MS211 - Cálculo Numérico

Laboratório 07

1 de abril de 2021

Guilherme Nunes Trofino
217276

1 Questão

1.1 Introdução

Problema Notou-se que o exército de Leônidas I era composto por M guerreiros espartanos, cuja distribuição $N(h)$ das alturas pode ser representada pela seguinte equação de distribuição normal:

$$N(h) = \frac{M}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(h-\mu)^2}{2\sigma^2}} \tag{1}$$

Onde:

1. μ: Representa a **Média das Alturas**;
2. σ: Representa o **Desvio Padrão**;

Assim, pode-se obter quantos guerreiros possuem altura entre a e b integrando a distribuição normal no intervalo desejado como representado na equação:

$$N = \int_a^b N(h)dh \tag{2}$$

Determine Toma-se $M = 300$, $\mu = 1,7m$ e $\sigma = 0,1m$.

1. Estime quantos guerreiros possuem altura entre $1,8m$ e $2,0m$?
2. Estime o erro cometido na estimativa.

1.2 Desenvolvimento

Teoria Sabe-se do Cálculo I que uma integral definida será formalmente dada, por definição, pela seguinte equação:

$$\boxed{\int_a^b f(x)dx = \lim_{n\to\infty} \sum_{k=1}^{n} f(x_k) \cdot \Delta x} \tag{3}$$

Onde:

$$\Delta x = \frac{b-a}{n} \text{ e } x_k \in [a,b]$$

Todavia, esta definição pode ser aproximada numericamente através da **Fórmula de Quadratura**. Desta forma temos a seguinte equação para $n+1$ pontos no intervalo desejado:

$$\boxed{\int_a^b f(x)dx = \sum_{k=0}^{n} w_k \cdot f(x_k) + R_{n+1}} \tag{4}$$

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} w_k \cdot f(x_k)$$

Onde:

1. $x_0, x_1, \ldots, x_n \in [a,b]$ são **Nós de Integração**;
2. $w_0, w_1, \ldots, w_n$ são os **Pesos** dos respectivos nós;
3. R_{n+1} será o **Resto** da aproximação;

Fórmula de Newton-Cotes Aplicam-se interpolações polinomiais nos pontos igualmente espaçados, $x_0 < x_1 < \cdots < x_n$, do intervalo fechado $[a,b]$. Assim, os pontos podem ser relacionados através de um polinômio interpolador de f no intervalo:

$$x_k = a + \frac{b-a}{n}k, \quad \forall k = 0, 1, \ldots, n$$

Regra dos Trapézios Repetidas Nesta formulação aproxima-se f por um polinômio linear interpolado por partes $\Lambda(x)$ em $x_0, x_1, \ldots, x_n$. Nesta abordagem tem-se a seguinte equação:

$$\int_a^b f(x)dx = \sum_{k=1}^{n} \int_{x_{k-1}}^{x_k} \Lambda(x)dx$$

$$\int_a^b f(x)dx = \sum_{k-1}^{n} \left(\frac{h}{2} \left(f(x_{k-1}) + f(x_k) \right) - \frac{h^3}{12} f''(\eta_k) \right)$$

$$\boxed{\int_a^b f(x)dx = \underbrace{\frac{h}{2} \sum_{k-1}^{n} \left(f(x_{k-1}) + f(x_k) \right)}_{T_n(f)} - \underbrace{\frac{h^3}{12} \sum_{k-1}^{n} f''(\eta_k)}_{R_T}} \tag{5}$$

Aproximação Desta maneira, pode-se representar uma aproximação numérica para integral desejada, através dos **Trapézios Repetidos**, pode ser calculado pela seguinte equação:

$$\boxed{T_n(f) = \frac{h}{2} \left(f(x_0) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(x_n) \right)}$$

Erro Desta maneira, pode-se representar uma aproximação numérica para o erro da integral, através dos **Trapézios Repetidos**, pode ser calculado pelas seguintes equações:

$$\boxed{R_T \leq \frac{b-a}{12} h^2 M_2}$$

O teorema do valor intermediário garante que, dado f'' contínua em $[a, b]$, existe $\epsilon \in (a, b)$ tal que o **Majorante** seja obtido pela seguinte equação:

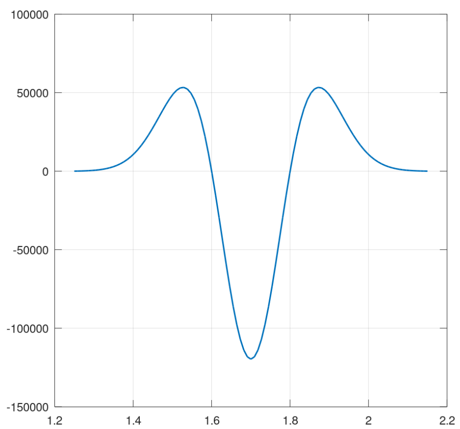
$$M_2 = \max_{a \leq \epsilon \leq b} |f''(\epsilon)|$$

1.3 Solução

Método dos Trapézios A integral de uma função, como apresentado no desenvolvimento, pode ser calculada através do **Método dos Trapézios Repetidos**. Optou-se por este por sua simplicidade de implementação e compreensão.

Observação Considerou-se $n = 300$ intervalos para interação do método, entretanto para o erro dos trapézios repetidos calculou-se a derivada de segunda ordem da função $N(h)$ através do **WolframAlpha**, obtendo a seguinte equação:

$$\boxed{f''(x) = 1.19683 \cdot 10^7 \cdot (x - 1.8) \cdot (x - 1.6) \cdot e^{-50 \cdot (-1.7 + x)^2}}$$



Análise A partir do gráfico pode-se concluir que o **Majorante** M_2, valor de pico da função, ocorre em $x = 1.7$. Assim pode-se concluir:

$$\boxed{M_2 = \max_{a \leq \epsilon \leq b} |f''(1.7)| = 119683}$$

Aproximação Desta forma, pelo Método dos Trapézios Repetidos, haveriam 47.19 guerreiros, aproximadamente 47, que possuem alturas entre $1,8m$ e $2m$, apresentando aproximadamente 0.0079722 de erro nesta estimativa.

1.4 Códigos

```
%=====================================================================================
function [I, E] = IntegralTrapezioRepetido(ddf, f, m, u, s, a, b, n)
  %===================================================================================
    %Recebe
      %Funcao: f
      %Constantes:
        %Qtde. Total da Amostra: m
        %Media da Amostra: u
        %Desvio Padrrao da Amostra: s
      %Intervalo de Integracao:
        %Inicio: a
        %Final:  b
      %Qtde. de Intervalos: n
      %Majorante da Segunda Derivada de f: ddf
    %Retorna
      %Aproximacao da Integral: I
      %Erro da Aproximacao: E
  %===================================================================================
  h = (b-a)/n; %Largura do Intervalo

  X = linspace(a, b, n+1); %Divisao do Espaco na Qtde. de Intervalos
  Y = f(X, m, u, s);                %Vetor de Respostas

  I = (h/2)*(f(X(1),m,u,s) + 2*sum(f(X(2:n),m,u,s)) + f(X(n+1),m,u,s));
  E = (b-a)*h^2/12*ddf;
endfunction
```

```
%=====================================================================================
a = 1.8; %Altura Minima a ser Avaliada
b = 2;   %Altura Maxima a ser Avaliada

m = 300; %Quantidade de Guerreiros
u = 1.7; %Altura Media dos Guerreiros
s = 0.1; %Desvio Padrao das Alturas

%Declaracao da Funcao de Desvio Padrao
f = @(x, m, u, s) (m./(s*sqrt(2*pi)))*(e.^(-((x-u).^2)./(2*(s^2))));

[I, E] = IntegralTrapezioRepetido(119583, f, m, u, s, a, b, 100)
```

2 Questão

2.1 Introdução

Problema Considera-se a função gamma definida pela seguinte equação:

$$\Gamma(x) = \int_0^\infty t^{x-1} \cdot e^{-t} dt, \quad x > 0 \tag{6}$$

Pode-se mostrar, usando integração por partes, que $\Gamma(x) = (x-1)\Gamma(x-1)$ e $\Gamma(1) = 1$, satisfazendo assim $\Gamma(n) = (n-1)!$ para todo natural $n > 0$ como mostrado na tabela:

x	1	2	3	4	5
$\Gamma(x)$	1	1	2	6	24

Determine Interpola-se a tabela acima, obtém-se uma aproximação da função Γ.

1. Determine o polinômio de grau 4 que interpola a tabela acima.
2. Determine uma spline cúbica que interpola a tabela acima.
3. Faça, na mesma figura:
   (a) Os pontos tabelados;
   (b) Polinômio de grau 4;
   (c) A spline cúbida;
4. Qual das duas funções fornece a melhor aproximação da função gamma no intervalo $[1, 5]$?
5. Qual das duas funções melhor aproxima a função gamme no intervalo $[1, 2]$?

2.2 Desenvolvimento

Teoria Sabe-se que, a partir de uma tabela de dados, pode-se obter uma função φ que interpola, isto é, relacionara todos os pontos fornecidos como descrito:

x	x_1	$\dots$	x_n
y	y_1	$\dots$	y_n

$$\boxed{\varphi(x_k) = y_k, \quad \forall k = 1, \dots, n} \tag{7}$$

Há diferentes métodos para se estimar a Interpolação Linear para um conjunto de dados, dos quais são notáveis:

1. Forma de Vandermonde;
2. Forma de Lagrande;
3. Forma de Newton;

Utiliza-se a Forma de Newton por sua simplicidade e reduzido custo computacional de execução quando comparado com o Forma de Lagrange, por exemplo, que apesar de direta é computacionalmente custosa.

Forma de Newton Nesta abordagem o sistema linear resultante será triangular inferior descrito pelas seguintes funções base:

$$\begin{aligned}
N_0 &= 1 \\
N_1 &= (x - x_0) \\
N_2 &= (x - x_0)(x - x_1) \\
&\vdots \\
N_n &= (x - x_0)(x - x_1) \cdots (x - x_{n-2})(x - x_{n-1})
\end{aligned}$$

Consequentemente os coeficientes do polinômio interpolador φ podem ser obtidos solucionando o sistemas linear apresentado anteriormente:

$$\boxed{\varphi_n(x) = \alpha_0 + \alpha_1(x - x_1) + \cdots + \alpha_n(x - x_1)(x - x_2) \cdots (x - x_{n-1})}$$

Operador Diferenças Dividas Alternativamente os coeficientes do polinômio interpolador podem ser calculados utilizando **Diferenças Dividas** denotadas em sequência:

$$\boxed{\alpha_k = f[x_0, x_1, \ldots, x_k], \quad \forall k = 0, 1, \ldots, n}$$

Fenômeno de Runge Nota-se que em interpolações realizadas em intervalos igualmente espaçadas há oscilação nas bordas de um intervalo em polinomios de grau elevado. Isso implica que a medida que o grau do polinômio interpolador aumenta cresce também o erro associado.

2.3 Solução

Interpolação Pode-se obter o polinômio interpolador de grau 4 adequado para os dados tabela pelos **Coeficientes de Newton**, expressos na seguinte tabela:

$$T = \begin{bmatrix} 1 & 0 & 0.5 & 0.33 & 0.375 \\ 1 & 1 & 1.5 & 1.83 & 0 \\ 2 & 4 & 7 & 0 & 0 \\ 6 & 18 & 0 & 0 & 0 \\ 24 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Sabe-se que o polinômio $\varphi_4(x)$ será construído através dos coeficientes de newton localizados na primeira linha da matriz T.

$$\alpha_k = f[x_0, x_1, \ldots, x_k] = T(1, k), \quad \forall k = 0, 1, \ldots, 4$$

Note que as entradas da matriz em **Octave** se iniciam em 1, portanto haverá um deslocamento da definição apresentada como mostrado:

$$\varphi_4(x) = \alpha_0 + \alpha_1(x - x_1) + \alpha_2(x - x_1)(x - x_2) + \alpha_3(x - x_1)(x - x_2)(x - x_3) + \alpha_4(x - x_1)(x - x_2)(x - x_3)(x - x_4)$$

Afim de simplificar a notação, aplica-se a representação de **Chaves Encadeadas** para a equação, substituindo as entradas x_k e $T_{1,k}$ pelos respectivos valores como segue:

$$\varphi_4(x) = T_{1,1} + (x - 1)\Big[T_{1,2} + (x - 2)\Big[T_{1,3} + (x - 3)\big[T_{1,4} + T_{1,5}(x - 4)\big]\Big]\Big]$$

$$\varphi_4(x) = 1 + (x - 1)\Big[0 + (x - 2)\Big[0.5 + (x - 3)\big[0.33 + 0.375(x - 4)\big]\Big]\Big]$$

Desenvolvendo a equação, obtém-se o seguinte **Polinômio Interpolador** de grau 4 demonstrada abaixo:

$$\boxed{\varphi_4(x) = 0.375x^4 - 3.417x^3 + 11.625x^2 - 16.583x + 9}$$

Spline Pode-se obter uma função adequada aos dados apresentados por uma **Spline Cúbica**, ou seja, um polinômio S_3 de grau menor ou igual à 3 definido por partes, com derivadas de primeira e segunda ordem contínuas, nos intervalos $[x_{k-1}, x_k] \in [x_0, x_n]$. Este polinômio será definido pela seguinte equação:

$$S_3(x) = \begin{cases} s_1(x), & x_0 \leq x \leq x_1 \\ \vdots & \\ s_k(x), & x_{k-1} \leq x \leq x_k \\ \vdots & \\ s_n(x), & x_{n-1} \leq x \leq x_n \end{cases}$$

Onde:

$$s_k = a_k(x - x_k)^3 + b_k(x - x_k)^2 + c_k(x - x_k) + d_k$$

Utilizando o comando `spline`, nativo do **Octave**, pode-se obter, a partir de um conjunto de dados, a Spline Cúbica para os dados apresentados. Aplicou-se este comando, pois apresenta os coeficientes dos polinômios definidos por partes, representados na seguinte matriz:

$$C[c_{ij}] = \begin{bmatrix} -0.04167 & 0.6250 & -0.58333 & 1.0000 \\ -0.04167 & 0.5000 & 0.54167 & 1.0000 \\ 2.20833 & 0.3750 & 1.41667 & 2.0000 \\ 2.20833 & 7.0000 & 8.79167 & 6.0000 \end{bmatrix}$$

Estes valores estão relacionados com os coeficientes dos polinômios. Cada coluna representa uma variável e cada linha representa um polinômio. Assim, temos a seguinte relação das entradas de c_{ij} e as variáveis:

$$c_{k1} = a_k, \quad c_{k2} = b_k, \quad c_{k3} = c_k, \quad c_{k4} = d_k$$

Assim a **Spline Cúbica** pode ser representada pelo seguinte sistema de equações:

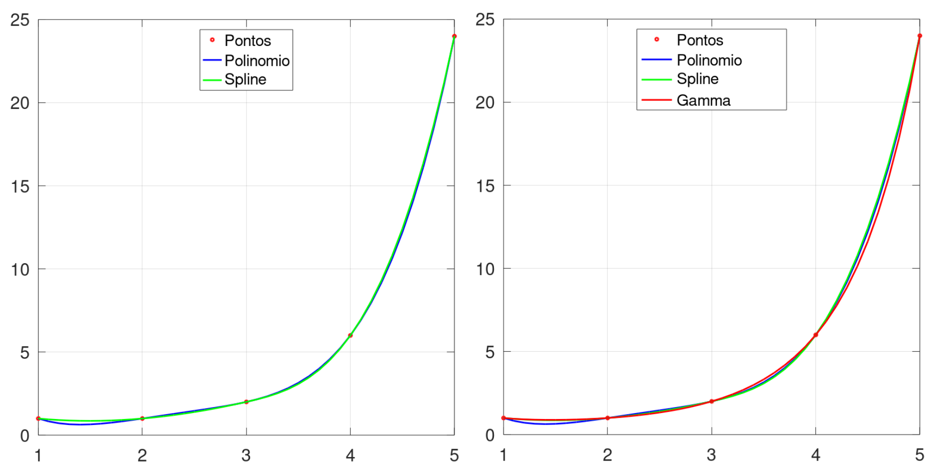
$$S_3(x) = \begin{cases} s_1(x) = a_1(x-2)^3 + b_1(x-2)^2 + c_1(x-2) + d_1, & 1 \le x \le 2 \\ s_2(x) = a_2(x-3)^3 + b_2(x-3)^2 + c_2(x-3) + d_2, & 2 \le x \le 3 \\ s_3(x) = a_3(x-4)^3 + b_3(x-4)^2 + c_3(x-4) + d_3, & 3 \le x \le 4 \\ s_4(x) = a_4(x-5)^3 + b_4(x-5)^2 + c_4(x-5) + d_4, & 4 \le x \le 5 \end{cases}$$

$$S_3(x) = \begin{cases} s_1(x) = -0.04167(x-2)^3 + 0.6250(x-2)^2 - 0.58333(x-2) + 1, & 1 \le x \le 2 \\ s_2(x) = -0.04167(x-3)^3 + 0.5000(x-3)^2 + 0.54167(x-3) + 1, & 2 \le x \le 3 \\ s_3(x) = +2.20833(x-4)^3 + 0.3750(x-4)^2 + 1.41667(x-4) + 2, & 3 \le x \le 4 \\ s_4(x) = +2.20833(x-5)^3 + 7.0000(x-5)^2 + 8.79167(x-5) + 6, & 4 \le x \le 5 \end{cases}$$

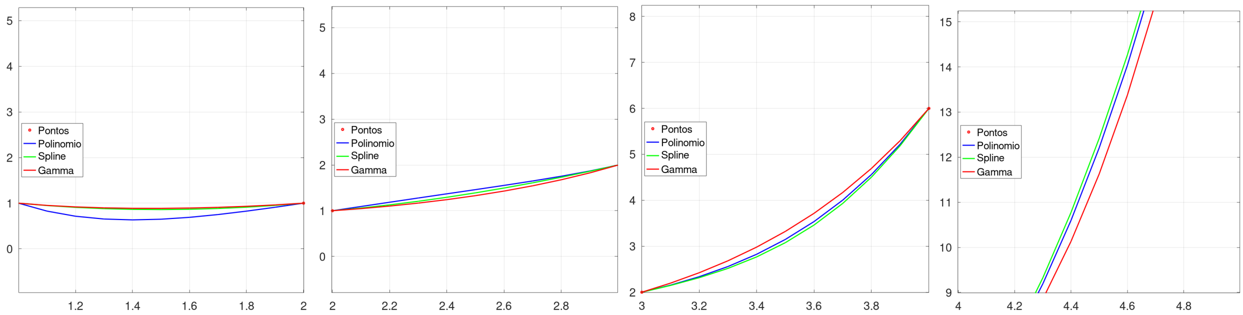
Desenvolvendo as equações, realizando as devidas manipulações, chegamos nos seguintes polinômios para cada um dos intervalos desejados:

$$\boxed{S_3(x) = \begin{cases} s_1(x) = -0.04167x^3 + 0.87502x^2 - 3.58337x + 5.00002, & 1 \le x \le 2 \\ s_2(x) = -0.04167x^3 + 0.87503x^2 - 3.58342x + 5.00008, & 2 \le x \le 3 \\ s_3(x) = +2.20833x^3 - 26.1250x^2 + 104.417x - 139, & 3 \le x \le 4 \\ s_4(x) = +2.20833x^3 - 26.1250x^2 + 104.416x - 139, & 4 \le x \le 5 \end{cases}}$$

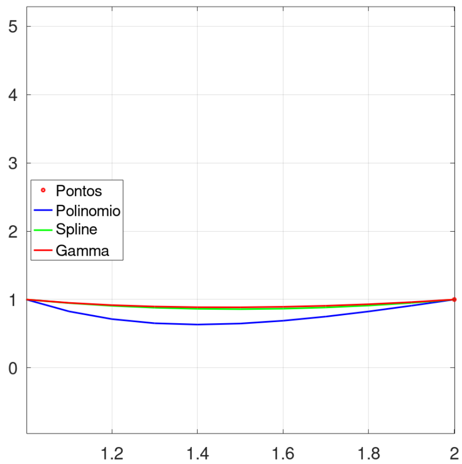
Gráfico Inserindo os pontos tabelados, o polinômio interpolador e a spline cúbica, respectivamente representadas pelas cores vermelho, azul e verde, no gráfico a esquerda. O gráfico da direita inclui o plot da função `Gamma`, nativa do **Octave**, imprimindo os valores precisos utilizados como benchmark das demais aproximações.



Intervalo $[1, 5]$ Nota-se, através da inspeção visual, que a distância, consequentemente o erro, entre `Gamma` e a **Splice Cúbica**, no intervalo $[1, 3]$, é menor. Entretanto, a distância entre **Gamma** e o **Polinômio**, no intervalo $[3, 5]$, é menor. Ambas representam boas aproximações, a **Splice Cúbica** será superior por apresentar muito pouco erro no intervalo $[1, 3]$.



Intervalo $[1, 2]$ A **Splice Cúbica** melhor aproxima a função `Gamma` no intervalo $[1, 2]$. Nota-se que esta aproximação possui comportamento mais estável e suave, visto que está definida por partes. Isso garante maior precisão e menor sucessão ao **Efeito de Runge**.



2.4 Códigos

```
%=================================================================================
function [T] = CoeficientesNewton (X, Y)
    %=============================================================================
    %Recebe Vetores: X e Y
    %Retorna um Vetor C com os Coeficientes de Newton do Conjunto
    %=============================================================================
  nPontos = length(X);           %Qtde de Dados
  T = zeros(nPontos, nPontos); %Matriz das Diferencas Dividas
  T(:,1) = Y;                    %Inicializa a Primeira Coluna

  for(j = 2:nPontos)
    for(i = 1:(nPontos-j+1))

      T(i,j) = (T(i+1,j-1) - T(i,j-1))/(X(j+i-1) - X(i)); %Calculo das Entradas

    endfor
  endfor
endfunction
```

```
%=================================================================================
X = [1, 2, 3, 4,  5]; %Coordenadas em X
Y = [1, 1, 2, 6, 24]; %Coordenadas em Y
x = 1:0.1:5; %Intervalo de Avaliacao


%INTERPOLACAO=====================================================================
T = CoeficientesNewton(X, Y) %Coeficientes de Newton
p = @(t)
    T(1,1)+(t-X(1)).*(T(1,2)+(t-X(2)).*(T(1,3)+(t-X(3)).*(T(1,4)+(t-X(4)).*T(1,5))));


%SPLINE===========================================================================
y = spline(X, Y, x);
C = spline(X, Y).coefs %Imprimi os Coeficientes


%=================================================================================
LW = 3; FS = 32; %Largura das Linhas | Tamanho da Fonte
plot(X, Y,          'linewidth', LW, 'or', %Pontos
     x, p(x),       'linewidth', LW, 'b',  %Polinomio Interpolador
     x, y,          'linewidth', LW, 'g',  %Spline Cubica
     x, gamma(x), 'linewidth', LW, 'r'); %Funcao Gamma

axis("square"); grid; set(gca, "fontsize", FS);
legend("Pontos","Polinomio","Spline","Gamma","location", "west")
```