1);
hold on plot(x1,y1,'r*');
w=11.25-3;
p=lagrange$_i nterp(x1,y1,w);$
$s = spline(x1,y1,w);$
$fprintf('UsingLagrangeinterpolationvalueoflog(11.25) =$
$fprintf('ErrorinLagrangeinterpolation = fprintf('UsingSplineinterpolationvalueoflog(11.25) = fprin$
for i=1:n
s1=1;
s2=1;
for j=1:n
if i =j
s1=(xx-xi(j))*s1;
s2=(xi(i)-xi(j))*s2;
end
end
z(i)=(s1./s2)*yi(i);
end
f(xx)=sum(z);
for i=1:length(x)
y(i)=double(f(x(i)));
end
end

Outcome of the Code example:

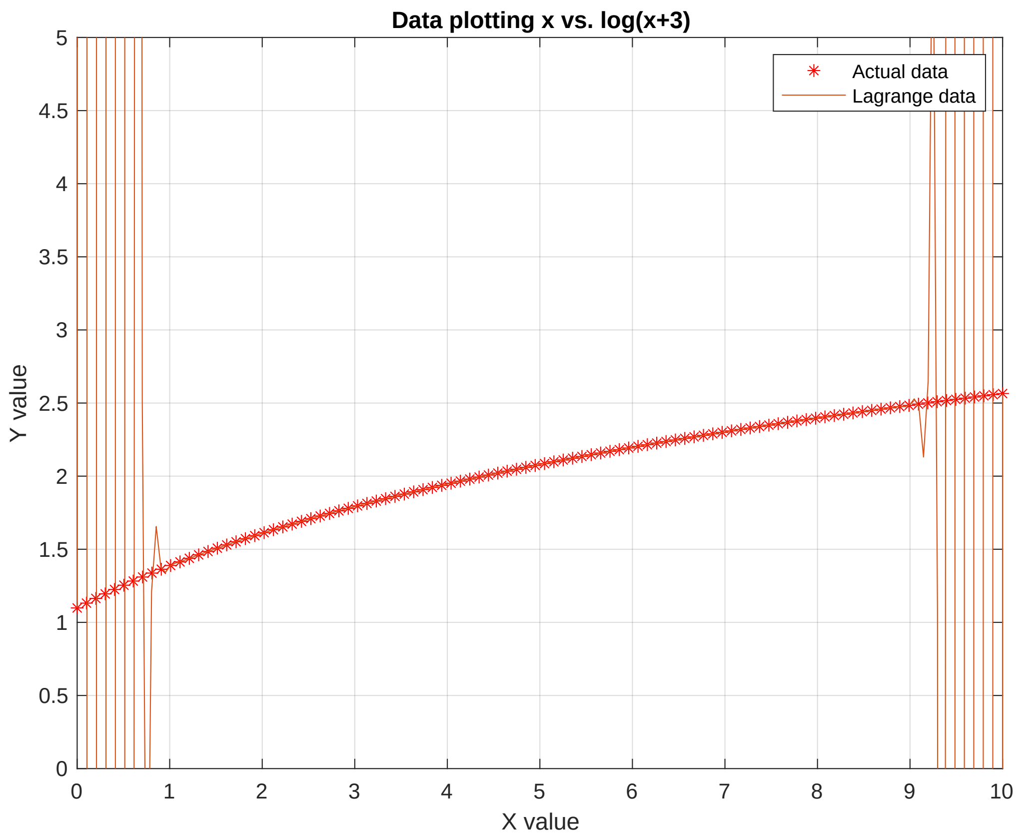

Figure 2: Lagrange Graph of example

Now lets put the given formulas to work!

We are given f(x) = cos(8πx) on intervals of [0,1]

We will start with f(10^{-5}) by using Lagrange interpolation polynomial...
The number of nodes we are going to have is three.
So we plug in three nodes...

It will execute something like this!

$L_0(x) = \frac{(x-10^{-4})(x-10^{-5})}{(10^{-6}-10^{-4})(10^{-6}-10^{-3})} \longrightarrow \frac{10^{12}(x-10^{-4})(x-10^{-5})}{(-10+1)(1-1000)} \longrightarrow \frac{10^{12}(x-10^{-4})(x-10^{-3})}{999*-199}$

$L_1(x) = \frac{(x-10^{-6})(x-10^{-5})}{(10^{-4}-10^{-6})(10^{-4}-10^{-5})} \longrightarrow \frac{10^{8}(x-10^{-6})(x-10^{-3})}{(-10+1)(1-1000)} \longrightarrow 1*10^7*(x-10^{-6})(x-10^{-4})$

$L_2(x) = \frac{(x-10^{-6})(x-10^{-4})}{(10^{-5}-10^{-6})(10^{-5}-10^{-4})} \longrightarrow \frac{10^{6}(x-10^{-6})(x-10^{-4})}{(-10+1)(1-1000)} \longrightarrow 1.11*10^6*(x-10^{-6})(x-10^{-4})$

Now its program will plug its values in its give function to plots is graphs

$Given f(x) = cos(8\pi x)$
$f_0(10^{-6}) = cos(8\pi*10^{-6}) \longrightarrow 0.99$
$f_1(10^{-4}) = cos(8\pi*10^{-4}) \longrightarrow 0.99$
$f_2(10^{-3}) = cos(8\pi*10^{-4}) \longrightarrow 0.99$

Now use sigma in order to find its approximated value

$f(x) = \sum_{i=1}^{2} f(x)_i L_i(x) \longrightarrow f(x_0)L_0(x)+f(x_1)L_1(x)+f(x_2)L_2(x) \longrightarrow by sub. from its original values we end up$
$0.99((1.01*10^9(x-1o^{-4})(x-10^{-3})+1*10^7*(x-10^{-6})(x-10^{-4})+1.11*10^6*(x-10^{-6})(x-10^{-4})) \longrightarrow 0.99(1.11*10^6*10^{-6}*9*10^{-5}) \longrightarrow 0.9999$
So the approximated value of 10^{-5} of f(x) = cos(8 π x) is 0.9999

Here is the graph of the outcome

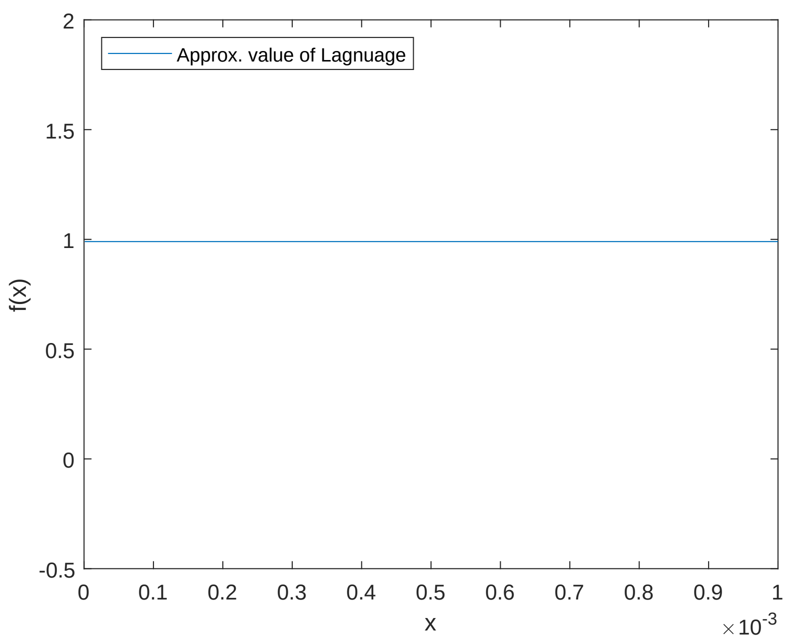

Figure 3: Lagrange Graph of $\cos(8\pi x)$

We are also given f(x) = $\sqrt{x-x^2}$ with intervals of 0 to 1.

Starting off with Lagrange interpolation polynomial
It will execute something like this!
$L_0(x) = \frac{(x-10^{-6})(x-10^{-3})}{(10^{-6}-10^{-4})(10^{-6}-10^{-3})} \longrightarrow \frac{10^{12}(x-10^{-4})(x-10^{-3})}{(-10+1)(1-100)} \longrightarrow \frac{10^{12}(x-10^{-4})(x-10^{-3})}{99*999} \longrightarrow 1.01*10^9*(x-10^{-4})(x-10^{-3})$

$L_1(x) = \frac{(x-10^{-6})(x-10^{-3})}{(10^{-4}-10^{-6})(10^{-4}-10^{-3})} \longrightarrow \frac{10^{8}(x-10^{-6})(x-10^{-3})}{(-10+1)(1-100)} \longrightarrow 1*10^7*(x-10^{-6})(x-10^{-4})$

$L_2(x) = \frac{(x-10^{-6})(x-10^{-4})}{(10^{-3}-10^{-6})(10^{-3}-10^{-4})} \longrightarrow \frac{10^{6}(x-10^{-6})(x-10^{-4})}{(-10+1)(1-1000)} \longrightarrow 1.11*10^6*(x-10^{-6})(x-10^{-4})$

Plug the values in we are given
f(x) = $\sqrt{x-x^2} \longrightarrow$
$f(10^{-6}) = \sqrt{10^{-6}-10^{-6*2}} \rightarrow 9.9*10^{-4}$
$f(10^{-4}) = \sqrt{10^{-4}-10^{-4*2}} \rightarrow 9.99*10^{-4}$

Now use sigma in order to find its approximated value
$f(x) = \sum_{i=1}^{2} f(x)_i L_i(x) \longrightarrow f(x_0)L_0(x)+f(x_1)L_1(x)+f(x_2)L_2(x) \longrightarrow bysub.fromitsoriginalvaluesweendup$
$0.99((1.01*10^9*(x-10^{-4})(x-10^{-3})(1*10^7*(x-10^{-6})(x-10^{-4}))(1.11*10^6*(x-10^{-6})(x-10^{-4}))) \longrightarrow 9.99*10^{-4}$
So the approximated value of 10^{-5} of f(x) = $\sqrt{x-x^2}$ is 9.99 * 10^{-4}

Here is the graph of the outcome

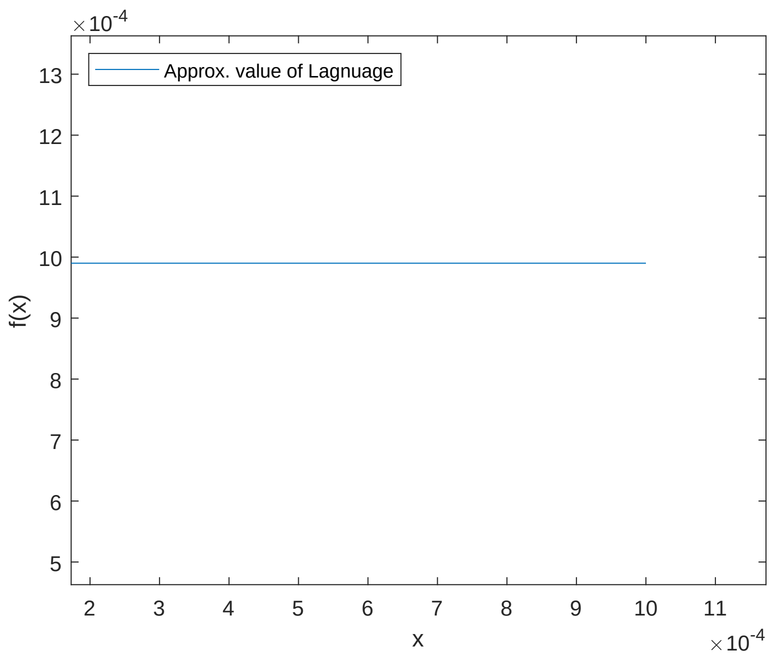

Figure 4: Lagrange Graph of $\sqrt{x - x^2}$

If you tried more nodes, it will keep continuously reach at its approximated value which is 9.99 * 10^{-4}. As you can tell in the graph, it basically is y = 9.99. Very similar except its x values go on depending on its nodes. if it was 20 nodes, it would end up at 10^{-19}.

So comparing with error's...

Outcome for given f(x) = cos(8πx):
test
Using Lagrange interpolation value of cos(8*pi*(10^-5)) = -2492295.436521
$ErrorinLagrangeinterpolation = 2.492296e + 06$
$UsingSplineinterpolationvalueofcos(8 * pi * (10^-5)) = 0.734136$
$ErrorinSplineinterpolation = 2.658642e - 01$

$Graph$:

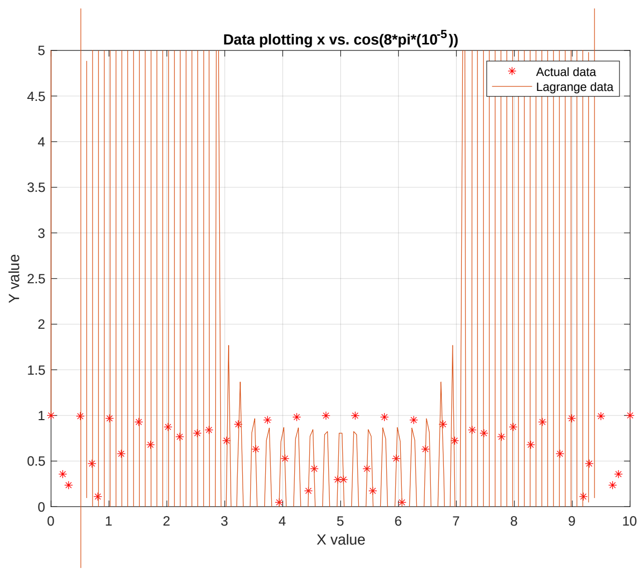

Figure 6: Lagrange Graph of cos(8πx)

Outcome for given f(x) = $\sqrt{x - x^2}$: