```
%
 ========================================================================
% AUTHOR ..... [Lishan Huang]
% UPDATED .... [Jan 23]
% Task 2
```

lagrange interpolation formula

slagrange.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

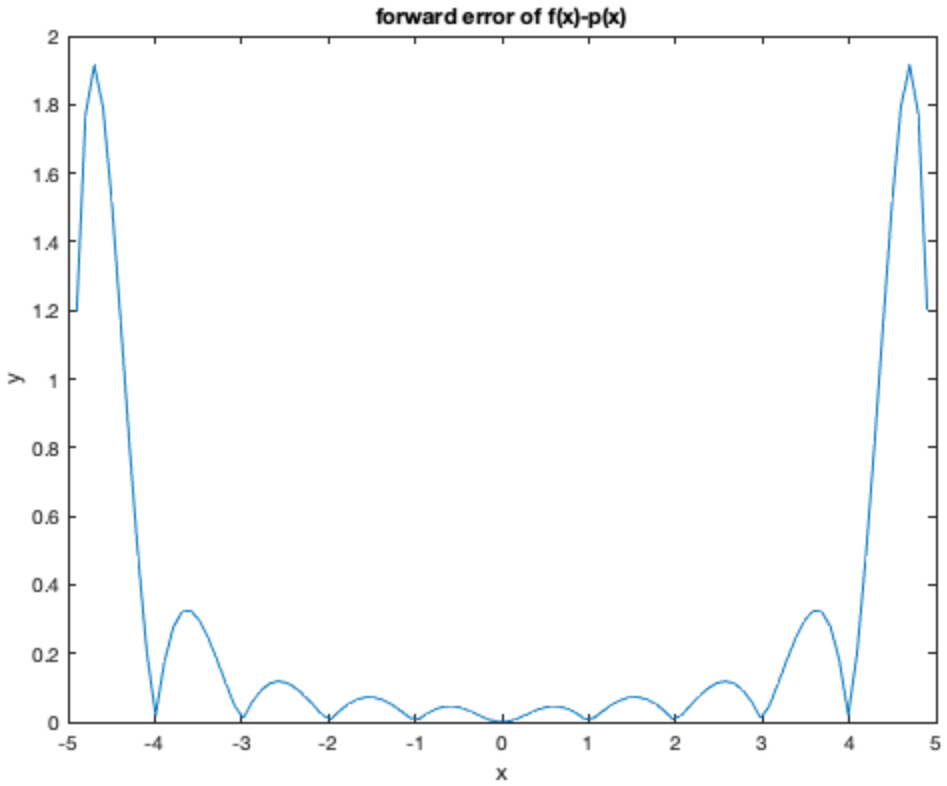
```
%
 ============================================================================
% AUTHOR ..... [Lishan Huang]
% UPDATED .... [Jan 23]
%
%  Evaluate the first form of the barycentric formula
%
% INPUT
% tau .... The vector of interpolation nodes (length n)
% rho .... The vector of values at the interpolation 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 using firstbaryeval function to
 interpolate
%the function f(x)=1/(1+x^2) and plot the forward error between f(x)
 and p(z)
```

```
%
%create a vector with 11 equal point between -5 and 5
tau=linspace(-5,5,11);
%
%calculate the value of rho by given function
rho=1./(1+tau.^2);
%create a vector with 100 equal point between -5 and 5
z=linspace(-5,5,100);
%calculate f(x) of vector z by given function and store in k
k=1./(1+z.^2);
%definition of first barycentric  form interpolation
t=firstbaryeval(tau,rho,z);
%plot result of the interpolation poin
plot(z,abs(k-t))
xlabel('x')
ylabel('y')
title('forward error of f(x)-p(x)')
```




```
% *Text Answers*
%No, it is not a good interpolant, because when the absolute value of
 x is away from 0, the error between adjacent interpolation points
 will increase, then it falses to  predit the values between adjacent
 interpolation points. In especially, when the absolute value of x
 is greater than 4, the errors of the interpolant begins to skyrocket
 thus it also can not predict the output for x that is out of the
 range of interpolation point
```

Published with MATLAB® R2017b