# Interpolacja wielomianowa

Kamil Górny, Wojciech Jachimowski, Mateusz Jendernal, Dominika Glugla

# Spis treści

1	$\mathbf{W}\mathbf{s}^{\dagger}$	tęp Opis implementacji algorytmu realizującego metodę	3
2	Dzi	ałanie programu	4
3	Opi	s programu (	6
	3.1	Struktura programu	ô
	3.2	Wejście-wyjście	ô
	3.3	Kod programu	7
		3.3.1 JavaScript	
		3.3.2 HTML	
		3.3.3 CSS - fragment	2

## 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)

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

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

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

 $f[t_i,t_{i+1},...,t_{i+n}]$  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:

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

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

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

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

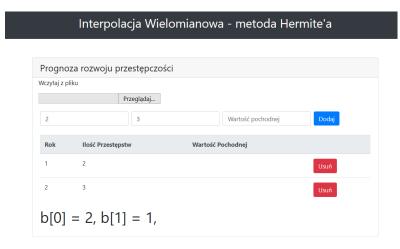
Otrzymujemy:  $b_0 = 4$ ,  $b_1 = 5$ ,  $b_2 = -3$ ,  $b_3 = 4$ , gdzie  $b_i$  to odpowiednio  $f[x_0, ..., x_i]$ 

### 1.1 Opis implementacji algorytmu realizującego metodę

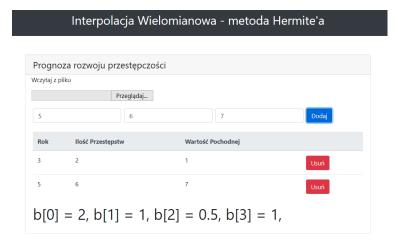
Algorytm jest algorytmem rekurencyjnym, który dla różnych danych wejściowych rozgałęzia się na dwie ścieżki w zależności czy  $t_0=t_m$  czy nie (gdzie  $t_0$  to pierwszy element, a  $t_m$  jest elementem ostatnim. Algorytm zwraca listę

elementów b(i). Algorytm wylicza wartość dla różnych danych wejściowych, począwszy 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

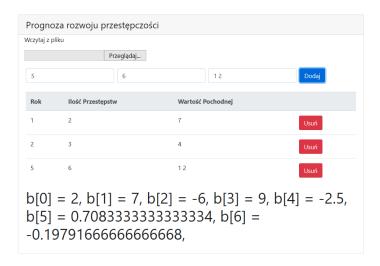


Rys. 1: Dodawanie wartości bez pochodnych



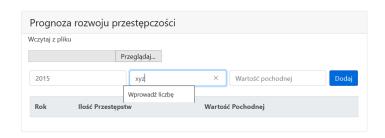
Rys. 2: Dodawanie wartości z jedną pochodną

## Interpolacja Wielomianowa - metoda Hermite'a



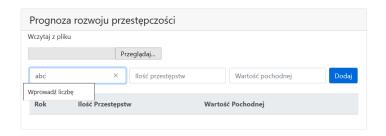
Rys. 3: Dodawanie wartości z dwiema pochodnymi

# Interpolacja Wielomianowa - metoda Hermite'a



Rys. 4: Dodawanie niewłaściwego formatu danych

## Interpolacja Wielomianowa - metoda Hermite'a



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/projektzespolowy-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')
   let derivative = document.getElementById('pochodna')
   let list = document.getElementById('tableBody')
   var data = []
   let polyniomaBox = document.getElementById('polyniomial')
   form.addEventListener('submit', createList);
   let input = document.querySelector('fileselector')
10
   function readSingleFile(e) {
        document.getElementById("add").disabled = true;
11
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++){</pre>
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;
        var year = p.childNodes[0].firstChild.nodeValue
45
46
        var crime_number = p.childNodes[1].firstChild.nodeValue
        var derivative = p.childNodes[2].firstChild.nodeValue
for(i=0; i<data.length; i++){</pre>
47
48
            if(data[i][0] == year && data[i][1] == crime_number && data[i
49
                ][2] == derivative) {
50
                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){
            document.getElementById("add").disabled = false;
58
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=""+year+""+""+crime_number
                 +""+""+""+""+""+"<button
                 type=\"button\" class=\"btn btn-danger\" onclick=\"
                 deleteRow(this)\">Usuń</button>"+""
68
             list.appendChild(list_row)
69
             var temp_table =[]
             if (derivative != ""){
70
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)
        if (year != '' && crime_number !=''){
101
102
            let list_row = document.createElement('tr')
```

```
103
            list_row.innerHTML=""+year.value+""+""+
                 crime_number.value+""+""+derivative.value+"//pre>
                 td>"+""+"<button type=\"button\" class=\"btn btn-
                 danger\" onclick=\"deleteRow(this)\">Usun</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 = []
            if (temp_table.length != 0){
110
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
            console.log("Added values: ["+year.value+", "+crime_number.
120
                 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){
            j = ele.length-1+(ele[2].length-1)
137
138
139
            else{
            j = ele.length-1
140
141
142
            while(j>0){
143
                 data_cloning.push(ele)
144
145
146
            }
147
148
        for (var ele of data_cloning){
149
            console.log(ele)
        }
150
151
        b=[]
        var i=data_cloning.length-1
152
```

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

```
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

```
<!DOCTYPE html>
   <html lang="en">
2
3
   <head>
4
        <meta charset="UTF-8">
5
       <meta name="viewport" content="width=device-width, initial-</pre>
           scale=1.0">
6
       <link rel="stylesheet" href="css/bootstrap.css">
7
8
       <title>Interpolacja Wielomianowa</title>
   </head>
9
10
   <body>
       <nav class="navbar navbar-dark bg-dark">
11
            <span class="navbar-text text-white mx-auto h2">
12
13
               Interpolacja Wielomianowa - metoda Hermite'a
14
            </span>
15
        </nav>
16
        <div class="card mt-5 w-50 mx-auto">
17
18
            <div class="card-header h4">
19
                Prognoza rozwoju przestępczości
20
              </div>
              <div class="mx-3">
21
22
                Wczytaj z pliku
                <input type="file" id="file-selector" accept=".txt, .</pre>
23
                    json">
24
              </div>
25
            <div class="card-body">
```

```
26
              <form class="form-inline">
27
                 <label class="sr-only" for="rok">Rok</label>
                 <input type="number" class="form-control mb-2 mr-sm</pre>
28
                     -2" id="rok" placeholder="Rok">
29
30
                 <label class="sr-only" for="iloscPrzestepstw">Ilość
                      przestępstw</label>
                  <input type="number" class="form-control mb-2 mr-sm</pre>
31
                     -2" id="iloscPrzestepstw" placeholder="Ilość
                     przestępstw">
32
                 <label class="sr-only" for="iloscPrzestepstw">
33
                     Wartość pochodnej pierwszego stopnia</label>
34
                 <input type="text" class="form-control mb-2 mr-sm</pre>
                     -2" id="pochodna" placeholder="Wartość
                     pochodnej">
35
36
                 <button type="submit" id="add" class="btn btn-</pre>
                     primary mb-2">Dodaj</button>
37
              </form>
38
              39
                 <thead class="thead-light">
40
                   Rok
41
42
                     Ilość Przestępstw
                     Wartość Pochodnej
43
44
                     45
                   46
                 </thead>
47
                 48
                 49
50
                51
                <div id="polyniomial"></div>
52
          </div>
53
        </div>
54
        <script src="script.js"></script>
55
   </body>
56
   </html>
```

#### 3.3.3 CSS - fragment

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

```
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;
     line-height: 1.5;
17
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
   hr {
27
   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] {
     text-decoration: underline;
      -webkit-text-decoration: underline dotted;
46
47
     text-decoration: underline dotted;
48
     cursor: help;
     border-bottom: 0;
49
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,
   d1 {
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  }
```