Metody Numeryczne - ćwiczenie 1

Sprawozdanie

Dawid Chmielewski 311188

Algorytmy Interpolacji oraz aproksymacji.

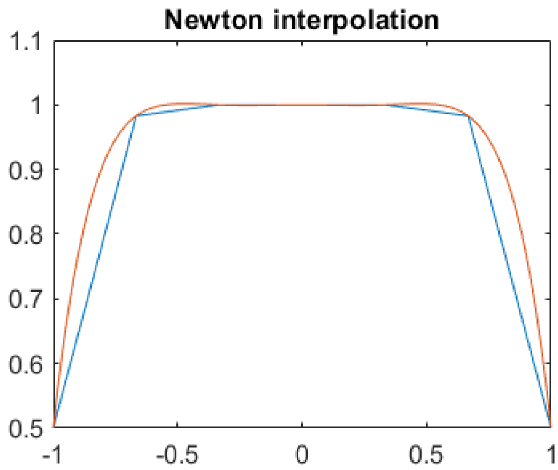
Do realizacji tego ćwiczenia zostały podane następujące dane: funkcja f(x) = 1/(x^10 + 1); przedział: <-1, 1>; liczba N węzłów: 7.

```
clc
clear
close all
format compact

N = 7;

x = linspace(-1,1,N);
f = @(x)(1./(1+x.^10));
y = f(x);

xpts = linspace (-1, 1, 100);
```

zadanie 1. Interpolacja Newtona.

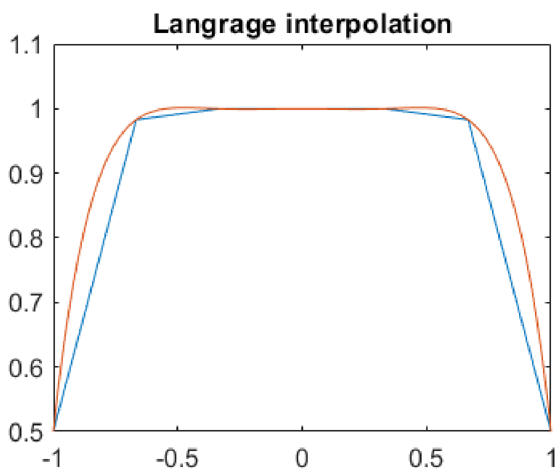
W ramach zadania 1 zaimplementowałem interpolację Newtona, która przyjmuje współrzędne x i y danego zestawu punktów interpolowanej funkcji oraz zbiór współrzędnych x punktów dla funkcji interpolującej i zwraca wartości y tych punktów jako wektor. Interpolacja Lagrange'a była natomiast przedstawiana w trakcie zajęć, moja ingerencja w nią sprowadzała się do zmodyfikowania jej tak, aby przyjmowała i zwracała identyczne argumenty.

```
axis equal
plot(x,y);
hold on;
yn = newton(x,y,xpts);
plot(xpts,yn);
title('Newton interpolation');
hold off;
```



Interpolacja Langrage'a:

```
figure
plot(x,y)
hold on;
yl = langrage(x,y,xpts);
plot(xpts,yl);
title('Langrage interpolation');
hold off;
```



Obie metody interpolacji dają taki sam rezultat. Otrzymane funkcje z ich pomocą są indentyczne.

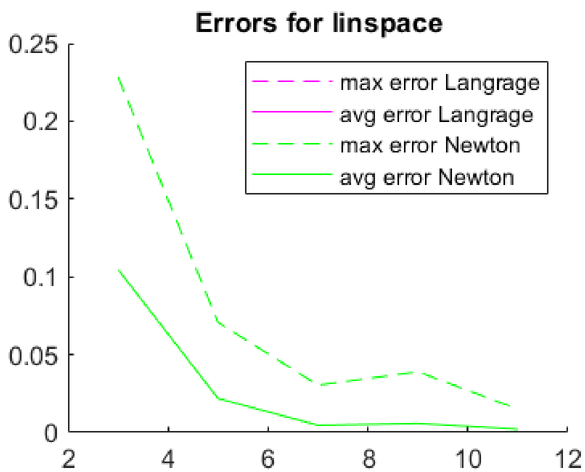
zadanie 2.

Eksperyment przeprowadziłem dla pięciu różnych ilości węzłów: 3, 5, 7, 9 i 11.

Dla równoodległych węzłów, uzyskanych za pomocą linspace:

```
nodesNmbrs = [3, 5, 7, 9, 11];
maxENewt = 1:5;
avgENewt = 1:5;
maxELang = 1:5;
avgELang = 1:5;
```

```
for i = 1:5
    linearErrors = error(f, linspace(-1,1,nodesNmbrs(i)), 1000);
    maxENewt(i) = linearErrors(1);
    avgENewt(i) = linearErrors(2);
    maxELang(i) = linearErrors(3);
    avgELang(i) = linearErrors(4);
end

figure
hold on;

plot(nodesNmbrs, maxELang, '--', color='magenta');
plot(nodesNmbrs, avgELang, '-', color='magenta');
plot(nodesNmbrs, maxENewt, '--', color='green');
plot(nodesNmbrs, avgENewt, '-', color='green');

title('Errors for linspace');
legend('max error Langrage', 'avg error Langrage', 'max error Newton', 'avg error Newton');

hold off;
```

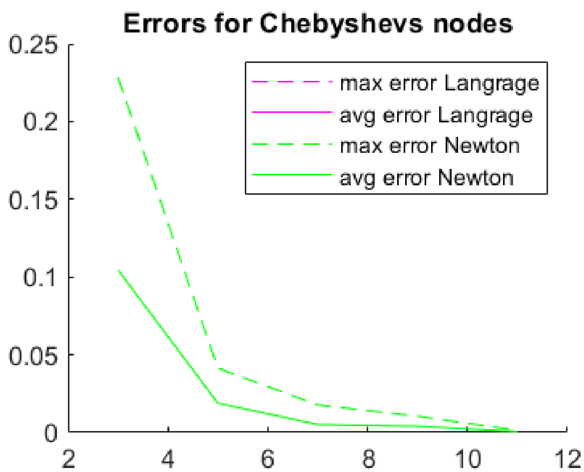


Możemy zauważyć, iż błędy maksymalne i średnie są dla obu interpolacji równe. Pokrywanie się tych linii na wykresie owocuje zasłonięciem jednej z nich (purpurowej) przez drugą (zieloną).

zadanie 3.

```
for i = 1:5
    chebyshevErrors = error(f, chebyshev(-1,1,nodesNmbrs(i)), 1000);
    maxENewt(i) = chebyshevErrors(1);
    avgENewt(i) = chebyshevErrors(2);
    maxELang(i) = chebyshevErrors(3);
    avgELang(i) = chebyshevErrors(4);
end

figure
```

```
hold on;

plot(nodesNmbrs, maxELang, '--', color='magenta');
plot(nodesNmbrs, avgELang, '-', color='magenta');
plot(nodesNmbrs, maxENewt, '--', color='green');
plot(nodesNmbrs, avgENewt, '-', color='green');

title('Errors for Chebyshevs nodes');
legend('max error Langrage', 'avg error Langrage', 'max error Newton', 'avg error Newton');

hold off;
```



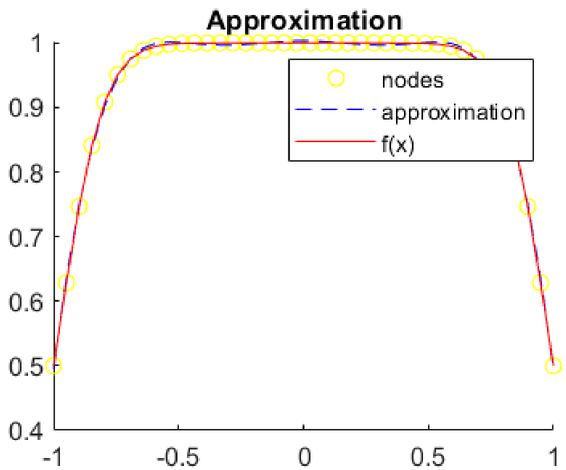
Jak możemy zaobserwować na wykresach, również dla węzłów Czebyszewa wartości błędów są identyczne i linie wykresów pokrywają się. Możemy natomiast dostrzec wyraźnie mniejszy maksymalny błąd w przypadku zastosowania węzłów Czebyszewa.

zadanie 4.

Aproksymację średniokwadratową przetestowałem dla przykładowego zestawu wartości: k=10, n=40.

```
n = 40;
k = 10;
x = linspace(-1, 1, n);
y = f(x);

A = zeros(k);
B = zeros(k, 1);

for i = 1:k
    for j = 1:k
        A(i, j) = sum(x.^(i + j - 2));
    end
    for j = 1:n
        B(i) = B(i) + (x(j)^(i-1)) * y(j);
    end
end
```

```
X = A \ B;
X = flip(X);

figure
hold on;

plot(x, f(x), 'o', color= 'yellow');
plot(x, polyval(X, x),'--',color = 'blue');
plot(x, f(x),'-', color = 'red');

legend('nodes', 'approximation','f(x)');
title('Approximation');

hold off;
```



Jak widać na wykresie, aproksymacja dobrze pokrywa się z pierwotną funkcją.

Wszystkie powyższe zadania zostały wykonane w oparciu o poniższe funkcje.

```
function Y = newton(x, y, X)
    n = length(x);
    matr = zeros(n,n);
    for i = 1:n
        matr(i) = y(i);
    end

    for i = 2:n
        for j = 1:n-i+1
            matr(j, i) = (matr(j+1,i-1)-matr(j,i-1))/(x(j+i-1)-x(j));
        end
    end
    v = matr(1, :);
    Y = newtonconf(v, x, X);
end

function YY = newtonconf(a, xp, x)
```

```
YY = zeros(length(x), 1);
    for i = 1:length(x)
        cc = 0;
        for j=1:length(a)
           ccf = 1;
                for k=1:(j-1)
                    ccf = ccf * (x(i) - xp(k));
                end
             cc = cc + a(j) * ccf;
        end
        YY(i) = cc;
    end
end

function Y = langrage(x, y, X)
N = numel(x)-1;
L = zeros(N+1,N+1);
    for k = 0:N
        nodes = x;
        nodes(k+1) = [];
        num = poly(nodes);
        den = prod(x(k+1)-nodes);
        L(k+1,:) = num/den;
    end
L = y'.*L;
P = sum(L);
Y = polyval(P,X);
end

function errorVectors = error(f, x, n)
xpts = linspace( min(x), max(x), n);
maxErrNewt = 0;
sumErrsNewt = 0;
maxErrLang = 0;
sunErrsLang = 0;
for i = 1:n
    errorNewt = abs(f(xpts(i)) - newton(x,f(x),xpts(i)));
    errorLang = abs(f(xpts(i)) - langrage(x,f(x),xpts(i)));
    if (errorNewt > maxErrNewt)
        maxErrNewt = errorNewt;
    end
    if (errorLang > maxErrLang)
        maxErrLang = errorLang;
    end
    sumErrsNewt = sumErrsNewt + errorNewt;
    sunErrsLang = sunErrsLang + errorLang;
end

avgErrNewt = sumErrsNewt/n;
avgErrLang = sumErrsNewt/n;
```

```
errorVectors = [ maxErrNewt, avgErrNewt, maxErrLang, avgErrLang];
end

function x = chebyshev(a, b, n)
x = zeros(1, n);
    for i = 1:n
        x(i) = 0.5*(a+b) + 0.5*(b-a) * cos((i-1)/(n-1)*pi);
    end
end
```