

# Interpolacja wielomianowa

Kamil Górny, Wojciech Jachimowski, Mateusz Jendernal, Dominika Glugla

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>3</b>
1.1	Opis implementacji algorytmu realizującego metodę . . . . .	3
<b>2</b>	<b>Działanie programu</b>	<b>4</b>
<b>3</b>	<b>Opis programu</b>	<b>6</b>
3.1	Struktura programu . . . . .	6
3.2	Wejście-wyjście . . . . .	6
3.3	Kod programu . . . . .	7
3.3.1	JavaScript . . . . .	7
3.3.2	HTML . . . . .	11
3.3.3	CSS - fragment . . . . .	12

# 1 Wstęp

Treść zadania: "Program, który przewidzi rozwój przestępczości na świecie. Węzły mają przedstawiać liczbę popełnianych przestępstw na świecie zmieniającą się w czasie."

Wykorzystywana metoda numeryczna: metoda Hermite'a (postać Newtona z wykorzystaniem węzłów dwukrotnych)

Interpolacja Hermite'a pozwala na znalezienie wielomianu przybliżającego, gdy spełnione są warunki:

$$\begin{aligned}\phi^{(i)}(x_k) &= f^{(i)}(x_k) \\ k &= 0, 1, \dots, n \\ i &= 0, 1, \dots, m, \\ \text{gdzie } m &\leq n\end{aligned}$$

„Niech  $x_0 < x_1 < \dots < x_n$  i każdy  $x_j$  ma krotność  $m_j \leq 1$ . Definiujemy  $t_0 \leq t_1 \leq \dots \leq t_m$ , gdzie  $m+1 = \sum_{j=0}^n m_j$ , jako  $t_0 = t_1 = \dots = t_{m_0-1} = x_0$ ,  $t_{m_0} = t_{m_0+1} = \dots = t_{m_0+m_1-1} = x_1$ , ...” (cyt. z wykładu dr Danuty Jaruszewskiej-Właczak)

$f[t_i, t_{i+1}, \dots, t_{i+n}]$  wyliczamy w zależności: jeżeli  $t_i = t_{i+n}$ , zakładając, że  $t_i$  to pierwszy element, a  $t_{i+n}$  to ostatni element, możemy wyrazić za pomocą poniższego wzoru:

$$f[t_i, t_{i+1}, \dots, t_{i+n}] = \frac{f^{(n)}(t_i)}{n!}$$

Jeżeli natomiast  $t_i$  jest różne od  $t_{i+n}$ , to wyrażamy to w następującej rekurencyjnej formule:

$$f[t_i, t_{i+1}, \dots, t_{i+n}] = \frac{f[t_{i+1}, \dots, t_{i+n}] - f[t_i, \dots, t_{i+n-1}]}{t_{i+n} - t_i}$$

Przykład. Interpolujemy dane:

$x_i$	1	2
$f(x_i)$	4	6
$f'(x_i)$	5	3

Współczynniki dla  $P(x) = f[x_0] + f[x_0, x_1](x-x_0) + f[x_0, x_1, x_2](x-x_0)(x-x_1) \dots$  wyznaczymy za pomocą ilorazów różnicowych:

Otrzymujemy:  $b_0 = 4, b_1 = 5, b_2 = -3, b_3 = 4$ , gdzie  $b_i$  to odpowiednio  $f[x_0, \dots, x_i]$

## 1.1 Opis implementacji algorytmu realizującego metodę

Algorytm jest algorytmem rekurencyjnym, który dla różnych danych wejściowych rozgałęzia się na dwie ścieżki w zależności czy  $t_0 = t_m$  czy nie (gdzie  $t_0$  to pierwszy element, a  $t_m$  jest elementem ostatnim. Algorytm zwraca listę

elementów  $b(i)$ . Algorytm wylicza wartość dla różnych danych wejściowych, počzawszy od danych bez pochodnych, kończąc na danych z wieloma pochodnymi. W pierwszym formularzu należy wpisać  $x_i$ , w drugim natomiast  $f(x_i)$ , a w trzecim w zależności od liczby pochodnych: nic lub pierwszą pochodną, lub każdą kolejną pochodną, oddzielając je spacjami.

## 2 Działanie programu

Interpolacja Wielomianowa - metoda Hermite'a

Proгноza rozwoju przestępczości

Wczytaj z pliku

Przeglądaj...

Rok	Ilość Przestępcstw	Wartość Pochodnej
1	2	
2	3	

$b[0] = 2, b[1] = 1,$

Rys. 1: Dodawanie wartości bez pochodnych

Interpolacja Wielomianowa - metoda Hermite'a

Proгноza rozwoju przestępczości

Wczytaj z pliku

Przeglądaj...

Rok	Ilość Przestępcstw	Wartość Pochodnej
3	2	1
5	6	7

$b[0] = 2, b[1] = 1, b[2] = 0.5, b[3] = 1,$

Rys. 2: Dodawanie wartości z jedną pochodną

## Interpolacja Wielomianowa - metoda Hermite'a

Prognoza rozwoju przestępczości

Wczytaj z pliku

Przeglądaj...

Rok	Ilość Przestępstw	Wartość Pochodnej	
1	2	7	<input type="button" value="Usuń"/>
2	3	4	<input type="button" value="Usuń"/>
5	6	1 2	<input type="button" value="Usuń"/>

$b[0] = 2, b[1] = 7, b[2] = -6, b[3] = 9, b[4] = -2.5,$   
 $b[5] = 0.7083333333333334, b[6] =$   
 $-0.19791666666666668,$

Rys. 3: Dodawanie wartości z dwiema pochodnymi

## Interpolacja Wielomianowa - metoda Hermite'a

Prognoza rozwoju przestępczości

Wczytaj z pliku

Przeglądaj...

Wprowadź liczbę

Rok	Ilość Przestępstw	Wartość Pochodnej
-----	-------------------	-------------------

Rys. 4: Dodawanie niewłaściwego formatu danych

## Interpolacja Wielomianowa - metoda Hermite'a

Prognoza rozwoju przestępczości

Wczytaj z pliku

Przeglądaj...

abc × Ilość przestępstw Wartość pochodnej Dodaj

Wprowadź liczbę

Rok	Ilość Przestępstw	Wartość Pochodnej
-----	-------------------	-------------------

Rys. 5: Dodawanie niewłaściwego formatu danych

### 3 Opis programu

#### 3.1 Struktura programu

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

#### 3.2 Wejście-wyjście

Wejście programu to dane w postaci  $x_i$ ,  $f(x_i)$ ,  $f'(x_i)$ ,..., a po dodaniu dowolnej liczby zmiennych w owej postaci program na wyjściu wyświetla kolejno współczynniki  $b$ , które służą do wyznaczanie wielomianu ze wzoru podanego w rozdziale 1. Dane mogą zostać wprowadzone przez formularz przeglądarki lub poprzez wczytanie pliku. Zawartość pliku należy podać w odpowiednim formacie, czyli traktując jeden rząd jako wektor zmiennych, które oddzielone są spacjami; np. 0 1 2 dla logiki programu będzie traktowane jako:  $x_0 = 0$ ,  $f(x_0) = 1$ ,  $f'(x_0) = 2$ . Aby przeciwdziałać wprowadzeniu nieprawidłowych danych w formularzu ustawione są typy pól, które uniemożliwiają wpisywanie niepoprawnych danych. Dane dodawane i wyświetlane są po wciśnięciu przycisku "dodaj". Dane wprowadzone przez użytkownika muszą być wprowadzone w taki sposób, aby algorytm mógł wyliczyć z nich wielomian, tzn. musi występować odpowiednia liczba elementów dodawanych.

## 3.3 Kod programu

### 3.3.1 JavaScript

```
1 let form = document.querySelector('.form-inline')
2 let year = document.getElementById('rok')
3 let crime_number = document.getElementById('iloscPrzestepstw')
4 let derivative = document.getElementById('pochodna')
5 let list = document.getElementById('tableBody')
6 var data = []
7 let polyniomaBox = document.getElementById('polyniomial')
8 form.addEventListener('submit', createList);
9 let input = document.querySelector('fileselector')
10 function readSingleFile(e) {
11     document.getElementById("add").disabled = true;
12     var file = e.target.files[0];
13     if (!file) {
14         return;
15     }
16     var reader = new FileReader();
17     reader.onload = function(e){
18         var content = e.target.result;
19         displayContents(content);
20     };
21
22     reader.readAsText(file);
23
24 }
25
26 function displayContents(content){
27     var lines = content.split(/\r\n|\n/);
28
29     for (i=0; i<lines.length;i++){
30         var number = lines[i].split(" ");
31         var year = number[0];
32         var crime_number = number[1];
33         var deriv = number[2];
34         createListFromFile(year, crime_number, deriv);
35     }
36
37
38 }
39
40 document.getElementById('file-selector').addEventListener('change',
    readSingleFile, false);
41
42 function deleteRow(o){
43
44     var p = o.parentNode.parentNode;
45     var year = p.childNodes[0].firstChild.nodeValue
46     var crime_number = p.childNodes[1].firstChild.nodeValue
47     var derivative = p.childNodes[2].firstChild.nodeValue
48     for(i=0; i<data.length; i++){
49         if(data[i][0]==year && data[i][1]==crime_number && data[i]
50             ] [2]==derivative){
51             data.splice(i,1)
```

```

51         console.log("Deleted values: [" + year + ", " + crime_number +
                    derivative+"]")
52     }
53 }
54
55 p.parentNode.removeChild(p);
56 console.log("Array after deletion: " + data)
57 if (data.length == 0){
58     document.getElementById("add").disabled = false;
59 }
60
61 }
62
63 function createListFromFile(year, crime_number, derivative){
64
65     if(year != '' && crime_number != ''){
66         let list_row = document.createElement('tr')
67         list_row.innerHTML = "<td>" + year + "</td>" + "<td>" + crime_number
        + "</td>" + "<td>" + derivative + "</td>" + "<td>" + "<button
        type='\"button\"' class='\"btn btn-danger\"' onclick='\"
        deleteRow(this)\">Usar</button>" + "</td>"
68         list.appendChild(list_row)
69         var temp_table = []
70         if (derivative != ""){
71             temp_table = derivative.split(" ")
72         }
73         var temp = []
74         if (temp_table.length != 0){
75             for(var element of temp_table){
76                 temp.push(parseInt(element))
77             }
78             data.push([parseInt(year), parseInt(crime_number), temp])
79         }
80         else{
81             data.push([parseInt(year), parseInt(crime_number)])
82         }
83
84         console.log("Added values: [" + year + ", " + crime_number + ", " +
                    derivative+"]")
85         console.log("Array after addition: " + data);
86
87         if(data.length == 2)
88             interpolation(data)
89     }
90 }
91
92
93
94
95 }
96
97 function createList(e){
98     e.preventDefault();
99
100     // console.log(typeof year.value)
101     if(year != '' && crime_number != ''){
102         let list_row = document.createElement('tr')

```



```

103     list_row.innerHTML+"<td>"+year.value+"</td>"+<td>"+
        crime_number.value+"</td>"+<td>"+derivative.value+"</
        td>"+<td>"+<button type=\"button\" class=\"btn btn-
        danger\" onclick=\"deleteRow(this)\">Usuar</button>"+</
        td>
104     list.appendChild(list_row)
105     var temp_table =[]
106     if (derivative.value != ""){
107         temp_table=derivative.value.split(" ")
108     }
109     var temp = []
110     if (temp_table.length != 0){
111         for(var element of temp_table){
112             temp.push(parseInt(element))
113         }
114         data.push([parseInt(year.value), parseInt(crime_number.
            value), temp])
115     }
116     else{
117         data.push([parseInt(year.value), parseInt(crime_number.
            value)])
118     }
119
120     console.log("Added values: ["+year.value+", "+crime_number.
        value+", "+derivative.value+"]")
121     console.log("Array after addition: "+data);
122
123     if(data.length==2)
124         interpolation(data)
125
126 }
127
128
129
130 }
131
132 function interpolation(data){
133     var data_cloning = []
134     for (var ele of data){
135         var j=0
136         if (ele.length==3){
137             j = ele.length-1+(ele[2].length-1)
138         }
139         else{
140             j = ele.length-1
141         }
142         while(j>0){
143             data_cloning.push(ele)
144             j--
145         }
146     }
147
148     for (var ele of data_cloning){
149         console.log(ele)
150     }
151     b=[]
152     var i=data_cloning.length-1

```

```

153     var len=0
154     while(i >= len){
155         b.push(iloraz_roznicowa(data_cloning.slice(0,data_cloning.
156             length-i)))
157         i--
158     }
159     display_status(b,data_cloning)
160 }
161
162 function display_status(b,t){
163     j=0
164     for (var i of b){
165         console.log("b["+ j++ +"] = "+i)
166     }
167     lista_znakow=[]
168     if ((b[0]-b[1]*t[0][0]+b[2]*t[0][0]*t[1][0]+b[3]*t[0][0]*t
169         [2][0]-
170         b[3]*t[0][0]*t[1][0]*t[2][0])>= 0){
171         lista_znakow.push("+")
172     }
173     else{
174         lista_znakow.push("")
175     }
176     if (b[1]-b[1]*t[1][0]-b[2]*t[0][0]+b[3]*t[1][0]*t[2][0] >= 0){
177         lista_znakow.push("+")
178     }
179     else{
180         lista_znakow.push("")
181     }
182     if (b[2]-t[2][0]*b[3]-b[3]*t[1][0]>=0){
183         lista_znakow.push("+")
184     }
185     else{
186         lista_znakow.push("")
187     }
188     var polynomial = "P(x) = "+(b[3])+"(x^3) "+lista_znakow[2]+" "
189     +(b[2]-t[2][0]*b[3]-b[3]*t[1][0]-t[0][0]*b[3])+"(x^2) "+
190     lista_znakow[1]+
191     "(b[1]-b[1]*t[1][0]-b[2]*t[0][0]+b[3]*t[1][0]*t[2][0])+"x "+
192     lista_znakow[0]+" "+(b[0]-b[1]*t[0][0]+b[2]*t[0][0]*t
193     [1][0]+b[3]*t[0][0]*t[2][0]-
194     b[3]*t[0][0]*t[1][0]*t[2][0])
195     printPolynomial(polynomial)
196 }
197
198 function iloraz_roznicowa(data){
199     if (data.length >=2 ){
200         if(data[0][0]==data[data.length-1][0]){
201             return(data[0][2][data.length-2]/factorial(data.length
202                 -1))
203         }

```

```

203     else{
204         return iloraz_roznicowy_rozny(data)
205     }
206 }
207 else{
208     return data[0][1]
209 }
210 }
211 function iloraz_roznicowy_rozny(data){
212     return (iloraz_roznicowa(data.slice(1)) - iloraz_roznicowa(data
        .slice(0,data.length-1)))/(data[data.length-1][0]-data
        [0][0])
213 }
214
215 function factorial(n){
216     if(n==0){
217         return 1
218     }
219     else{
220         return n * factorial(n-1)
221     }
222 }
223
224 function printPolynomial(polyniomial){
225     polyniomaBox.innerHTML+"<h1>" + polyniomial + "</h1>"
226 }

```

### 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-
        scale=1.0">
6      <link rel="stylesheet" href="css/bootstrap.css">
7
8      <title>Interpolacja Wielomianowa</title>
9  </head>
10 <body>
11     <nav class="navbar navbar-dark bg-dark">
12         <span class="navbar-text text-white mx-auto h2">
13             Interpolacja Wielomianowa - metoda Hermite'a
14         </span>
15     </nav>
16
17     <div class="card mt-5 w-50 mx-auto">
18         <div class="card-header h4">
19             Prognoza rozwoju przestępczości
20         </div>
21         <div class="mx-3">
22             <p>Wczytaj z pliku</p>
23             <input type="file" id="file-selector" accept=".txt, .
                json">
24         </div>
25     <div class="card-body">

```

```

26     <form class="form-inline">
27         <label class="sr-only" for="rok">Rok</label>
28         <input type="number" class="form-control mb-2 mr-sm
29             -2" id="rok" placeholder="Rok">
30
31         <label class="sr-only" for="iloscPrzestepstw">Ilość
32             przestępstw</label>
33         <input type="number" class="form-control mb-2 mr-sm
34             -2" id="iloscPrzestepstw" placeholder="Ilość
35             przestępstw">
36
37         <label class="sr-only" for="iloscPrzestepstw">
38             Wartość pochodnej pierwszego stopnia</label>
39         <input type="text" class="form-control mb-2 mr-sm
40             -2" id="pochodna" placeholder="Wartość
41             pochodnej">
42
43         <button type="submit" id="add" class="btn btn-
44             primary mb-2">Dodaj</button>
45     </form>
46     <table class="table mt-3">
47         <thead class="thead-light">
48             <tr>
49                 <th scope="col">Rok</th>
50                 <th scope="col">Ilość Przestępstw</th>
51                 <th scope="col">Wartość Pochodnej</th>
52                 <th scope="col"></th>
53             </tr>
54         </thead>
55         <tbody id="tableBody">
56
57             </tbody>
58         </table>
59         <div id="polynomial"></div>
60     </div>
61 </div>
62 <script src="script.js"></script>
63 </body>
64 </html>

```

### 3.3.3 CSS - fragment

```

1  html {
2      font-family: sans-serif;
3      line-height: 1.15;
4      -webkit-text-size-adjust: 100%;
5      -webkit-tap-highlight-color: rgba(0, 0, 0, 0);
6  }
7
8  article, aside, figcaption, figure, footer, header, hgroup, main,
9      nav, section {
10     display: block;
11 }
12
13 body {
14     margin: 0;

```

```

14 font-family: -apple-system, BlinkMacSystemFont, "Segoe UI",
    Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif, "
    Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol", "
    Noto Color Emoji";
15 font-size: 1rem;
16 font-weight: 400;
17 line-height: 1.5;
18 color: #212529;
19 text-align: left;
20 background-color: #fff;
21 }
22
23 [tabindex="-1"]:focus:not(:focus-visible) {
24     outline: 0 !important;
25 }
26
27 hr {
28     box-sizing: content-box;
29     height: 0;
30     overflow: visible;
31 }
32
33 h1, h2, h3, h4, h5, h6 {
34     margin-top: 0;
35     margin-bottom: 0.5rem;
36 }
37
38 p {
39     margin-top: 0;
40     margin-bottom: 1rem;
41 }
42
43 abbr[title],
44 abbr[data-original-title] {
45     text-decoration: underline;
46     -webkit-text-decoration: underline dotted;
47     text-decoration: underline dotted;
48     cursor: help;
49     border-bottom: 0;
50     -webkit-text-decoration-skip-ink: none;
51     text-decoration-skip-ink: none;
52 }
53
54 address {
55     margin-bottom: 1rem;
56     font-style: normal;
57     line-height: inherit;
58 }
59
60 ol,
61 ul,
62 dl {
63     margin-top: 0;
64     margin-bottom: 1rem;
65 }
66
67 ol ol,

```

```
68 ul ul,  
69 ol ul,  
70 ul ol {  
71     margin-bottom: 0;  
72 }  
73  
74 dt {  
75     font-weight: 700;  
76 }  
77  
78 dd {  
79     margin-bottom: .5rem;  
80     margin-left: 0;  
81 }  
82  
83 blockquote {  
84     margin: 0 0 1rem;  
85 }
```