

1 Introdução

Sua tarefa é implementar um programa que faça a interpolação de pontos através da resolução de um sistema linear de equações.

O trabalho pode ser feito individualmente ou em duplas.

Os métodos de interpolação são usados para encontrar um polinômio que passe pelos pontos conhecidos de uma função f que é desconhecida. Para interpolar $n + 1$ pontos, podemos criar uma matriz de Vandermonde.

Pontos conhecidos da função f :

- $(x_0, f(x_0))$
- $(x_1, f(x_1))$
- ...
- $(x_n, f(x_n))$

\Rightarrow

Matriz de Vandermonde correspondente:

$$\left(\begin{array}{cccccc|c} 1 & x_0 & x_0^2 & x_0^3 & \dots & x_0^n & f(x_0) \\ 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^n & f(x_1) \\ 1 & x_2 & x_2^2 & x_2^3 & \dots & x_2^n & f(x_2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^n & f(x_n) \end{array} \right)$$

Ao resolver o sistema acima, você encontrará como solução os coeficientes $a_0, a_1, a_2, \dots, a_n$ do polinômio interpolador de grau n :

$$p_n(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_nx^n \quad (1)$$

2 O seu trabalho

Seu trabalho é ler os $n + 1$ pontos conhecidos da função, construir a matriz de Vandermonde e resolver o sistema linear utilizando *eliminação de Gauss com pivoteamento parcial*.

2.1 Entrada

A entrada do seu programa consiste de uma lista de pontos conhecidos da função a ser interpolada.

A primeira linha contém um inteiro p que informa o número pontos conhecidos. As próximas p linhas descrevem os pontos. Cada ponto possui dois valores reais: o primeiro para o parâmetro/coordenada x e o segundo para o resultado da função ($f(x)$).

2.1.1 Exemplo de entrada

Suponha que conhecemos os 5 pontos a seguir:

- $(1, 2)$
- $(1.3, 4.4)$
- $(1.7, 4.56)$
- $(1.9, 5)$

- (2.3, 7)

A entrada para o programa seria

```
5
1 2
1.3 4.4
1.7 4.56
1.9 5
2.3 7
```

2.2 Saída

A saída do seu programa deve ser o vetor solução com os coeficientes do polinômio interpolador: são p números reais.

2.2.1 Exemplo de saída

Para o exemplo de entrada anterior, o polinômio interpolador é:

$$p_4(x) = -98.49 + 241.08x - 207.739x^2 + 77.711x^3 - 10.562x^4 \quad (2)$$

Portanto, a saída do seu programa deve ser

```
-98.49
241.08
-207.739
77.711
-10.562
```

3 Implementação

Você deve ler e escrever os dados na entrada e saída padrão, de forma que seu programa pode ser executado com o comando:

```
$ ./interpola < entrada.txt > saida.txt
```

Você deve implementar, na linguagem C/C++, a solução para o problema, usando eliminação de Gauss com pivoteamento parcial e retrossubstituição.

3.1 Considerações

1. Você não deve usar bibliotecas que executam as operações do problema;
2. Não use `vector` ou qualquer outro *container* do C++;
3. Seu programa deve funcionar para grandes quantidades de pontos (centenas de milhões e até bilhões de pontos).
4. Você deve usar variáveis do tipo `double` para os pontos e coeficientes.

4 Discussão

O seu relatório deve discutir os resultados que seu programa gera. Para isso, você deve usar a saída que seu programa gera (coeficientes do polinômio interpolador) para criar o gráfico do polinômio. Nesta parte, você pode usar `gnuplot`, `Python` ou outra ferramenta. *Se usar ferramentas que usam código, envie também esse código na entrega do trabalho.*

Você deve plotar o gráfico com o polinômio e os pontos conhecidos e verificar se o polinômio de fato intercepta todos os pontos. Além disso você deve discutir se o polinômio tem um bom comportamento.

Além disso, discuta o erro do seu polinômio em pelo menos 5 pontos aleatórios diferentes dos pontos que foram informados. Faça isso comparando o valor de $p(x)$ com $f(x)$, para os exemplos das seções 6.1 e 6.2.

5 O que entregar

Você deverá entregar, até 07/julho, um arquivo `.zip` contendo uma pasta, que por sua vez contém:

- Um arquivo C/C++ chamado `interpola.c` ou `interpola.cpp` que implementa a sua solução para este trabalho.
- Um arquivo chamado `relatorio.pdf` contendo o nome completo dos integrantes do grupo e as discussões pontuadas nesta especificação.
 - Esse arquivo deve conter uma introdução ao problema, a descrição de como você resolveu e detalhes de implementação relevantes, problemas conhecidos, dificuldades encontradas, os gráficos dos polinômios interpoladores, a discussão do erro para os exemplos requeridos.
- Outros arquivos `.txt` contendo outros exemplos de entrada e saída que o grupo usou para testar o trabalho.

O arquivo compactado descrito acima deve ser nomeado com as iniciais do nome de cada integrante separados por um hífen. Por exemplo: se o Fulano da Silva fez o trabalho com o João de Souza, o arquivo deve ser nomeado `fs-js.zip`. A pasta, dentro do arquivo compactado, deve ter o mesmo nome, a não ser pela extensão.

6 Exemplos de teste

Abaixo estão 3 exemplos para teste do seu programa. Os dois primeiros exemplos possuem a função que gerou os pontos, apenas para a análise de erro.

Note que, por conta dos erros de ponto flutuante, podem haver pequenas diferenças de precisão nos resultados.

6.1 Exemplo 1

6.1.1 Função f que gerou os pontos, para comparação

$$f(x) = x^6 - 2x^5 - 3x^4 + 5x^3 + 2x^2 - 3x + 1$$

6.1.2 Pontos

- $(-1, 1)$
- $(-0.8, 1.808704)$
- $(0.2, 0.514624)$
- $(0.5, 0.390625)$

- (1.3, 0.297649)
- (2.5, 15.765625)

6.1.3 Entrada do programa

```
6
-1 1
-0.8 1.808704
0.2 0.514624
0.5 0.390625
1.3 0.297649
2.5 15.765625
```

6.1.4 Saída do programa

```
0.74000000000000102
-1.4610000000000006
1.6459999999999978
0.4370000000000001
-1.7099999999999989
0.6999999999999956
```

6.1.5 Polinômio interpolador

$$p_5(x) = 0.74000000000000102 - 1.4610000000000006x + 1.6459999999999978x^2 + 0.4370000000000001x^3 - 1.7099999999999989x^4 + 0.6999999999999956x^5$$

6.2 Exemplo 2

6.2.1 Função f que gerou os pontos, para comparação

$$f(x) = x^5 + 3x^4 + 1.5x^3 - 2x^2 - 3x + 2$$

6.2.2 Pontos

- $(-1, 3.5)$
- $(-0.8, 3.25312)$
- $(0.2, 1.33712)$
- $(0.5, 0.40625)$

6.2.3 Entrada do programa

```
4
-1 3.5
-0.8 3.25312
0.2 1.33712
0.5 0.40625
```

6.2.4 Saída do programa

```
1.8479999999999999
-2.358
-0.93599999999999991
-0.22999999999999992
```

6.2.5 Polinômio interpolador

$$p_3(x) = 1.8479999999999999 - 2.358x - 0.93599999999999991x^2 - 0.22999999999999992x^3$$

6.3 Exemplo 3

6.3.1 Pontos

- (2.096564322979868, 9.745443487783383)
- (6.922977157751151, 1.242846887386261)
- (4.220441410259928, 3.789449024006479)
- (1.802878236405755, 0.8368959365195927)
- (1.324377887241789, 5.936317566366855)
- (1.877280316779064, 9.774407080764629)
- (7.557760165820275, 6.712616033725468)
- (4.720106905544402, 1.230350867632133)
- (2.356347202074704, 8.896870895536257)
- (3.905720613358983, 3.180695896049662)
- (2.893813967551298, 9.832444876731104)
- (9.364300979225938, 0.7215618713442351)

6.3.2 Entrada do programa

```
12
2.096564322979868 9.745443487783383
6.922977157751151 1.242846887386261
4.220441410259928 3.789449024006479
1.802878236405755 0.8368959365195927
1.324377887241789 5.936317566366855
1.877280316779064 9.774407080764629
7.557760165820275 6.712616033725468
4.720106905544402 1.230350867632133
2.356347202074704 8.896870895536257
3.905720613358983 3.180695896049662
2.893813967551298 9.832444876731104
9.364300979225938 0.7215618713442351
```

6.3.3 Saída do programa

553845.9855656959
-2003335.478904233
3181892.698628633
-2929692.783456058
1737428.132587893
-696749.125034249
192772.1166684658
-36792.60004408728
4747.063040546158
-394.3310129347495
18.98378542303456
-0.4013834577339842