

Projeto Final de Análise Numérica

Alisson Cordeiro e Gabriel Gaspar

1 Introdução

Este projeto tem como objetivo apresentar o estudo de Interpolação Polinomial, para a matéria de Análise Numérica, do 3º semestre do curso de Ciência da Computação do Instituto Federal do Paraná - Campus Pinhais. A Interpolação Polinomial é uma técnica utilizada para encontrar um polinômio que passa por um conjunto de pontos dados, permitindo a estimativa de valores intermediários e a análise do comportamento da função.

2 Código-fonte

```
1 #include <iostream>
2 #include <iomanip>
3 #include <cmath>
4 using namespace std;
5
6 void pivoteamentoParcial(double** matriz, int n, int coluna) {
7     int linhaMax = coluna;
8     for (int i = coluna + 1; i < n; ++i) {
9         if (abs(matriz[i][coluna]) > abs(matriz[linhaMax][coluna])) {
10             linhaMax = i;
11         }
12     }
13     if (linhaMax != coluna) {
14         for (int j = 0; j <= n; ++j) {
15             double temp = matriz[coluna][j];
16             matriz[coluna][j] = matriz[linhaMax][j];
17             matriz[linhaMax][j] = temp;
18         }
19     }
20 }
21
22 int main() {
23     int n;
24     cin >> n;
25
26     double** matriz = new double*[n];
27
28     for (int i = 0; i < n; ++i) {
29         double x, y;
30         cin >> x >> y;
31
32         matriz[i] = new double[n + 1];
33         for (int j = 0; j < n; ++j)
34             matriz[i][j] = pow(x, j);
35
36         matriz[i][n] = y;
37     }
38
39     for (int i = 0; i < n; ++i) {
40         pivoteamentoParcial(matriz, n, i);
41
42         for (int j = i + 1; j < n; ++j) {
43             double fator = matriz[j][i] / matriz[i][i];
44             for (int k = i; k <= n; ++k) {
45                 matriz[j][k] -= fator * matriz[i][k];
46             }
47         }
48     }
49
50     double* resultado = new double[n];
```

```

51     for (int i = n - 1; i >= 0; --i) {
52         resultado[i] = matriz[i][n];
53         for (int j = i + 1; j < n; ++j)
54             resultado[i] -= matriz[i][j] * resultado[j];
55         resultado[i] /= matriz[i][i];
56     }
57
58     for (int i = 0; i < n; ++i) {
59         cout << resultado[i] << endl;
60     }
61
62     for (int i = 0; i < n; ++i)
63         delete[] matriz[i];
64     delete[] matriz;
65     delete[] resultado;
66
67     return 0;
68 }

```

3 Utilização do programa

Para compilar e utilizar o programa, foi usado o seguinte código:

```

1  g++ -o interpola interpola.cpp

```

Para uso do programa, foi usado um arquivo de entrada, contendo os valores necessários, sendo o primeiro valor, a quantidade de pontos conhecidos e o restante sendo os pares ordenados representando os pontos. Um exemplo de entrada está disponível abaixo, em [Resultados](#).

```

1  ./interpola < entrada.txt

```

A saída do programa está de acordo com as especificações do trabalho, com os valores sendo exibidos em ordem crescente do grau da variável, sendo o primeiro valor referente a x^0 , seguido por x^1 , x^2 , até x^{n-1} .

4 Resultados

Usando a tabela abaixo como entrada, o número 5 representa a quantidade de pontos e o resto são os pontos em si. Com isso, o programa retorna a função abaixo da tabela.

5	
1	2
1.3	4.4
1.7	4.56
1.9	5
2.3	7

$$f(x) = -10,561661x^4 + 77,710623x^3 - 207,739316x^2 + 241,080037x - 98,489683$$

A figura abaixo mostra o gráfico da função interpoladora, com os cinco pontos previamente dados plotados sobre ele para verificar o ajuste:

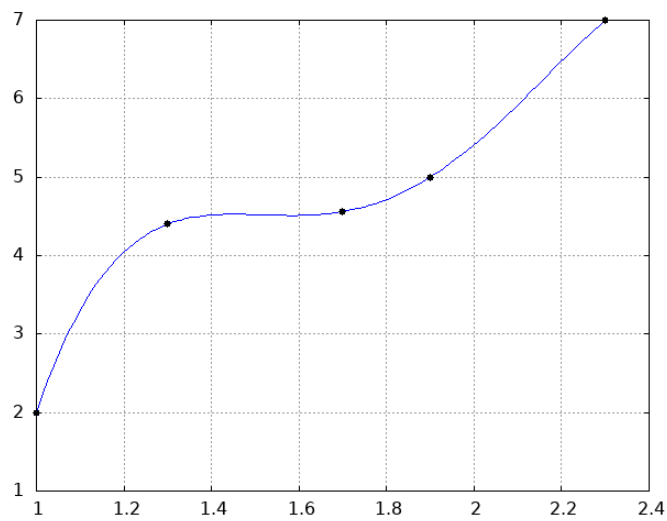


Figura 1: Gráfico da função com os pontos dados previamente