Cálculo Numérico
Atividade avaliativa 2

Luís Otávio Lopes Amorim

17 de outubro de 2020

1. Em um determinado experimento foram observados os seguintes dados:

x_i	-2	0	1	3
y_i	-1	1	-2	-1

Sabendo que o comportamenteo da função que descreve tal fenômeno é semelhante ao de uma função polinomial de grau 3, pede-se:

(a) **(4,0)** Faça, pelo método dos mínimos quadrados, o ajuste da curva que descreve o fenômeno observado por uma função da forma $f(x) = ax^3 + bx^2 + cx + d$.

SOLUÇÃO

O exercício pede a função

$$\varphi(x) = \alpha x^3 + \beta x^2 \gamma x + \theta$$

. Assim, temos as funções $g(x)$:

$$g_1(x) = x^0$$
$$g_2(x) = x^1$$
$$g_3(x) = x^2$$
$$g_4(x) = x^3$$

Portanto, para encontrar os coeficientes α β γ e θ basta resolver o sistema:

$$\begin{cases}
\alpha \sum_{k=1}^{4} x_k^0 + \beta \sum_{k=1}^{4} x_k^1 + \gamma \sum_{k=1}^{4} x_k^2 + \theta \sum_{k=1}^{4} x_k^3 = \sum_{k=1}^{4} x_k^0 \cdot y_k \\
\alpha \sum_{k=1}^{4} x_k^1 + \beta \sum_{k=1}^{4} x_k^2 + \gamma \sum_{k=1}^{4} x_k^3 + \theta \sum_{k=1}^{4} x_k^4 = \sum_{k=1}^{4} x_k^1 \cdot y_k \\
\alpha \sum_{k=1}^{4} x_k^2 + \beta \sum_{k=1}^{4} x_k^3 + \gamma \sum_{k=1}^{4} x_k^4 + \theta \sum_{k=1}^{4} x_k^5 = \sum_{k=1}^{4} x_k^2 \cdot y_k \\
\alpha \sum_{k=1}^{4} x_k^3 + \beta \sum_{k=1}^{4} x_k^4 + \gamma \sum_{k=1}^{4} x_k^5 + \theta \sum_{k=1}^{4} x_k^6 = \sum_{k=1}^{4} x_k^3 \cdot y_k
\end{cases}$$

Assim, primeiro devemos calcular os somatórios:

$$\sum_{k=1}^{4} x_k^0 = 4 \qquad \sum_{k=1}^{4} x_k^1 = 2 \qquad \sum_{k=1}^{4} x_k^2 = 14 \qquad \sum_{k=1}^{4} x_k^3 = 20 \qquad \sum_{k=1}^{4} x_k^4 = 98$$

$$\sum_{k=1}^{4} x_k^5 = 212 \qquad \sum_{k=1}^{4} x_k^6 = 794 \quad \sum_{k=1}^{4} x_k^0 \cdot y_k = -3 \quad \sum_{k=1}^{4} x_k^1 \cdot y_l = -3 \quad \sum_{k=1}^{4} x_k^2 \cdot y_k = -15$$

$$\sum_{k=1}^{4} x_k^3 \cdot y_k = -21$$

Para então criar a matriz dos coeficientes e escaloná-la:

$$\begin{pmatrix} 4 & 2 & 14 & 20 & -3 \\ 2 & 14 & 20 & 98 & -3 \\ 14 & 20 & 98 & 212 & -15 \\ 20 & 98 & 212 & 794 & -21 \end{pmatrix} \begin{matrix} \\ L_2 = 2L_2 - L_1 \\ L_3 = 2L_3 - 7L_1 \\ L_4 = L_4 - 5L_1 \end{matrix} \sim \begin{pmatrix} 4 & 2 & 14 & 20 & -3 \\ 0 & 26 & 26 & 176 & -3 \\ 0 & 26 & 98 & 284 & -9 \\ 0 & 88 & 142 & 694 & -6 \end{pmatrix}$$

$$\begin{pmatrix} 4 & 2 & 14 & 20 & -3 \\ 0 & 26 & 26 & 176 & -3 \\ 0 & 26 & 98 & 284 & -9 \\ 0 & 44 & 71 & 347 & -3 \end{pmatrix} \begin{matrix} \\ \\ L_3 = L_3 - L_3 \\ L_4 = 13L_4 - 22L_2 \end{matrix} \sim \begin{pmatrix} 4 & 2 & 14 & 20 & -3 \\ 0 & 26 & 26 & 176 & -3 \\ 0 & 0 & 72 & 108 & -6 \\ 0 & 0 & 351 & 639 & 27 \end{pmatrix}$$

$$\begin{pmatrix} 4 & 2 & 14 & 20 & -3 \\ 0 & 26 & 26 & 176 & -3 \\ 0 & 0 & 12 & 18 & -1 \\ 0 & 0 & 39 & 71 & 3 \end{pmatrix} \begin{matrix} \\ \\ \\ L_4 = 4L_4 - 13L_2 \end{matrix} \sim \begin{pmatrix} 4 & 2 & 14 & 20 & -3 \\ 0 & 26 & 26 & 176 & -3 \\ 0 & 0 & 12 & 18 & -1 \\ 0 & 0 & 0 & 50 & 25 \end{pmatrix}$$

Assim, temos:

$$\alpha = \frac{1}{2} \beta = \frac{-5}{6} \gamma = \frac{-8}{3} \theta = 1$$

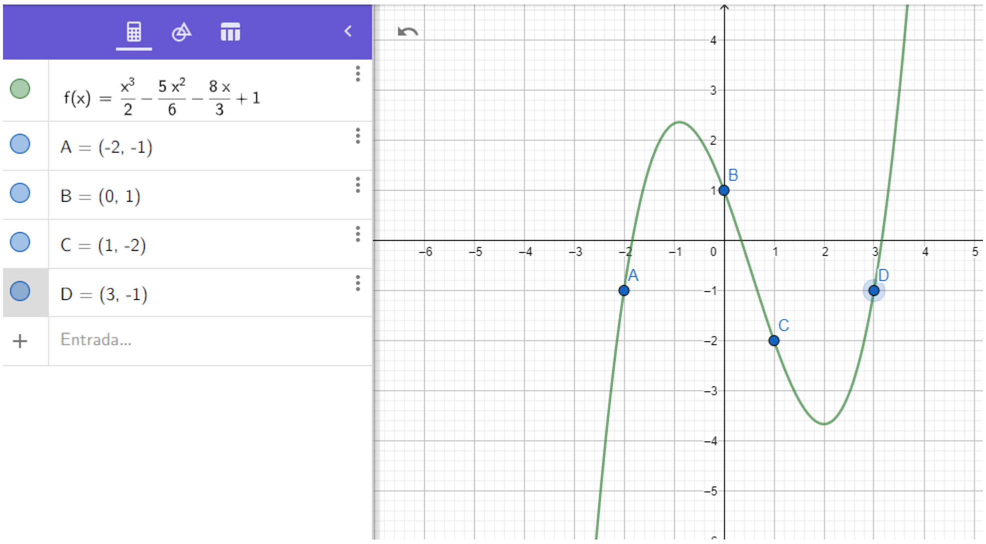
Então:

$$\varphi(x) = \frac{x^3}{2} - \frac{5x^2}{6} - \frac{8x}{3} + 1$$

(b) **(3,0)** Construa o gráfico da função encontrada e calcule o erro quadrático envolvido no processo.

SOLUÇÃO

Observando o gráfico, podemos perceber que todos os pontos tabelados estão presentes no gráfico. Assim, $f(x_k) - \varphi(x_k) = 0$ para todo k. Portanto, o erro quadrático é 0.



(c) **(3,0)** Usando um software e um método de sua escolha (dentre os três métodos vistos em aula), contrua um algoritmo e calcule TODAS as raízes de f(x) com precisão de 0,001.

SOLUÇÃO

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

float funcao(float x);
float derivada(float x);
float newton(float x_k);

int main(void) {

    float x_k1 = newton(-2);
    float x_k2 = newton(0);
    float x_k3 = newton(3);
    // Resposta final

    printf("As raizes de f(x) sao: %f %f e %f", x_k1, x_k2, x_k3);
}

float funcao(float x) {
    return pow(x,3)/2 - 5 * pow(x,2) /6 - 8 * x / 3 + 1;
}

float derivada(float x) {
    return 3 * pow(x,2) / 2 - 5 * x / 3  - 8 / 3;
}

float newton(float x_k) {
```

```
    float x_k1 = x_k;
    float f;
    float erro = INFINITY;
    int iteracao = 0;
    // Itera ate que o erro seja menor que o desejado
    while (erro >= 0.001) {
        // Atualiza os valores de x e f(x)
        x_k = x_k1;
        f = funcao(x_k);
        x_k1 = x_k - f / derivada(x_k);
        // Atualiza o valor do erro
        erro = fmin(fabs(x_k1 - x_k), fabs(f));
        iteracao++;
        // Prints da iteracao
        printf("\nIteracao: %d\n", iteracao);
        printf("\tErro: %f\n", erro);
        printf("\tx_%d: %f\n", iteracao, x_k1);
        printf("\tf(x_%d): %f\n", iteracao, f);
    }
    return x_k1;
}
```