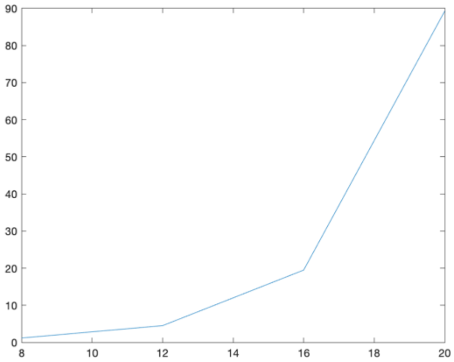
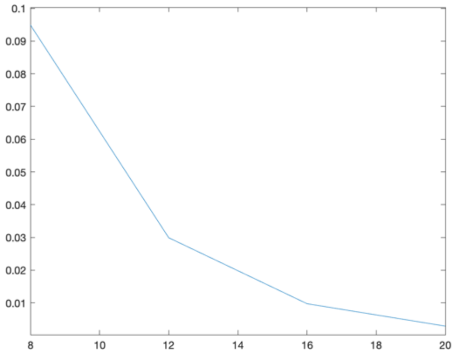
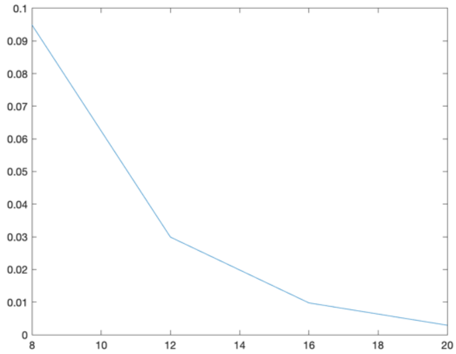
P1.1 M1 error v.s. N



P1.2 M2 error v.s. N

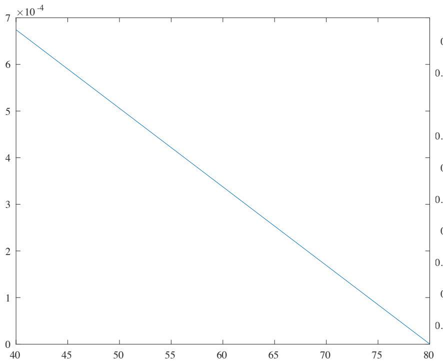


P1.3 M3 error v.s. N

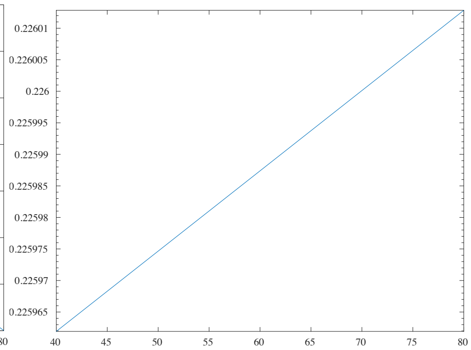


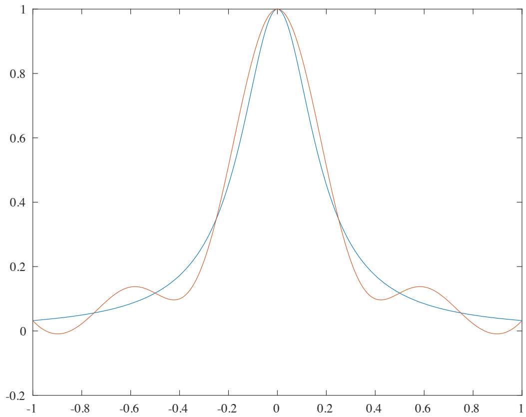
M1.p1 With function f(x) = 1/(1+30*x^2), on the interval [-1,1], as N increases, using equidistance points and weights(P1.1) leads to increase of error, which is caused by error in region close to endpoints -1 and 1. However, if we use Chebyshev weights(M2 and M3), no matter we use equidistance points or Chebyshev points, the error gets smaller, with exponential decay.

P2.1 M2 error v.s. N



P2.2 M3 error v.s. N

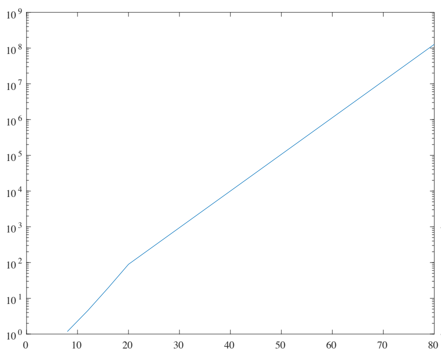




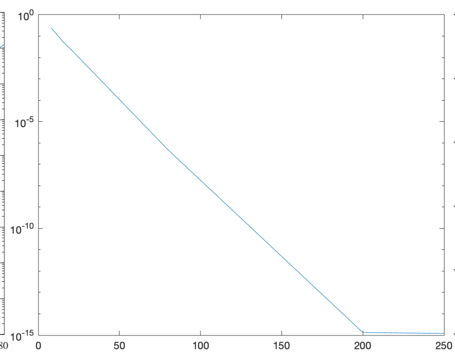
M3 N = 8 f and interpolation graph

M1.p2 With same function but smaller interval [-1/2, 1/2] and higher order polynomials N = 40,80. If we use equidistance points and weights, error gets to infinity rapidly, so it cannot be plotted, but it drops to 2.142170326963773 at N = 80. If we use Chebyshev points and weights, the error decreases as N increase, and smallest error reaches 4.74538377592815e-07 when N = 80. However, if we use equidistance points with Chebyshev weights, the error increases, and smallest error 0.225961928599247 is at N = 40.

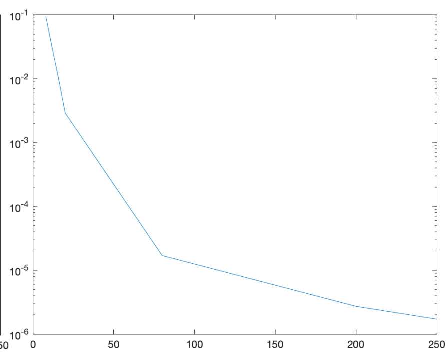
P3.1 M1 error v.s. N



P3.2 M2 error v.s. N



P3.3 M3 error v.s. N



M2 With same function f(x) = 1/(1+30*x^2) and interval [-1,1], but higher degrees of N = [8,1620,80,200,250], using equidistance points and weights(P3.1) causes larger error with exponential growth, and the error even achieves infinity at N = 200 and 250 which cannot be plotted further. Obviously, we cannot gain accuracy by increasing N from 200 to 250, and the lowest error = 1.17306813475113 is achieved at N = 8 in this method.

If we use Chebyshev weights(P3.2 and P3.3), the error gets smaller as well, with exponential decay. In both methods, we can get better accuracy by increasing N from 200 to 250. By using Chebyshev points and weights(P3.2), we can get smallest error with fastest rate of decay among these three methods. The lowest error = 1.221245327087672e-15 is achieved at N = 250. However, if we use equidistance points with Chebyshev weights(P3.3), lowest error 1.712350589655465e-06 is achieved at N = 250.

```matlab
f = @(x) 1./(1 + 30*x.^2 );
N = [8, 12, 16, 20, 80, 200, 250];
error = [];
for i = 1:length(N)
    error = [error barycentric(f,-1,1,N(i))];
end
semilogy(N,error);

function [maxError] = barycentric(f,a,b,N)
    syms x m;
    xvalue = [];
    % x with equidistance points
    xvalue = linspace(a,b,N+1);

    % Chebysheve weights
    for k = 1:N+1
        weight(k) = (-1)^(k-1);
        % x with Chebysheve points
        %xvalue = [xvalue cos((k-1)*pi/N)];
    end

    weight(1) = 1/2;
    weight(N+1) = ((-1)^N)/2;
    xdata = f(xvalue);
    % equidistance weights
    %weight = @(k) (-1)^(k-1)*nchoosek(N,k-1);

    pa = 0;
    pb = 0;
    % calculate nominator
    for m = 1:N+1
        pa = pa + weight(m)*xdata(m)/(x-xvalue(m));
    end
    % calculate denominator
    for m = 1:N+1
        pb = pb + (weight(m)/(x-xvalue(m)));
    end
    p = pa/pb;
    p = matlabFunction(p);
    t = linspace(-1,1,321);
    fex = f(t);
    interpolation = p(t);
    for i = 1:321
        if ismember(t(i),xvalue)
            interpolation(i) = f(t(i));
        end
    end
    figure
    plot(t,fex,t,interpolation);
    maxError = max(abs(interpolation-fex));
end
```