```
%
 ========================================================================
% AUTHOR ..... [Lishan Huang]
% UPDATED .... [Jan 23]
% Task 1
```

lagrange interpolation formula

lagrange.m

```
%
 ========================================================================
% AUTHOR ..... [Lishan Huang]
% UPDATED .... [Jan 23]
%
% Evaluate the Lagrange interpolation formula
%
% INPUT
% tau .... The vector of interpolation nodes (length n)
% rho .... The vector of values at the interpolation nodes (length n)
% x   .... A vector of values to evaluate the interpolating polynomial
 at (length 1 to many (probably not n!))
%
% OUTPUT
% T :
%
 ========================================================================
function F = lagrange(tau, rho, x)
%initialize F=0
t=tau;
p=rho;
F=0;
%create a for loop
    for k=1:length(t)
        %make L=1 before each loop where l means Lk in the function
        L=1;
        for i=1:length(t)
            if i~=k
                L=L.*(x-t(i))./(t(k)-t(i));
            end
        end
        F=F+p(k).*L;
    end
end
```

The first form of the barycentric formula

firstbaryeval.m

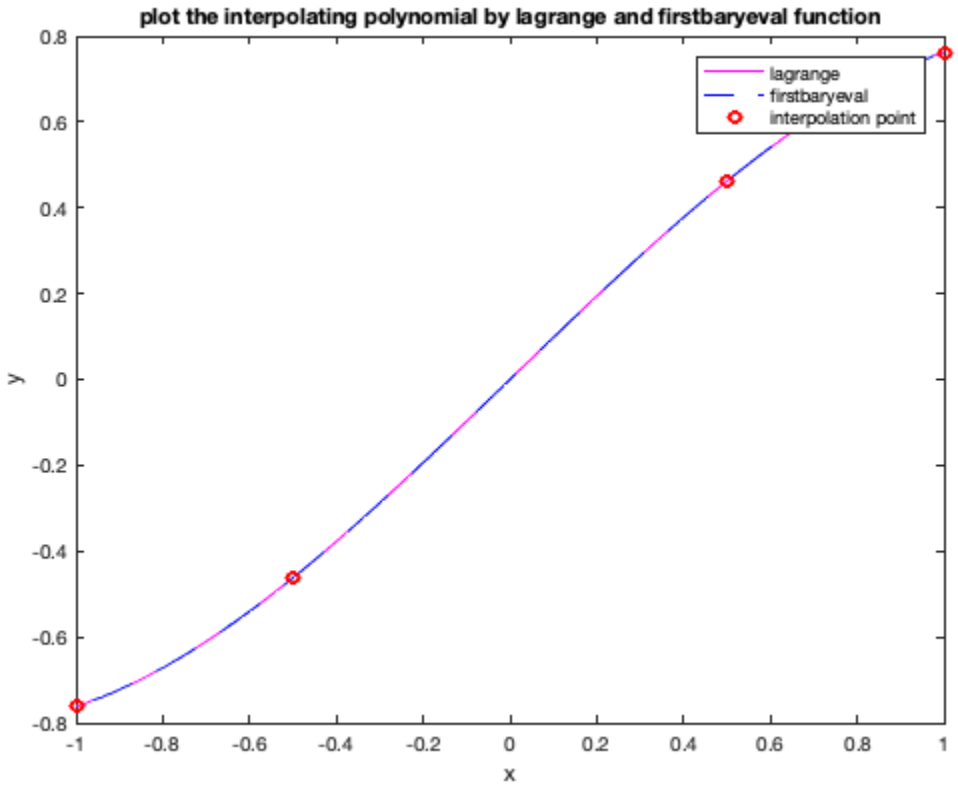
```
%
 ========================================================================
% AUTHOR ..... [Lishan Huang]
% UPDATED .... [Jan 23]
%
%  Evaluate the first form of the barycentric formula
%
% INPUT
% tau .... The vector of interpolation nodes (length n)
% rho .... The vector of values at the interpolation nodes (length n)
% x   .... A vector of values to evaluate the interpolating polynomial
 at (length 1 to many (probably not n!))
%
% OUTPUT
% T :
%
 ========================================================================

function T = firstbaryeval(tau, rho, x)
%initialize t p T and a
    t=tau;
    p=rho;
    T=0;
    a=1;
    %calculate w(x) and store in a
        for k=1:length(t)
            a=a.*(x-t(k));
        end
    %create for loop to add up the function
        for k=1:length(t)
            b=1;
            %for loop for b(k)
            for j=1:length(t)
               if j~=k
                   b=b.*((t(k)-t(j))).^-1;
               end
            end
            T=T+b.*p(k)./(x-t(k));
        end
    T=T.*a;
end
```

Script

```
%The following is a code that use lagrange and firstbaryeval function
 to
%plot the interpolating polynomial
%
```

```
%
%define tau and rho
tau=[-1, -1/2, 1/2, 1];
rho=[-0.7616, -0.4621, 0.4621, 0.7616];
%create 100 equal point between -1 and 1
x=linspace(-1,1,100);
%use lagrange interpolation formula to evalue x
y=lagrange(tau,rho,x);
%use the first form of the barycentric formula to evalue x
z=firstbaryeval(tau,rho,x);
%plot the x y z
plot(x,y,'m',x,z,'--b')
%keep in the same graph
hold on
%plot tau rho
plot(tau,rho,'or')
xlabel('x')
ylabel('y')
title('plot the interpolating polynomial by lagrange and firstbaryeval
 function')
legend('lagrange','firstbaryeval','interpolation point')
```



Published with MATLAB® R2017b