1 Método de Chebyshev (15/11/2017)

Seja $f(x)$ uma função limitada no intervalo (a,b). Podemos expandir $f(x)$ da seguinte maneira:

$$f(x(t))_{(a\leq x\leq b)} = F(t) = \frac{1}{2}a_0 + a_1T_1(t) + a_2T_2(t) + \dots \tag{1}$$

em que

$$T_r(t) = \cos\left(r\cos^{-1}(t)\right)$$

$$t = \frac{2x-(b+a)}{b-a}$$

Integrando a Eq. 1, temos:

$$\frac{2}{b-a}\int_a^x f(x)dx = \int_{-1}^t F(t)dt = \frac{1}{2}b_0 + b_1T_1(t) + b_2T_2(t) + \dots \tag{2}$$

em que

$$b_r = \frac{a_{r-1}-a_{r+1}}{2r}, \qquad r = 1,2,3,\dots$$

O valor de b_0 é determinado pelo limite inferior de integração, então:

$$b_0 = 2b_1 - 2b_2 + 2b_3 - 2b_4 + \dots$$

A integral é definida por:

$$\frac{2}{b-a}\int_a^b f(x)dx = \int_{-1}^1 F(t)dt = \frac{1}{2}b_0 + b_1 + b_2 + \dots = 2\left(b_1 + b_3 + b_5 + \dots\right) \tag{3}$$

Os coeficientes da expansão 1 podem ser calculados usando a observação de que qualquer polinômio de grau N pode ser escrito na forma

$$f(x(t)) = F(t) = \frac{1}{2}a_0 + a_1T_1(t) + a_2T_2(t) + \dots + a_{N-1}T_{N-1}(t) + \frac{1}{2}a_NT_N(t) = \sum_{r=0}^{N}{}'' a_rT_r(t) \tag{4}$$

Aqui, $\sum''$ denota a soma finita sujo primeiro e último termos são multiplicados por $\frac{1}{2}$.
Os coeficientes em 4 são dados por

$$a_r = \frac{2}{N}\sum_{s=0}^{N}{}'' F\left(\cos\frac{\pi s}{N}\right)\cos\left(\frac{\pi r s}{N}\right)$$

Isto é uma consequência da ortogonalidade da função cosseno com respeito aos pontos $t_s = \cos\frac{\pi s}{N}$, expressada pela equação:

$$\sum_{s=0}^{N}{}'' \cos\left(\frac{\pi i s}{N}\right)\cos\left(\frac{\pi j s}{N}\right) = \begin{cases} 0, & i\neq j \\ N, & i=j=0 \text{ ou } N \\ \frac{1}{2}N, & i=j\neq 0 \text{ ou } N \end{cases}$$

2 Exercícios (16/11/2017)

Use o método de expansão de Chebyshev para calcular as integrais definidas abaixo com até 6 dígitos de precisão. As integrais são:

$$\int_{-1}^1 \frac{1}{x^2+x^2+0.9}dx$$

$$\int_{-1}^1 \sqrt{\left|x+\frac{1}{2}\right|}dx$$

Para cada caso, construa uma tabela de valores para os coeficientes a_r e b_r. Obtenha, também, um gráfico da função aproximada pelo método e compare com o gráfico da função no integrando.

3 Resolução

O método de Chebyshev para cálculo de integrais definidas foi implementado em MATLAB

3.1 Exercício 1

Os valores de a_r e b_r são dados na tabela abaixo com $N = 22$. Temos a seguinte integral definida:

$$\int_{-1}^{1} \frac{1}{x^4 + x^2 + 0.9} dx.$$

Tabela 1: Resultados numéricos do primeiro exercício.

r	a_r	b_r
0	1.336666674225290	-
1	0.000000000000000	0.858441127690537
2	-0.380215581155785	0.000000000000000
3	-0.000000000000000	-0.073545576067158
4	0.061057875247165	-0.000000000000000
5	0.000000000000000	0.006451618504673
6	-0.003458309799560	0.000000000000000
7	0.000000000000000	-0.000152793299825
8	-0.001319203602016	-0.000000000000000
9	0.000000000000000	-0.000102300550229
10	0.000522206302106	0.000000000000000
11	-0.000000000000001	0.000028448694199
12	-0.000103664970276	-0.000000000000000
13	0.000000000000000	-0.000004394783399
14	0.000010599398091	0.000000000000000
15	-0.000000000000000	0.000000325208060
16	0.000000843156299	0.000000000000000
17	-0.000000000000001	0.000000044054481
18	-0.000000654696062	-0.000000000000000
19	0.000000000000000	-0.000000021484507
20	0.000000161715189	-0.000000000000000
21	0.000000000000001	0.000000004921833
22	-0.000000045001792	-

O valor da integral é dado por:

$$\int_{-1}^{1} \frac{1}{x^4 + x^2 + 0.9} dx = 2\left(b_1 + b_3 + b_5 + \dots\right) = 1.582232965777331$$

Na aproximação da função do integrando, a raiz quadrada do erro quadrático médio foi de:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(\hat{y}_t - y_t\right)^2}{n}} = 1.902886989865449 \cdot 10^{-7}$$

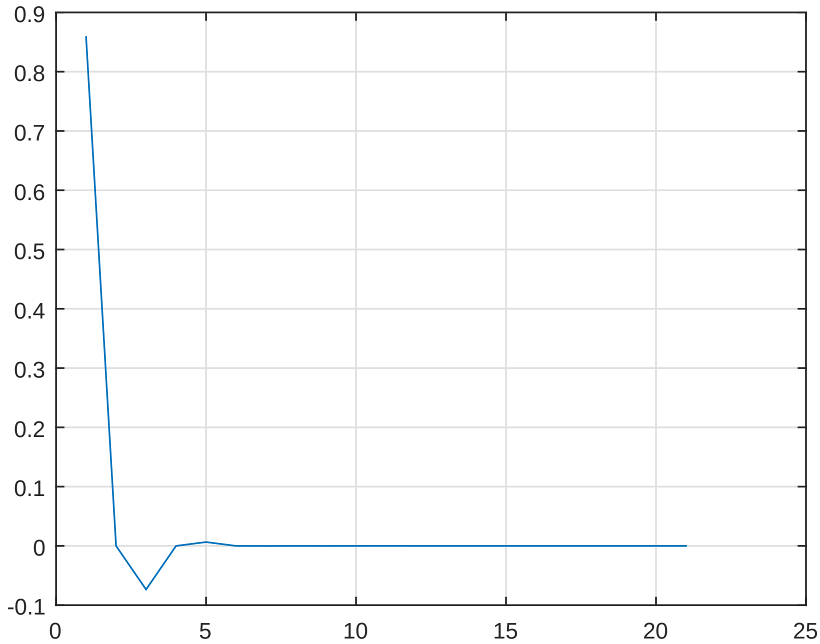

Figura 1: Convergência dos valores de b_r no exercício 1.

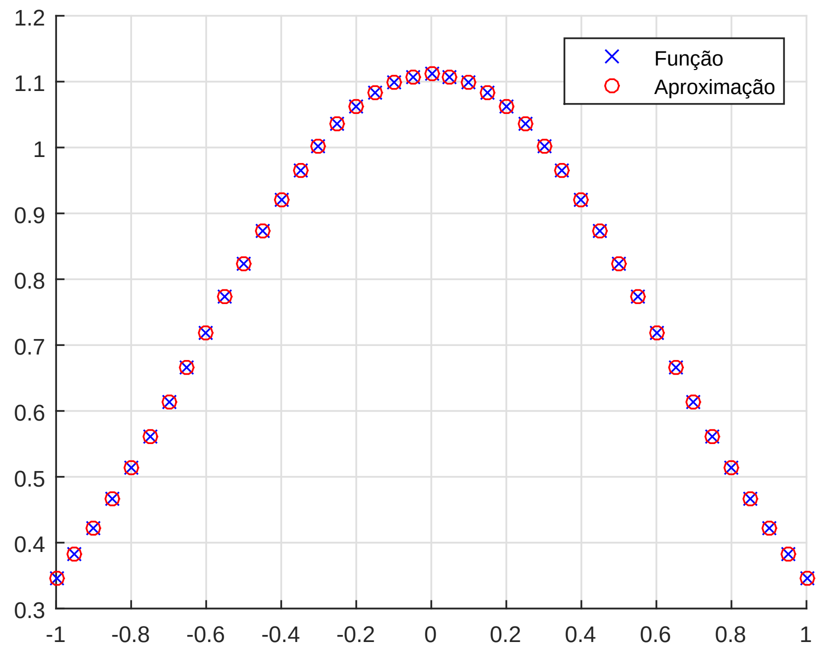

Figura 2: Função do exercício 1 aproximada.

3.2 Exercício 2

Os valores de a_r e b_r são dados na tabela abaixo com $N = 19$. Temos a seguinte integral definida:

$$\int_{-1}^{1} \sqrt{\left|x + \frac{1}{2}\right|} dx.$$

Tabela 2: Resultados numéricos do segundo exercício.

r	a_r	b_r
0	1.579218322193437	-
1	0.366965847474833	0.706798543267637
2	0.165621235658162	0.127971400146460
3	-0.144919753111006	0.020615630793443
4	0.041927450897503	-0.022306267769967
5	0.033530389048728	0.008925014411228
6	-0.047322693214777	0.001260725493651
7	0.018401683124920	-0.004380237292847
8	0.014000628885079	0.002631724712030
9	-0.023705912267560	0.000161211385072
10	0.011098823953786	-0.001510158617662
11	0.006497260085674	0.001124821018122
12	-0.013647238444891	-0.000062835382376
13	0.008005309262690	-0.000610000383275
14	0.002212771520264	0.000572724555622
15	-0.008030978294732	-0.000147278518457
16	0.006631127073971	-0.000218432570725
17	-0.001041136031548	0.000318722594433
18	-0.004205441136746	-0.000202050212572
19	0.006232671621041	-

O valor da integral é dado por:

$$\int_{-1}^{1} \sqrt{\left|x + \frac{1}{2}\right|} dx = 2\left(b_1 + b_3 + b_5 + \dots\right) = 1.465612854550713$$

Na aproximação da função do integrando, a raiz quadrada do erro quadrático médio foi de:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} \left(\hat{y}_t - y_t\right)^2}{n}} = 0.033624284161012$$

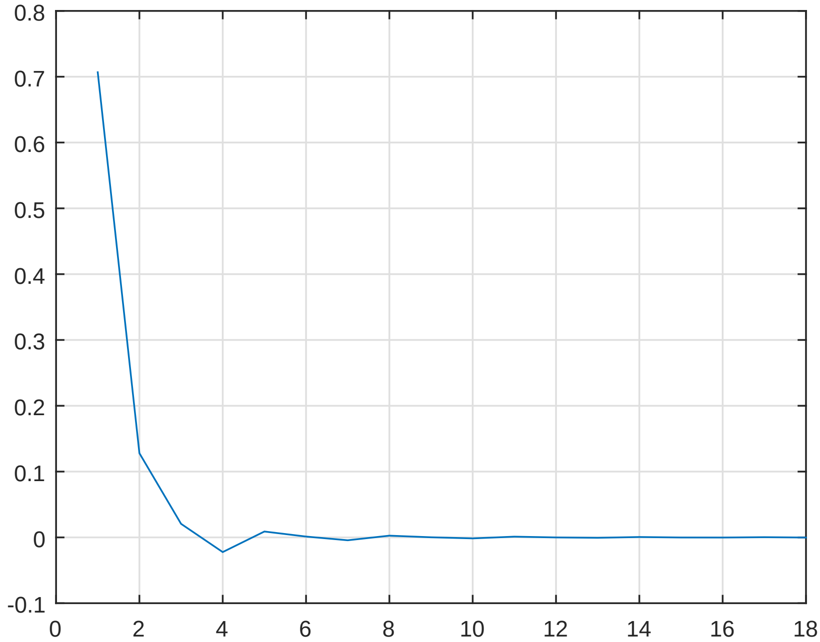

Figura 3: Convergência dos valores de b_r no exercício 2.

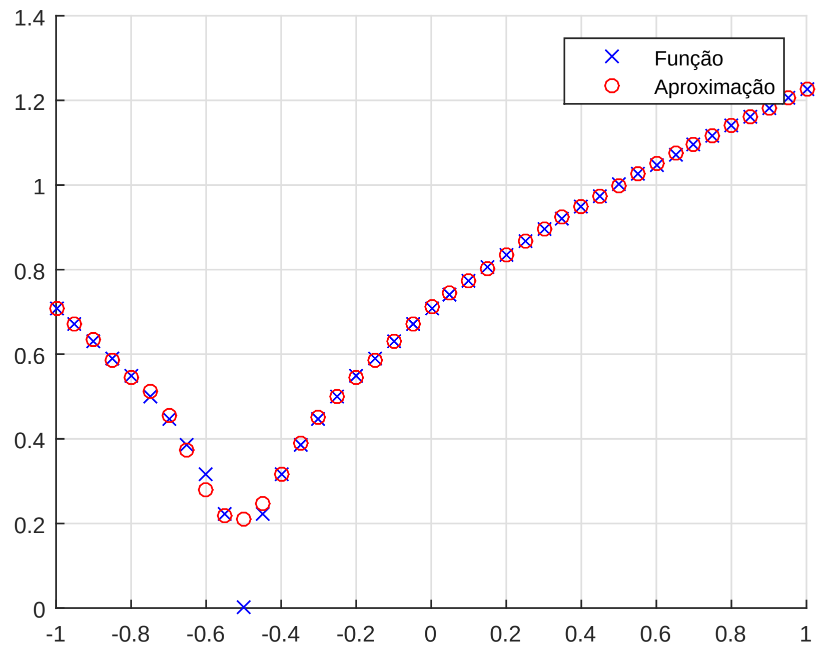

Figura 4: Função do exercício 2 aproximada.

3.3 Código em MATLAB

```
clear
clc
format long

%% Definições iniciais
F = @(x) 1./(x.^4+x.^2+0.9);
%F = @(x) sqrt(abs(x+0.5));
N = 22;

%% Matriz de pesos da soma finita com primeiro e último pesos iguais a 0.5
w = [.5 ones(1, N-1) .5];

%% Cria iteradores de 0 a N
[s,r] = meshgrid(0:N, 0:N);

%% Calcula os valores dos coeficientes a_r
ar = F(cos(s * pi / N)) .* cos(pi * s .* r / N);
ar = (2/N) * ar*w';

%% Calcula os valores dos coeficientes b_r
r = 2:N;
br = (ar(r-1) - ar(r+1))./(2*(r-1)');

%% Calcula o valor da integral
I = 2*sum(br(1:2:end))

%% Plota a função real e a aproximada
x = -1:0.05:1;
yr = F(x);
yt = cos((0:N)'*acos(x))' * (w' .* ar);
hold on;
plot(x,yr,'bx');
plot(x,yt,'ro');
grid on;
hold off;
legend('Função', 'Aproximação');

%% Calcula o erro quadrático mádio da aproximação
rmse = sqrt(mean((yr-yt').^2))
```

3.4 Código em Python

```python
import numpy as np
import matplotlib.pyplot as plt

# Definicoes iniciais
N = 22
#F = lambda x: 1/(np.power(x, 4) + np.power(x, 2) + 0.9)
F = lambda x: np.sqrt(np.abs(x + 0.5))

# Matriz de pesos da soma finita com primeiro e ultimos pesos iguais a 0.5
w = np.array([0.5] + [1 for i in range(N-1)] + [0.5])

# Cria iteradores de 0 a N
s, r = np.meshgrid(np.arange(N+1), np.arange(N+1))

# Calcula os valores dos coeficientes a_r
ar = F(np.cos(s * np.pi / N)) * np.cos(np.pi * s * r / N)
ar = (2/N) * np.dot(ar, w)

# Calcula os valores dos coeficientes b_r
r = np.arange(1, N)
br = (ar[r-1] - ar[r+1]) / (2*r)

# Calcula o valor da integral
I = 2 * np.sum( br[ np.arange(0, len(br), 2) ] )


# Plota a funcao real e a aproximada
x = np.linspace(-1,1,41)
yr = F(x)
yt = np.cos(np.dot(np.arange(N+1).reshape((N+1, 1)), np.arccos(x).reshape
    (1, x.shape[0]))).T
yt = np.dot(yt, (ar * w).reshape((N+1, 1)))

# Calcula o erro quadratico medio da aproximacao
rmse = np.mean((yr - yt.T)**2)**0.5

print('Integral:', I)
print('RMSE:', rmse)

# Plota a funcao real e a aproximada
plt.scatter(x, yr, edgecolors='b', marker='o', facecolors='none', label='
    Função')
plt.scatter(x, yt, color='r', marker='.', label='Aproximação')
plt.grid()
plt.legend()
plt.show()
```