Szymon Ociepka *Data wykonania: 22.05.2024*

Metoda Heuna & RK4

Rozwiązywanie równań różniczkowych

Metody numeryczne

1. Wprowadzenie teoretyczne

1.1 metoda Heuna

Metoda Heuna, znana również jako metoda trapezów, to jedna z metod numerycznych stosowanych do rozwiązywania równań różniczkowych. Jest to metoda jednokrokowa i jest ulepszoną wersją metody Eulera.

Metoda Heuna polega na użyciu średniej ważonej pomiędzy punktem początkowym i przewidywanym punktem końcowym do obliczenia wartości funkcji.

$$y_{n+1}=y_n+\frac{h}{2}\cdot[f(x_n,y_n)+f(x_n+h,y_n+h\cdot f(x_n,y_n))] \tag{1}$$

$$N=\frac{(b-x_0)}{h} \tag{2}$$

gdzie:

$n=0,1,\dots,N$

$f(x,y)-\mathit{dane\ równanie}$

$y_0-\mathit{warunek\ początkowy}$

$x_0-\mathit{początek\ badanego\ zakresu}$

$b-\mathit{koniec\ badanego\ zakresu}$

$h-\mathit{rozmiar\ kroków}$

Metoda Heuna jest znacznie dokładniejsza niż metoda Eulera, ale jest też nieco bardziej złożona obliczeniowo. Mimo to, jest często stosowana ze względu na swoją efektywność i dokładność.

1.2 metoda RK4

Metoda Rungego-Kutty czwartego rzędu, znana również jako RK4, to jedna z najpopularniejszych metod numerycznych stosowanych do rozwiązywania równań różniczkowych. Jest to metoda jednokrokowa, która oferuje dobrą równowagę pomiędzy dokładnością a złożonością obliczeniową.

Metoda RK4 polega na obliczeniu czterech przybliżeń (stąd nazwa “czwartego rzędu”) wartości funkcji w danym punkcie, a następnie użyciu ich do obliczenia następnego kroku. W praktyce wygląda to tak:

$$y_{i+1}=y_i+\frac{(k_1+k_2+k_3+k_4)}{6} \tag{3}$$

gdzie:

$$i=0,1,2,\ldots$$

$$k_1=h\cdot f(x_i,y_i) \tag{4}$$

$$k_2=h\cdot f(x_i+\frac{h}{2},y_i+\frac{k_1}{2}) \tag{5}$$

$$k_3=h\cdot f(x_i+\frac{h}{2},y_i+\frac{k_2}{2}) \tag{6}$$

$$k_4=h\cdot f(x_i+h,y_i+k_3) \tag{7}$$

Metoda RK4 jest znacznie dokładniejsza niż metoda Eulera czy metoda Heuna, ale jest też nieco bardziej złożona obliczeniowo. Mimo to, jest często stosowana ze względu na swoją efektywność i dokładność.

2. Opis implementacji numerycznej

```
double f(double x, double y) {
    return cos(x) - sin(x) - y;
}
```

Fragment kodu 2.1: funkcja równania różniczkowego

Ta funkcja reprezentuje funkcję w równaniu różniczkowym.

W tym przypadku $f(x,y)=\cos(x)-\sin(x)-y$.

2.1 metoda Heuna

```
cout << "Metoda Heuna" << endl;
for (int i = 1; i < N; i++) {
    double k1 = f(x[i - 1], y[i - 1]);
    double k2 = f(x[i - 1] + h, y[i - 1] + h * k1);

    y[i] = y[i - 1] + (h / 2) * (k1 + k2);
    x[i] = x[i - 1] + h;
    printf("y%d = %lf\n", i, y[i]);
}
```

Fragment kodu 2.2: implementacja metody Heuna

Każda iteracja pętli **for** odpowiada jednemu kroku metody Heuna. W zmiennej **double k_1** obliczana jest wartość funkcji **f** w punkcie **(x[i-1], y[i-1])**. Ta wartość reprezentuje nachylenie stycznej do krzywej rozwiązania w tym punkcie. W zmiennej **double k_2** obliczana jest wartość funkcji **f** w punkcie **(x[i-1]+h, y[i-1]+h*k1)**. Ta wartość reprezentuje nachylenie stycznej do krzywej rozwiązania w punkcie, do którego prowadziłoby nas zastosowanie metody Eulera. Następnie jest **wartość y[i]** na podstawie poprzedniej **wartości y[i-1]** oraz średniego nachylenia stycznego obliczonego na podstawie **k_1** i **k_2**. W ostatnim kroku obliczana jest nowa **wartość x[i]** jako suma poprzedniej **wartości x[i-1]** oraz **kroku h**. Wartości **x** i **y** są tablicami przechowującymi **wartości x** i **y** w kolejnych punktach. Wartości **x[0]** oraz **y[0]** są już zainicjowane.

2.2 metoda RK4

```
cout << "Metoda RK4" << endl;
for (int i = 1; i < N; i++) {
    double k1 = h * f(x[i - 1], y[i - 1]);
    double k2 = h * f(x[i - 1] + 0.5 * h, y[i - 1] + 0.5 * k1);
    double k3 = h * f(x[i - 1] + 0.5 * h, y[i - 1] + 0.5 * k2);
    double k4 = h * f(x[i - 1] + h, y[i - 1] + k3);

    y[i] = y[i - 1] + (k1 + 2 * k2 + 2 * k3 + k4) / 6;

    x[i] = x[i - 1] + h;
    printf("y%d = %lf\n", i, y[i]);
}
```

Fragment kodu 2.3: implementacja metody RK4

W petli **for** obliczane są cztery wartości: **k_1**, **k_2**, **k_3**, **k_4** według wzorów (4-7). Każda z nich reprezentuje różne przybliżenia nachylenia funkcji w danym punkcie. Następnie obliczana jest nowa wartość **y[i]** jako suma **wartości y[i-1]** oraz średniej ważonej wartości **k_1**, **$2k_2$**, **$2k_3$**, **k_4**. W ostatnim kroku obliczana jest nowa **wartość x[i]** jako suma poprzedniej

wartości x[i-1] oraz **kroku h**. Wartości **x** i **y** są tablicami przechowującymi **wartości x** i **y** w kolejnych punktach. Wartości **x[0]** oraz **y[0]** są już zainicjowane.

3. Testy kodu numerycznego

Test I

Dane:

$f(x,y)=\cos(x)-\sin(x)-y,$

x_0 = 0; y_0 = 2; b = 0.3; h = 0.1

Wynik Wolfram Alpha:

Wynik zaimplementowanego programu:

metoda Heuna

y_1 = 1.89984
y_2 = 1.79877
y_3 = 1.6961

```
Metoda Heuna
y1 = 1.899759
y2 = 1.798634
y3 = 1.695917
```

Fragment konsoli 3.1: wynik metody Heuna dla testu I

metoda RK4

y_1 = 1.89984
y_2 = 1.7988
y_3 = 1.69615

```
Metoda RK4
y1 = 1.899842
y2 = 1.798797
y3 = 1.696155
```

Fragment konsoli 3.2: wynik metody RK4 dla testu I

Test II

Dane:

$f(x,y)=x+y,$

x_0 = 0; y_0 = 1; b = 0.2; h = 0.1

Wynik Wolfram Alpha:

Wynik zaimplementowanego programu:

metoda Heuna

y_1 = 1.11
y_2 = 1.24205

```
Metoda Heuna
y1 = 1.110000
y2 = 1.242050
```

Fragment konsoli 3.3: wynik metody Heuna dla testu II

metoda RK4

y_1 = 1.11034
y_2 = 1.24281

```
Metoda RK4
y1 = 1.110342
y2 = 1.242805
```

Fragment konsoli 3.4: wynik metody RK4 dla testu II

Test III

Dane:

$$f(x,y)=\frac{y}{x}+x,$$

x_0 = ;1 y_0 = 2; b = 2; h = 0.1

Wynik Wolfram Alpha:

Wynik zaimplementowanego programu:

metoda Heuna

y_1 = 2.30969
y_2 = 2.63937
y_3 = 2.98906
y_4 = 3.35874
y_5 = 3.74842
y_6 = 4.15811
y_7 = 4.58779
y_8 = 5.03747
y_9 = 5.50715
y_{10} = 5.99683

```
Metoda Heuna
y1 = 2.309545
y2 = 2.639087
y3 = 2.988627
y4 = 3.358164
y5 = 3.747700
y6 = 4.157234
y7 = 4.586767
y8 = 5.036299
y9 = 5.505830
y10 = 5.995360
```

Fragment konsoli 3.5: wynik metody Heuna dla testu III

metoda RK4

y_1 = 2.31
y_2 = 2.64
y_3 = 2.99
y_4 = 3.36
y_5 = 3.75
y_6 = 4.16
y_7 = 4.59
y_8 = 5.04
y_9 = 5.51
y_{10} = 6

```
Metoda RK4
y1 = 2.310000
y2 = 2.639999
y3 = 2.989999
y4 = 3.359999
y5 = 3.749999
y6 = 4.159998
y7 = 4.589998
y8 = 5.039998
y9 = 5.509998
y10 = 5.999998
```

Fragment konsoli 3.6: wynik metody RK4 dla testu III

Test IV

Dane:

$f(x,y)=y-2\cdot x^2\cdot e^{-x}$,

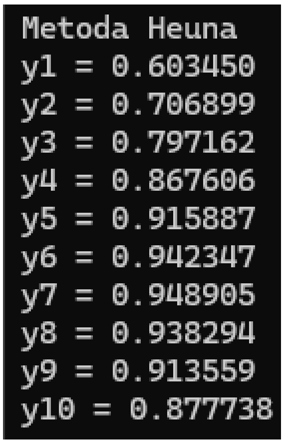
$x_0 = 0$; $y_0 = 0.5$; $b = 2$; $h = 0.2$

Wynik Wolfram Alpha:

Wynik zaimplementowanego programu:

metoda Heuna

$y_1 = 0.605332$
$y_2 = 0.710036$
$y_3 = 0.801169$
$y_4 = 0.872276$
$y_5 = 0.92115$
$y_6 = 0.948239$
$y_7 = 0.955546$
$y_8 = 0.945878$
$y_9 = 0.922351$
$y_{10} = 0.888071$

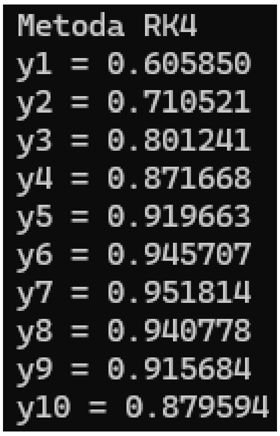


```
Metoda Heuna
y1 = 0.603450
y2 = 0.706899
y3 = 0.797162
y4 = 0.867606
y5 = 0.915887
y6 = 0.942347
y7 = 0.948905
y8 = 0.938294
y9 = 0.913559
y10 = 0.877738
```

Fragment konsoli 3.7: wynik metody Heuna dla testu IV

metoda RK4

$y_1 = 0.60585$
$y_2 = 0.710521$
$y_3 = 0.801241$
$y_4 = 0.871668$
$y_5 = 0.919663$
$y_6 = 0.945707$
$y_7 = 0.951814$
$y_8 = 0.940778$
$y_9 = 0.915684$
$y_{10} = 0.879594$



```
Metoda RK4
y1 = 0.605850
y2 = 0.710521
y3 = 0.801241
y4 = 0.871668
y5 = 0.919663
y6 = 0.945707
y7 = 0.951814
y8 = 0.940778
y9 = 0.915684
y10 = 0.879594
```

Fragment konsoli 3.8: wynik metody RK4 dla testu IV

Test V

Dane:

$f(x,y)=\frac{y}{x}\cdot(\ln(y)-\ln(x))$,

$x_0 = 1$; $y_0 = e^2$; $b = 3$; $h = 0.1$

Wynik Wolfram Alpha: *Wynik zaimplementowanego programu:*

metoda Heuna

y_1 = 8.97184
y_2 = 10.8038
y_3 = 12.9198
y_4 = 15.3588
y_5 = 18.1653
y_6 = 21.3892
y_7 = 25.087
y_8 = 29.3222
y_9 = 34.1667
y_{10} = 39.7012
y_{11} = 46.0169
y_{12} = 53.2161
y_{13} = 61.4145
y_{14} = 70.7417
y_{15} = 81.3438
y_{16} = 93.3848
y_{17} = 107.049
y_{18} = 122.545
y_{19} = 140.104
y_{20} = 159.988

```
Metoda Heuna
y1 = 8.969110
y2 = 10.797444
y3 = 12.908549
y4 = 15.341397
y5 = 18.139995
y6 = 21.354017
y7 = 25.039500
y8 = 29.259632
y9 = 34.085631
y10 = 39.597732
y11 = 45.886287
y12 = 53.053005
y13 = 61.212330
y14 = 70.492989
y15 = 81.039726
y16 = 93.015234
y17 = 106.602322
y18 = 122.006334
y19 = 139.457859
y20 = 159.215753
```

Fragment konsoli 3.9: wynik metody Heuna dla testu V

metoda RK4

y_1 = 8.98273
y_2 = 10.8299
y_3 = 12.9662
y_4 = 15.4321
y_5 = 18.2733
y_6 = 21.5414
y_7 = 25.2947
y_8 = 29.5993
y_9 = 34.5295
y_{10} = 40.1693
y_{11} = 46.6135
y_{12} = 53.9688
y_{13} = 62.3557
y_{14} = 71.9097
y_{15} = 82.7836
y_{16} = 95.1494
y_{17} = 109.2
y_{18} = 125.155
y_{19} = 143.257
y_{20} = 163.782

```
Metoda RK4
y1 = 8.982731
y2 = 10.829890
y3 = 12.966222
y4 = 15.432122
y5 = 18.273279
y6 = 21.541351
y7 = 25.294708
y8 = 29.599277
y9 = 34.529485
y10 = 40.169315
y11 = 46.613498
y12 = 53.968838
y13 = 62.355711
y14 = 71.909732
y15 = 82.783632
y16 = 95.149360
y17 = 109.200431
y18 = 125.154564
y19 = 143.256632
y20 = 163.781966
```

Fragment konsoli 3.10: wynik metody RK4 dla testu V

4. Analiza wyników

W celu głębszej analizy powstała tabela, w której porównane zostały wyniki zaimplementowanych metod z wynikami programu „Wolfram Alpha” na podstawie powyższych testów.

Tabela 1: wyniki Testu I

	Wolfram Alpha	Program	Wolfram Alpha	Program
wartości	Heun		RK4	
x_1	1.89984	1.899759	1.89984	1.899842
x_2	1.79877	1.798634	1.7988	1.798797
x_3	1.6961	1.695917	1.69615	1.696155

Tabela 2: wyniki Testu II

	Wolfram Alpha	Program	Wolfram Alpha	Program
wartości	Heun		RK4	
x_1	1.11	1.11	1.11034	1.110342
x_2	1.24205	1.24205	1.24281	1.242805

Tabela 3: wyniki Testu III

	Wolfram Alpha	Program	Wolfram Alpha	Program
wartości	Heun		RK4	
x_1	2.30969	2.309545	2.31	2.31
x_2	2.63937	2.639087	2.64	2.639999
x_3	2.98906	2.988627	2.99	2.989999
x_4	3.35874	3.358164	3.36	3.359999
x_5	3.74842	3.7477	3.75	3.749999
x_6	4.15811	4.157234	4.16	4.159998
x_7	4.58779	4.586767	4.59	4.589998
x_8	5.03747	5.036299	5.04	5.039998
x_9	5.50715	5.50583	5.51	5.509998

x_{10}	5.99683	5.99536	6	5.999998

Tabela 4: wyniki Testu IV

	Wolfram Alpha	Program	Wolfram Alpha	Program
wartości	Heun		RK4	
x_1	0.605332	0.60345	0.60585	0.60585
x_2	0.710036	0.706899	0.710521	0.710521
x_3	0.801169	0.797162	0.801241	0.801241
x_4	0.872276	0.867606	0.871668	0.871668
x_5	0.92115	0.915887	0.919663	0.919663
x_6	0.948239	0.942347	0.945707	0.945707
x_7	0.955546	0.948905	0.951814	0.951814
x_8	0.945878	0.938294	0.940778	0.940778
x_9	0.922351	0.913559	0.915684	0.915684
x_{10}	0.888071	0.877738	0.879594	0.879594

Tabela 5: wyniki Testu V

	Wolfram Alpha	Program	Wolfram Alpha	Program
wartości	Heun		RK4	
x_1	8.97184	8.96911	8.98273	8.982731
x_2	10.8038	10.797444	10.8299	10.82989
x_3	12.9198	12.908549	12.9662	12.966222
x_4	15.3588	15.341397	15.4321	15.432122
x_5	18.1653	18.139995	18.2733	18.273279
x_6	21.3892	21.354017	21.5414	21.541351
x_7	25.087	25.0395	25.2947	25.294708
x_8	29.3222	29.259632	29.5993	29.599277
x_9	34.1667	34.085631	34.5295	34.529485

x_{10}	39.7012	39.597732	40.1693	40.169315
x_{11}	46.0169	45.886287	46.6135	46.613498
x_{12}	53.2161	53.053005	53.9688	53.968838
x_{13}	61.4145	61.21233	62.3557	62.355711
x_{14}	70.7417	70.492989	71.9097	71.909732
x_{15}	81.3438	81.039726	82.7836	82.783632
x_{16}	93.3848	93.015234	95.1494	95.14936
x_{17}	107.049	106.602322	109.2	109.200431
x_{18}	122.545	122.006334	125.155	125.154564
x_{19}	140.104	139.457859	143.257	143.256632
x_{20}	159.988	159.215753	163.782	163.781966

Dzięki analizie wyników można stwierdzić, że przedstawiona metoda została zaimplementowana poprawnie. Wyniki uzyskane za pomocą zaimplementowanego programu są niemal identyczne względem wyników uzyskanych za pomocą programu Wolfram Alpha. Jednakże, występuje parę przypadków gdzie zauważono minimalne rozbieżności, które mogą wynikać z precyzji elementów - liczby zmiennoprzecinkowe o dużej liczbie miejsc po przecinku, zainicjowana stała Eulera używana w programie.
W trakcie obliczeń numerycznych, między innymi w metodzie Heuna lub Rungego-Kutty czwartego rzędu obliczania równań różniczkowych, mogą wystąpić błędy zaokrąglenia wynikające z ograniczonej precyzji liczb zmiennoprzecinkowych. Te błędy mogą się kumulować w trakcie obliczeń, prowadząc do zaokrąglania wyników. Analizując wyniki metoda RK4 jest znacznie dokładniejsza niż metoda Heuna, ale jest też nieco bardziej złożona obliczeniowo. Mimo to, jest często stosowana ze względu na swoją efektywność i dokładność.

5. Wnioski

W tym sprawozdaniu przedstawiono oraz omówiono metody Heuna oraz Rungego-Kutty czwartego rzędu służące do rozwiązywania równań różniczkowych. Pokazano oraz wytłumaczono ich prostą implementację w języku c++. Z analizy przeprowadzonych testów wynika że metody te są skutecznym narzędziem do rozwiązywania równań różniczkowych. Testy wykazały, że zastosowana implementacja jest szczególnie efektywna pod względem uzyskanych wyników, natomiast mogą wystąpić minimalne błędy zaokrąglenia wynikające z ograniczonej precyzji liczb zmiennoprzecinkowych.

6. Źródła

- prezentacja z zajęć „lab 12.pdf”

- Beata Pańczyk, Edyta Łukasik, Jan Sikora, Teresa Guziak, Metody numeryczne w przykładach, Politechnika Lubelska, 2012
- www.wolframalpha.com/input?i=use+Heun+method+y%27+%3D+cos%28x%29+-+sin%28x%29+-+y%2C+y%280%29+%3D+2%2C+from+0+to+0.3%2C+h+%3D+0.1
- www.wolframalpha.com/input?i=use+RK4+method+y%27+%3D+cos%28x%29+-+sin%28x%29+-+y%2C+y%280%29+%3D+2%2C+from+0+to+0.3%2C+h+%3D+0.1