Largrange poly_interpolation.m

```
function pn=Lagrange_poly_interpolation(xi,yi,x)
num=length(xi);%number of sample points
pn=0;
for i=1:num
    li=1;
    for j=1:num
        if j~=i
            li=li.*(x-xi(j))/(xi(i)-xi(j));
        end
    end
    pn=pn+yi(i)*li;
end
```

Naïve poly_interpolation.m

```
function pn=Naive_poly_interpolation(xi,yi,x)
n=length(xi)-1;
A=vander(xi);
b=yi';
a=A\b;
pn=0;
for i=1:n+1
    pn=pn+a(i)*x.^(n-i+1);
end
```

Poly_interpolation for f = cosx

```
a=[5 10 20 40];
for j=1:length(a)
n=a(j);
h=2/n;
xi=-1:h:1;
yi=cos(xi);
plot(xi,yi,'ro')
hold on
x=-1:0.01:1;
pn=Naive_poly_interpolation(xi,yi,x);
plot(x,pn,'k','linewidth',3)
norm(pn-cos(x))
pn=Lagrange_poly_interpolation(xi,yi,x);
plot(x,pn,'y','linewidth',3)
norm(pn-cos(x))
end
```

Poly_interpolation for f =1/(1+x^2)

```
a=[5 10 20 40];
for j=1:length(a)
n=a(j);
h=2/n;
xi=-1:h:1;
yi=(1./(1+xi.^2));
```

```
plot(xi,yi,'ro')
hold on
x=-1:0.01:1;
pn=Naive_poly_interpolation(xi,yi,x);
plot(x,pn,'k','linewidth',3)
norm(pn-(1./(1+x.^2)))
pn=Lagrange_poly_interpolation(xi,yi,x);
plot(x,pn,'y','linewidth',3)
norm(pn-(1./(1+x.^2)))
end
```