

RA145107

July 29, 2025

Trabalho 2 - Geometria Analítica

Aluna: Cintia da Silva Bulcao

Ra: 145107

1. A rotina desenvolvida tem como objetivo calcular e exibir os principais elementos de uma hipérbole centrada na origem, cuja equação geral é dada por $\pm m(x^2) \pm n(y^2) + k = 0$. Os coeficientes m , n e k devem ser inseridos pelo usuário em forma de vetor $[m, n, k]$, sendo obrigatório que m e n possuam sinais contrários e que o valor de k seja negativo. Com base nesses coeficientes, o programa determina as coordenadas dos focos, dos vértices reais e dos vértices imaginários da hipérbole, além de calcular as equações das retas assíntotas. Ao final, todos esses elementos são representados graficamente no plano cartesiano, permitindo uma visualização clara e completa da hipérbole construída.

```
[3]: def main():
    print("Informe os coeficientes a, b e c da equação  $\pm a(x^2) \pm b(y^2) + c = 0$ ,  
↪separados por vírgula.")
    print("Atenção: a e b devem ter sinais diferentes e c precisa ser negativo.  
↪")

    dados = receber_coeficientes() # [a, b, c]

    pontos_foco = calcular_focos(dados) # [P1, P2]
    verticais = calcular_vertices_reais(dados) # [V1, V2]
    laterais = calcular_vertices_imaginarios(dados) # [U1, U2]
    retas_ass = gerar_assintotas(dados) # [reta1, reta2]

    print("--" * 40)
    print(f"Hipérbole registrada com equação: {mostrar_equacao(dados)}")
    print(f"Focos localizados em:\n- P1 = {pontos_foco[0]}\n- P2 =  
↪{pontos_foco[1]}")
    print(f"Vértices reais:\n- V1 = {verticais[0]}\n- V2 = {verticais[1]}")
    print(f"Vértices imaginários:\n- U1 = {laterais[0]}\n- U2 = {laterais[1]}")
    print(f"Assíntotas:\n- r: {retas_ass[0]}\n- s: {retas_ass[1]}")

    # Visualização gráfica
    x, y = var('x y')
    intervalo_x = (x, -medida(pontos_foco[0]) - 2, medida(pontos_foco[0]) + 2)
```

```

intervalo_y = (y, -medida(laterais[0]) - 2, medida(laterais[0]) + 2)

curva = dados[0]*x^2 + dados[1]*y^2 + dados[2] == 0
graf_curva = implicit_plot(curva, intervalo_x, intervalo_y)

graf_focos = points(pontos_foco, size=30, color='black')
graf_verticais = points(verticais, size=30, color='red')
graf_laterais = points(laterais, size=30, color='orange')
graf_r1 = implicit_plot(retas_ass[0], intervalo_x, intervalo_y,
↳color='blue', linestyle='--')
graf_r2 = implicit_plot(retas_ass[1], intervalo_x, intervalo_y,
↳color='blue', linestyle='--')

total = graf_curva + graf_focos + graf_verticais + graf_laterais + graf_r1
↳+ graf_r2
total.show(axes=True)

def receber_coeficientes() -> list[float]:
    while True:
        entrada = input("Digite os três coeficientes separados por vírgula: ").
↳split(',')
        try:
            coef = list(map(float, entrada))
        except:
            print("Erro na leitura. Digite três números válidos.")
            continue

        if len(coef) != 3:
            print("Erro: Você deve informar exatamente 3 valores.")
        elif (coef[0] > 0 and coef[1] > 0) or (coef[0] < 0 and coef[1] < 0):
            print("Erro: os coeficientes a e b devem ter sinais opostos.")
        elif coef[2] >= 0:
            print("Erro: o coeficiente c precisa ser negativo.")
        elif 0 in coef:
            print("Erro: Nenhum coeficiente pode ser zero.")
        else:
            return coef

        print("Tente novamente.")

def calcular_focos(info: list[float]) -> list[list[float]]:
    num_a = abs(info[0] / info[2])^-1
    num_b = abs(info[1] / info[2])^-1
    distancia = sqrt(num_a + num_b)

```

```

    if info[0] > info[1]:
        return [[distancia, 0], [-distancia, 0]]
    else:
        return [[0, distancia], [0, -distancia]]

def calcular_vertices_reais(info: list[float]) -> list[list[float]]:
    if info[0] > info[1]:
        a = abs(info[0] / info[2])(-1/2)
        return [[a, 0], [-a, 0]]
    else:
        a = abs(info[1] / info[2])(-1/2)
        return [[0, a], [0, -a]]

def calcular_vertices_imaginaris(info: list[float]) -> list[list[float]]:
    if info[0] > info[1]:
        b = abs(info[1] / info[2])(-1/2)
        return [[0, b], [0, -b]]
    else:
        b = abs(info[0] / info[2])(-1/2)
        return [[b, 0], [-b, 0]]

def gerar_assintotas(info: list[float]) -> list[Expression]:
    x, y = var('x y')
    a = abs(info[0] / info[2])(-1/2)
    b = abs(info[1] / info[2])(-1/2)
    reta_1 = y == (b / a) * x
    reta_2 = y == -(b / a) * x
    return [reta_1, reta_2]

def mostrar_equacao(info: list[float]) -> str:
    parte_x = f"{'-' if info[0] < 0 else ''}{abs(info[0]) if abs(info[0]) != 1
↪else ''}x2"
    parte_y = f"{'+' if info[1] > 0 else '-'}{abs(info[1]) if abs(info[1]) != 1
↪else ''}y2"
    parte_k = f"{'+' if info[2] > 0 else '-'}{abs(info[2])}"
    return parte_x + parte_y + parte_k + " = 0"

def medida(ponto: list[float]) -> float:
    return sqrt(ponto[0]2 + ponto[1]2)

main()

```

Informe os coeficientes a, b e c da equação $\pm a(x^2) \pm b(y^2) + c = 0$, separados por vírgula.

Atenção: a e b devem ter sinais diferentes e c precisa ser negativo.

Digite os três coeficientes separados por vírgula: 2,7,-1

Digite os três coeficientes separados por vírgula:

Erro: os coeficientes a e b devem ter sinais opostos.

Tente novamente.

Erro na leitura. Digite três números válidos.

Digite os três coeficientes separados por vírgula: 1,-3,-2

Hipérbole registrada com equação: $x^2 - 3.0y^2 - 2.0 = 0$

Focos localizados em:

- P1 = [1.632993161855452, 0]

- P2 = [-1.632993161855452, 0]

Vértices reais:

- V1 = [1.4142135623730951, 0]

- V2 = [-1.4142135623730951, 0]

Vértices imaginários:

- U1 = [0, 0.816496580927726]

- U2 = [0, -0.816496580927726]

Assíntotas:

- r: $y == 0.5773502691896257 * x$

- s: $y == -0.5773502691896257 * x$

[3]:

