

Aproksymacja równań różniczkowych

Kamil Górny, Wojciech Jachimowski, Mateusz Jendernal, Dominika Gługła

Spis treści

1	Wstęp	3
1.1	Opis implementacji algorytmu realizującego metodę	3
2	Działanie programu	4
3	Opis programu	4
3.1	Struktura programu	4
3.2	Wejście-wyjście	4
3.3	Kod programu	5
3.3.1	JavaScript	5
3.3.2	HTML	5
3.3.3	CSS - fragment	5

1 Wstęp

Treść zadania: „Napisz program, który rozwiąże trzema metodami (Eulera, zmodyfikowaną Eulera oraz Heuna) zagadnienie różniczkowe:

$y'(x) = f(x, y(x))$, $y = 6$, gdzie $f(x, y(x)) = 2(y - 2x - 1) + 2$. Program ma również obliczyć dokładność dla każdej z tych metod, porównując je z dokładnym rozwiązaniem: $f(x) = 3x^2 + 2x + 1$. Dane wejściowe:

- a. Liczba $n \geq 1$, określająca podział odcinka $[a, b]$
- b. Liczba b , informująca o końcu odcinka $[a, b]$

Dane wyjściowe:

- a. Przybliżone rozwiązanie zagadnienia dla każdej z metod.
- b. Dokładność rozwiązania dla każdej z metod. ”

Wykorzystywana metoda numeryczna: metoda Eulera, zmodyfikowana metoda Eulera oraz metoda Heuna.

Zakładając, że mamy funkcję $f(x_k)$, dzięki podanym metodom jesteśmy w stanie zaproksymować $f'(x_k)$.

Metoda Eulera

Zakładając, że mamy podaną funkcję $y'(x_k) = f(x_k, y(x_k))$, możemy zaproksymować ją z zależności

$y(x_{k+1}) = y_k + hf(x_k, y_k)$, $0 \leq k \leq N - 1$, gdzie y_0 otrzymane jest w momencie wejścia i jest to x-owa współrzędna punktu należącego do funkcji pochodnej, która jest podana jako dana wejściowa.

Następne metody różnią się od siebie tylko zależnością rekurencyjną, gdzie dla poszczególnych metod obliczane są w następujący sposób:

Zmodyfikowana metoda Eulera:

$$y(x_{k+1}) = y_k + hf(x_k + \frac{h}{2}, y_k + \frac{h}{2}f(x_k, y_k)), \text{ gdzie } 0 \leq k \leq N - 1$$

Metoda Heuna:

$$y(x_{k+1}) = y_k + \frac{h}{2}(f(x_k, y_k) + f(x_k + h, y_k + hf(x_k, y_k))), \text{ gdzie } 0 \leq k \leq N - 1$$

1.1 Opis implementacji algorytmu realizującego metodę

Zależności rekurencyjne algorytmu są realizowane przez pętlę for. Wartości h funkcji bazowej i jej pochodnej są obliczane w oddzielnych funkcjach programu zgodnie z założeniami algorytmu.

2 Działanie programu

Metody numeryczne dla zagadnień różniczkowych

Zagadnienie różniczkowe: $y'(x) = f(x, y(x))$, $y(1) = 6$, gdzie $f(x, y(x)) = 2(y - 2x - 1) + 2$

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

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

Euler method
x0 = 1
x1 = 1.6666666666666665
x2 = 2.3333333333333333
x3 = 2.9999999999999996
y0 = 6
y1 = 11.333333333333332
y2 = 21.999999999999996
y3 = 45.111111111111111
(Błąd y(x1)) = 1.3333333333333321
(Błąd y(x2)) = 3.552713678800501e-15
(Błąd y(x3)) = 11.111111111111114
Błąd maksymalny metody: 11.111111111111114

Rys. 1: Działanie programu po wpisaniu danych wejściowych

3 Opis programu

3.1 Struktura programu

Program napisany jest w JavaScript, wykorzystuje HTML, CSS, Bootstrap. W celu realizowania zadania program wykorzystuje pętle, listy i funkcje. Program można wyświetlić w przeglądarce uruchamiając plik index.html. Można pobrać kod źródłowy z GitHuba: <https://github.com/projekt-zespolowy-rodeo/rownania-rozniczkowe>.

3.2 Wejście-wyjście

Wejście programu to liczba n określająca podział odcinka oraz liczba b informująca o końcu odcinka. Wyjście programu to przybliżone rozwiązanie zagadnienia oraz dokładność rozwiązania dla każdej z podanych metod.

3.3 Kod programu

3.3.1 JavaScript

```
1 let solution = document.getElementById("solution");
2 let br = document.createElement('br').value;
3
4 const eulerMethod = (n,a,b)=>{
5     solution.innerHTML = "";
6     solution.appendChild(document.createTextNode("Euler method"));
7     solution.appendChild(document.createElement('br'));
8     console.log("Euler method");
9     h = countH(a,b,n);
10    list_of_x = [];
11    list_of_x.push(a);
12
13
14    solution.appendChild(document.createTextNode("x"+0+" = "+
15        list_of_x[0]))
16    solution.appendChild(document.createElement('br'));
17    console.log("x"+0+" = "+list_of_x[0]);
18    for(var i=1;i<=n;i++){
19        list_of_x.push(list_of_x[i-1]+h);
20        solution.appendChild(document.createTextNode("x"+i+" = "+
21            list_of_x[i]))
22        solution.appendChild(document.createElement('br'));
23        console.log("x"+i+" = "+list_of_x[i]);
24    }
25    list_of_y = [];
26    list_of_y.push(func(a));
27    for(var i=1;i<=n;i++){
28        list_of_y.push(list_of_y[i-1]+h*dfunct(list_of_x[i-1],
29            list_of_y[i-1]));
30    }
31    for(var i=0;i<=n;i++){
32        solution.appendChild(document.createTextNode("y"+i+" = "+
33            list_of_y[i]))
34        solution.appendChild(document.createElement('br'));
35        console.log("y"+i+" = "+list_of_y[i]);
36    }
37    list_of_underestimates = []
38    for(var i=1;i<=n;i++){
39        var blad = Math.abs(list_of_y[i] - funct(list_of_x[i]));
40        list_of_underestimates.push(blad);
41        solution.appendChild(document.createTextNode("(Blad y(x"+i+
42            ") = "+blad))
43        solution.appendChild(document.createElement('br'));
44        console.log("(Blad y(x"+i+" = "+blad);
45    }
46    solution.appendChild(document.createTextNode("Błąd maksymalny
47        metody: "))
48    solution.appendChild(document.createTextNode(Math.max(...
49        list_of_underestimates)));
50    console.log(Math.max(...list_of_underestimates));
51    solution.appendChild(document.createElement('br'));
52    solution.appendChild(document.createElement('br'));
```

```

47
48 const modifiedEulerMethod = (n,a,b)=>{
49     solution.appendChild(document.createTextNode("Modified Euler
50         method"));
51     solution.appendChild(document.createElement('br'));
52     console.log("Modified Euler method");
53     h = countH(a,b,n);
54     list_of_x = [];
55     list_of_x.push(a);
56     solution.appendChild(document.createTextNode("x"+0+ " = "+
57         list_of_x[0]));
58     solution.appendChild(document.createElement('br'));
59     console.log("x"+0+ " = "+list_of_x[0]);
60     for(var i=1;i<=n;i++){
61         list_of_x.push(list_of_x[i-1]+h);
62         solution.appendChild(document.createTextNode("x"+i+ " = "+
63             list_of_x[i]));
64         solution.appendChild(document.createElement('br'));
65         console.log("x"+i+ " = "+list_of_x[i]);
66     }
67     list_of_y = [];
68     list_of_y.push(func(a));
69     for(var i=1;i<=n;i++){
70         list_of_y.push(list_of_y[i-1]+h*dfunct(list_of_x[i-1]+(h/2)
71             ,(h/2)*dfunct(list_of_x[i-1],list_of_y[i-1])));
72     }
73     for(var i=0;i<=n;i++){
74         solution.appendChild(document.createTextNode("y"+i+ " = "+
75             list_of_y[i]));
76         solution.appendChild(document.createElement('br'));
77         console.log("y"+i+ " = "+list_of_y[i]);
78     }
79     list_of_underestimates = []
80     for(var i=1;i<=n;i++){
81         var blad = Math.abs(list_of_y[i] - funct(list_of_x[i]));
82         list_of_underestimates.push(blad);
83         solution.appendChild(document.createTextNode("(Blad y(x"+i+
84             ") = "+blad));
85         solution.appendChild(document.createElement('br'));
86         console.log("(Blad y(x"+i+ ") = "+blad);
87     }
88     solution.appendChild(document.createTextNode('Błąd maksymalny
89         metody: '));
90     solution.appendChild(document.createTextNode(Math.max(...
91         list_of_underestimates)));
92     solution.appendChild(document.createElement('br'));
93     console.log(Math.max(...list_of_underestimates));
94 }
95
96 const heunsMethod = (n,a,b)=>{
97     solution.appendChild(document.createTextNode("Heun's method"));
98     solution.appendChild(document.createElement('br'));
99     console.log("Heun's method");
100     h = countH(a,b,n);
101     list_of_x = [];

```

```

96     list_of_x.push(a);
97     solution.appendChild(document.createTextNode("x"+0+ " = "+
100         list_of_x[0]));
98     solution.appendChild(document.createElement('br'));
99     console.log("x"+0+ " = "+list_of_x[0]);
100     for(var i=1;i<=n;i++){
101         list_of_x.push(list_of_x[i-1]+h);
102         solution.appendChild(document.createTextNode("x"+i+ " = "+
103             list_of_x[i]));
104         solution.appendChild(document.createElement('br'));
105         console.log("x"+i+ " = "+list_of_x[i]);
106     }
107     list_of_y = [];
108     list_of_y.push(func(a));
109     for(var i=1;i<=n;i++){
110         list_of_y.push(list_of_y[i-1]+(h/2)*(dfunct(list_of_x[i-1],
111             list_of_y[i-1])+dfunct(list_of_x[i-1]+h,h*dfunct(
112                 list_of_x[i-1],list_of_y[i-1]))));
113     }
114     for(var i=0;i<=n;i++){
115         solution.appendChild(document.createTextNode("y"+i+ " = "+
116             list_of_y[i]));
117         solution.appendChild(document.createElement('br'));
118         console.log("y"+i+ " = "+list_of_y[i]);
119     }
120     list_of_underestimates = []
121     for(var i=1;i<=n;i++){
122         var blad = Math.abs(list_of_y[i] - funct(list_of_x[i]));
123         list_of_underestimates.push(blad);
124         solution.appendChild(document.createTextNode("(Blad y(x"+i+
125             ") = "+blad));
126         solution.appendChild(document.createElement('br'));
127         console.log("(Blad y(x"+i+") = "+blad);
128     }
129     solution.appendChild(document.createTextNode('Błąd maksymalny
130         metody: '));
131     solution.appendChild(document.createTextNode(Math.max(...
132         list_of_underestimates)));
133     console.log(Math.max(...list_of_underestimates));
134 }
135
136 const countH=(a,b,n)=>{
137     return (b-a)/n;
138 }
139
140 const dfunct = (x,y)=>{
141     return 2*(y-2*x-1)+2;
142 }
143
144 const funct = (x) =>
145 {
146     return 3*x*x+2*x+1
147 }
148
149 function solve(){
150     let n = document.getElementById("n").value;

```

```

145     let b = document.getElementById("b").value;
146     eulerMethod(n,1,b);
147     modifiedEulerMethod(n,1,b);
148     heunsMethod(n,1,b);
149 }

```

3.3.2 HTML

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-
6          scale=1.0">
7      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com
8          /bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-
9          gg0yR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
10          iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
11      <title>Metody numeryczne dla zagadnień różniczkowych</title>
12  </head>
13  <body>
14      <nav class="navbar navbar-dark bg-dark">
15          <span class="navbar-text text-white mx-auto h2">
16              Metody numeryczne dla zagadnień różniczkowych
17          </span>
18      </nav>
19      <div class="card mt-5 w-50 mx-auto">
20          <div class="card-header h4">
21              <div>Zagadnienie różniczkowe:  $y'(x) = f(x, y(x))$ ,  $y(1) =$ 
22              6, gdzie  $f(x, y(x)) = 2(y^2x^?1) + 2$ </div>
23          <br>
24          <div class="inline-block">
25              <form class="form-inline">
26                  <div class="form-group">
27                      <label for="n">Liczba n >=1, określająca podział
28                      odcinka [a, b]</label>
29                      <input type="number" id="n" class="form-control
30                          mx-sm-3">
31                  </div>
32                  <div class="form-group">
33                      <label for="b">Liczba b, informująca o końcu
34                      odcinka [a, b]</label>
35                      <input type="number" id="b" class="form-control
36                          mx-sm-3">
37                  </div>
38                  <a id="solve" class="btn btn-dark text-white" style
39                      ="cursor: pointer;" onclick="solve()">Rozwiąż</
40                      a>
41              </form>
42          </div>
43      </div>
44      <div class="card-body">
45          <div id="solution"></div>

```



```
38         </div>
39
40
41     </div>
42     <script src="src/main.js"></script>
43 </body>
44 </html>
```