

Лабораторная работа по дисциплине:
«Компьютерная геометрия и графика»

Тема работы:
«Расчёт сплайна методом прогонки»

Выполнил:

Студент ИЛиП, 3 курс,
группа ЗЛПБ-ИСиТ19-1
№ зачетной книжки 319014,
Галкин Андрей Андреевич

Санкт-Петербург

2021 г.

Можно задать произвольное количество точек для построения сплайна:

```
11 // Можно задать произвольное количество точек
12 $x = array(50, 175, 220, 329, 450, 531, 625, 710, 799);
13 $y = array(200, 450, 120, 587, 650, 415, 567, 79, 699);
14 $n = count($x);
```

Строим систему уравнений для определения коэффициентов s , для этого воспользуемся формулой:

$$S(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

$$\frac{h_{i-1}}{3} C_{i-1} + \frac{2}{3}(h_{i-1} + h_i)C_i + \frac{h_i}{3} C_{i+1} = \frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}$$

Данное действие происходит в функции **CreateMatrixC(\$x, \$y, \$n)**:

```
77 //Заполняем матрицу коэффициентами
78 for($j = 0; $j < ($n - 2); $j++){
79     if($j == 0){
80         $i = 1;
81         $SLAE[$j][$i] = 2/3*($h[$i-1] + $h[$i]);
82         $SLAE[$j][$i+1] = $h[$i]/3;
83         $SLAE[$j][$n] = ($y[$i+1] - $y[$i])/ $h[$i] - ($y[$i] - $y[$i-1])/ $h[$i-1];
84     }
85     else if ($j == ($n - 3)){
86         $i = $n - 2;
87         $SLAE[$j][$i-1] = $h[$i-1]/3;
88         $SLAE[$j][$i] = 2/3*($h[$i-1] + $h[$i]);
89         $SLAE[$j][$n] = ($y[$i+1] - $y[$i])/ $h[$i] - ($y[$i] - $y[$i-1])/ $h[$i-1];
90     }
91     else{
92         $i = $j + 1;
93         $SLAE[$j][$i-1] = $h[$i-1]/3;
94         $SLAE[$j][$i] = 2/3*($h[$i-1] + $h[$i]);
95         $SLAE[$j][$i+1] = $h[$i]/3;
96         $SLAE[$j][$n] = ($y[$i+1] - $y[$i])/ $h[$i] - ($y[$i] - $y[$i-1])/ $h[$i-1];
97     }
98 }

119 // Создаём массив h
120 function CreateH($x, $n){
121     $h = array();
122     for($i = 0; $i < $n - 1; $i++){
123         $h[$i] = $x[$i+1] - $x[$i];
124     }
125     return $h;
126 }
```

После небольших преобразований, на выходе получаем СЛАУ в виде трехдиагональной матрицы, решениями которой будут коэффициенты s .

Матрица A - трёхдиагональная

$$A = \begin{pmatrix} -b_1 & c_1 & 0 & 0 & 0 & \dots & f_1 \\ a_2 & -b_2 & c_2 & 0 & 0 & \dots & f_2 \\ 0 & a_3 & -b_3 & c_3 & 0 & \dots & f_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \dots & a_n & -b_n & f_n \end{pmatrix}$$

Примем следующее допущение: В крайних точках функции линейны, т.е. там нет кривизны, а это значит что вторая производная равна нулю. Тогда c_0 и c_n равны нулю:

$$\begin{aligned} S'_i(x) &= b + 2c(x - x_i) + 3d(x - x_i)^2 \\ S''_i(x) &= 2c + 6d(x - x_i) \end{aligned}$$

$$c_0 = 0 \quad c_n = 0,$$

Решение СЛАУ осуществляется методом прогонки в функции **SolvingSLAE(\$M)**:

```

131 // Для решения СЛАУ
132 function SolvingSLAE ($M) {
133     $nj = count($M);
134     $ni = count($M[0]);
135
136     $y = array();
137     $x = array();
138
139     $a = array();
140     $b = array();
141
142     // Прямой ход
143     $y[0] = $M[0][0];
144     $a[0] = -$M[0][1]/$y[0];
145     $b[0] = $M[0][$ni-1]/$y[0];
146
147     for($i = 1; $i < $nj; $i++){
148         $y[$i] = $M[$i][$i] + ($M[$i][$i-1] * $a[$i-1]);
149         $a[$i] = -$M[$i][$i+1]/$y[$i];
150         $b[$i] = ($M[$i][$ni-1] - $M[$i][$i-1]*$b[$i-1])/ $y[$i];
151     }
152
153     // Обратный ход
154     $x[$nj-1] = $b[$nj-1];
155
156     for($i = ($nj - 2); $i >= 0; $i--){
157         $x[$i] = $a[$i]*$x[$i+1] + $b[$i];
158     }
159
160     return $x;
161 }

```

Далее создаём массив со всеми коэффициентами для всех кубических многочленов (коэффициенты a равны координатам у соответствующих точек, а коэффициенты d и b вычисляются по формулам):

$$a_{i-1} = y_{i-1}; \quad a_i = y_i, \text{ т.е. } a_i = f(x_i)$$

$$d_{i-1} = \frac{(C_i - C_{i-1})}{3h_{i-1}}$$

$$b_{i-1} = \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{h_{i-1}}{3}(C_i + 2C_{i-1})$$

Создание данной матрицы происходит в функции **CreateMatrixABCD(\$x, \$y, \$n, \$c)**:

```

166 // Для создания матрицы всех коэффициентов кубических многочленов
167 function CreateMatrixABCD($x, $y, $n, $c){
168     $h = CreateH($x, $n);
169
170     // Массив коэффициентов a
171     $a = array();
172     for($i = 0; $i < $n; $i++){
173         $a[$i] = $y[$i];
174     }
175
176     // Массив коэффициентов b и d
177     $d = array();
178     $b = array();
179     for($i = 0; $i < $n - 1; $i++){
180         $d[$i] = ($c[$i+1] - $c[$i]) / (3*$h[$i]);
181
182         $b[$i] = ($y[$i+1] - $y[$i]) / $h[$i] - ($h[$i] / 3) * ($c[$i+1] + 2*$c[$i]);
183     }
184
185     // Создаём матрицу со всеми коэффициентами
186     $ABCD = array();
187
188     for($i = 0; $i < count($c)-1; $i++){
189         $ABCD[$i][0] = $a[$i];
190         $ABCD[$i][1] = $b[$i];
191         $ABCD[$i][2] = $c[$i];
192         $ABCD[$i][3] = $d[$i];
193     }
194
195     return $ABCD;
196 }

```

Зная коэффициенты многочлена для промежутков между каждых двух точек строим график функции (построение происходит в функции **DrawSpline(\$f, \$x_min, \$x_max)**):

```

201 // Построения сплайна между 3-мя точками
202 function DrawSpline($f, $x_min, $x_max){
203     global $img, $black;
204
205     $s = ($x_max - $x_min) / 200;
206     $x1 = $x_min;
207
208     do{
209         $x2 = $x1 + $s;
210
211         $y1 = F($f, $x1, $x_min);
212         $y2 = F($f, $x2, $x_min);
213
214         imageLine($img, X($x1), Y($y1), X($x2), Y($y2), $black);
215
216         $x1 = $x2;
217     }while($x1 < $x_max);
218 }
219
220 // функция кубического четырёхчлена для построения сплайна
221 function F($f, $x, $x0){
222     $y = $f[0] + $f[1]*($x-$x0) + $f[2]*pow(($x-$x0), 2) + $f[3]*pow(($x-$x0), 3);
223     return $y;
224 }
225

```

Пример работы программы:

