Warszawa, 9.05-3.06.2020

PROJEKT WNUM

ZADANIE 3 #25

Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych

Przedmiot: WNUM

Prowadzący projekt: dr inż. Andrzej Miękina

Wykonawca: Maciej Kaczkowski

Oświadczam, że niniejsza praca, stanowiąca podstawę do uznania osiągnięcia efektów uczenia się z przedmiotu **Wstęp do metod numerycznych** została wykonana przeze mnie samodzielnie.

Maciej Kaczkowski

300660

1. CEL DOŚWIADCZEŃ, WSTĘP TEORETYCZNY

Celem doświadczeń było wyznaczenie estymaty pola powierzchni figury F ograniczonej wielomianem

$$w(x) = x^6 + 9x^5 - 59x^4 - 5511x^3 + 1316x^2 + 44308x - 162720$$

w przedziale [rl, ru], gdzie rl i ru to odpowiednio mniejszy i większy pierwiastek wielomianu. Jako referencyjną, dokładną wartość przyjęto wartość całki

$$I = \left| \int_{rl}^{ru} w(x)dx \right|$$

obliczoną za pomocą wyrażeń symbolicznych.

Metoda kwadratur

Polega na zastąpieniu obliczanej całki dokładnym wynikiem całkowania wielomianu interpolującego. Ten efekt uzyskujemy przy pomocy kwadratury – sumy iloczynów współczynników (ustalonych przy budowie kwadratury) i wartości funkcji interpolującej w węzłach Najważniejszą cechą kwadratury jest jej zbieżność – kwadratura jest zbieżna jeśli wraz ze wzrostem liczby węzłów błąd rozwiązania maleje. Szybkość zbieżności kwadratury określa rząd kwadratury p. Mówimy, że kwadratura jest rzędu p jeśli za jej pomocą można w sposób dokładny odwzorować wszystkie wielomiany stopni niższych niż p, ale nie wszystkie wielomiany stopnia p. W doświadczeniu użyto kwadratur złożonych o rzędach N = 2, …, 6 i zbadano zależność dokładności rozwiązania od ilości węzłów zmienianej z krokiem 3.

Metoda Monte-Carlo (wersja „orzeł-reszka")

Jest to metoda estymacji statystycznej wartości oczekiwanej zmiennej statystycznej na podstawie jej realizacji. Metody tego typu znajdują szerokie zastosowanie w problemie całkowania numerycznego, zazwyczaj dla całek których nie można obliczyć innymi metodami (lub jest bardzo trudno). W doświadczeniu użyto metody „orzeł-reszka", dla której przyjmujemy, że rozkład prawdopodobieństwa zmiennej losowej jest funkcją dwuwartościową i przyjmuje wartość 1 jeśli należy do obszaru (pola pod wykresem), a wartość 0 jeśli nie należy. Punkty w których wykonujemy losowanie mogą być ułożone losowo lub regularnie. Metoda jest tym dokładniejsza im więcej punktów przyjmiemy, jednak kosztem wzrastającego czasu obliczeń.

2. PUNKT 1

Na podstawie rozwiązania zadania 2 z projektu #25 przyjęto:

$ru = 8$

$rl = -9$

Za pomocą wyrażeń symbolicznych obliczono wartość całki:

$$\dot{I} = \left| \int_{-9}^{8} w(x)dx \right| = \frac{1004310547}{420}$$

Obliczoną wartość wykorzystano w dalszych punktach jako wartość dokładną.

3. PUNKT 2

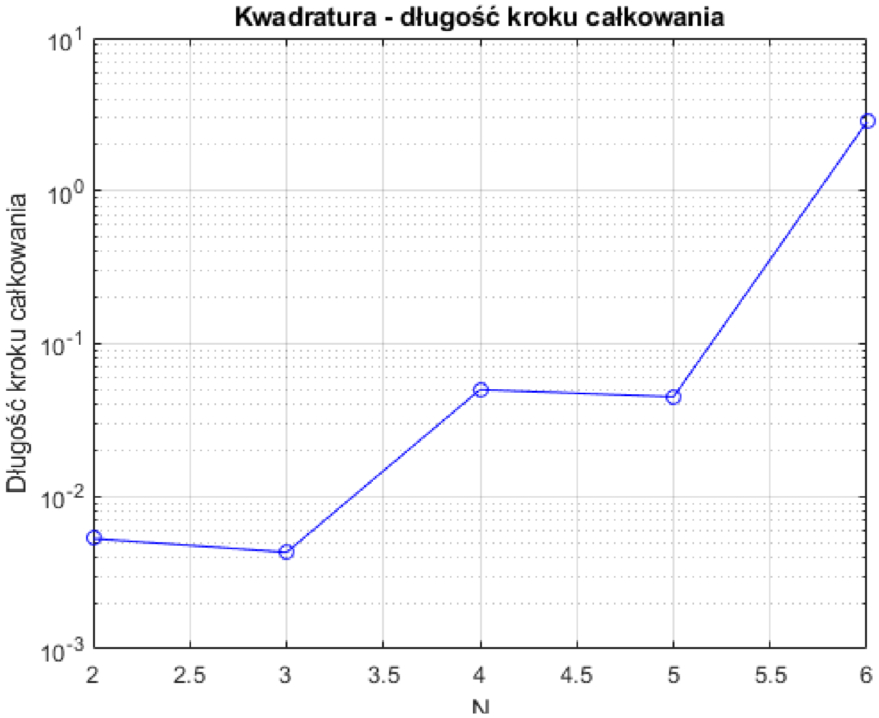
Określono błąd bezwzględny rozwiązania jako:

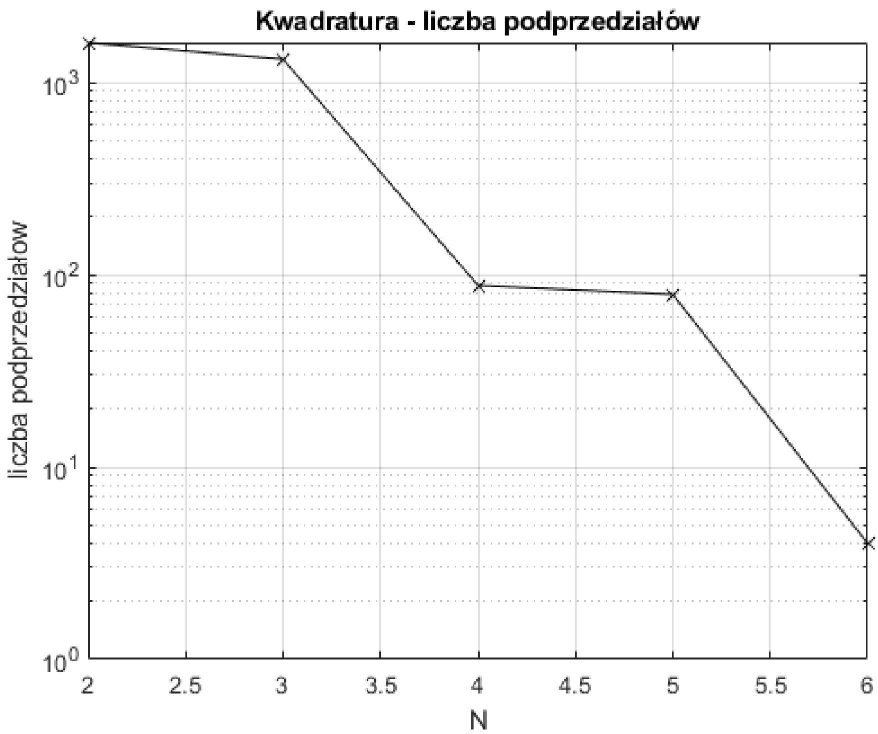
$\Delta\tilde{I} \stackrel{\text{def}}{=} \tilde{I} - \dot{I}$

Przyjęto następujący warunek określający wymaganą dokładność rozwiązania:

$|\Delta\tilde{I}| < 5 \cdot 10^{-7}$

Ilość podprzedziałów zwiększano z krokiem 3.

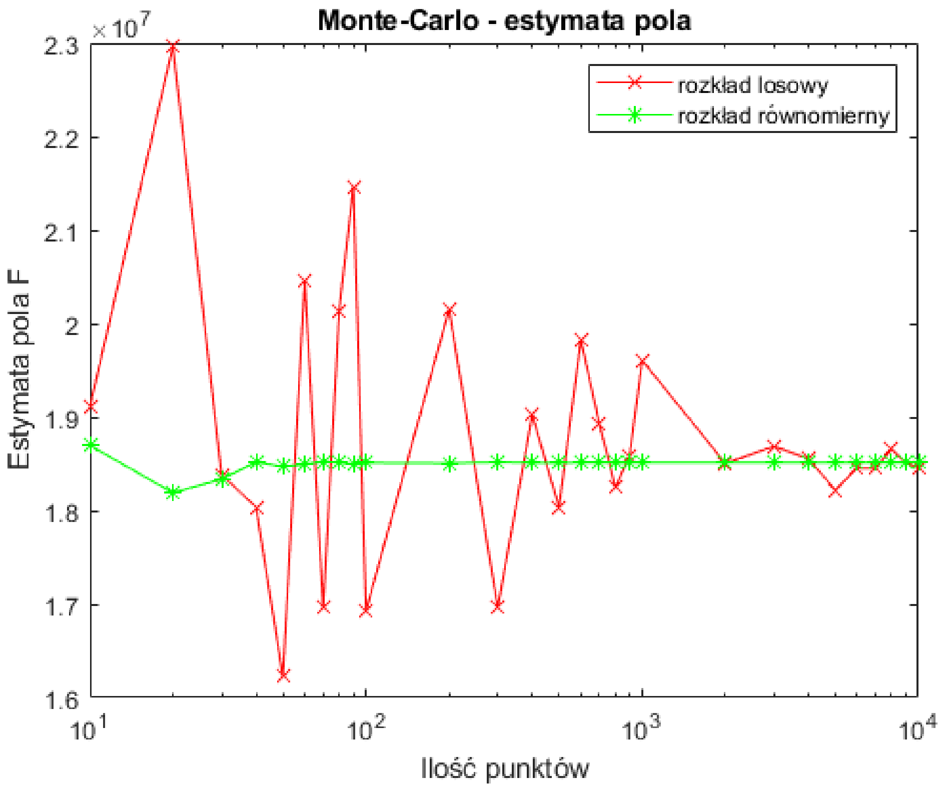


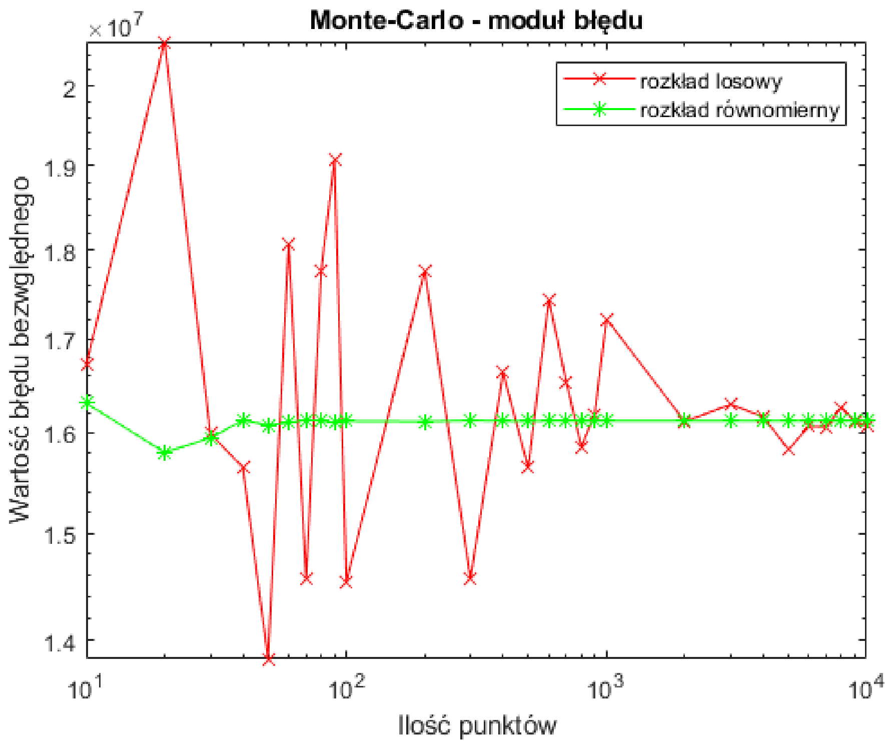


4. PUNKT 3

Wykonano całkowanie metodą Monte-Carlo dla wzrastających wartości parametru N.

Uzyskane zależności przestawiają poniższe wykresy:





5. OBSERWACJE, WNIOSKI OGÓLNE

Wraz ze wzrostem rzędu kwadratury spada liczba podprzedziałów, a rośnie długość kroku. Jest to zgodne z przewidywaniami teoretycznymi – im wyższy stopień wielomianu interpolującego tym bardziej jego całka przypomina całkę badanej funkcji. Umożliwia to wydłużenie kroku całkowania i zmniejszenie liczby podprzedziałów, co pozwala na zmniejszenie czasu obliczeń. Warto zauważyć, że ilość podprzedziałów zmniejsza się o 2 rzędy wielkości – od wartości rzędu 10^3 do wartości rzędu 10^1.

W warunku zakończenia pętli należy porównywać różnicę wartości obliczonej numerycznie i wartość $\dot{I} = \int_{-9}^{8} w(x)dx = -\frac{1004310547}{420}$, a nie wartość bezwzględną. W przeciwnym wypadku może dojść do nieskończonej iteracji pętli.

Zgodnie z przewidywaniami wzrost ilości punktów w metodzie Monte-Carlo powodował wzrost dokładności estymacji. Metoda z losowaniem N^2 punktów osiągała dokładniejsze wyniki później niż metoda z siatką $N \times N$ równomiernie rozłożonych punktów. Oprócz tego dawała różne efekty w kolejnych wywołaniach, co jest spowodowane losowym rozkładem punktów. Błąd rozwiązania nadal był jednak duży, co może wynikać z faktu, że jest to błąd bezwzględny – nie uwzględnia wielkości błędu w stosunku do wielkości dokładnej, a jedynie ich różnicę.

Metoda kwadratur w użytej implementacji charakteryzuje się krótkim czasem wykonania. Metoda Monte-Carlo w użytej implementacji charakteryzuje się długim czasem wykonania – być może jest to spowodowane zbyt częstym używaniem konstrukcji for … end zamiast operacji na macierzach. Wiadomo, że MatLab jest środowiskiem zorientowanym macierzowo, więc użycie operacji na macierzach mogłoby przyspieszyć działanie programu. Można wyciągnąć ogólny wniosek, że jeśli to możliwe należy unikać metod Monte-Carlo, w zamian stosując inne metody, np. kwadratury.

Wykresy wykonano w skali półlogarytmicznej – semilogy (liczba podprzedziałów, długość kroku całkowania), półlogarytmicznej – semilogx (estymata pola F) oraz logarytmicznej – loglog (błąd bezwzględny).

6. LISTING KODU

```
clc
clear all

%-----------------------------------------------
-------------------------
%-----------------------------PUNKT 1---------
-------------------------
%-----------------------------------------------
-------------------------

ru = 8;
rl = -9;
coeff = [1, 9, -59, -1155, 1316, 44308, -
162720];

%obliczenie całki za pomocą wyrażeń
symbolicznych
syms x;
%y = symfun(x.^6 + 9.*x.^5 - 59.*x.^4 -
1155.*x.^3 + 1316.*x.^2 + 44308.*x - 162720,
[x]);
w = x.^6 + 9.*x.^5 - 59.*x.^4 -1155.*x.^3 +
1316.*x.^2 + 44308.*x - 162720;

%rzeczywiste pierwiastki wielomianu (uzyskane
z zadania 2) to: -9 i 8
p_int = int(w, [rl ru]);
display(abs(p_int));
display(1004310547/420);

%-----------------------------------------------
-------------------------
%-----------------------------PUNKT 2---------
-------------------------
%-----------------------------------------------
-------------------------

DELTA = 5*10^-7;

%liczba podprzedzialow
subs_num = zeros(1,5);

%dlugosc kroku calkowania
h = zeros(1,5);

for N=2:6
   I=0;
   subs=1;

   while(abs(I-p_int) > DELTA)
       I=0;
       %podzial przedzialu na podprzedzialy
       x=(ru-rl)/subs;

       for i = 1:subs
           a = rl + (i-1) * x;
           b = rl + i * x;
           %wykonanie kwadratury
           I = I + kwadratura(coeff, a, b, N);
       end

   subs=subs+3;

   end

   h(N-1) = x/N;
   subs_num(N-1) = subs;

end

%wykonanie wykresow
figure(1);
semilogy((2:6),h, 'b-o');
xlabel("N");
ylabel("Długość kroku całkowania");
title('Kwadratura - długość kroku
całkowania');
grid on;

figure(2);
semilogy((2:6),subs_num, 'k-x') ;
xlabel("N");
ylabel("liczba podprzedziałow");
title('Kwadratura - liczba podprzedziałów');
grid on;

%-----------------------------------------------
-------------------------
%-----------------------------PUNKT 3---------
-------------------------
%-----------------------------------------------
-------------------------

%UWAGA UWAGA UWAGA UWAGA UWAGA UWAGA UWAGA
UWAGA UWAGA UWAGA
%wykonywanie kodu z punktu 3 zajmuje dużo
czasu, około 20 min

N = [linspace(10,100,10),
linspace(200,1000,9), linspace(2000,10000,9)];
c_eq = 0;
c_rand = 0;
estimate_eq = zeros(1,length(N));
estimate_rand =zeros(1, length(N));
delta_abs_equal = zeros(1, length(N));
delta_abs_random = zeros(1, length(N));

%wyznaczenie obszaru Omega:
%x nalezy do [-9, 8]
w_int = @(x) x.^7./7 + (3.*x.^6)./2 -
(59.*x.^5)./5 - (1155.*x.^4)./4 +
(1316.*x.^3)./3 + 22154.*x.^2 - 162720.*x;
num = linspace(rl, ru, 1000);
%oszacowanie zbioru wartosci funkcji w
przedziale [-9,8]
MIN = min(w_int(num));
MAX = max(w_int(num));
%y nalezy do (MIN, MAX) oraz wiemy ze MIN < 0
i MAX > 0 zatem pole:
Omega = (abs(ru) + abs(rl))*(abs(MIN) +
abs(MAX));

for k = 1:length(N)

%losowanie punktów
x_rand = rand(1,N(k)).*(abs(ru) + abs(rl)) +
rl;
y_rand = rand(N(k),1).*(abs(MIN) + abs(MAX)) +
MIN;

%wybor punktow
x_eq = linspace(-9, 8, N(k));
y_eq = (linspace(MIN, MAX, N(k))).';

%sprawdzenie czy punkty należą do pola
ograniczonego wykresem
for i = 1:N(k)
    for j = 1:N(k)
        if (abs(w_int(x_rand(1,i))) >
abs(y_rand(j,1)))
            c_rand=c_rand+1;
        end

        if (abs(w_int(x_eq(1,i))) >
abs(y_eq(j,1)))
            c_eq=c_eq+1;
        end
    end
end

%oszacowanie wartosci calki
estimate_rand(k) = (c_rand/(N(k)^2))*Omega;
```

```
delta_abs_random(k) = abs(p_int-
estimate_rand(k));
estimate_eq(k) = (c_eq/(N(k)^2))*Omega;
delta_abs_equal(k) = abs(p_int-
estimate_eq(k));
c_eq = 0;
c_rand = 0;

end

%wykonanie wykresu
figure(2)
loglog(N, delta_abs_random, 'r-x');
hold on;
loglog(N, delta_abs_equal, 'g-*');
hold off;
title('Monte-Carlo - moduł błędu');
xlabel('Ilość punktów')
ylabel('Wartość błędu bezwględnego');
legend('rozkład losowy', 'rozkład
równomierny', 'Location', 'NorthEast');

figure(3)
semilogx(N, estimate_rand, 'r-x');
hold on;
semilogx(N, estimate_eq, 'g-*');
hold off;
title('Monte-Carlo - estymata pola');
xlabel('Ilość punktów')
ylabel('Estymata pola F');
legend('rozkład losowy', 'rozkład
równomierny', 'Location', 'NorthEast');

%---------------------------------------------
----------------------------
%--------------------------------FUNKCJE-------
----------------------------
%---------------------------------------------
----------------------------

function [wynik] = kwadratura(coeff, a, b, N)
H=(b-a)/N;
if (N==2) %kwadratura Simpsona
    wynik = 1/6*polyval(coeff,a) +
4/6*polyval(coeff,a+H) + 1/6*polyval(coeff,b);
end
if (N==3) %kwadratura trzech ósmych
    wynik =1/8*polyval(coeff,a) +
3/8*polyval(coeff,a+H) +
3/8*polyval(coeff,a+2*H) +
1/8*polyval(coeff,b);
end
if (N==4) %kwadratura Milne'a
    wynik =7/90*polyval(coeff,a) +
32/90*polyval(coeff,a+H) +
12/90*polyval(coeff,a+2*H) +
32/90*polyval(coeff,a+3*H) +
7/90*polyval(coeff,b);
end
if (N==5) %kwadratura Bode'a
    wynik =19/288*polyval(coeff,a) +
75/288*polyval(coeff,a+H) +
50/288*polyval(coeff,a+2*H) +
50/288*polyval(coeff,a+3*H) +
75/288*polyval(coeff,a+4*H) +
19/288*polyval(coeff,b);
end
if (N==6) %kwadratura Weddle'a
    wynik =41/840*polyval(coeff,a) +
216/840*polyval(coeff,a+H) +
27/840*polyval(coeff,a+2*H) +
272/840*polyval(coeff,a+3*H) +
27/840*polyval(coeff,a+4*H) +
216/840*polyval(coeff,a+5*H) +
41/840*polyval(coeff,b);
end
wynik=wynik*(b-a);
end
```