Wydział Inżynierii Metali i Informatyki Przemysłowej

<table>
<tr><td>Imię i nazwisko
Meg Paskowski</td><td rowspan="2">Temat laboratorium
Rozwiązywanie równań różniczkowych</td><td>Data oddania
2.06.2024r.</td><td>Data ćwiczeń
28.05.2024r.</td></tr>
<tr><td>Prowadzący
dr hab. inż. Marcin Hojny</td><td colspan="2">Grupa laboratoryjna
4</td></tr>
</table>

1. Cel ćwiczenia

Celem laboratorium nr. 11 było zapoznanie się z pojęciem rozwiązywania równań różniczkowych metodą Euliera, Heuna i klasyczną RK4 oraz zaimplementowanie tych algorytmów w wybranym języku programowania.

2. Wstęp teoretyczny

Rozwiązywanie równań różniczkowych jest fundamentalnym zagadnieniem w matematyce i naukach przyrodniczych. Wiele zjawisk fizycznych, chemicznych, biologicznych i inżynierskich można opisać za pomocą równań różniczkowych. Jednak znalezienie dokładnego, analitycznego rozwiązania tych równań często nie jest możliwe. W takich przypadkach stosuje się metody numeryczne, które pozwalają na przybliżone rozwiązanie problemu.

Problem początkowy dla równania różniczkowego rzędu pierwszego.

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0 \tag{1}$$

Gdzie:

- f(x, y) jest daną funkcją,
- (x_0, y_0) są warunkami początkowymi.

Ponieważ znalezienie dokładnego rozwiązania analitycznego często nie jest możliwe, stosuje się metody numeryczne, które pozwalają na przybliżone rozwiązanie problemu początkowego. Dwie popularne metody to metoda Eulera i metoda Heuna.

Metoda Eulera → jest jedną z najprostszych metod numerycznych do rozwiązywania równań różniczkowych. Opiera się na aproksymacji pochodnej za pomocą ilorazu różnicowego. Dla małego kroku h, wartość y(x + h) można przybliżyć.

$$\mathbf{y(x + h) \approx y(x) + h \cdot f(x, y(x))} \tag{2}$$

Gdzie:

- y(x) - przybliżona wartość rozwiązania w punkcie x
- h - krok całkowania (przyrost zmiennej niezależnej x)
- f(x, y(x)) - wartość funkcji prawej strony równania różniczkowego y' = f(x, y) w punkcie (x, y(x)).

Iteracyjnie stosując ten wzór, można uzyskać przybliżone rozwiązanie problemu początkowego. Metoda Eulera jest prosta w implementacji. Ma błąd rzędu $O(h^2)$, co oznacza, że jej dokładność jest ograniczona.

Metoda Heuna → jest ulepszeniem metody Eulera i ma wyższą dokładność. Zamiast aproksymować pochodną w punkcie (x, y(x)), metoda Heuna uwzględnia również nachylenie funkcji w punkcie (bierze średnią z dwóch nachyleń funkcji).

$$\mathbf{f(x + h, y(x) + h \cdot f(x, y(x)))} \tag{3}$$

Gdzie:

- x + h jest kolejną wartością zmiennej niezależnej x po dodaniu kroku h
- y(x) + h * f(x, y(x)) jest przybliżeniem wartości y(x + h) obliczonym metodą Eulera.

Przybliżenie y(x + h):

$$y(x + h) \approx y(x) + \left(\frac{h}{2}\right) \cdot (f(x, y(x)) + f(x + h, y(x) + h \cdot f(x, y(x)))) \tag{4}$$

Gdzie:

- y(x) - przybliżona wartość rozwiązania w punkcie x
- h - krok całkowania (przyrost zmiennej niezależnej x)
- (x, y(x)) - wartość f w punkcie (x, y(x))
- f(x + h, y(x) + h * f(x, y(x))) - wartość f w punkcie (x + h, y(x) + h * f(x, y(x))), gdzie y(x) + h * f(x, y(x)) jest przybliżeniem Eulera

Zarówno metoda Eulera, jak i metoda Heuna wymagają iteracyjnego obliczania kolejnych przybliżeń. Liczba kroków N potrzebna do znalezienia rozwiązania na przedziale $[x_0, b]$

$$N=\frac{(b-x_0)}{h} \tag{5}$$

Gdzie:

- h to krok całkowania.
- x_0 to końcowa wartość zmiennej x
- b jest początkową wartością zmiennej x

Metoda RK4 (Runge-Kutta 4. Rzędu) → jest jedną z najbardziej popularnych metod numerycznych do rozwiązywania równań różniczkowych. Jest znacznie dokładniejsza niż metody Eulera i Heuna i ma błąd rzędu $O(h^5)$. Metoda ta wykorzystuje cztery pośrednie obliczenia nachyleń, aby osiągnąć wysoką dokładność.

Algorytm RK4 dla danego kroku h można przedstawić w następujący sposób:

Obliczanie nachylenia na początku przedziału.

$$k_1=h\cdot f(x_n,y_n) \tag{6}$$

Obliczenie nachylenia w środku przedziału, na podstawie (6).

$$k_2=h\cdot f\left(x_n+\frac{h}{2},y_n+\frac{k_1}{2}\right) \tag{7}$$

Obliczanie nachylenia w środku przedziału na podstawie (7).

$$k_3=h\cdot f\left(x_n+\frac{h}{2},y_n+\frac{k_2}{2}\right) \tag{8}$$

Obliczanie nachylenia na końcu przedziału, na podstawie (8).

$$k_4=h\cdot f(x_n+h,y_n+k_3) \tag{9}$$

Przybliżone rozwiązania y w punkcie x_n + h jest obliczane za pomocą wzoru (10) i opiera się na poszczególnych wynikach z wzorów (6-9).

$$y_{n+1}=y_n+\frac{1}{6}(k_1+2k_2+2k_3+k_4) \tag{10}$$

Gdzie:

- y_n jest przybliżoną wartością rozwiązania w punkcie x_n.
- h to krok całkowania
- k_1,k_2,k_3,k_4 to pośrednie obliczenia nachyleń (6-9).

Metoda RK4, pomimo swojej złożoności, jest bardzo efektywna i szeroko stosowana w praktyce ze względu na swoją wysoką dokładność i stabilność.

3. Implementacja

W ramach ćwiczeń zaimplementowano program w języku C++, implementuje trzy metody numeryczne do rozwiązywania równań różniczkowych: metodę Eulera, metodę Heuna i klasyczną RK4.

Funkcja *„differential_equation()”* jest wykorzystywana do obliczania wartości pochodnej dla podanej funkcji.

```
double differential_equation(double x, double y) {
    return 3 * x;
}
```

Funkcja *„analytical_solution()*” reprezentuje analityczne rozwiązanie równania różniczkowego. Przyjmuje wartość „x” dla którego szukana jest wartość.

```
// Analytical solution
double analytical_solution(double x) {
    return (1.5 * pow(x, 2)) - 8.5;
}
```

Funkcja *„euler_solution()”* implementuje metodę Eulera do rozwiązywania równań różniczkowych. Przyjmuje cztery argumenty: *„initial_x*” (początkowa wartość zmiennej niezależnej x), *„final_x*” (końcowa wartość x), *„initial_y*” (początkowa wartość zmiennej zależnej y) oraz *„step*” (krok całkowania). Funkcja na sam początek oblicza liczbę kroków *„num_steps*” potrzebnych do przejścia z *„initial_x*” do *„final_x*” z danym krokiem *„step*”. Następnie przy wykorzystaniu pętli „for’ obliczane są kolejne wartości *„y*” za pomocą wzoru (2). Wynik odpowiednich kroków jest wyświetlany w konsoli.

```
// Euler's method
double euler_solution(double initial_x, double final_x, double initial_y, double step) {
    int num_steps = (final_x - initial_x) / step;
    double x = initial_x;
    double y = initial_y;

    cout << "Euler's method:" << endl;

    for (int i = 0; i < num_steps; i++) {
        y += step * differential_equation(x, y);

        cout << "\tY_" << i << " solution: " << y << endl;
        x += step;
    }

    return y;
}
```

Funkcja *„heun_solution()”* implementuje metodę Heuna do rozwiązywania równań różniczkowych. Przyjmuje te same argumenty co funkcja *„euler_solution()”.* Funkcja oblicza liczbę kroków *„num_steps*” w taki sam sposób jak w metodzie Eulera. Następnie kolejne wartości nachylenia „k_1” i „k_2” (3) oraz *„y*” za pomocą wzoru metody Heuna (4). Wynik stopniowych kroków także wyświetlany jest w konsoli.

```
double heun_solution(double initial_x, double final_x, double initial_y, double step) {
    int num_steps = (final_x - initial_x) / step;
    double x = initial_x;
    double y = initial_y;

    cout << "Heun's method:" << endl;

    for (int i = 0; i < num_steps; i++) {
        double k1 = differential_equation(x, y);
        double k2 = differential_equation(x + step, y + step * k1);

        y += step / 2.0 * (k1 + k2);
        cout << "\tY_" << i << " solution: " << y << endl;
        x += step;
    }

    return y;
}
```

Kolejną funkcja implementuje metodę klasyczną RK4, *„RK4_solution()”.* Na początku funkcji definiowane są parametry wejściowe – te same co w poprzedzających metodach. Następnie obliczana jest liczba kroków potrzebnych do przejścia od „*initial_x*” do „*final_x*” przy podanym krok „*h*”. Wykorzystana pętla „*for*” iteruje przez „*num_steps*” liczbę kroków. Obliczane są po kolei nachylenia (6-9) oraz do wartości „*y*” zapisywany jest wynik (10). Na koniec poprzez wydruk jest wyświetlany wynik odpowiedniego kroku.

```
//RK4's method
double RK4_solution(double initial_x, double final_x, double initial_y, double step) {
    int num_steps = (final_x - initial_x) / step;
    double x = initial_x;
    double y = initial_y;

    cout << "RK4's method:" << endl;

    for (int i = 0; i < num_steps; i++) {
        double k1 = step * differential_equation(x, y);
        double k2 = step * differential_equation(x + 0.5 * step, y + 0.5 * k1);
        double k3 = step * differential_equation(x + 0.5 * step, y + 0.5 * k2);
        double k4 = step * differential_equation(x + step, y + k3);

        y += (k1 + 2 * k2 + 2 * k3 + k4) / 6.0;
        cout << "\tY_" << i << " solution: " << y << endl;
        x += step;
    }

    return y;
}
```

Funkcja główna „*main()*” jest punktem wejścia programu. W tym przypadku, program rozwiązuje równanie różniczkowe z warunkiem początkowym y z podanym krokiem, wykorzystując zarówno metodę Eulera, jak i metodę Heuna. Wyniki są wyświetlane na konsoli z poprzedzającym oznaczeniem metody. Na sam konieć obliczany jest wynik również metodą analityczną.

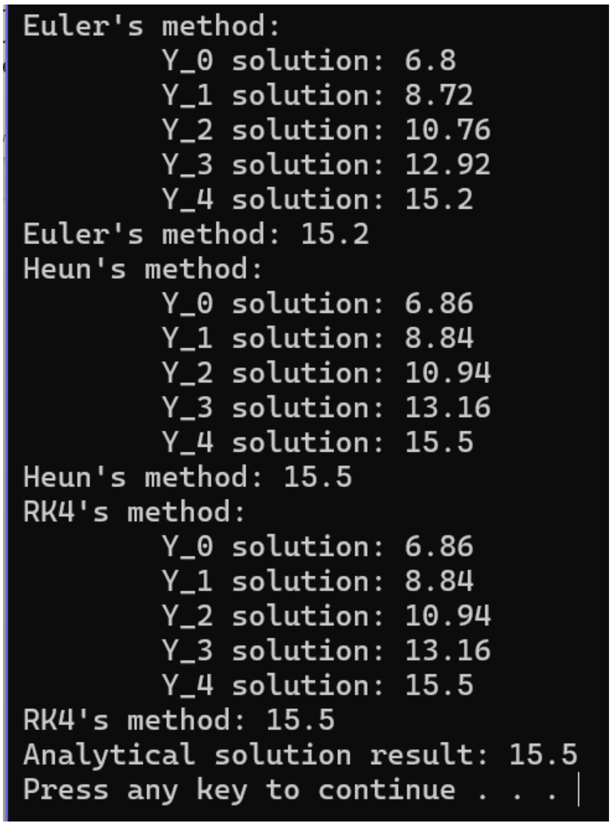
```
// Solving differential equations
    cout << "Euler's method:" << euler_solution(3, 4, 5, 0.2) << endl;
    cout << "Heun's method:" << heun_solution(3, 4, 5, 0.2) << endl;
    cout << "Analytical solution result: " << analytical_solution(4) << endl;

    system("PAUSE");
    return 0;
```

5. Testy jednostkowe i opracowanie wyników

Testy zostały przeprowadzone dla różnych funkcji z określonym krokiem czasowym oraz warunkiem początkowym, a następnie porównane zostały do rozwiązania analitycznego.

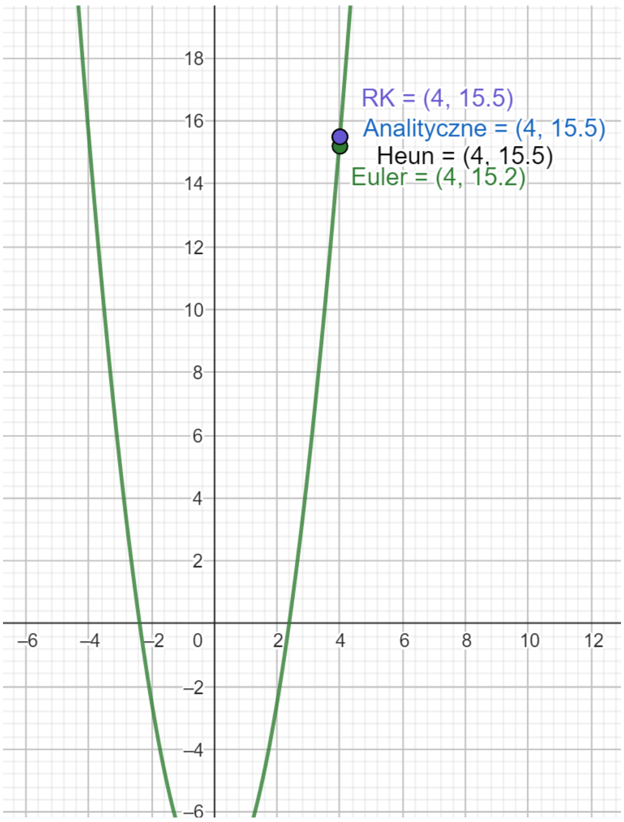
I. Test 1 dla funkcji f(x,y)=3*x z warunkiem początkowym y(3) = 5
Wartość b = 4
Krok czasowy h = 0.2

Wynik metodą Eulera = 15.2
Wynik metodą Heuna = 15.5
Wynik metodą klasyczną RK4 = 15.5
Wynik dla rozwiązania analitycznego = 15.5



```
Euler's method:
        Y_0 solution: 6.8
        Y_1 solution: 8.72
        Y_2 solution: 10.76
        Y_3 solution: 12.92
        Y_4 solution: 15.2
Euler's method: 15.2
Heun's method:
        Y_0 solution: 6.86
        Y_1 solution: 8.84
        Y_2 solution: 10.94
        Y_3 solution: 13.16
        Y_4 solution: 15.5
Heun's method: 15.5
RK4's method:
        Y_0 solution: 6.86
        Y_1 solution: 8.84
        Y_2 solution: 10.94
        Y_3 solution: 13.16
        Y_4 solution: 15.5
RK4's method: 15.5
Analytical solution result: 15.5
Press any key to continue . . .
```

Rysunek 1 Wyniki dla testu I



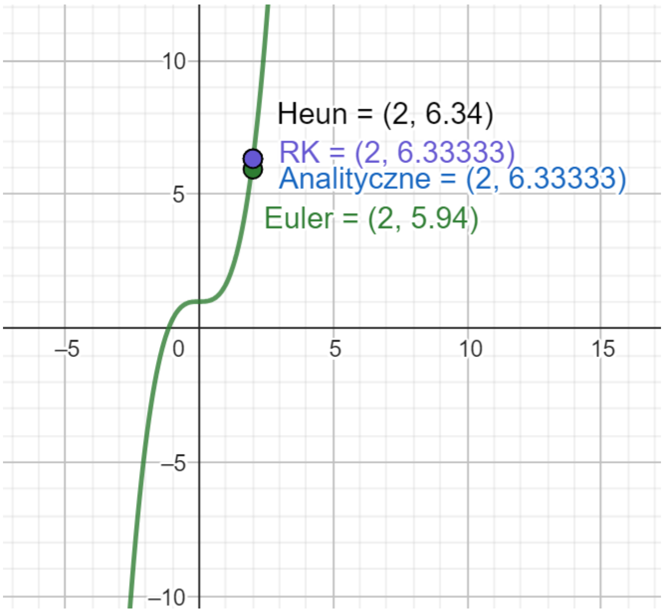
Rysunek 2 Przedstawienie wyników dla testu I "Geogebra.com"

II. Test 2 dla funkcji y'(x)=2*x^2 z warunkiem początkowym y(0) = 1
Wartość b = 2
Krok czasowy h = 0.1

Wynik metodą Eulera = 5.94
Wynik metodą Heuna = 6.34
Wyniki metodą klasyczną RK4 = 6.33333
Wynik dla rozwiązania analitycznego = 6.33333

```
Euler's method:
        Y_0 solution: 1
        Y_1 solution: 1.002
        Y_2 solution: 1.01
        Y_3 solution: 1.028
        Y_4 solution: 1.06
        Y_5 solution: 1.11
        Y_6 solution: 1.182
        Y_7 solution: 1.28
        Y_8 solution: 1.408
        Y_9 solution: 1.57
        Y_10 solution: 1.77
        Y_11 solution: 2.012
        Y_12 solution: 2.3
        Y_13 solution: 2.638
        Y_14 solution: 3.03
        Y_15 solution: 3.48
        Y_16 solution: 3.992
        Y_17 solution: 4.57
        Y_18 solution: 5.218
        Y_19 solution: 5.94
Euler's method: 5.94
Heun's method:
        Y_0 solution: 1.001
        Y_1 solution: 1.006
        Y_2 solution: 1.019
        Y_3 solution: 1.044
        Y_4 solution: 1.085
        Y_5 solution: 1.146
        Y_6 solution: 1.231
        Y_7 solution: 1.344
        Y_8 solution: 1.489
        Y_9 solution: 1.67
        Y_10 solution: 1.891
        Y_11 solution: 2.156
        Y_12 solution: 2.469
        Y_13 solution: 2.834
        Y_14 solution: 3.255
        Y_15 solution: 3.736
        Y_16 solution: 4.281
        Y_17 solution: 4.894
        Y_18 solution: 5.579
        Y_19 solution: 6.34
Heun's method: 6.34
RK4's method:
        Y_0 solution: 1.00067
        Y_1 solution: 1.00533
        Y_2 solution: 1.018
        Y_3 solution: 1.04267
        Y_4 solution: 1.08333
        Y_5 solution: 1.144
        Y_6 solution: 1.22867
        Y_7 solution: 1.34133
        Y_8 solution: 1.486
        Y_9 solution: 1.66667
        Y_10 solution: 1.88733
        Y_11 solution: 2.152
        Y_12 solution: 2.46467
        Y_13 solution: 2.82933
        Y_14 solution: 3.25
        Y_15 solution: 3.73067
        Y_16 solution: 4.27533
        Y_17 solution: 4.888
        Y_18 solution: 5.57267
        Y_19 solution: 6.33333
RK4's method: 6.33333
Analytical solution result: 6.33333
Press any key to continue . . .
```

Rysunek 3 Wyniki dla testu II



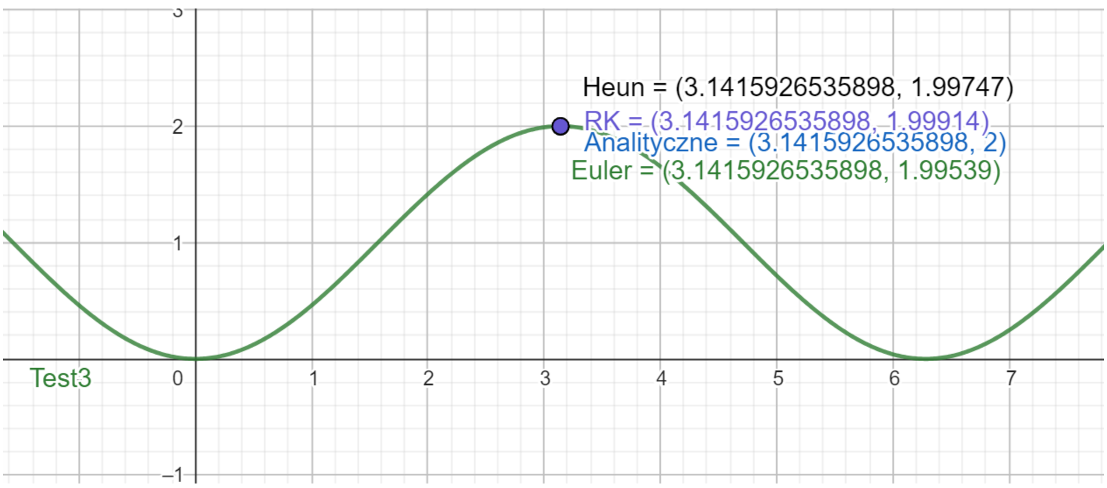
Rysunek 4 Przedstawienie wyników dla testu II "Geogebra.com"

III. Test 3 dla funkcji y'(x)=sinx z warunkiem początkowym y(0) = 0
Wartość b = π
Krok czasowy h = 0.1

Wynik metodą Eulera = 1.99539
Wynik metodą Heuna = 1.99747
Wyniki metodą klasyczną RK4 = 1.99914
Wynik dla rozwiązania analitycznego = 2

```
Euler's method:
        Y_0 solution: 0
        Y_1 solution: 0.00998334
        Y_2 solution: 0.0298503
        Y_3 solution: 0.0594023
        Y_4 solution: 0.0983441
        Y_5 solution: 0.146287
        Y_6 solution: 0.202751
        Y_7 solution: 0.267173
        Y_8 solution: 0.338908
        Y_9 solution: 0.417241
        Y_10 solution: 0.501388
        Y_11 solution: 0.590509
        Y_12 solution: 0.683713
        Y_13 solution: 0.780069
        Y_14 solution: 0.878614
        Y_15 solution: 0.978363
        Y_16 solution: 1.07832
        Y_17 solution: 1.17749
        Y_18 solution: 1.27487
        Y_19 solution: 1.3695
        Y_20 solution: 1.46043
        Y_21 solution: 1.54675
        Y_22 solution: 1.6276
        Y_23 solution: 1.70217
        Y_24 solution: 1.76972
        Y_25 solution: 1.82957
        Y_26 solution: 1.88112
        Y_27 solution: 1.92385
        Y_28 solution: 1.95735
        Y_29 solution: 1.98128
        Y_30 solution: 1.99539
Euler's method: 1.99539
Heun's method:
        Y_0 solution: 0.00499167
        Y_1 solution: 0.0199168
        Y_2 solution: 0.0446263
        Y_3 solution: 0.0788732
        Y_4 solution: 0.122315
        Y_5 solution: 0.174519
        Y_6 solution: 0.234962
        Y_7 solution: 0.303041
        Y_8 solution: 0.378075
        Y_9 solution: 0.459315
        Y_10 solution: 0.545948
        Y_11 solution: 0.637111
        Y_12 solution: 0.731891
        Y_13 solution: 0.829341
        Y_14 solution: 0.928488
        Y_15 solution: 1.02834
        Y_16 solution: 1.1279
        Y_17 solution: 1.22618
        Y_18 solution: 1.32219
        Y_14 solution: 0.928488
        Y_15 solution: 1.02834
        Y_16 solution: 1.1279
        Y_17 solution: 1.22618
        Y_18 solution: 1.32219
        Y_19 solution: 1.41497
        Y_20 solution: 1.50359
        Y_21 solution: 1.58718
        Y_22 solution: 1.66489
        Y_23 solution: 1.73595
        Y_24 solution: 1.79964
        Y_25 solution: 1.85534
        Y_26 solution: 1.90249
        Y_27 solution: 1.9406
        Y_28 solution: 1.96932
        Y_29 solution: 1.98833
        Y_30 solution: 1.99747
Heun's method: 1.99747
RK4's method:
        Y_0 solution: 0.00499583
        Y_1 solution: 0.0199334
        Y_2 solution: 0.0446635
        Y_3 solution: 0.078939
        Y_4 solution: 0.122417
        Y_5 solution: 0.174664
        Y_6 solution: 0.235158
        Y_7 solution: 0.303293
        Y_8 solution: 0.37839
        Y_9 solution: 0.459698
        Y_10 solution: 0.546404
        Y_11 solution: 0.637642
        Y_12 solution: 0.732501
        Y_13 solution: 0.830033
        Y_14 solution: 0.929263
        Y_15 solution: 1.0292
        Y_16 solution: 1.12884
        Y_17 solution: 1.2272
        Y_18 solution: 1.32329
        Y_19 solution: 1.41615
        Y_20 solution: 1.50485
        Y_21 solution: 1.5885
        Y_22 solution: 1.66628
        Y_23 solution: 1.73739
        Y_24 solution: 1.80114
        Y_25 solution: 1.85689
        Y_26 solution: 1.90407
        Y_27 solution: 1.94222
        Y_28 solution: 1.97096
        Y_29 solution: 1.98999
        Y_30 solution: 1.99914
RK4's method: 1.99914
Analytical solution result: 2
Press any key to continue . . .
```

Rysunek 5 Wyniki dla testu III



Rysunek 6 Przedstawienie wyników dla testu III „Geogebra.com”

6. Analiza wyników

Uzyskane wyniki w testach (I-III) sugerują poprawną implementacje omawianych metod. W tabeli (1) umieszczone jest zestawienie uzyskanych wyników.

Tabela 1 Uzyskane wyniki dla testów I-IV

Numer Testu	Funkcja	Wynik metoda Eulera	Wynik metoda Heuna	Wyniki metoda klasyczna RK4	Wynik dla rozwiązania analitycznego
I	3x	15.2	15.5	15.5	15.5
II	$2x^2$	5.94	6.34	6.33333	6.33333
III	sinx	1.99539	1.99747	1.99914	2

Opracowanie wyników:

- Test funkcji y'=3x

Wynik uzyskany metodą Eulera wynosi 15.2, co jest nieco mniejsze od wyniku analitycznego 15.5. Metoda Heuna daje wynik dokładnie równy wynikowi analitycznemu (15.5), podobnie jak metoda RK4. To pokazuje, że zarówno metoda Heuna, jak i metoda RK4 są bardziej dokładne od metody Eulera, szczególnie przy prostych funkcjach liniowych.

- Test funkcji $y'=2x^2$

Wynik uzyskany metodą Eulera 5.94 jest mniejszy niż wynik analityczny 6.33333, podczas gdy metoda Heuna daje wynik 6.34, który jest bardzo bliski wynikowi analitycznemu. Metoda RK4 daje wynik 6.33333, który jest dokładnie równy wynikowi analitycznemu. Ponownie, zarówno metoda Heuna, jak i metoda RK4 wykazują wyższą dokładność niż metoda Eulera.

- Test funkcji y'=sinx

Dla tej funkcji wyniki wszystkich trzech metod są bardzo bliskie wynikowi analitycznemu wynoszącemu 2. Metoda Eulera daje wynik 1.99955, metoda Heuna 1.99663, a metoda RK4 1.99914. Wszystkie metody dają przybliżone wyniki z bardzo małym błędem, Dla funkcji trygonometrycznych metoda RK4 jest najdokładniejsza, jednak metoda Eulera i metoda Heuna również mogą być wystarczająco dokładne przy odpowiednio małym kroku.

7. Wnioski

W testach potwierdzono, że metoda Heuna zazwyczaj zapewnia lepszą dokładność niż metoda Eulera dzięki uwzględnieniu średniej nachyleń funkcji w dwóch puntach. Ważny przy tych metodach jest również odpowiednio dobrany krok czasowy. Dla prostych funkcji liniowych i trygonometrycznych obie metody mogą dawać zadowalające wyniki, ale dla funkcji wykładniczych, kwadratowych, czy trygonometrycznych różnica w dokładności jest bardziej zauważalna. Metoda RK4 jest szczególnie skuteczna dla funkcji o bardziej złożonych kształtach, takich jak funkcje wykładnicze i kwadratowe, trygonometryczne co widać po przeprowadzonym teście III. Porównując wszystkie użyte metody, RK4 oferuje najwyższą dokładność numeryczną spośród omawianych, jednak jest też bardziej złożona obliczeniowo.

8. Źródła

- Prezentacja autorstwa dr hab. inż. Marcina Hojnego „Rozwiązywanie równań różniczkowych (cz. 1) – metoda Eulera".
- Prezentacja autorstwa dr hab. inż. Marcina Hojnego „Rozwiązywanie równań różniczkowych (cz. 2) – metoda Heuna".