# Lista 4 De Análise Numérica

Prof. Tatiane Amaral

*Aluno: Patrick Duarte Pimenta*

---

1. **A)**

**"Valor interpolado: "**

**17.877143**

==========

```
clc

/*
Esse código define uma função chamada interpolacaoLagrange que
recebe
os pontos x e y e o valor xx para interpolar. O algoritmo
percorre cada
ponto de x e calcula o polinômio de Lagrange correspondente. Em
seguida, multiplica o valor y do ponto pelo polinômio de
Lagrange e
soma ao valor interpolado. No final, o valor interpolado é
exibido
no console.
*/

function y_interp=interpolacaoLagrange(x, y, xx)
    n = length(x) - 1;  // Grau do polinômio interpolador
    y_interp = 0;  // Valor interpolado inicializado

    for i = 1:n+1
        L = 1;  // Inicializa o polinômio de Lagrange para o
ponto i

        for j = 1:n+1
```

```
            if i ~= j
                L = L * (xx - x(j)) / (x(i) - x(j));  // Atualiza o polinômio de Lagrange
            end
        end

        y_interp = y_interp + y(i) * L;  // Atualiza o valor interpolado
    end

    disp("Valor interpolado: ", y_interp);
end

// Pontos de entrada
x = [
    0.81;
    0.83;
    0.86;
    0.87
    ];
y = [
    16.94410;
    17.56492;
    18.50515;
    18.82091
    ];
xx = 0.84;

// Chamada da função de interpolação de Lagrange
interpolacaoLagrange(x, y, xx);
```

---

1. **B)**

**"Coeficientes do polinômio interpolador:"**

**31.576  5.875  -2.0833333**

**"Valor interpolado:"**

**31.751625**

==========

```
clc

/*
Este código irá calcular os coeficientes do polinômio
interpolador de
Lagrange usando o métod0 de diferenças divididas de Newton e,
em
seguida, avaliar o polinômio interpolador no ponto dado xx. Os
resultados serão exibidos no console.
*/

// Função para calcular o coeficiente do polinômio interpolador
de Lagrange
function coeficientes=coeficientesNewton(x, y)
    n = length(x) - 1;  // Grau do polinômio interpolador

    // Inicialização da tabela de diferenças divididas
    tabelaDD = zeros(n+1, n+1);
    tabelaDD(:,1) = y';

    // Cálculo das diferenças divididas
    for j = 2:n+1
        for i = j:n+1
            tabelaDD(i,j) = (tabelaDD(i,j-1) - tabelaDD(i-1,j-
1)) / (x(i) - x(i-j+1));
        end
    end

    // Coeficientes do polinômio interpolador
    coeficientes = tabelaDD(n+1, 2:n+1);

    // Retornar os coeficientes calculados
    coeficientesNewton = coeficientes;
endfunction

// Função para avaliar o polinômio interpolador de Lagrange no
ponto xx
function valorInterpolado=interpolaNewton(x, coeficientes, xx)
    n = length(coeficientes);  // Grau do polinômio
interpolador
```

```
    // Inicialização do valor interpolado
    valorInterpolado = coeficientes(n);

    // Cálculo do valor interpolado usando a forma de Newton
    for i = n-1:-1:1
        valorInterpolado = coeficientes(i) + (xx - x(i)) *
valorInterpolado;
    end
endfunction

// Pontos x e y
// Pontos de entrada
x = [
    0.81;
    0.83;
    0.86;
    0.87
    ];
y = [
    16.94410;
    17.56492;
    18.50515;
    18.82091
    ];
xx = 0.84;

// Valor para interpolar
xx = 0.84;

// Calcular coeficientes do polinômio interpolador de Lagrange
coeficientes = coeficientesNewton(x, y);

// Interpolar o valor xx usando os coeficientes calculados
valorInterpolado = interpolaNewton(x, coeficientes, xx);

// Exibir resultado
disp("Coeficientes do polinômio interpolador:");
disp(coeficientes);
disp("Valor interpolado:");
disp(valorInterpolado);
```

---

2. **A)**

**"O valor interpolado é:"**

**173.53846**

---

```
clc

// Função para calcular o Polinômio Interpolador de Grau Dois
function y_interp=polinomioInterpoladorGrauDois(x, y, xx)
    // Verifica se os vetores x e y têm o mesmo tamanho
    if length(x) ~= length(y) then
        error("Os vetores x e y devem ter o mesmo tamanho!");
    end

    // Número de pontos de interpolação
    n = length(x);

    // Calcula as diferenças divididas
    for i = 1:n-1
        h(i) = x(i+1) - x(i);
        delta(i) = (y(i+1) - y(i))/h(i);
    end

    // Calcula as diferenças divididas de segunda ordem
    for i = 1:n-2
        delta2(i) = (delta(i+1) - delta(i))/(x(i+2) - x(i));
    end

    // Calcula o valor interpolado
    y_interp = y(1) + delta(1)*(xx - x(1)) + delta2(1)*(xx -
x(1))*(xx - x(2));
end

// Chama a função para calcular o valor interpolado
y_interp = polinomioInterpoladorGrauDois(x, y, xx);

// Ponto x
x = [
    79;
    69;
    82;
    63;
    73
```

```
];

// Ponto y
y = [
    183;
    173;
    188;
    163;
    178
];

// Valor a ser interpolado
xx = 70;

// Exibe o resultado
disp("O valor interpolado é:");
disp(y_interp);
```

---

2. B)

"A estimativa de erro é:"

0.4615385

---

```
// Calcula a estimativa de erro
M = max(abs(delta2));
erro = M*abs(xx - x(1))*abs(xx - x(2));
```

---

2. C)

"Os parâmetros do ajuste são:"

"Alpha (α):"

-467.78523

**"Beta (β):"**

**189.82266**

**"A altura aproximada de uma pessoa com peso de 70 kg é:"**

**-241.79629**

---

```
clc

// Função para ajuste da curva por mínimos quadrados
function [alpha, beta]=ajusteMinimosQuadrados(x, y)
    // Monta a matriz A
    A = [sin(x)', cos(x)'];

    // Calcula a pseudo-inversa de A
    A_pseudo_inv = pinv(A);

    // Calcula os parâmetros alpha e beta
    parametros = A_pseudo_inv * y';

    // Separa os valores dos parâmetros
    alpha = parametros(1);
    beta = parametros(2);
endfunction

// Ponto x
x = [79, 69, 82, 63, 73];

y = [183, 173, 188, 163, 178];

// Chama a função para ajustar a curva e obter os parâmetros
alpha e beta
[alpha, beta] = ajusteMinimosQuadrados(x, y);

// Exibe os valores dos parâmetros
disp("Os parâmetros do ajuste são:");
```

```
disp("Alpha (α):");
disp(alpha);
disp("Beta (β):");
disp(beta);

// Calcula a altura aproximada de uma pessoa com peso de 70
kg
altura_aproximada = alpha*sin(70) + beta*cos(70);

// Exibe a altura aproximada
disp("A altura aproximada de uma pessoa com peso de 70 kg
é:");
disp(altura_aproximada);
```

---

3. A)

"Estimativa de infectados em março de 2002:"

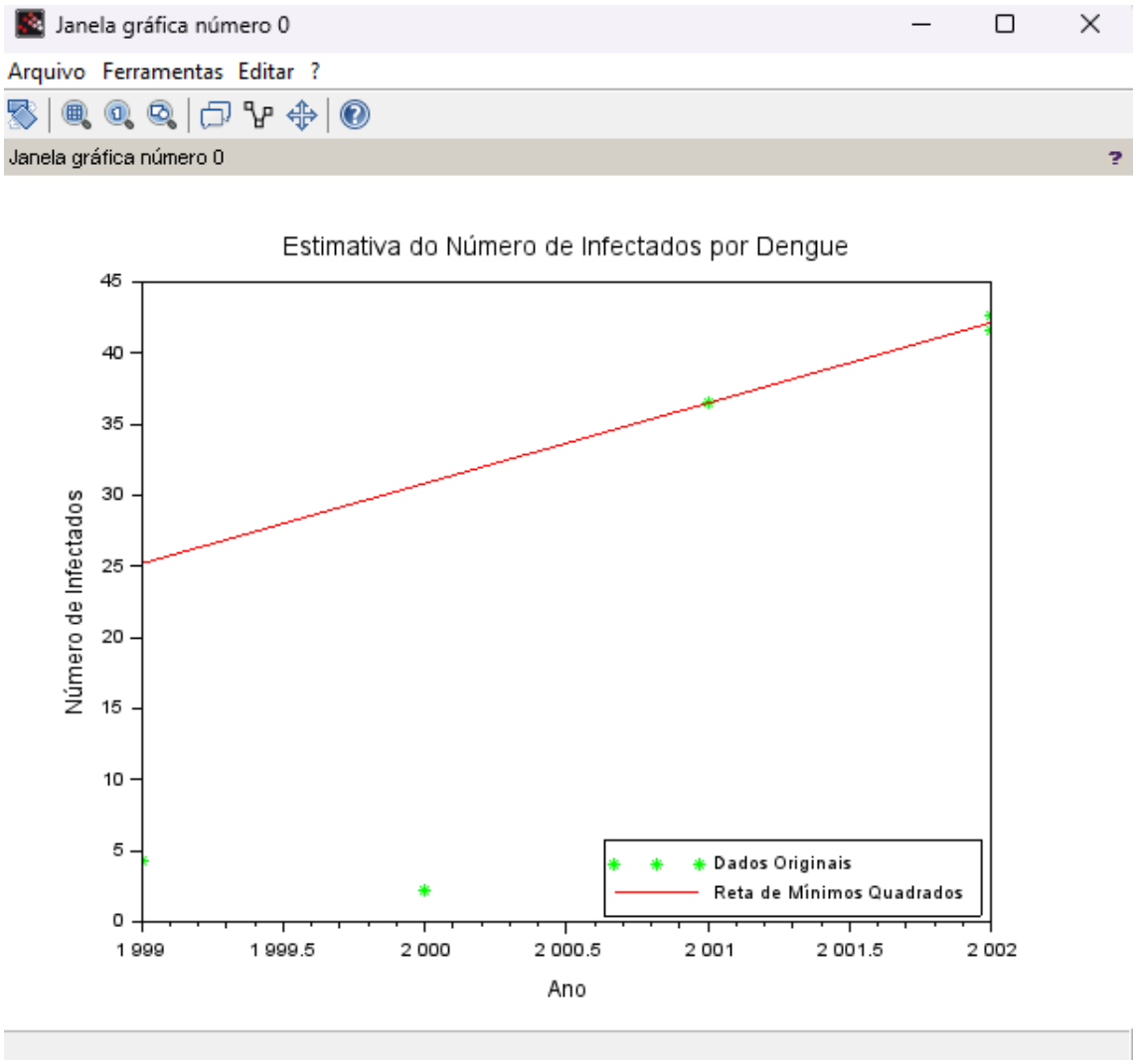
42.150000

---

```
clc

// Definir os dados
data = [1999, 2000, 2001, 2002, 2002];
numeros = [4.300, 2.200, 36.500, 41.600, 42.700];

// Criar uma matriz com os valores x e y
x = data(3:5);   // Seleciona os valores correspondentes a
2001, 2002(1) e 2002(2)
y = numeros(3:5);

// Calcular as somas necessárias
n = length(x);
sum_x = sum(x);
sum_y = sum(y);
sum_xy = sum(x .* y);
sum_x2 = sum(x.^2);

// Calcular os coeficientes da reta de mínimos quadrados
a1 = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x^2);
// Coeficiente angular
```

```
a0 = (sum_y - a1 * sum_x) / n;  // Coeficiente linear

// Estimar o número de infectados em março de 2002
mes_marco_2002 = 2002;  // Mês de março de 2002
numero_infectados_marco_2002 = a0 + a1 * mes_marco_2002;  //
Estima o número de infectados em março de 2002

disp("Estimativa de infectados em março de 2002:");
disp(numero_infectados_marco_2002);  // Exibe o resultado da
estimativa

//Gráfico
// Cria um vetor com os valores x para plotagem da reta
x_plot = linspace(min(data), max(data), 100);

// Calcula os valores y correspondentes para plotagem da
reta
y_plot = a0 + a1 * x_plot;

// Plota os dados originais como pontos
plot(data, numeros, 'g*');

// Plota a reta estimada

plot(x_plot, y_plot, 'r');

// Configurações do gráfico
title('Estimativa do Número de Infectados por Dengue');
xlabel('Ano');
ylabel('Número de Infectados');
legend('Dados Originais', 'Reta de Mínimos Quadrados',
"in_lower_right");
```

---




3. **B)**

**"Estimativa para o número de infectados em fevereiro de 2001: 28.361711"**

---

```
clc

// Dados de entrada
data = [1999, 2000, 2001, 2002, 2002];
numeros = [4.300, 2.200, 36.500, 41.600, 42.700];

// Tamanho dos dados
n = length(data);

// Cálculo das somas
```

```
sum_x = sum(data);
sum_y = sum(numeros);
sum_x2 = sum(data.^2);
sum_x3 = sum(data.^3);
sum_x4 = sum(data.^4);
sum_xy = sum(data .* numeros);
sum_x2y = sum(data.^2 .* numeros);

// Montagem do sistema de equações
A = [n, sum_x, sum_x2; sum_x, sum_x2, sum_x3; sum_x2,
sum_x3, sum_x4];
B = [sum_y; sum_xy; sum_x2y];

// Resolução do sistema de equações
coeff = A \ B;

// Estimativa para fevereiro de 2001
mes_2001_2 = 2001; // Fevereiro
estimativa_fevereiro_2001 = coeff(1) + coeff(2) * mes_2001_2
+ coeff(3) * mes_2001_2^2;

// Resultado
disp("Estimativa para o número de infectados em fevereiro de
2001: " + string(estimativa_fevereiro_2001));
```

---

3. **C) Item c????**

4. **A)**

   **"Velocidade aproximada no ponto de interpolação: 91.333333"**

   **"Erro de interpolação no ponto de interpolação: 0"**

   **"O carro será multado."**

==========

```
clc
```

```
// Dados de entrada
distancia = [0, 0.2, 0.3, 0.5, 0.8, 1.0];
velocidade = [80, 85, 88, 92, 85, 80];

// Ponto onde queremos interpolar
ponto_interpolacao = 0.4;

// Cálculo do Polinômio Interpolador
n = 2; // Grau do polinômio interpolador
x = distancia(1:n+1);
y = velocidade(1:n+1);

A = zeros(n+1, n+1);
B = zeros(n+1, 1);

for i = 1:n+1
    for j = 1:n+1
        A(i, j) = x(i)^(j-1);
    end
    B(i) = y(i);
end

coeficientes = A\B; // Resolvendo o sistema linear

// Avaliação do polinômio interpolador no ponto de interpolação
velocidade_aproximada = 0;
for i = 1:n+1
    velocidade_aproximada = velocidade_aproximada +
coeficientes(i)*ponto_interpolacao^(i-1);
end

// Cálculo do Erro de Interpolação
x_interp = 0.4;
velocidade_real = coeficientes(1);

for i = 2:n+1
    velocidade_real = velocidade_real +
coeficientes(i)*x_interp^(i-1);
end

erro_interp = abs(velocidade_real - velocidade_aproximada);

// Saída dos resultados

// i)
```

```
disp("Velocidade aproximada no ponto de interpolação: " +
string(velocidade_aproximada));

// ii)
disp("Erro de interpolação no ponto de interpolação: " +
string(erro_interp));

// iii) Verificação de multa
velocidade_maxima = 90;

if velocidade_aproximada <= velocidade_maxima then
    disp("O carro não será multado.")
else
    disp("O carro será multado.")
end
```

---

4. **B)**

**"A equação da reta é: y = -0.8411215x + 85.392523"**

**"A velocidade esperada em d = 1.1 é aproximadamente 84.46729 km/h."**

==========

```
clc

distancia = [0, 0.2, 0.3, 0.5, 0.8, 1.0];
velocidade = [80, 85, 88, 92, 85, 80];

// Calcular as médias das distâncias e velocidades
mean_dist = mean(distancia);
mean_vel = mean(velocidade);

// Calcular os valores para os mínimos quadrados
Sxx = sum((distancia - mean_dist).^2);
Sxy = sum((distancia - mean_dist) .* (velocidade - mean_vel));

// Calcular os coeficientes da reta
m = Sxy / Sxx;
b = mean_vel - m * mean_dist;

// Calcular a velocidade esperada em d = 1.1
d = 1.1;
```

```
vel_esperada = m * d + b;

disp("A equação da reta é: y = " + string(m) + "x + " +
string(b));
disp("A velocidade esperada em d = 1.1 é aproximadamente " +
string(vel_esperada) + " km/h.");
```

---

4. **C)**

**"Coeficientes do polinômio de segundo grau:"**

**79.460776**

**42.334734**

**-42.228671**

**"Velocidade esperada em d = 1.1:"**

**74.932293**

---

```
clc

// Dados de distância e velocidade
distancia = [0, 0.2, 0.3, 0.5, 0.8, 1.0];
velocidade = [80, 85, 88, 92, 85, 80];

// Número de pontos de dados
n = length(distancia);

// Matriz de Vandermonde
X = [ones(n, 1), distancia', distancia'.^2];

// Vetor de velocidades
y = velocidade';

// Cálculo dos coeficientes
coefs = (X'*X) \ (X'*y);
```

```
// Cálculo da velocidade esperada em d = 1.1
d = 1.1;
velocidade_esperada = coefs(1) + coefs(2)*d + coefs(3)*d^2;

// Exibição dos resultados
disp("Coeficientes do polinômio de segundo grau:");
disp(coefs);
disp("Velocidade esperada em d = 1.1:");
disp(velocidade_esperada);
```

---

4. **D)**

**A notícia afirma que a estimativa de erro do radar é de 10% e conclui que os carros poderiam andar a uma velocidade máxima de 99 km/h sem serem multados. Essa afirmação parece levar em consideração a margem de erro do radar ao estabelecer um limite de velocidade seguro para evitar multas.**

**No entanto, é importante destacar que a estimativa de erro do radar não é o único fator a ser considerado na determinação dos limites de velocidade e na aplicação de multas. As autoridades de trânsito geralmente estabelecem limites de velocidade máxima com base em critérios de segurança, levando em conta vários fatores, como condições da via, fluxo de tráfego, sinalização e outros aspectos.**

**Embora a margem de erro do radar possa influenciar a precisão das medições de velocidade, os limites de velocidade legalmente estabelecidos geralmente não são definidos com base nessa margem de erro. Em vez disso, eles são definidos com base em critérios de segurança e regulamentações específicas.**

5.

**A interpolação polinomial e o ajuste de curvas pelo método dos mínimos quadrados são duas técnicas diferentes utilizadas para encontrar uma função que se ajuste aos dados de um conjunto de pontos.**

**A interpolação polinomial envolve o cálculo de um polinômio que passa exatamente por todos os pontos dados. Essa técnica é baseada no uso de polinômios de grau n, onde n é o número de pontos dados. A ideia é encontrar os coeficientes do polinômio de modo que ele satisfaça as condições de interpolação, ou seja, que ele passe exatamente por todos os pontos.**

**Por outro lado, o ajuste de curvas pelo método dos mínimos quadrados busca encontrar uma função que minimize a soma dos quadrados das diferenças entre os valores observados e os valores preditos pela função. Essa técnica é utilizada quando não é necessário que a função passe exatamente por todos os pontos, mas sim que se aproxime o melhor possível deles.**

**É possível obter um mesmo polinômio que realize tanto a interpolação como o ajuste de curvas pelo método dos mínimos quadrados, dependendo das características dos pontos dados. Em alguns casos, os pontos podem ser dispostos de forma que um polinômio de grau suficientemente alto consiga passar por todos eles e, ao mesmo tempo, minimizar a soma dos quadrados das diferenças. No entanto, essa coincidência é rara e geralmente é necessário utilizar técnicas específicas para interpolação polinomial e ajuste de curvas separadamente.**

**Em resumo, a interpolação polinomial busca um polinômio que passe exatamente por todos os pontos, enquanto o ajuste de curvas pelo método dos mínimos quadrados busca uma função que se ajuste aos pontos de forma mais geral, minimizando a soma dos quadrados das diferenças. Embora seja possível obter um mesmo polinômio que atenda a ambos os critérios, isso é excepcional e normalmente são utilizadas técnicas diferentes para cada caso.**