Projekt nr 3 - Metody numeryczne dla zagadnień różniczkowych

Dominik Kulwicki, Piotr Szpak

28 stycznia 2021

Spis treści

1 Treść zadania

Napisz program, który rozwiąże trzema metodami(Eulera, zmodyfikowaną Eulera oraz Heuna) zagadnienia różniczkowe:

$$y'(x) = f(x, y(x)),\ y(1) = 6\ ,$$

gdzie

$$f(x, y(x)) = 6x^5 + 16x^3 - 2\sqrt{x^6 + 4x^4 + 2 - y}\ .$$

Program ma również obliczyć dokładność dla każdej z metod, porównując je z dokładnym rozwiązaniem:

$$y'(x) = x^6 + 4x^4 - x^2 + 2\ .$$

Dane wejściowe:

a. Liczba $n \geqslant 1$, określająca podział odcinka $[a, b]$.

b. Liczba b, informująca o końcu odcinka $[a, b]$.

Dane wyjściowe:

a. Przybliżone rozwiązanie zagadnienia dla każdej metod.

b. Dokładność rozwiązania dla każdej z metod.

1.1 Opis metody numerycznej

Rozważamy zagadnienie początkowe

$$y'(x) = f(x, y(x)),\ y(a) = y_0, \tag{1}$$

na przedziale $[a, b]$, gdzie $f : [a, b] \times R \to R$ jest daną funkcją, a $y_0 \in R$ jest daną liczbą.

Dla ustalonej liczby naturalnej $N \geqslant 1$ definiujemy $h = \frac{b-a}{N}$ i punkty

$$x_k = a + k\,h,\ 0 \leqslant k \leqslant N.$$

Wtedy x_k są równo rozłożone w przedziale $[a, b]$. Punkty takie nazywamy węzłami równoległymi.

Niech $y = y(x)$ będzie rozwiązaniem zagadnienia (1). Skonstruujemy tzw. metody różnicowe na wyznaczanie wartości y_k, które przybliżać będą wartości $y(x_k)$ rozwiązania dokładnego dla $0 \leqslant k \leqslant N$.

Różnicowa metoda Eulera

Z równania różniczkowego dla $x = x_k$ otrzymujemy

$$y'(x_k) = f(x_k, y(x_k)).$$

Wartość $y'(x)$ przybliżamy za pomocą ilorazu różnicowego

$$y'(x_k) \approx \frac{y(x_k + h) - y(x_k)}{h}.$$

Ponieważ $x_k + h = x_{k+1}$, to otrzymujemy $y'(x_k) \approx \frac{1}{h}(y(x_{k+1}) - y(x_k))$, a po wstawieniu do równania $\frac{1}{h}(y(x_{k+1}) - y(x_k)) \approx f(x_k, y(x_k))$ lub

$$y(x_{k+1}) \approx y(x_k) + hf(x_k, y(x_k)).$$

Otrzymana równość przybliżona jest źródłem zależności rekurencyjnej

$$y_{k+1} = y_k + hf(x_k, y_k),\ 0 \leqslant k \leqslant N - 1, \tag{2}$$

gdzie y_0 jest dane w zagadnienu (1). Zależność (2) nazywamy różnicową metodą Eulera. Punkty (x_k, y_k), $0 \leqslant k \leqslant N$, wyznaczają przybliżone rozwiązanie zagadnienia (1).

Zmodyfikowana metoda Eulera

Po scałkowaniu równania $y'(x) = f(x, y(x))$ w granicach od x_k do x_{k+1} otrzymujemy $y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} f(x, y(x))dx$. Przybliżamy całkę "polem prostokąta" z punktem środkowym

$$\int_{x_k}^{x_{k+1}} f(x, y(x))\, dx \approx hf(x_k + \frac{h}{2}, y(x_k + \frac{h}{2})).$$

Wartość $y(x_k + \frac{h}{2})$ przybliżamy za pomocą rozwinięcia Taylora.

$$y(x_k + \frac{h}{2}) = y(x_k) + \frac{h}{2} f(x_k, y(x_k)) + \frac{h^2}{4}\frac{y''(\eta k)}{2},$$

biorąc $y(x_k + \frac{h}{2}) \approx y(x_k) + \frac{h}{2} f(x_k, y(x_k))$. Otrzymamy

$$y(x_{k+1}) \approx y(x_k) + hf\Big(x_k + \frac{h}{2}, y(x_k) + \frac{h}{2} f(x_k, y(x_k))\Big),$$

co jest źródłem zależności rekurencyjnej

$$y_{k+1} = y_k + hf\Big(x_k + \frac{h}{2}, y_k + \frac{h}{2} f(x_k, y_k)\Big),\ 0 \leqslant k \leqslant N - 1. \qquad (3)$$

Metodę (3) nazywamy zmodyfikowaną metodą Eulera.

Metoda Huena

W równości $y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} f(x, y(x))\, dx$ całkę przybliżymy "polem trapezu"

$$\int_{x_k}^{x_{k+1}} f(x, y(x))\, dx \approx \frac{h}{2}\Big(f(x_k, y(x_k)) + f(x_{k+1}, y(x_{k+1}))\Big).$$

Otrzymamy przybliżenie
$y(x_{k+1}) \approx y(x_k) + \frac{h}{2}\Big(f(x_k, y(x_k)) + f(x_{k+1}, y(x_{k+1}))\Big)$, które jest źródłem metody

$$y_{k+1} = y_k + \frac{h}{2}\Big(f(x_k, y_k) + f(x_{k+1}, y_{k+1})\Big),\ 0 \leqslant k \leqslant N - 1.$$

Jest to metoda niejawna - do wyznaczenia y_{k+1} trzeba rozwiązać równanie $u = \varphi(u)$, gdzie $\varphi(u) = y_k + \frac{h}{2}\Big(f(x_k, y_k) + f(x_{k+1}, u)\Big)$. Jeśli weźmiemy $u_0 = y_k + hf(x_k, y_k)$ i za y_{k+1} przyjmiemy $u_1 = \varphi(u)$, to otrzymamy

$$y_{k+1} = y_k + \frac{h}{2}\Big(f(x_k, y_k) + f(x_k + h, y_k + hf(x_k, y_k))\Big),$$

gdzie $0 \leqslant k \leqslant N - 1$. Jest to tzw. metoda Huena.

1.2 Przykład ilustrujący metodę

Przykład 1. Zagadnienie

$$y'(x) = 1 + 2\sqrt{y(x) - x - 1},\ y(1) = 3,$$

Rozwiązanie dokładne

$$y'(x) = x^2 + x + 1\ .$$

Na przedziale $[1, 2]$, przyjmujemy $N = 2$. Wtedy $h = \frac{2-1}{2} = \frac{1}{2},\ x_0 = 1,\ x_1 = \frac{3}{2},\ x_2 = 2$

Metoda Eulera

$$y_{k+1} = y_k + hf(x_k, y_k),\ k \geqslant 0$$

$y_0 = y(1) = 3$
$y_1 = y_0 + hf(x_0, y_0) = 3 + \frac{1}{2}f(1, 3) = 3 + \frac{1}{2} * 3 = \frac{9}{2}$
$y_2 = y_1 + hf(x_1, y_1) = \frac{9}{2} + \frac{1}{2}f(\frac{3}{2}, \frac{9}{2}) = \frac{9}{2} + \frac{1}{2}\Big(1 + 2\sqrt{2}\Big) = 5 + \sqrt{2}$
$y(x_1) = \frac{9}{4} + \frac{3}{2} + 1 = \frac{19}{4} \Longrightarrow |y_1 - y(x_1)| = |\frac{9}{2} - \frac{19}{4}| = \frac{1}{4}$
$y(x_2) = 2^2 + 2 + 1 = 7 \Longrightarrow |y_2 - y(x_2)| = |5 + \sqrt{2} - 7| = 2 - \sqrt{2} \approx 0.6$
błąd maksymalny: $\max\{\frac{1}{4}, 2 - \sqrt{2}\} = 2 - \sqrt{2} \approx 0.6$

Zmodyfikowana metoda Eulera

$$y_{k+1} = y_k + hf\Big(x_k + \frac{h}{2}, y_k + \frac{h}{2}f(x_k, y_k)\Big),\ k \geqslant 0$$

$y_0 = y(1) = 3$
$y_1 = y_0 + hf\Big(x_0 + \frac{h}{2}, y_0 + \frac{h}{2}f(x_0, y_0)\Big) = 3 + \frac{1}{2}f\Big(\frac{5}{4}, 3 + \frac{1}{4}f(1, 3)\Big) = 3 + \frac{1}{2}f(\frac{5}{4}, \frac{15}{4}) = 3 + \frac{1}{2}\Big(1 + 2\sqrt{\frac{6}{4}}\Big) = \frac{7}{2} + \frac{1}{2}\sqrt{6}$
$y(x_1) = \frac{9}{4} + \frac{3}{2} + 1 = \frac{19}{4} \Longrightarrow |y_1 - y(x_1)| = |\frac{7}{2} + \frac{1}{2}\sqrt{6} - \frac{19}{4}| = \frac{5}{4} - \frac{1}{2}\sqrt{6} \approx \frac{1}{40}$

Metoda Huena

$$y_{k+1} = y_k + \frac{h}{2}\Big(f(x_k, y_k) + f(x_k + h, y_k + hf(x_k, y_k))\Big),\ k \geqslant 0$$

$y_0 = y(1) = 3$
$y_1 = y_0 + \frac{h}{2}\Big(f(x_0, y_0) + f(x_0 + h, y_0 + hf(x_0, y_0))\Big) = 3 + \frac{1}{4}f\Big(3 + f(\frac{3}{2}, \frac{9}{2})\Big) = 4 + \frac{1}{2}\sqrt{2}$
$y(x_1) = \frac{9}{4} + \frac{3}{2} + 1 = \frac{19}{4} \Longrightarrow |y_1 - y(x_1)| = |4 + \frac{1}{2}\sqrt{2} - \frac{19}{4}| = \frac{3}{4} - \frac{1}{2}\sqrt{2} \approx \frac{1}{20}$

2 Opis implementacji algorytmu

Implementacja algorytmu opiera się na użyciu klasy **DifferentialEquation**, zawierającej konstruktor, trzy główne metody **eulerMethod**, **modifiedEulerMethod**, **heuenMethod** oraz metody pomocnicze **solveFunction** i **solutionPrecision**.
Konstruktor inicjuje zmienne, określające podział odcinka, jego początek i koniec oraz wartość $y0$. Oblicza także wartość zmiennej h oraz wypełnia tablicę rozwiązań dokładnych.
Metody główne implementują kolejne sposoby rozwiązywania zagadnienia różniczkowego zawartego w metododzie **solveFunction**. Metoda **solutionPrecision** oblicza dokładność rozwiązania.

3 Program komputerowy

3.1 Struktura danych i programu

Program jest podzielony na dwie główne częsci poprzez klasy **Main** oraz **DifferentialEquation**. Pierwsza zajmuje się pobraniem danych wejściowych od użytkownika wywołaniem trzech głównych metod obliczających algorytm oraz wypisaniem danych wyjściowych. Druga implementuje działanie algorytmu.

3.2 Opis wejścia-wyjścia

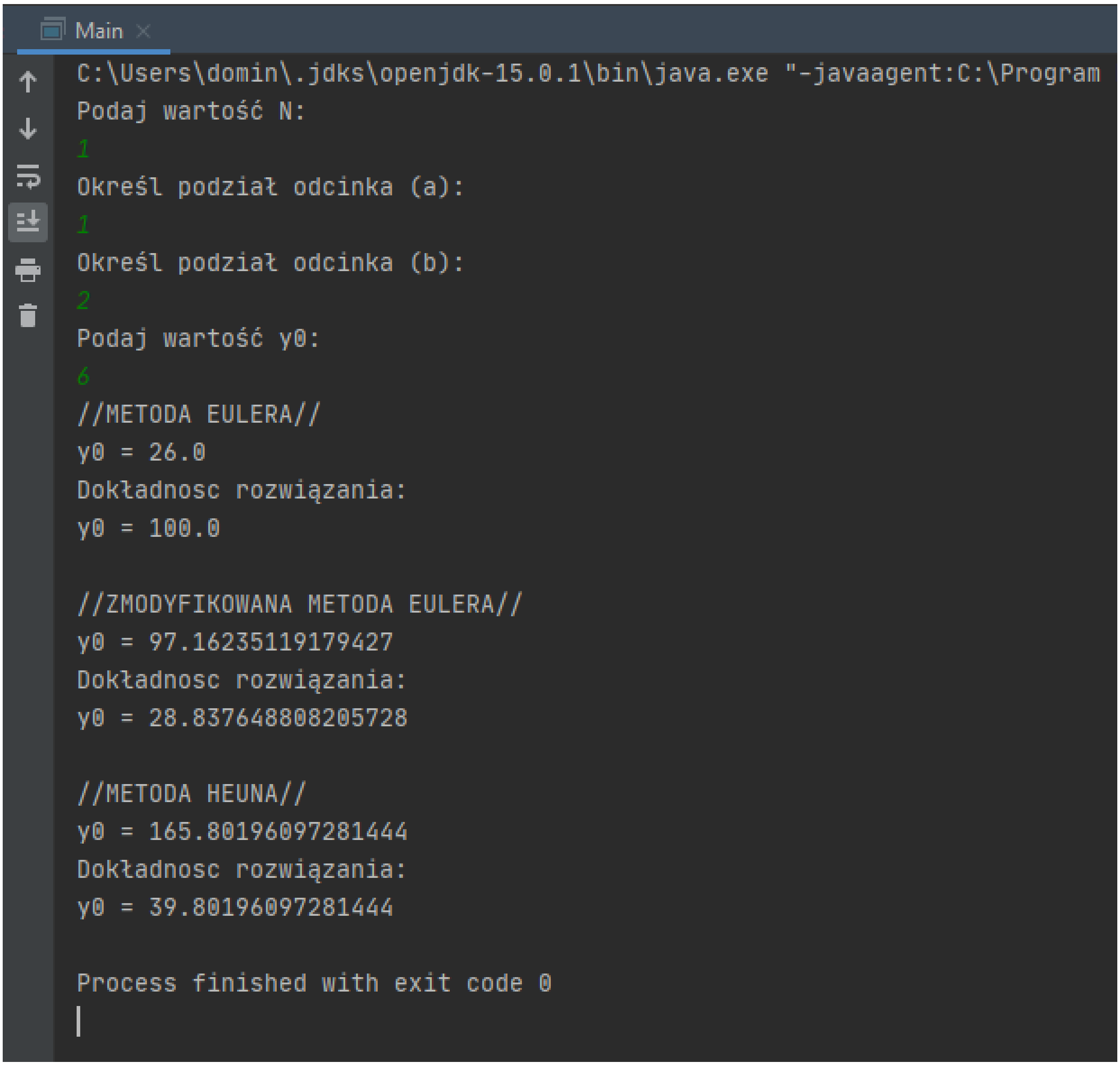
Do danych wejściowych należą trzy wartości typu **integer** — N, a, b oraz wartość typu **double** — $y0$. Po uruchomieniu programu użytkownik proszony jest o uzupełnienie kolejnych danych. Wprowadzenie danych nieprawidłowych skutkuję wyświetleniem odpowiedniego komunikatu i ponawia możliwość ich wprowadzenia. Następnie program przekazuje te dane do konstruktora i wywołuje kolejne metody wyliczając następne wartości y dla każdej z metod.

Danymi wyjściowymi są przybliżone rozwiązanie zagadnienia oraz dokładność rozwiązania dla każdej z metod.

3.3 Tekst kodu programu

```java
public double solveFunction(double x, double y) {
    double sqrtx = pow(x,6) + 4 * pow(x,4) + 2 - y;
    sqrtx = Math.sqrt(sqrtx);
    return 6 * pow(x,5) + 16 * pow(x,3) - 2 * sqrtx;
}

public double[] eulerMethod() {
    double x = this.a;
    double[] y = new double[this.N];

    y[0] = y0 + h * solveFunction(x,y0);
    x += h;

    for(int i=1; i < N; i++){
        y[i] = y[i-1] + h * solveFunction(x,y[i-1]);
        x += h;
    }
    return y;
}

public double[] modifiedEulerMethod() {
    double x = this.a;
    double[] y = new double[this.N];
    y[0] = y0 + h * solveFunction( x: x + h/2,  y: y0 + h/2 * solveFunction(x,y0));
    x += h;
    for (int i = 1; i < N; i++) {
        y[i] = y[i-1] + h * solveFunction( x: x + h/2,  y: y[i-1] + h/2 * solveFunction(x,y[i-1]));
        x += h;
    }
    return y;
}

public double[] heunMethod() {
    double x = this.a;
    double[] y = new double[this.N];
    y[0] = y0 + h/2 * (solveFunction(x,y0) + solveFunction( x: x + h,  y: y0 + h * solveFunction(x,y0)));
    x += h;
    for (int i = 1; i < N; i++) {
        y[i] = y[i-1] + h/2 * (solveFunction(x,y[i-1]) +
            solveFunction( x: x + h,  y: y[i-1] + h * solveFunction(x,y[i-1])));
        x += h;
    }
    return y;
}
```

3.4 Zademonstrowanie działania



```
Main
C:\Users\domin\.jdks\openjdk-15.0.1\bin\java.exe "-javaagent:C:\Program
Podaj wartość N:
1
Określ podział odcinka (a):
1
Określ podział odcinka (b):
2
Podaj wartość y0:
6
//METODA EULERA//
y0 = 26.0
Dokładnosc rozwiązania:
y0 = 100.0

//ZMODYFIKOWANA METODA EULERA//
y0 = 97.16235119179427
Dokładnosc rozwiązania:
y0 = 28.837648808205728

//METODA HEUNA//
y0 = 165.80196097281444
Dokładnosc rozwiązania:
y0 = 39.80196097281444

Process finished with exit code 0
```

3.5 Przykładowe wyniki programu

```
C:\Users\domin\.jdks\openjdk-15.0.1\bin\java.exe "-javaagent:C:\Program
Podaj wartość N:
3
Określ podział odcinka (a):
0
Określ podział odcinka (b):
1
Podaj wartość y0:
1
//METODA EULERA//
y0 = 0.33333333333333337
y1 = -0.3345747825327143З
y2 = 0.31415197051722865
Dokładnosc rozwiązania:
y0 = 1.606310013717421
y1 = 2.7680452900772954
y2 = 5.685848029482772

//ZMODYFIKOWANA METODA EULERA//
y0 = 0.2542515629905252
y1 = -0.03970807546423594
y2 = 2.525295794639238
Dokładnosc rozwiązania:
y0 = 1.6853917840602293
y1 = 2.473178583008817
y2 = 3.474704205360762

//METODA HEUNA//
y0 = 0.33271260873364283
y1 = 0.3229705964973194
y2 = 3.569044850791569
Dokładnosc rozwiązania:
y0 = 1.6069307383171116
y1 = 2.110499911047262
y2 = 2.430955149208431
```

```
Main
C:\Users\domin\.jdks\openjdk-15.0.1\bin\java.exe "-javaagent:C:\Program
Podaj wartość N:
-1
Podaj prawidłowe dane:
0
Podaj prawidłowe dane:
10
Określ podział odcinka (a):
0
Określ podział odcinka (b):
1
Podaj wartość y0:
1
//METODA EULERA//
y0 = 0.8
y1 = 0.5824803739312995
y2 = 0.3568108038309835
y3 = 0.14252338902940676
y4 = -0.02921622525944842
y5 = -0.11344089253441703
y6 = -0.04850717362963475
y7 = 0.2474934162595222
y8 = 0.8810423112670849
y9 = 1.9882245852552134
Dokładnosc rozwiązania:
y0 = 1.190401
y1 = 1.3839836260687006
y2 = 1.5863181961690165
y3 = 1.8039726109705931
y4 = 2.0448412252594483
y5 = 2.3184968925344176
y6 = 2.636556173629635
y7 = 3.013050583740478
y8 = 3.464798688732915
y9 = 4.011775414744786
```

Main

```
C:\Users\domin\.jdks\openjdk-15.0.1\bin\java.exe "-javaagent:C:\Program
Podaj wartość N:
2
Określ podział odcinka (a):
0
Określ podział odcinka (b):
1
Podaj wartość y0:
-2
//METODA EULERA//
y0 = -4.0
y1 = -5.409373049312599
Dokładnosc rozwiązania:
y0 = 6.015625
y1 = 11.409373049312599

//ZMODYFIKOWANA METODA EULERA//
y0 = -4.111683926636376
y1 = -2.900718256695518
Dokładnosc rozwiązania:
y0 = 6.127308926636376
y1 = 8.900718256695518

//METODA HEUNA//
y0 = -3.7046865246562994
y1 = -0.6154919414082651
Dokładnosc rozwiązania:
y0 = 5.720311524656299
y1 = 6.615491941408266

Process finished with exit code 0
```