Homework #6

Math 131-05D-Spring 2017

3/14/17 at 1:30pm

By Yeash Patel

Problem 1.

1. Write a MATLAB function, called Lagrange_poly that inputs a set of data points (x;y)=(datx,daty), a set x of numbers at which to interpolate, and outputs the polynomial interpolant, y, evaluated at x using Lagrange polynomial interpolation. Your function header should look something like:

function y = Lagrange_poly(x,datx,daty).

As mentioned in the problem the function takes the input x, datx, and, daty. All of these inputs are expected to be an array of numbers either integers, fractions, or floats. The function returns an array containing the interpolated values at x. to display how to use this function lets interpolate y=x^2 we have datx=[-2,-1,0,1,2], daty=[4,1,0,1,4], and x=[-4,-3,3,4] we shall have the output stored in y.

Please see appendix A problem one for the code or refer to the lagrange_poly.m file in folder homework six code.

Matlab intput and output

```
>> datx=[-2,-1,0,1,2]

datx =

   -2    -1     0     1     2

>> daty=[4,1,0,1,4]

daty =

    4     1     0     1     4

>> x=[-4,-3,3,4]

x =

   -4    -3     3     4

>> y=lagrange_poly (x, datx,daty)

y =

   16     9     9    16
```

Problem 2.

2. Use the code you developed in Problem 1 to interpolate the functions

(a) f(x)= e^{-1^2}

(b) f(x)= $\frac{1}{(1+x^2)}$

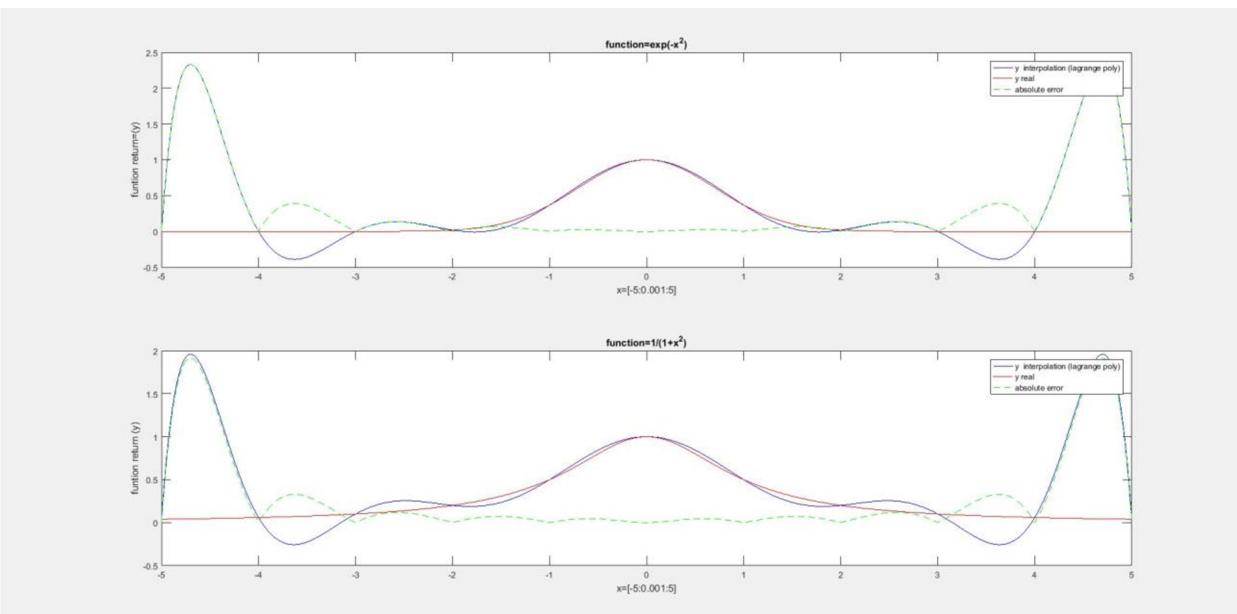
Using the data points datx=-5:1:5. Interpolate at the points x=-5:0.001:5. Plot the results and comment on the error.

> We are given a total of 11 reference points to interpolate 10001 points using a lagrangian interpolation method. For convenience I created a script, named pr2.m that calls on the function created in problem one with the appropriate inputs provided by the problem and creates the graphs necessary. The graphs display the real function values in red and the interpolated function values in blue. On both of the graph at about range $-1 \leq x \leq 1$ the error seems to be fairly small as one can see from the absolute error function plotted in green. However as we get father away from those bounds we see the interpolated function oscillate and the error increasing this though is to be expected since we are moving further out form our center.

Matlab intput

>> pr2

Matlab output



Problem 3.

3. Write a MATLAB function, called Newton_poly that inputs a set of data points (x;y)=(datx,daty), a set x of numbers at which to interpolate, and outputs the polynomial interpolant, y, evaluated at x using Newton polynomial interpolation. Your function header should look something like:

function y = Newton_poly(x,datx,daty)

As mentioned in the problem the function takes the input x, datx, and, daty. All of these inputs are expected to be an array of numbers either integers, fractions, or floats. The function returns an array containing the interpolated values at x. to display how to use this function lets interpolate y= $1+\frac{4}{3}x+\frac{2}{3}x^2$ we have datx=[-1,0,1], daty=[1/3,1,3], and x=[1,2,3,4] we shall have the output stored in y.

Please see appendix A problem three for the code or refer to the Newton_poly.m file in folder homework six code.

Matlab intput and output

```
>> datx=[-1,0,1]

datx =

    -1     0     1

>> daty=[1/3,1,3]

daty =

    0.3333    1.0000    3.0000

>> x=[1,2,3,4]

x =

     1     2     3     4

>> y= Newton_poly(x,datx,daty)

y =

    3.0000    6.3333   11.0000   17.0000
```

Problem 4.

4. Use the code you developed in Problem 3 to interpolate the function

$$f(x)=\frac{1}{(1+x^2)}$$

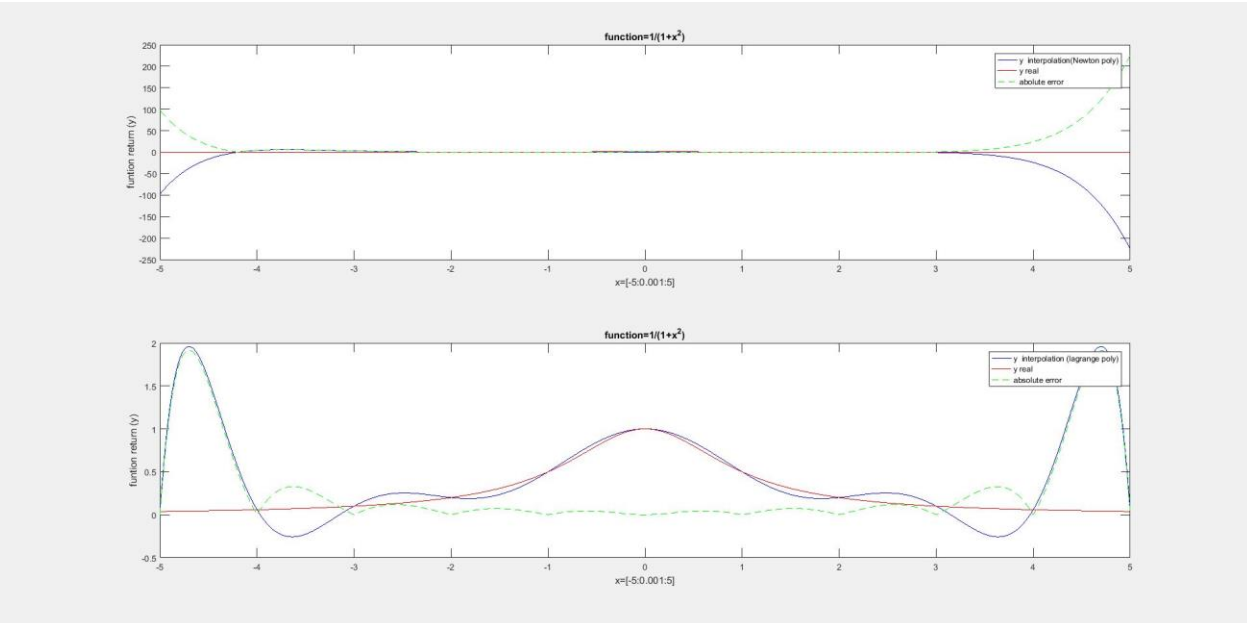
using the data points datx=-5:1:5. Interpolate at the points x=-5:0.001:5. Plot the results and compare them to what you got in problem 2(b). Explain why you get what you get.

> We find that Newton's polynomials interpolations gives us a very accurate results in the approximate range of $-2 \leq x \leq 3$. We can see the absolute error plotted as the green dashed line. This is most likely due to the recursive nature of this algorithm.

Matlab intput

>> **pr4**

Matlab output



APPENDIX A

PROBLEM 1

lagrange_poly.m:

```
function [w] = lagrange_poly (x, datx,daty)

n=length(datx);
q=length(x);
l=ones(1,n);
w=zeros(1,q);

for z=1:q
t=z;
l=ones(1,n);
y=0;
  for i=1:n
  k=i;
   for j=1:n
         if (j==k)
       continue;
         else
       l(k)=l(k)*((x(t)-datx(j))/(datx(k)-datx(j)));
       end
     end
   y=y+daty(k)*l(k);
   w(z)=y;
  end
end
end
```

PROBLEM 2

pr2.m

```
clc
clear all
close all
datx=[-5:1:5];
n=(length(datx));
x=[-5:0.001:5];
u=length(x);
for l=1:n
fun1daty(l)=(exp(-datx(l).^2));
end
y1i=lagrange_poly(x,datx,fun1daty);
for z=1:u
y1(z)=(exp(-x(z).^2));
end
for q=1:u
    abserror1(q)=abs(y1(q)-y1i(q));
end
subplot(2,1,1)
plot(x,y1i,'-b',x,y1,'-r',x,abserror1,'g--')
```

```
legend('y  interpolation (lagrange poly)','y real','absolute error')
xlabel('x=[-5:0.001:5]')
ylabel('funtion return=(y)')
title('function=exp(-x^2)')
for l=1:n
fun2daty(l)=(1./(1+datx(l).^2));
end
y2i=lagrange_poly(x,datx,fun2daty);
for z=1:u
y2(z)=(1./(1+x(z).^2));
end
for q=1:u
    abserror2(q)=abs(y2(q)-y2i(q));
end
subplot(2,1,2)
plot(x,y2i,'-b',x,y2,'-r',x,abserror2,'g--')
legend('y  interpolation (lagrange poly)','y real','absolute error')
xlabel('x=[-5:0.001:5]')
ylabel('funtion return (y)')
title('function=1/(1+x^2)')
```

PROBLEM 3

Newton_poly.m

```
function [y] = Newton_poly (x, datx,daty)

n=length(daty);
u=length(x);
k=1;
F=zeros(n);
y=zeros([1,u]);
p=zeros([1,u]);
g=ones([u,n]);

while(k<=n)
  F(k,1)=daty(k);
  k = k + 1;
end

for i=2:n
  for j=2:i
      if i==j
        F(i,j)=(F(i,j-1)-F(i-1,j-1))/(datx(i)-datx(1));
      else
       F(i,j)=(F(i,j-1)-F(i-1,j-1))/(datx(i)-datx(i-j+1));
       end
  end
end

for q=1:n
  a(q)=F(q,q);
end

for l=1:u
```

```
  for d=2:n
    if d==2
      g(l,d)=x(l)-datx(d-1);
    else
    g(l,d)=g(l,d-1)*x(l)-datx(d-1);
    end
  end
end

for r=1:u
  for c=1:n
   y(r)=g(r,c)*a(c)+y(r);
  end
end

end
```

PROBLEM 4

pr4.m

```
clc
clear all
close all
datx=[-5:1:5];
n=(length(datx));
x=[-5:0.001:5];
u=length(x);
for l=1:n
fun1daty(l)=(1./(1+datx(l).^2));
end
y1i=Newton_poly(x,datx,fun1daty);
for z=1:u
y1(z)=(1./(1+x(z).^2));
end
for q=1:u
    abserror1(q)=abs(y1(q)-y1i(q));
end
subplot(2,1,1)
plot(x,y1i,'-b',x,y1,'-r',x,abserror1,'--g')
legend('y  interpolation(Newton poly)','y real', 'abolute error')
xlabel('x=[-5:0.001:5]')
ylabel('funtion return (y)')
title('function=1/(1+x^2)')
for l=1:n
fun2daty(l)=(1./(1+datx(l).^2));
end
y2i=lagrange_poly(x,datx,fun2daty);
for z=1:u
y2(z)=(1./(1+x(z).^2));
end
for q=1:u
    abserror2(q)=abs(y2(q)-y2i(q));
end
subplot(2,1,2)
plot(x,y2i,'-b',x,y2,'-r',x,abserror2,'g--')
```

```
legend('y  interpolation (lagrange poly)','y real','absolute error')
xlabel('x=[-5:0.001:5]')
ylabel('funtion return (y)')
title('function=1/(1+x^2)')
```