**%% Przyk?adowe zadanie 1**

clc;

clear;

format long;

Fi = [11];

h = 0.5;

w = prostyNewtonCotes(Fi, h)

Fi = [0, 11];

h = 1;

w = prostyNewtonCotes(Fi, h)

Fi = [0, 0.0107, 11];

h = 0.5;

w = prostyNewtonCotes(Fi, h)

Fi = [0, 0.000186, 0.1907, 11];

h = 0.333;

w = prostyNewtonCotes(Fi, h)

Fi = [0, 0.00001, 0.0107, 0.61945, 11];

h = 0.25;

w = prostyNewtonCotes(Fi, h)

Fi = [11, 0.0107421875, 0, 0.0107421875, 11];

h = 0.5;

w = prostyNewtonCotes(Fi, h)

**%% Zadanie 2**

clc;

clear;

format long;

f = @(x)(11 \* x^10);

X = linspace(-1, 1, 201);

Y = arrayfun(f, X);

rzedy = [1 2 3 4];

for r = rzedy

w = zlozonyNewtonCotes(Y, 0.01, r)

end

%% Zadanie 3

clc;

clear;

format long;

rzedy = [1 2 3 4];

lr = length(rzedy);

W = zeros(lr, 1000);

a = -2;

b = 2;

f = @(x)(11 \* x^10);

for r = rzedy

for i = (r + 1):1000

n = r \* i + 1;

h = (b - a)/(n - 1);

X = linspace(a, b, n);

Y = arrayfun(f, X);

W(r, i) = zlozonyNewtonCotes(Y, h, r);

end

end

**%%**

loglog([1:1000], abs(W(1, :) - 2.0), '-r', [1:1000], abs(W(2, :) - 2.0), '-g', [1:1000], abs(W(3, :) - 2.0), '-b', [1:1000], abs(W(4, :) - 2.0), '-m')

grid on

legend('d = 1','d = 2','d = 3','d = 4')

**function wynik = prostyNewtonCotes(wartosci, krok)**

%PROSTYNEWTONCOTES Ca?kowanie metod? prost? Newtona-Cotesa

%Obs?ugiwane s? rz?dy od 0 do 5

switch length(wartosci)

case 1

wynik = wartosci(1) \* krok;

return;

case 2

sigma = [1 1];

ns = 2;

case 3

sigma = [1 4 1];

ns = 6;

case 4

sigma = [1 3 3 1];

ns = 8;

case 5

%TODO: wzór Booole'a

sigma = [7 32 12 32 7];

ns = 90;

otherwise

wynik = NaN;

return;

end

wynik = 0;

for i = 1:length(wartosci)

wynik = wynik + sigma(i) \* wartosci(i);

end

wynik = wynik \* (krok \* (length(wartosci)-1));

wynik = wynik / ns;

end

**function indeksy = utworzNakladajaceIndeksy(dlugoscWektora, krok)**

%UTWORZNAKLADAJACEINDEKY Generuje nakladajace indeksy dla danego wektora

% Przydatne do zlozonego NC

l = dlugoscWektora;

podstawoweIndeksy = 1:krok-1:l;

if l ~= podstawoweIndeksy(end)

indeksy = zeros(2, length(podstawoweIndeksy));

for i = 1:length(indeksy)-1

indeksy(:, i) = [podstawoweIndeksy(i), podstawoweIndeksy(i+1)];

end

indeksy(:, end) = [podstawoweIndeksy(end), l];

else

indeksy = zeros(2, length(podstawoweIndeksy)-1);

for i = 1:length(indeksy)

indeksy(:, i) = [podstawoweIndeksy(i), podstawoweIndeksy(i+1)];

end

end

end

**function wynik = zlozonyNewtonCotes(wartosci, krok, rzad)**

%ZLOZONYNEWTONCOTES Ca?kowanie metod? prost? Newtona-Cotesa

wynik = 0;

indeksy = utworzNakladajaceIndeksy(length(wartosci), rzad+1);

for i = indeksy

przedzial = wartosci(i(1):i(2));

wynik = wynik + prostyNewtonCotes(przedzial, krok);

end

end