11/6/17

Alejandro Stawsky

Math 104A HW4

1. (a) $l_0(x) = \frac{x-1 \; x-3}{-1 \cdot -3}$ $\quad f_0 = 1$

$l_1(x) = \frac{x \; x-3}{1 \cdot -2}$ $\quad f_1 = 1$

$l_2(x) = \frac{x \cdot x-1}{3 \cdot 2}$ $\quad f_3 = -5$

$p_2(x) = \frac{x^2-4x+3}{3} + \frac{\cancel{(-x^2)}+3x}{+2} + \frac{x^2-x}{6}(-5)$

$= \frac{2x^2-8x+6-3x^2+9x-5x^2+5x}{6}$

$= \frac{-6x^2+6x+6}{6} = -x^2+x+1$

(b) $p_2(2) = -4+2+1 = -1 \approx f(2)$

```
%2.a

% Look at the bottom

%2.b

K=[];
L=[];
M=[];

for j=0:4
    % appending the equispaced nodes for n=4
    K=[K, -1+j*(1/2)];
end

for j=0:10
    % appending the equispaced nodes for n=10
    L=[L, -1+j*(1/5)];
end

for j=0:20
    % appending the equispaced nodes for n=20
    M=[M, -1+j*(1/10)];
end

x = linspace(-1,1);

figure(1)
% The Lebesgue Constant is about 2.2
plot(x,Leb(K,x))
title('Graph of Lebesgue Function n=4')

figure(2)
% The Lebesgue Constant is about 30
plot(x,Leb(L,x))
title('Graph of Lebesgue Function n=10')

figure(3)
% The Lebesgue Constant is about 11,000
plot(x,Leb(M,x))
title('Graph of Lebesgue Function n=20')

%2.c

K1=[];
L1=[];
M1=[];

for j=0:4
    % appending the Chebyshev nodes for n=4
    K1=[K1, cos((j*pi)/4)];
end
```

```
for j=0:10
    % appending the Chebyshev nodes for n=10
    L1=[L1, cos((j*pi)/10)];
end

for j=0:20
    % appending the Chebyshev nodes for n=20
    M1=[M1, cos((j*pi)/20)];
end

x = linspace(-1,1);

figure(4)
% Lebesgue constant is about 1.8
plot(x,Leb(K1,x))
title('Graph of Lebesgue Function n=4 (Chebyshev)')

figure(5)
% Lebesgue constant is about 2.4
plot(x,Leb(L1,x))
title('Graph of Lebesgue Function n=10 (Chebyshev)')

figure(6)
% Lebesgue constant is about 2.6
plot(x,Leb(M1,x))
title('Graph of Lebesgue Function n=20 (Chebyshev)')

% The Lebesgue Constant for the
  % Chebyshev nodes are much smaller
    % than for the equispaced nodes

%3.a


% the Barycentric formula and

% Barycentric weights functions are below


%3.b


A = [0 1/4 .52 .74 1.28 1.5];
    % the set of nodes
C = [0 .7070 1 .7071 -.7074 -1];
    % the set of function values at corresponding nodes

% get the Barycentric Weights
B1 = Baryweights(A,0,0);
B2 = Baryweights(A,1,0);

% Use weights to compute Barycentric Formula
P2 = BaryFormula(A,B1,C,2)
```

```
%4.a

% create set of Equispaced Nodes
  % for n= 4, 8 and 12 respectively

EN4 = K; % Re-use K from 2.b
EN8 = [];
for j=0:8
    % appending the equispaced nodes for n=8
    EN8=[EN8, -1+j*(1/4)];
end
EN12 = [];
for j=0:12
    % appending the equispaced nodes for n=12
    EN12=[EN12, -1+j*(1/6)];
end
EN100 = [];
for j=0:100
    % appending the equispaced nodes for n=12
    EN100=[EN100, -1+j*(1/6)];
end

% Create the set of Bary weights for their
  % respective n's

BW4=Baryweights(EN4,0,0);
BW8=Baryweights(EN8,0,0);
BW12=Baryweights(EN12,0,0);

% create set of function values at equispaced
  % nodes for n = 4, 8 and 12 respectively

FV4=[];
for j=1:length(EN4)
    FV4=[FV4, Runge(EN4(j))];
end
FV8=[];
for j=1:length(EN8)
    FV8=[FV8, Runge(EN8(j))];
end
FV12=[];
for j=1:length(EN12)
    FV12=[FV12, Runge(EN12(j))];
end
FV100=[];
for j=1:length(EN100)
    FV100=[FV100, Runge(EN100(j))];
end
```

```
% Graph the Bary Formulas
x = -1 +2.*rand(1,1000);

% create the values of the BaryFormula Approx.
BF4=[];
BF8=[];
BF12=[];

for i=1:length(x)
    BF4 = [BF4, BaryFormula(EN4,BW4,FV4,x(i))];
    BF8 = [BF8, BaryFormula(EN8,BW8,FV8,x(i))];
    BF12 = [BF12, BaryFormula(EN12,BW12,FV12,x(i))];
end

% Plot the function and all approximations

figure(7)
scatter(x,BF4,5,'r','filled')
hold on
scatter(x,BF8,5,'b','filled')
hold on
scatter(x,BF12,5,'black','filled')
hold on
x1=linspace(-1,1);
h=plot(x1,Runge(x1),'green');
h(1).LineWidth = 2;
title('Graph of f(x) and the langrange interpolating polys (4.a)')
hold off

% 4.b

% create set of Chebyshev Nodes
  % for n= 4, 8, 12, and 100 respectively

CN4 = [];
for j=0:4
    % appending the Chebyshev nodes for n=4
    CN4=[CN4, cos((j*pi)/4)];
end
CN8 = [];
for j=0:8
    % appending the Chebyshev nodes for n=8
    CN8=[CN8, cos((j*pi)/8)];
end
CN12 = [];
for j=0:12
    % appending the Chebyshev nodes for n=12
    CN12=[CN12, cos((j*pi)/12)];
end
CN100 = [];
for j=0:100
    % appending the Chebyshev nodes for n=100
    CN100=[CN100, cos((j*pi)/100)];
end
```

```matlab
% Create the set of Bary weights for their
  % respective n's

BCW4=Baryweights(CN4,0,1);
BCW8=Baryweights(CN8,0,1);
BCW12=Baryweights(CN12,0,1);
BCW100=Baryweights(CN100,0,1);

% create set of function values at equispaced
  % nodes for n= 4, 8 and 12 respectively

FCV4=[];
for j=1:length(CN4)
    FCV4=[FCV4, Runge(CN4(j))];
end
FCV8=[];
for j=1:length(CN8)
    FCV8=[FCV8, Runge(CN8(j))];
end
FCV12=[];
for j=1:length(CN12)
    FCV12=[FCV12, Runge(CN12(j))];
end
FCV100=[];
for j=1:length(CN100)
    FCV100=[FCV100, Runge(CN100(j))];
end

% Graph the Bary Formulas
x = -1 +2.*rand(1,1000);

% create the values of the BaryFormula Approx.
BCF4=[];
BCF8=[];
BCF12=[];
BCF100=[];

for i=1:length(x)
    BCF4 = [BCF4, BaryFormula(CN4,BCW4,FCV4,x(i))];
    BCF8 = [BCF8, BaryFormula(CN8,BCW8,FCV8,x(i))];
    BCF12 = [BCF12, BaryFormula(CN12,BCW12,FCV12,x(i))];
    BCF100 = [BCF100, BaryFormula(CN100,BCW100,FCV100,x(i))];
end

% plot the function and all approximations

figure(8)
scatter(x,BCF4,3,'r','filled')
hold on
scatter(x,BCF8,3,'b','filled')
hold on
scatter(x,BCF12,3,'black','filled')
hold on
```

```
scatter(x,BCF100,3,'magenta','filled')
hold on
x1=linspace(-1,1);
h=plot(x1,Runge(x1),'green');
h(1).LineWidth = 1.3;
title('Graph of f(x) and the langrange interpolating polys (4.b)')
hold off

%4.c

% plot the errors

x = -1 + 2.*rand(1,1000);

% Graph the Bary Formulas
X = [];
for j=1:1000
    X=[X, -1 + j*(2/1000)];
end
x = -1 +2.*rand(1,1000);

% create the values of the BaryFormula Approx.
BF4=[];
BF8=[];
BF12=[];
BFC4=[];
BFC8=[];
BFC12=[];
BFC100=[];

for i=1:length(x)
    BF4 = [BF4, BaryFormula(EN4,BW4,FV4,X(i))];
    BF8 = [BF8, BaryFormula(EN8,BW8,FV8,X(i))];
    BF12 = [BF12, BaryFormula(EN12,BW12,FV12,X(i))];
    BFC4 = [BFC4, BaryFormula(CN4,BCW4,FCV4,X(i))];
    BFC8 = [BFC8, BaryFormula(CN8,BCW8,FCV8,X(i))];
    BFC12 = [BFC12, BaryFormula(CN12,BCW12,FCV12,X(i))];
    BFC100 = [BFC100, BaryFormula(CN100,BCW100,FCV100,X(i))];
end

figure(9)
scatter(X,(Runge(X) - BF4),3,'r','filled')
hold on
scatter(X,(Runge(X) - BF8),3,'b','filled')
hold on
scatter(X,(Runge(X) - BF12),3,'black','filled')
title('Graph of the error of 4.a')
hold off

figure(10)
scatter(X,Runge(X) - BFC4,3,'r','filled')
hold on
scatter(X,Runge(X) - BFC8,3,'b','filled')
hold on
```

```
scatter(X,Runge(X) - BFC12,3,'black','filled')
hold on
scatter(X,Runge(X) - BFC100,3,'magenta','filled')
title('Graph of the error of 4.b')
hold off

%4.d

% We will reuse the Baryweights from 4.a

% create set of function values at equispaced
  % nodes for n = 4, 8 and 12 respectively

FV4=[];
for j=1:length(EN4)
    FV4=[FV4, exp(-(EN4(j)).^2)];
end
FV8=[];
for j=1:length(EN8)
    FV8=[FV8, exp(-(EN8(j)).^2)];
end
FV12=[];
for j=1:length(EN12)
    FV12=[FV12, exp(-(EN12(j)).^2)];
end


% Graph the Bary Formulas
X = [];
for j=1:1000
    X=[X, -1 + j*(2/1000)];
end
x = -1 +2.*rand(1,1000);

% create the values of the BaryFormula Approx.
BF4=[];
BF8=[];
BF12=[];

for i=1:length(x)
    BF4 = [BF4, BaryFormula(EN4,BW4,FV4,X(i))];
    BF8 = [BF8, BaryFormula(EN8,BW8,FV8,X(i))];
    BF12 = [BF12, BaryFormula(EN12,BW12,FV12,X(i))];
end

% Plot the function and all approximations

figure(11)
plot(X,BF4,'red')
hold on
plot(X,BF8,'blue')
hold on
plot(X,BF12,'black')
hold on
```

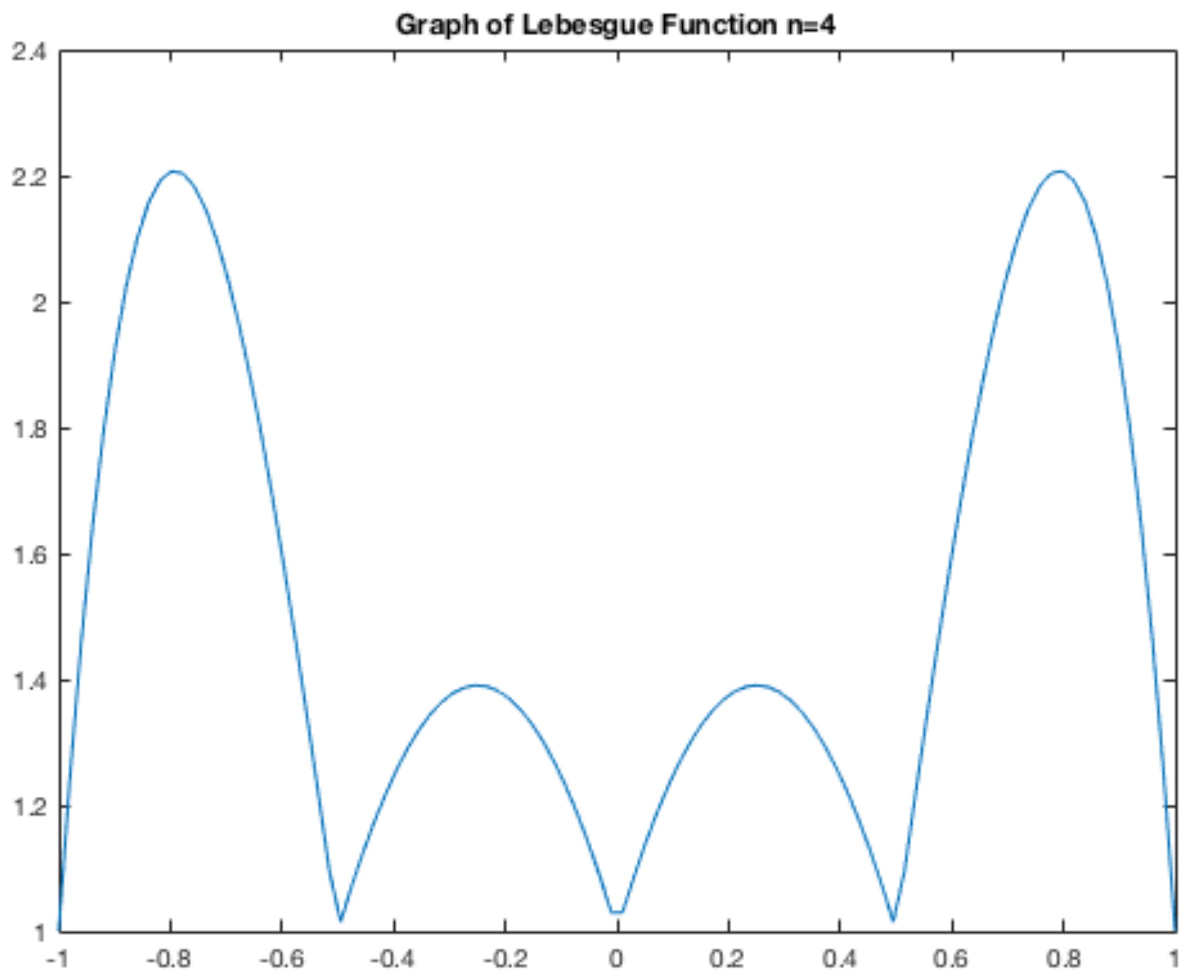
```
x1=linspace(-1,1);
y1 = exp(-(x1).^2);
plot(x1,exp(-(x1).^2),'green')
title('Graph of f(x) and the langrange interpolating polys (4.d)')
hold off

% All of the Lagrange Polynomials approximate it almost perfectly



% Functions used:

% 2.a:

function l = Leb(K,x)
%The Lebesgue Constant function
%   K is the list of nodes
%   x is where to evaluate the Leb func.

l=0;
for j=1:length(K)
    p=1;
    for i=1:length(K)
        if i ~= j

            p = p.*( (x - K(i)) / (K(j) - K(i)) );

        end
    end
    l = l + abs(p);
end

end

% 3.a:

function B = Baryweights(A,e,c)
%   A is the set of nodes
%   e=1 means the nodes are equidistributed
%   c=1 means the nodes are Chebyshev

B=[]; % The set of Bary weights

if e==1 % If the nodes are equispaced
    for j=0:length(A)-1
        Lambda = (-1)^(j).*(nchoosek(length(A)-1,j));
        B = [B, Lambda];
    end
end

if c==1 % If the nodes are Chebyshev
    for j=1:length(A)
        if j== 1 || j==length(A)
            Lambda = .5*(-1)^(j);
```

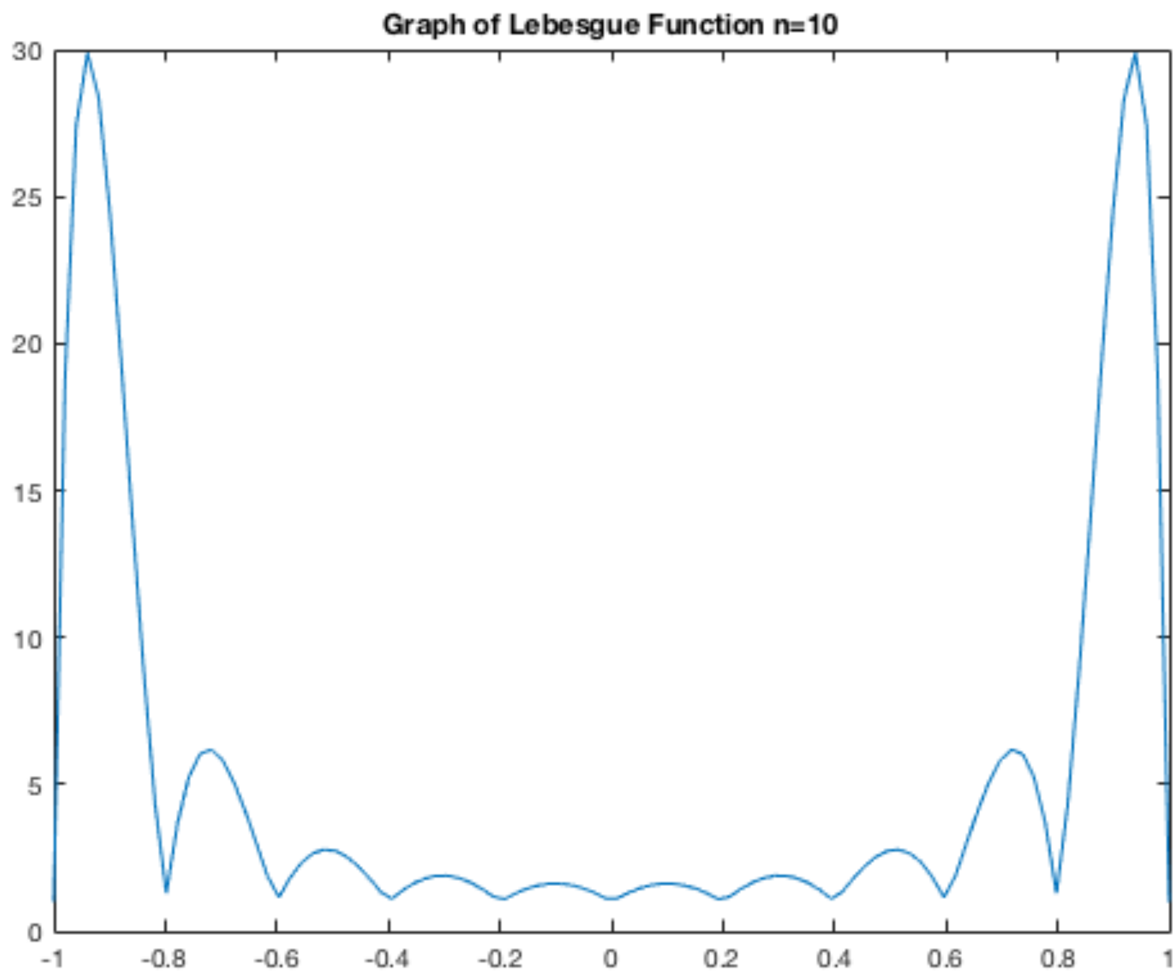
```
        end
        if j~=1 && j~=length(A)
            Lambda = (-1)^(j);
        end
        B = [B, Lambda];
    end
end

if e==0 && c==0 % If the nodes are arbitrary
    for j=1:1:length(A)
    P=1;
    for k=1:length(A)
        if k ~= j
            P = P*(A(j) - A(k));
        end
    end
    Lambda = (1/P);
    B = [B, Lambda];
    end
end
end

function P = BaryFormula(A,B,C,x)
%   A is the set of nodes, B is the baryweights,
%   C is the set of function values at the nodes,
%   x is the evaluated point
P=0;
T=0;
Q=0;
for j=1:length(B)
    T = T + C(j).*(B(j)./(x - A(j)));
end

for j=1:length(B)
    Q = Q + (B(j)./(x - A(j)));
end

P = T/Q;

end



% 4.a:

function y = Runge(x)
    y = (1./(1+25.*x.^2));
end
```

P2 =

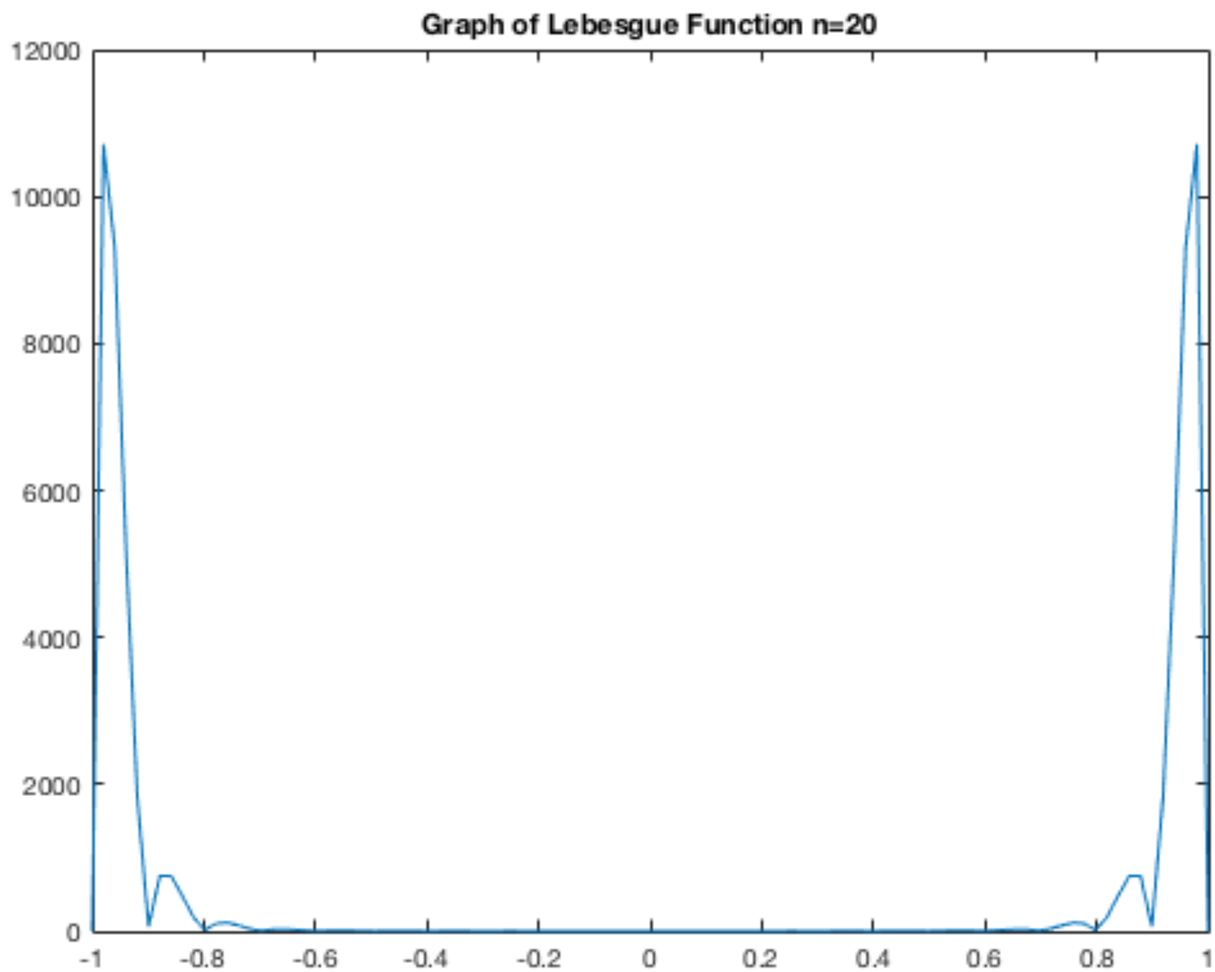
   -2.3438

Published with MATLAB® R2017b

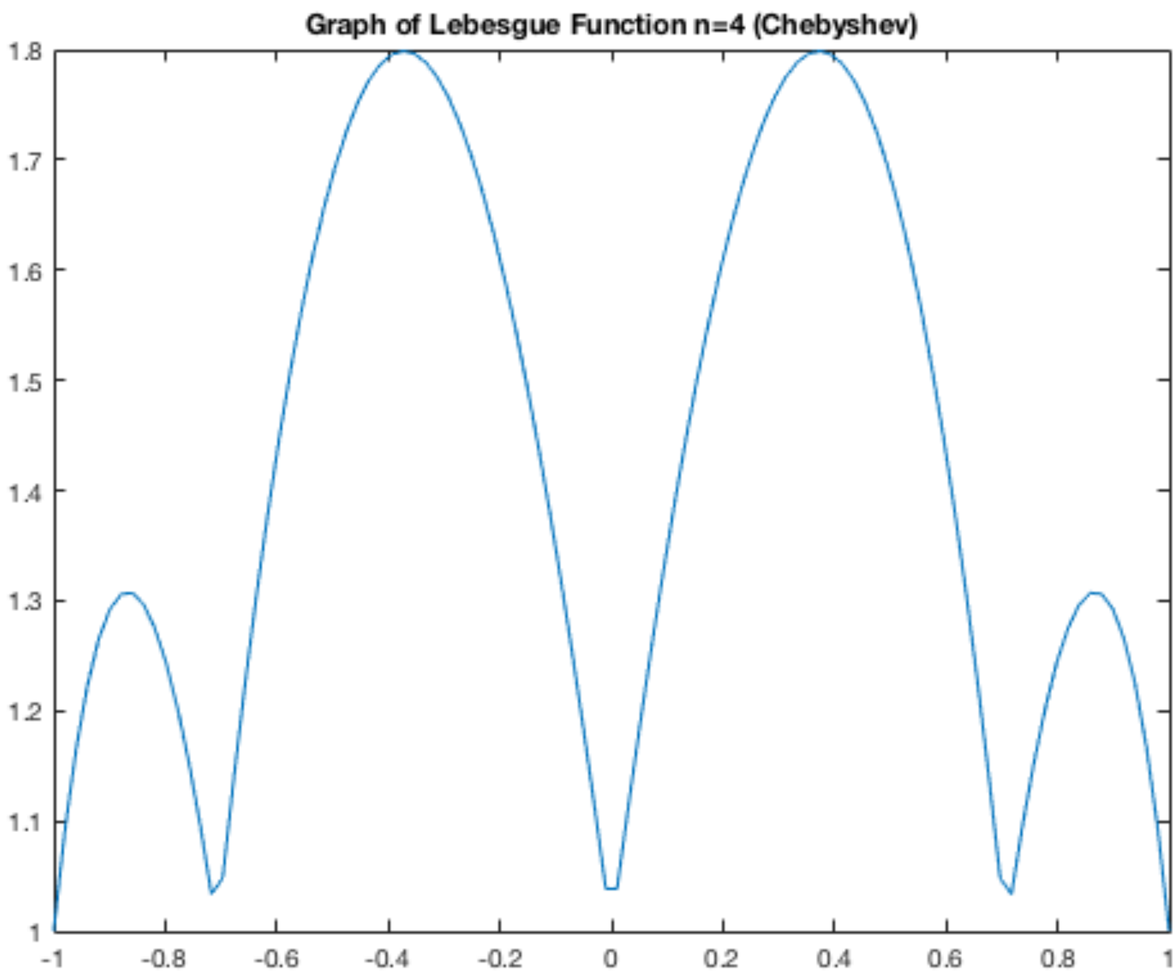

Graph of Lebesgue Function n=4
2.4
2.2
2
1.8
1.6
1.4
1.2
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

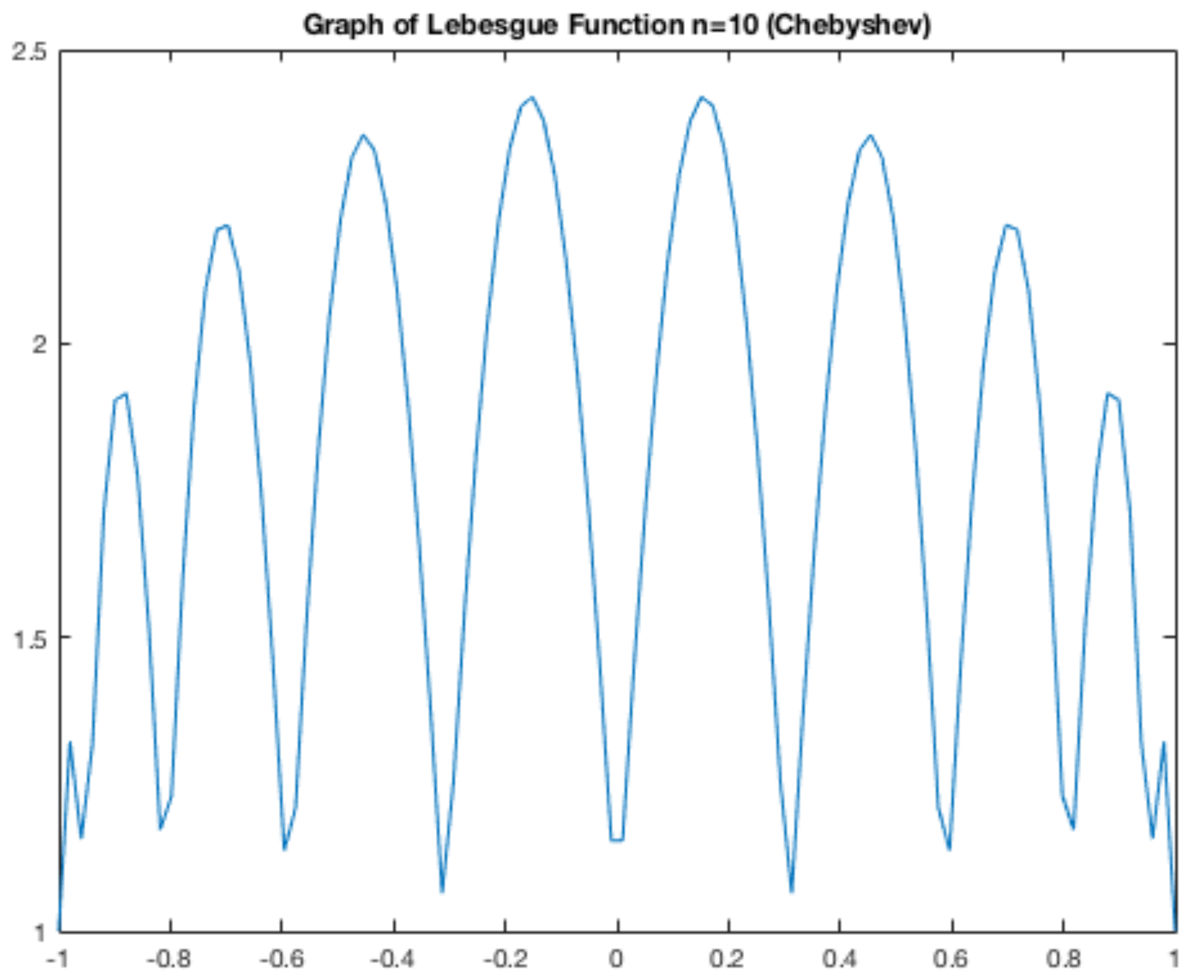


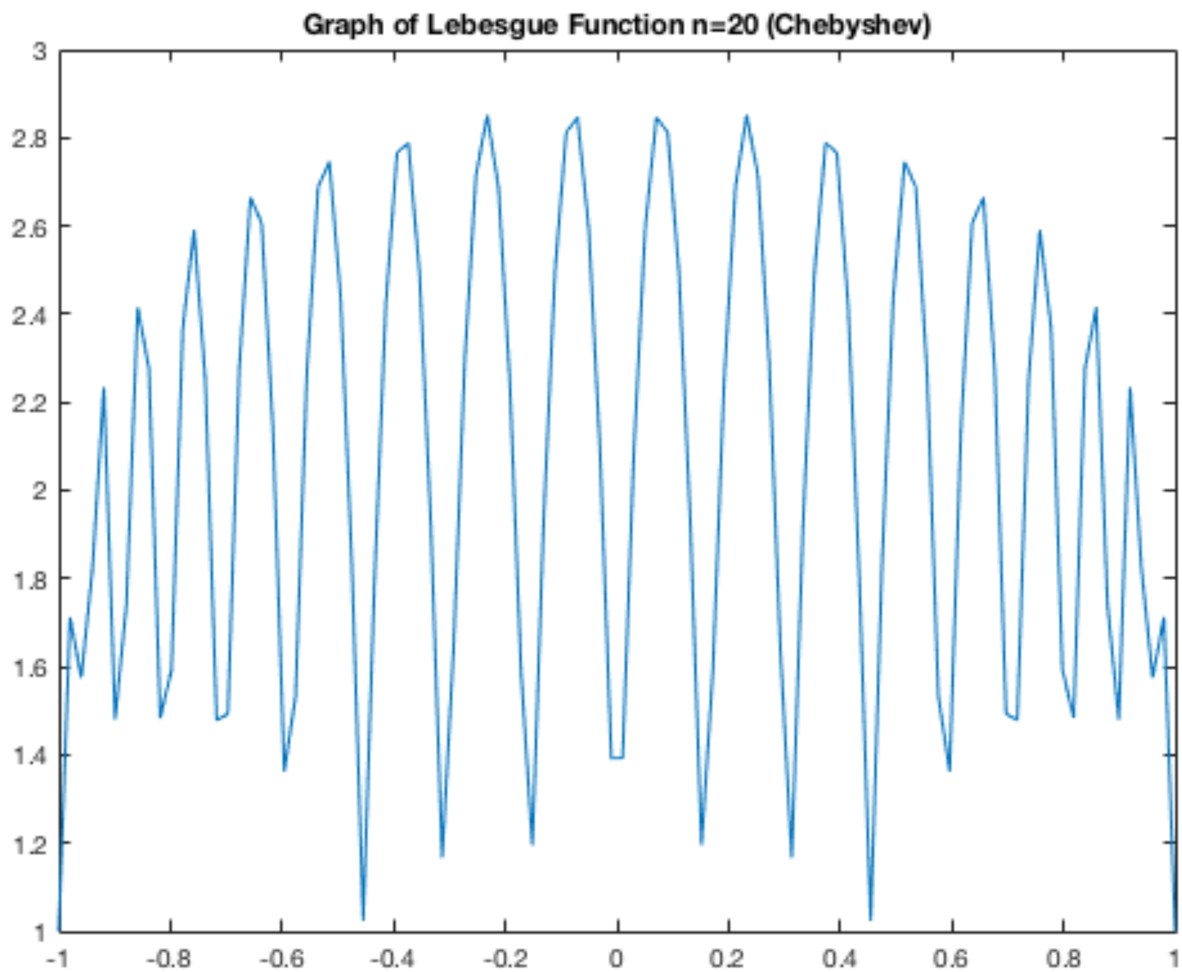
Graph of Lebesgue Function n=10
30
25
20
15
10
5
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1



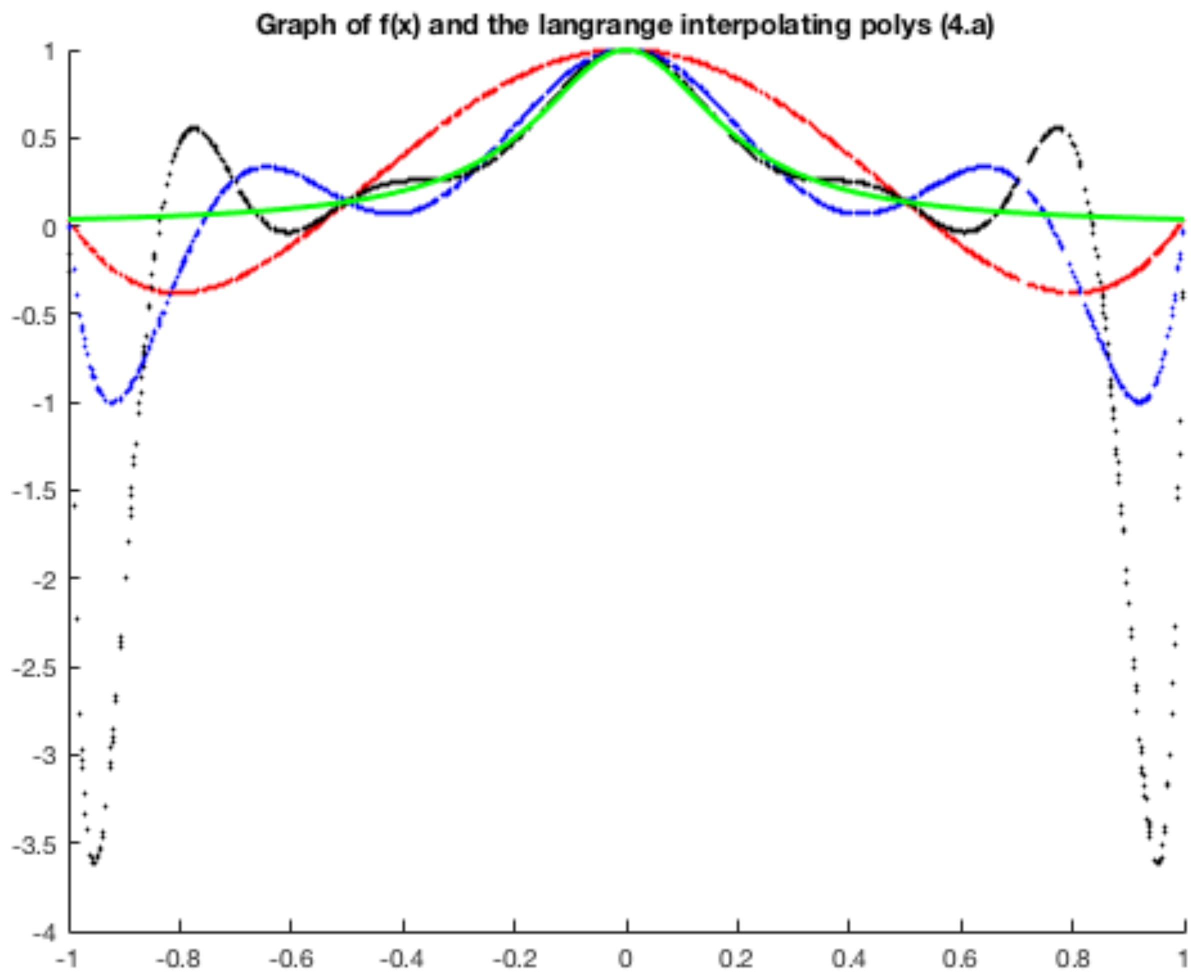
Graph of Lebesgue Function n=20
12000
10000
8000
6000
4000
2000
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

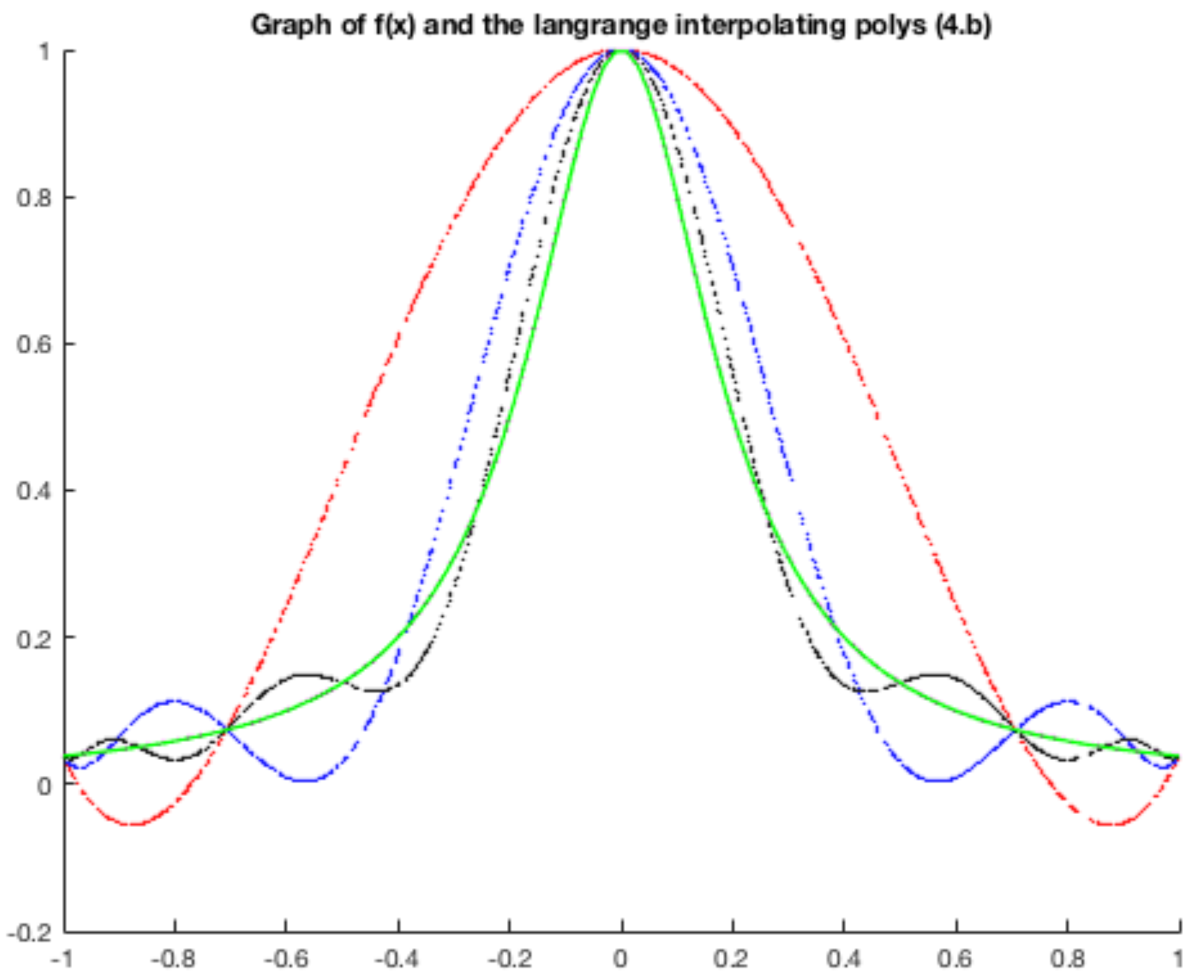

Graph of Lebesgue Function n=4 (Chebyshev)
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

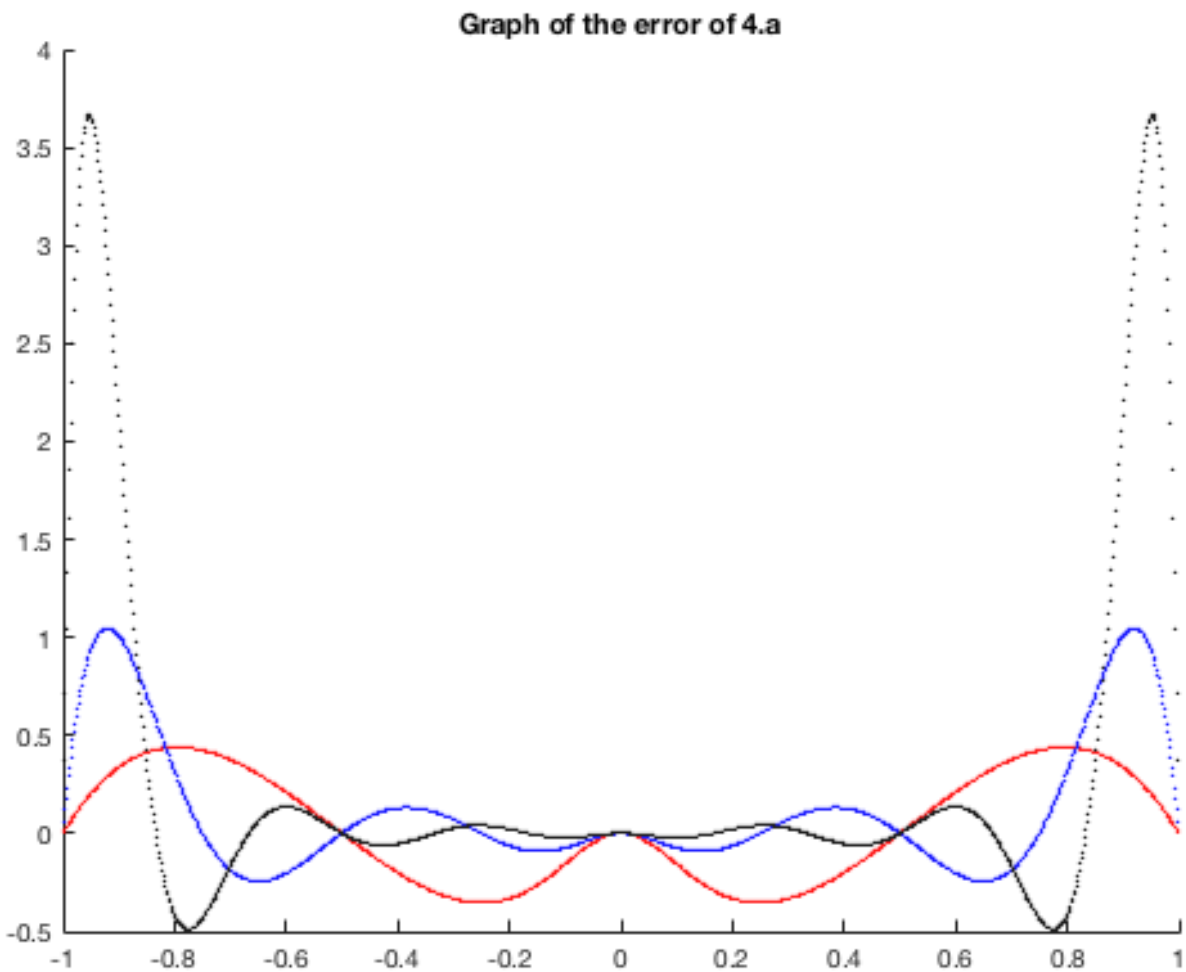

Graph of Lebesgue Function n=10 (Chebyshev)
2.5
2
1.5
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1



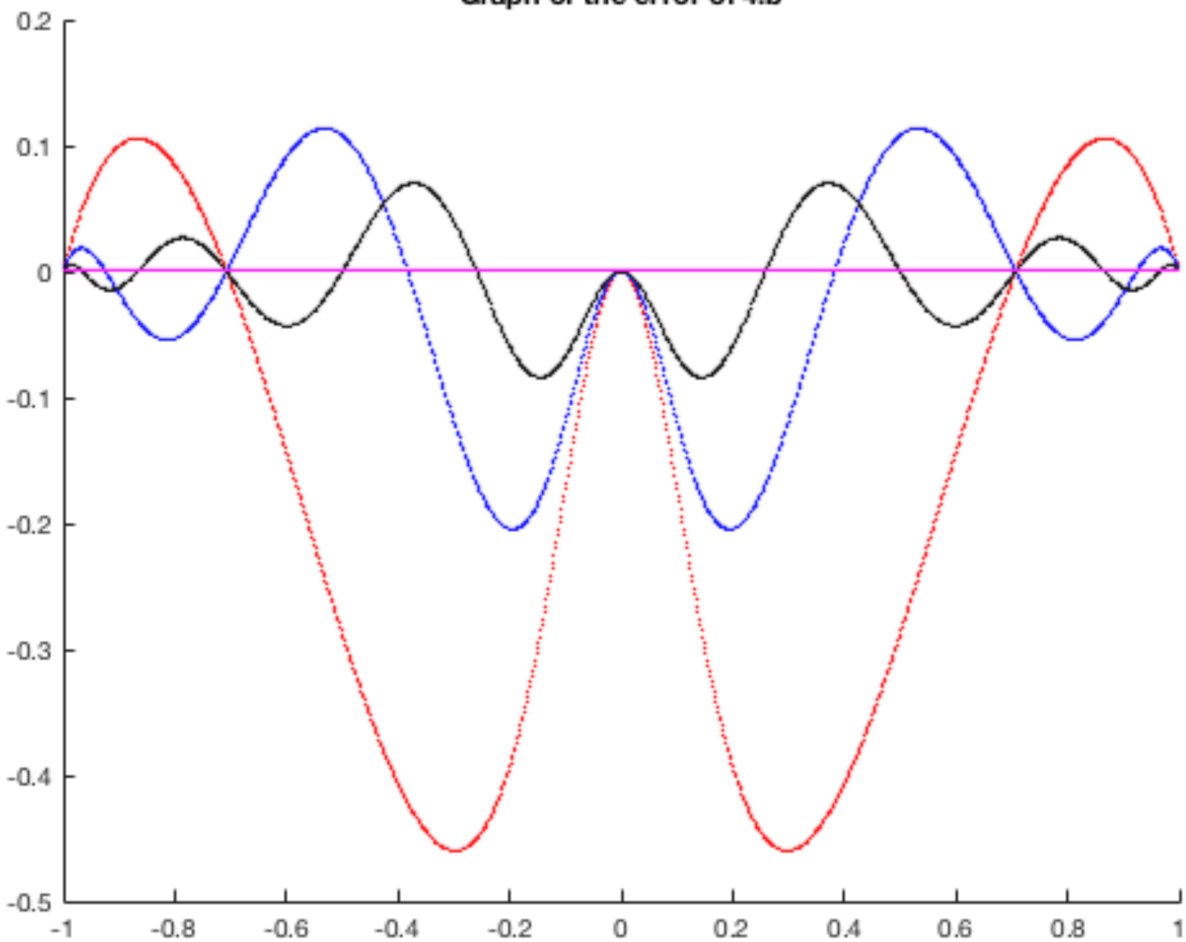
Graph of Lebesgue Function n=20 (Chebyshev)
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

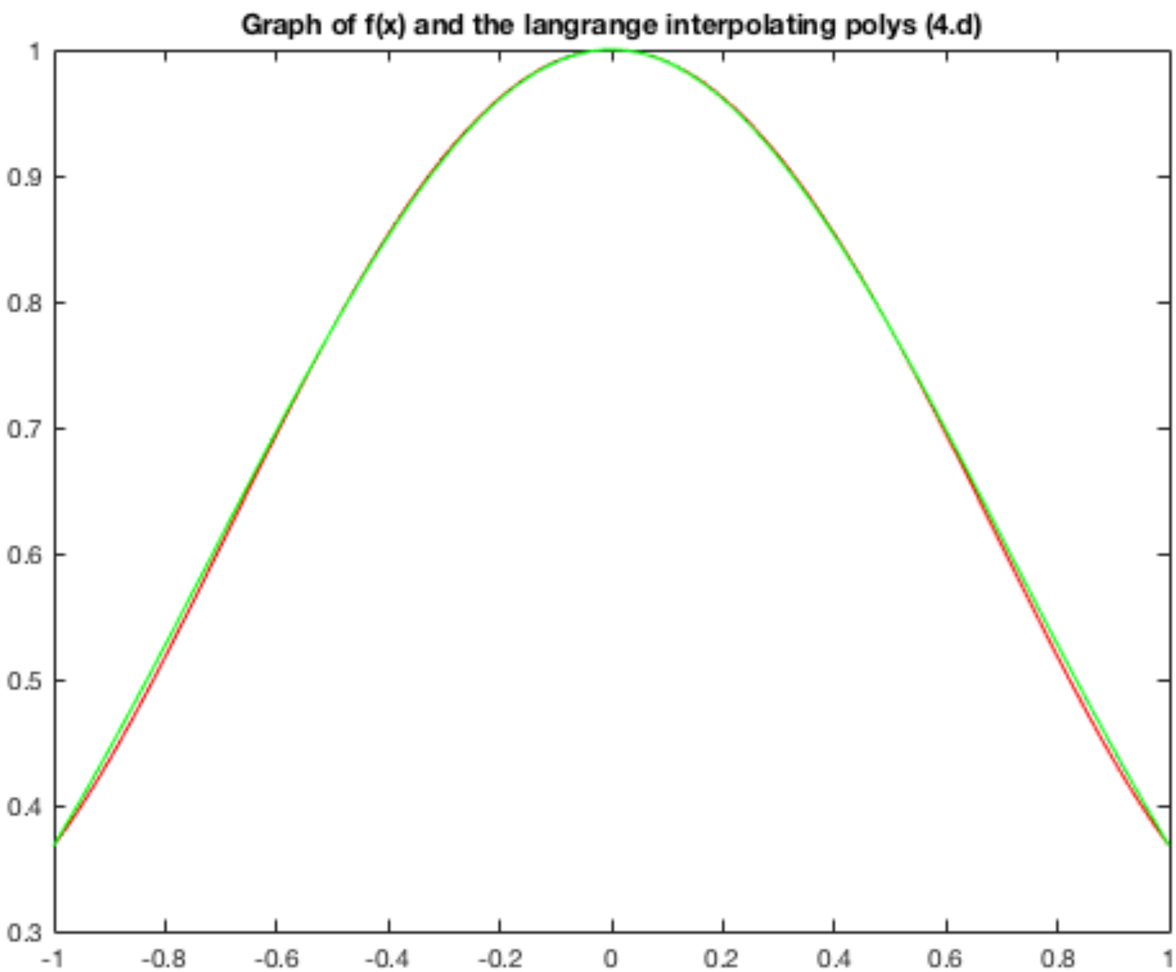

Graph of f(x) and the langrange interpolating polys (4.a)
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Graph of f(x) and the langrange interpolating polys (4.b)
1
0.8
0.6
0.4
0.2
0
-0.2
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Graph of the error of 4.a
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Graph of the error of 4.b




Graph of f(x) and the langrange interpolating polys (4.d)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1