SPRAWOZDANIE		
Laboratoria metod numerycznych i symulacji		
Temat: Różniczkowanie i całkowanie Pierwiastki równania nieliniowego	Numer: Lab 3, 4	Data: 04.04.2022
Wykonał: Roman Nawrot		

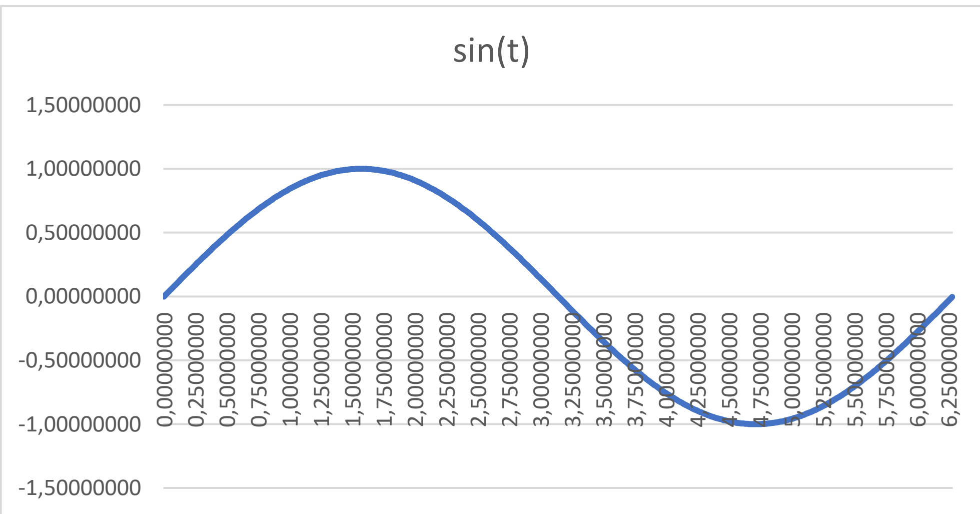
1. Dyskretyzacja sygnałów

a) W języku C++ dokonano dyskretyzacji sygnału sin(t) dla $\Delta t = 0.01$ na przedziale $t \in < 0; 2\Pi >$. Za liczbę Π przyjęto 3.14159

b) Kod źródłowy

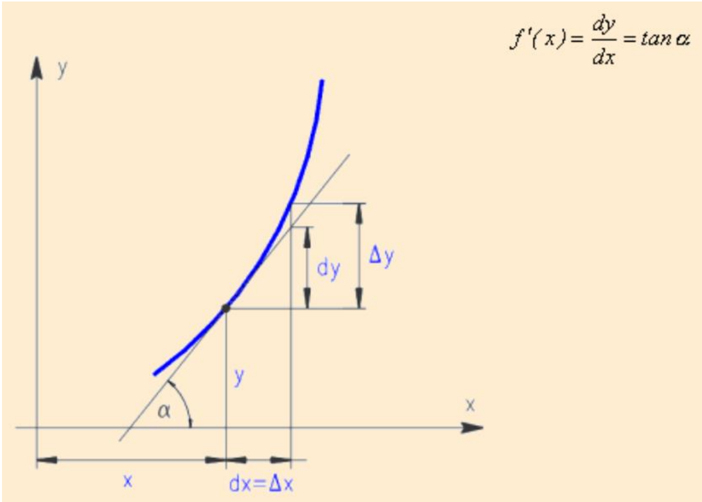
```
 double pi = 3.14159;
float delta_t = 0.01;  // odstep miedzy kolejnymi probkami

// ======== [ DYSKRETYZACJA ] =========

double x = 0;  // aktualna wartosc probki
int iter = 0;  // numer kroku

vector <long double> y;

while (x <= (2*pi))
{
    y.push_back(sin(x));

    cout << "wartosc sin dla x = " << x << " to: " << y[iter] << endl;

    x += delta_t;
    iter++;
}
```

c) Wykres



2. Obliczenie pochodnej

a) Obliczenie pochodnej sin(t) odbywa się za pomocą obliczenia tangensa kąta z wartości sin(t) / sin(Δt +t)



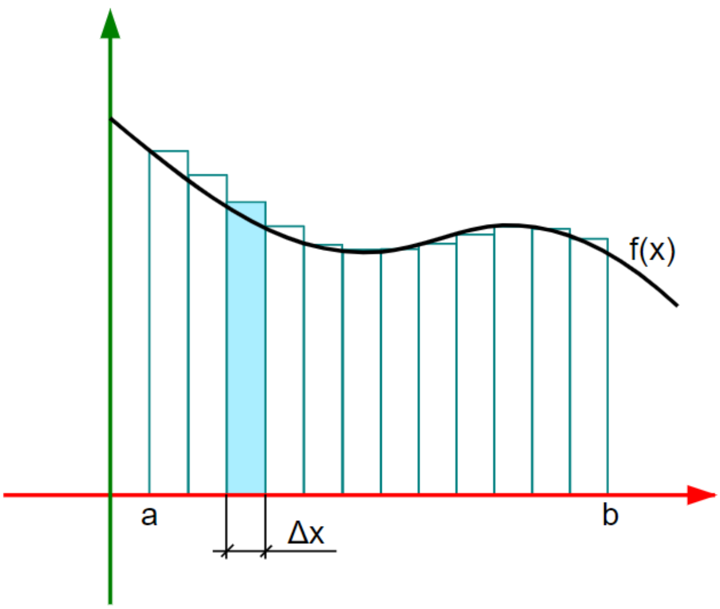
b) Kod źródłowy

```
// ======== [ POCHODNA ] =========

vector <double> pochodna;

x = 0;
iter = 0;

while (x <= 2*pi)
{

    pochodna.push_back((sin(x + delta_t) - sin(x)) / delta_t);

    cout << "pochodna w punkcie " << x << " wynosi: " << pochodna[iter] << endl;

    x += delta_t;
    iter++;

}
```

3. Obliczanie całki $\int_0^{\Pi} x\,dx$ dla Δ t ∈ {0.1, 0.01, 0.001, 0, 0001}

a) Metoda prostokątów

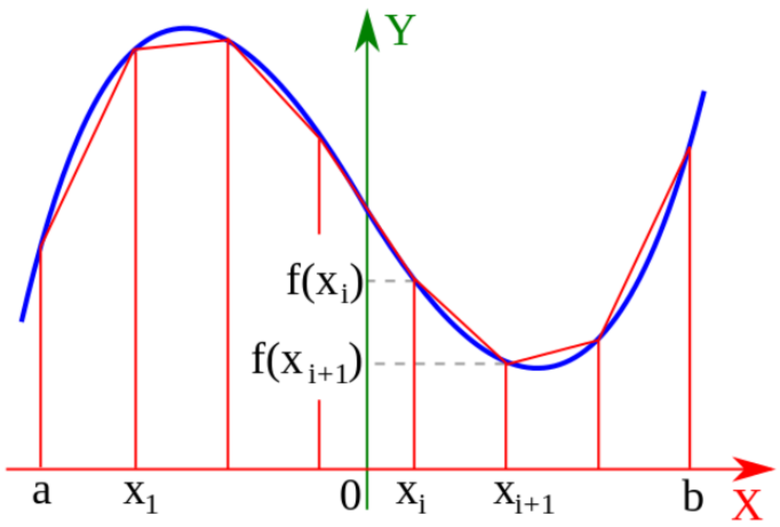
i. Wizualny opis działania



ii. Kod źródłowy

```
delta_t = 0.1;

// METODA PROSTOKATOW

double calka = 0;
x = 0;

while (x <= pi)
{

      calka += delta_t * x;

      x += delta_t;

}

cout << "[metoda prostokatow] wartosc calki: " << calka << endl;
```

b) Metoda trapezów
   i. Wizualny opis działania

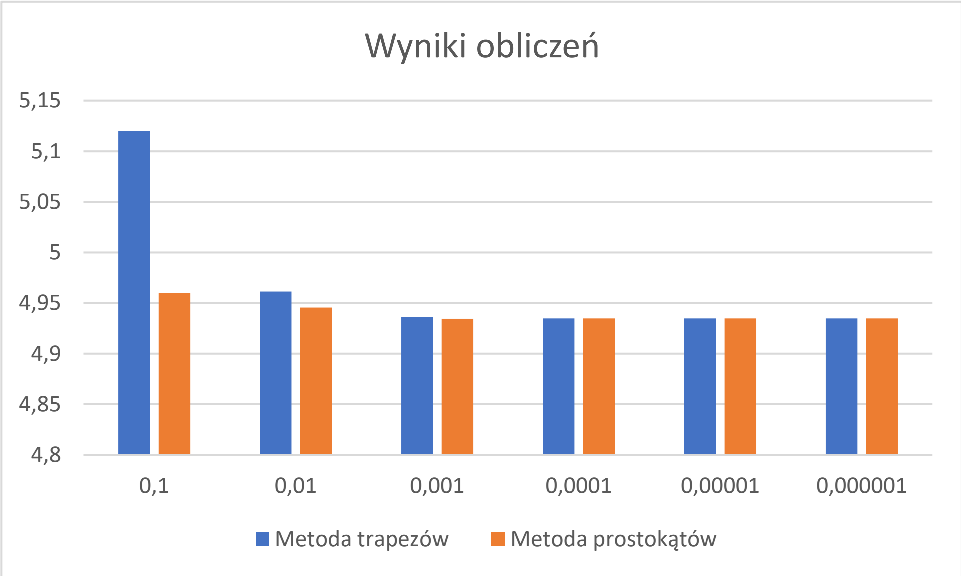


ii. Kod źródłowy

```
// METODA TRAPEZOW

calka = 0;
x = 0;

while (x <= pi)
{

    calka += 0.5 * (x + (x + delta_t)) * delta_t; // 1/2 * (a+b) * h

    x += delta_t;

}

cout << "[metoda trapezow] wartosc calki: " << calka << endl;
```

c) Zestawienie wyników dla Δ t ∈ {0.1, 0.01, 0.001, 0, 0001, 0,00001, 0,000001}

Δ t	Metoda trapezów	Metoda prostokątów
0,1	5,12	4,96
0,01	4,96125	4,9455
0,001	4,93608	4,93451
0,0001	4,93483	4,93467
0,00001	4,93483	4,93481
0,000001	4,9348	4,9348

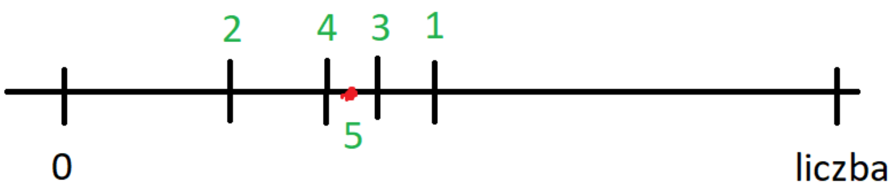


d) Konkluzja
Dla Δ t = 0,1 metoda trapezów jest w stosunku do metody prostokątów niedokładna, natomiast dla Δ t = 0,01 obie metody zrównują się wynikami i zwiększają swoje podobieństwo wraz ze wzrostem częstotliwości próbkowania (czyli zmniejszania Δ t).

4. Obliczanie pierwiastka kwadratowego z dowolnej liczby
a) Za pomocą biblioteki math.h

```
// MATH

pierwiastek = sqrt(liczba);

cout << "[MATH] Pierwiastek liczby " << liczba << " jest rowny: " << pierwiastek <<
endl;
```

b) Za pomocą ,,Zgadywania''
i. Wizualny opis działania



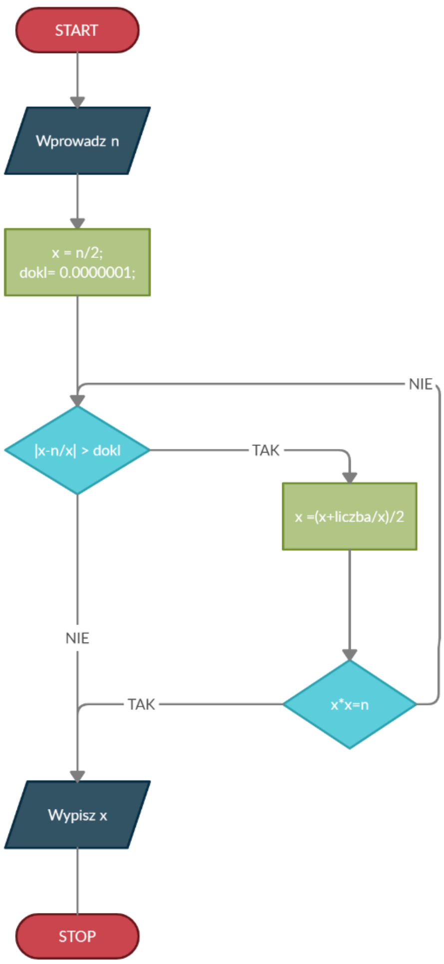
ii. Metoda polega na dzieleniu liczby przez 2 do momentu, aż wartość bezwzględna z różnicy dzielnika podniesionego do kwadratu i liczby będzie mniejsza niż zdefiniowany maksymalny błąd obliczeń eps.

$|x - y| \leq e$

iii. Kod źródłowy

```
// ZGADYWANIE

long double eps = 0.1;

long double granica_l = 0;
long double granica_p = liczba;

long double srodek = liczba / 2;

long double obliczona_liczba = srodek * srodek;

while (abs(obliczona_liczba - liczba) > eps)
{
   if (obliczona_liczba > liczba)
   {
         granica_p = srodek;
   }
   else if (obliczona_liczba < liczba)
   {
         granica_l = srodek;
   }
   srodek = granica_l + ((granica_p - granica_l) / 2);
   obliczona_liczba = srodek * srodek;

}

cout << "[ZGADYWANIE] Pierwiastek liczby " << liczba << " jest rowny: " <<
srodek << endl;
```

c) Za pomocą metody Newtona
   i. Wizualny opis działania

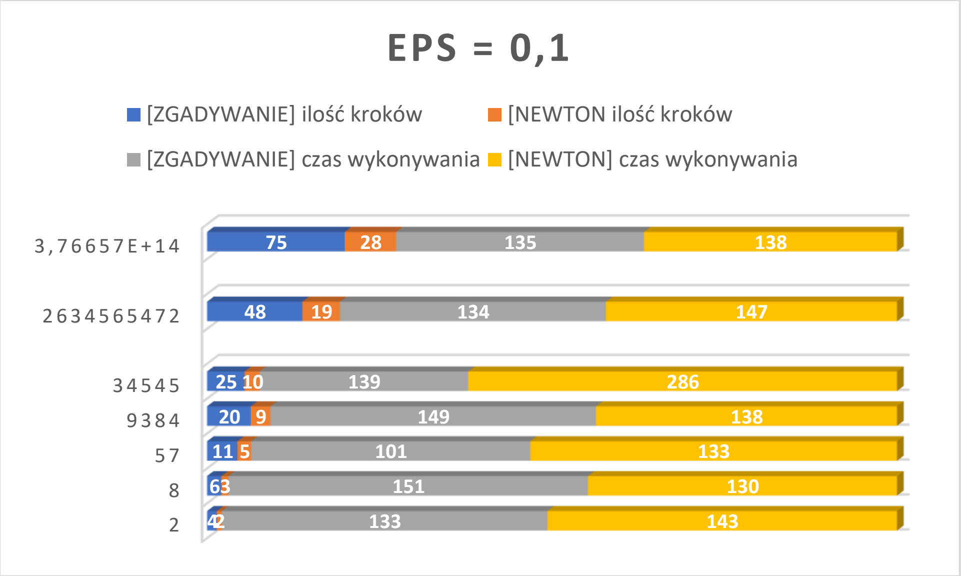


ii. Kod źródłowy

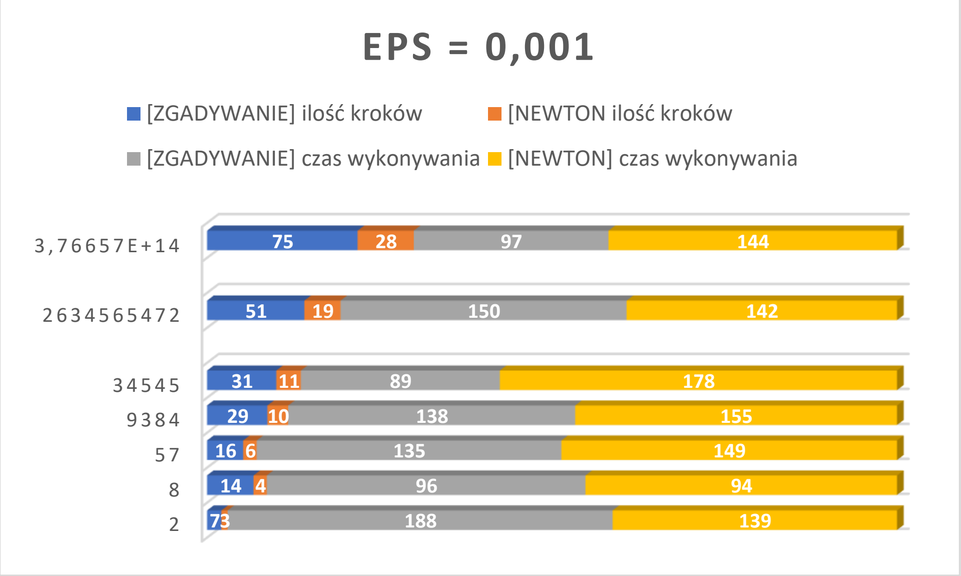
```
// METODA NEWTONA

double a = 1;
double b = liczba;

while (abs(a - b) >= eps)
{

    a = (a + b) / 2.;
    b = liczba / a;

}

cout << "[NEWTON] Pierwiastek liczby " << liczba << " jest rowny: " << a <<
endl;
```

d) Porównanie ilości kroków i czas wykonywania obliczeń dla różnych metod, dokładności oraz wartości do policzenia

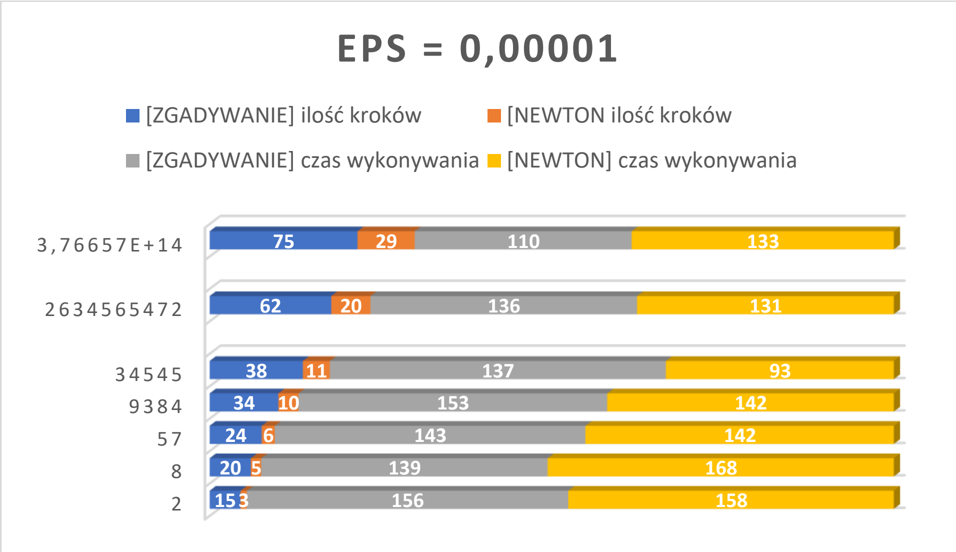
i. Dla eps = 0,1



ii. Dla eps = 0,001



iii. Dla eps = 0,00001



e) Wnioski

Badanie wykazało, że Metoda Newtona w obliczaniu pierwiastków wymaga dużo mniej kroków niż metoda na ,,zgadywanie''.
Natomiast nie udało się określić przewagi czasowej jednej metody nad drugą.
Podsumowując: Metoda Newtona to dużo mniej kodu, dużo mniej kroków do wykonania oraz prawdopodobny krótszy czas wykonania.