Estudo sobre a equação do segundo grau

Prof. Doherty Andrade

www.metodosnumericos.com.br

1. Introdução

Uma equação do segundo grau é uma equação polinomial de segundo grau na forma geral:

$$ax^2 + bx + c = 0, \tag{1}$$

onde $a \neq 0$, b e c são coeficientes reais, e x é a variável desconhecida.

A solução para a equação do segundo grau pode ser encontrada usando a fórmula de Bhaskara. A fórmula é dada por:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{2}$$

O discriminante, representado por Δ, é definido como $\Delta = b^2 - 4ac$.

O valor de Δ determina a natureza das raízes da equação.

Há 3 casos a considerar:

Se $\Delta > 0$, a equação possui duas raízes reais distintas.

Se $\Delta = 0$, a equação possui uma raiz real (raiz dupla).

Se $\Delta < 0$, a equação possui duas raízes complexas conjugadas.

Exemplo

Considere a equação $2x^2 - 5x + 2 = 0$. Os coeficientes são $a = 2$, $b = -5$ e $c = 2$. Podemos usar a fórmula de Bhaskara para encontrar as raízes:

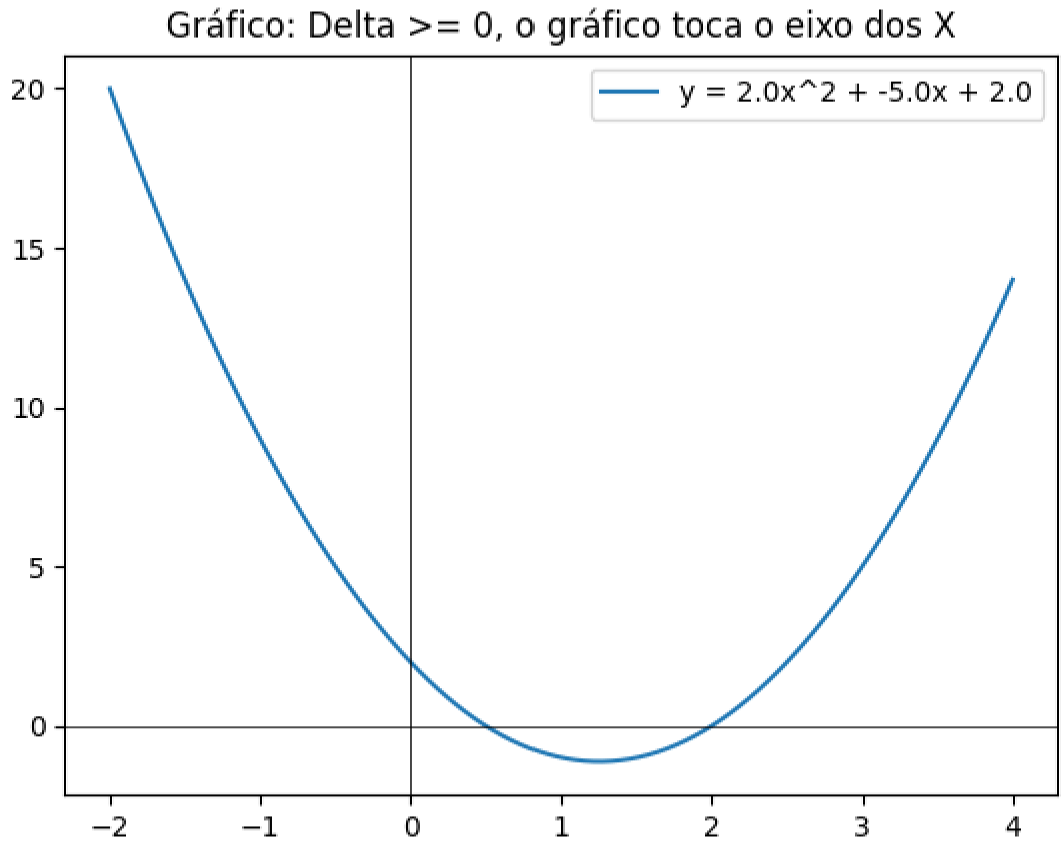
$$\begin{align} x &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{3}\\ x &= \frac{5 \pm \sqrt{(-5)^2 - 4(2)(2)}}{2(2)} \tag{4}\\ x &= \frac{5 \pm \sqrt{9}}{4} \tag{5}\\ x &= \frac{5 \pm 3}{4} \tag{6}\\ x &= \frac{5 - 3}{4} = 0.5 \text{ ou} \tag{7}\\ x &= \frac{5 + 3}{4} = 2.0. \tag{8} \end{align}$$

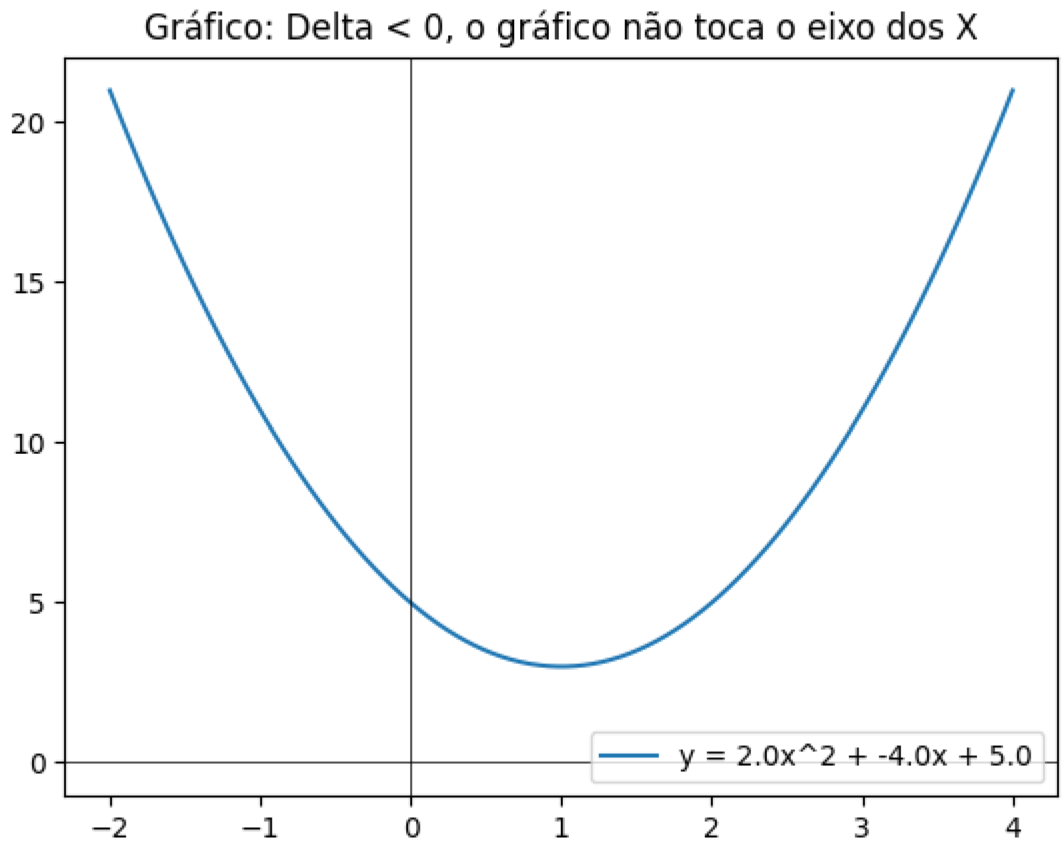
A equação do segundo grau é uma ferramenta importante na álgebra, permitindo a resolução de problemas que envolvem variáveis quadráticas. A fórmula de Bhaskara fornece uma maneira elegante de encontrar as raízes da equação. Além disso, o discriminante oferece informações sobre a natureza dessas raízes.

Graficamente, o discriminante também dá importantes respostas sobre o comportamento do gráfico do polinômio quadrático associado.

Se $\Delta \geq 0$, o gráfico toca o eixo dos X.

Se $\Delta < 0$, o gráfico não toca o eixod os X.





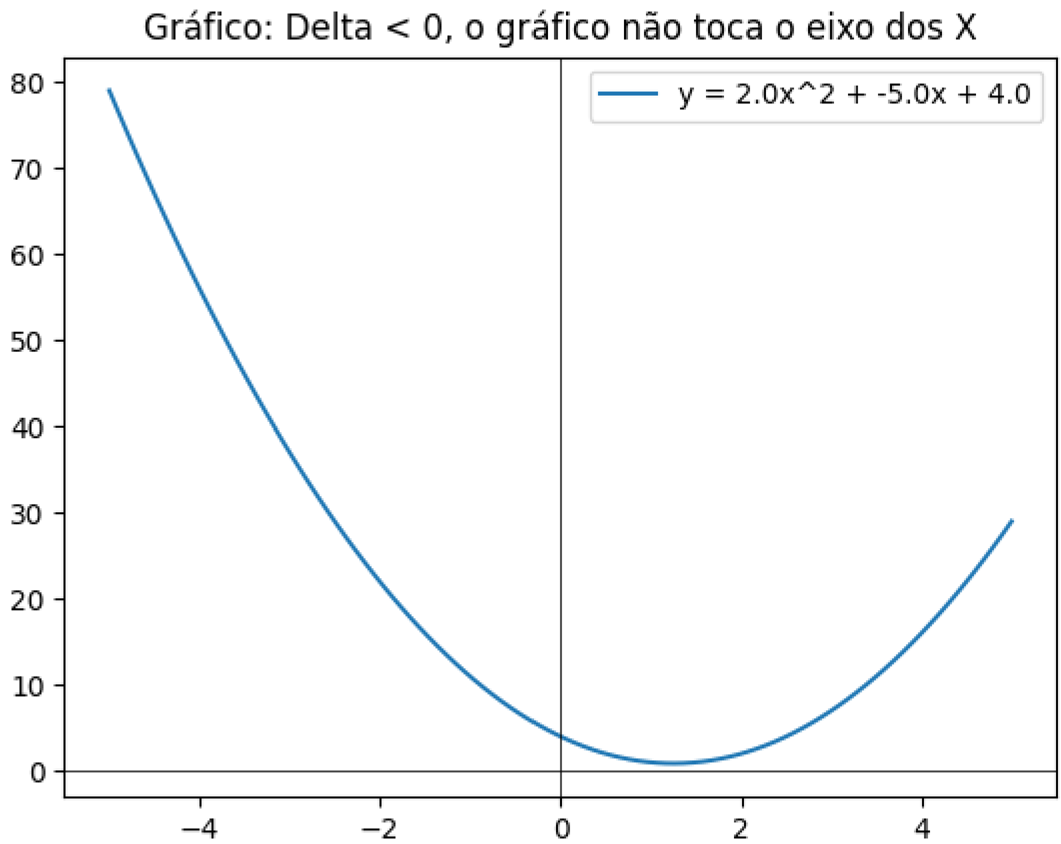
2. Determinando as Raízes

No código abaixo, vc entra com os coeficientes da equação $ax^2 + bx + c = 0$, ele determina as raizes e plota o gráfico.

In [12]:
```
import matplotlib.pyplot as plt
import numpy as np
import cmath  # Módulo para trabalhar com números complexos

def resolver_equacao_segundo_grau(a, b, c):
    # Calcula o discriminante
    delta = b**2 - 4*a*c

    # Verifica o sinal do discriminante
    if delta > 0:
        print("A equação possui duas raízes reais.")
    elif delta == 0:
        print("A equação possui uma raiz real.")
    else:
        print("A equação possui raízes complexas.")

    # Calcula as raízes usando a fórmula de Bhaskara
    if delta >= 0:
        x1 = (-b + np.sqrt(delta)) / (2*a)
        x2 = (-b - np.sqrt(delta)) / (2*a)
        print(f"Raízes reais: x1 = {x1}, x2 = {x2}")
    else:
        # Raízes complexas (usando números complexos)
        x1 = (-b + cmath.sqrt(delta)) / (2*a)
        x2 = (-b - cmath.sqrt(delta)) / (2*a)
        print(f"Raízes complexas: x1 = {x1}, x2 = {x2}")

    # Gera o gráfico da função
```

```
    x = np.linspace(-5, 5, 100)
    y = a*x**2 + b*x + c

    plt.plot(x, y, label=f'y = {a}x^2 + {b}x + {c}')
    plt.axhline(0, color='black',linewidth=0.5)
    plt.axvline(0, color='black',linewidth=0.5)

    if delta < 0:
        plt.title("Gráfico: Delta < 0, o gráfico não toca o eixo dos X")
    else:
        plt.title("Gráfico: Delta >= 0, o gráfico toca o eixo dos X")

    plt.legend()
    plt.show()

# Exemplo de uso
a = float(input("Digite o coeficiente 'a': "))
b = float(input("Digite o coeficiente 'b': "))
c = float(input("Digite o coeficiente 'c': "))

resolver_equacao_segundo_grau(a, b, c)
```

```
Digite o coeficiente 'a': 2
Digite o coeficiente 'b': -5
Digite o coeficiente 'c': 4
A equação possui raízes complexas.
Raízes complexas: x1 = (1.25+0.6614378277661477j), x2 = (1.25-0.6614378277661477j)
```



3. O Vértice de uma Parábola

Uma parábola é uma curva geométrica que pode ser descrita pela função quadrática $y = ax^2 + bx + c$, onde $a \neq 0$, b e c são coeficientes reais. O vértice é um ponto crucial em uma parábola, pois representa o ponto de mínimo ou máximo, dependendo da concavidade da parábola.

Se $a > 0$ o grpafico da parábola é voltado para cima.

Se (a<0$ o gráfico da parábola é voltado para baixo.

Para encontrar as coordenadas do vértice de uma parábola na forma $y = ax^2 + bx + c$, podemos usar a fórmula:

$$x_v = \frac{-b}{2a} \tag{9}$$

$$y_v = f(x_v) = a(x_v)^2 + b(x_v) + c = -\frac{\Delta}{4a}. \tag{10}$$

A coordenada x_v é a abscissa do vértice, enquanto y_v é a ordenada do vértice. O vértice é representado pelo ponto (x_v, y_v).

Se $a > 0$, a parábola abre para cima, e o vértice representa o ponto mínimo absoluto. Se $a < 0$, a parábola abre para baixo, e o vértice representa o ponto máximo absoluto.

Exemplo:

Considere a equação $y = 2x^2 - 4x + 1$. Podemos usar as fórmulas acima para encontrar as coordenadas do vértice:

$$x_v = \frac{-(-4)}{2 \cdot 2} = 1 \tag{11}$$

$$y_v = 2(1)^2 - 4(1) + 1 = -1 \tag{12}$$

Portanto, o vértice desta parábola é $(1, -1)$.

4. Aplicação

Um fazendeiro dispõe de 36 metros de tela para cercar uma parte retangular do seu quintal com a maior área possível. Quais devem ser as dimensões da parte do quintal?

Vejamos a solução.

A área é dada por $A = xy$ para um retângulo de dimensões x e y. Como o perímetro é dado por $2x + 2y = 36$, então temos que $x + y = 18$ e assim, $y = 18 - x$.

Substituindo na expressão da área, temos que $A = x(18 - x) = -x^2 + 18x$, uma parábola. Segue facilmente que o vértice V tem coordenadas

$$V = (9, 81).$$

Assim, as medidas do quintal devem ser $x = 9$ metros de largura e $y = 18 - 9 = 9$ metros com comprimento. Um quadrado, portanto.

Como $a = -1 < 0$ o gráfico da parábola é voltado para baixo, e o vértice indica um ponto de máximo absoluto. Isto é, a maior área que pode ser cercada com a tela disponível é de 81 metros quadrados.

5. Código pra determinar o vértice

O código a seguir determina o vértice e o classifica como ponto de máximo ou de mínimo.

In [23]:
```
def encontrar_vertice(a, b, c):
    # Calcular coordenada x do vértice
    x_vertice = -b / (2 * a)

    # Calcular coordenada y do vértice
    y_vertice = a * x_vertice**2 + b * x_vertice + c

    # Determinar se o vértice é um ponto de mínimo ou máximo
    tipo_vertice = "Ponto de Mínimo" if a > 0 else "Ponto de Máximo"

    # Retornar as coordenadas do vértice e o tipo
    return x_vertice, y_vertice, tipo_vertice

# Exemplo de uso
a = float(input("Digite o coeficiente 'a': "))
b = float(input("Digite o coeficiente 'b': "))
c = float(input("Digite o coeficiente 'c': "))

# Encontrar as coordenadas do vértice e o tipo
x_vertice, y_vertice, tipo_vertice = encontrar_vertice(a, b, c)

# Imprimir as coordenadas do vértice e o tipo
print(f"Coordenadas do vértice: ({x_vertice}, {y_vertice})")
print(f"Tipo do vértice: {tipo_vertice}")
```

```
Digite o coeficiente 'a': -1
Digite o coeficiente 'b': 18
Digite o coeficiente 'c': 0
Coordenadas do vértice: (9.0, 81.0)
Tipo do vértice: Ponto de Máximo
```

In [ ]: