Implementação dos métodos numéricos Newton-Raphson e Descida de Gradiente

Murillo Freitas Bouzon

***Resumo*—Em problemas computacionais onde é inviável o uso de soluções fechadas, os métodos numéricos são utilizados para se aproximar de uma solução para o problema. No entanto, método numéricos acabam sendo mais custoso devido a sua natureza iterativa para chegar até a solução. Neste trabalho é feita a implementação do método Newton-Raphson para encontrar a raiz de uma função, e do método Descida de Gradiente para encontrar o mínimo de uma função. Os dois métodos foram testados em duas funções diferentes, mostrando que é possível encontrar a raiz e o mínimo de uma função qualquer, porém dependendo da taxa de aprendizado utilizado, é possível que o mínimo tenha sido ultrapassado e não pode ser encontrado.**

***Index Terms*—Métodos numéricos, Descida de Gradiente, Newton-Raphson**

I. INTRODUÇÃO

Uma das abordagens para resolver problemas computacionais é utilizando métodos numéricos. Ao contrário das soluções analíticas, onde é encontrada uma solução exata através da resolução de um polinômio de grau 4 ou menor, as soluções numéricas tentam convergir para a solução, realizando diversas iterações para poder se aproximar da solução. No entanto, por mais que os métodos numéricos tendem a ser genéricos, eles possuem um custo computacional alto devido ao número de iterações para chegar até a solução.

Um método numérico consolidado é o método de Newton-Raphson, utilizado para estimar a raiz de uma função, sendo também utilizado para resolver sistemas não-lineares. Outro método conhecido é a Descida de Gradiente, utilizado para resolver problemas de otimização. É encontrado o valor mínimo de uma função, através de diversas iterações, seguindo a direção do gradiente.

Sendo assim, neste trabalho foram implementados os métodos de Newton-Raphson e Descida de Gradiente para encontrar a raiz e o mínimo de uma função.

II. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentadas as teorias relacionadas aos métodos numéricos implementados neste trabalho.

A. Newton-Raphson

O método de Newton-Raphson foi proposto por Isaac Newton no século XVII e modificado por Joseph Raphson em 1690 [1]. É um método iterativo para calcular a raiz de uma função iterando a Equação (1) enquanto $f(x_n) \approx 0$.

$$x_(n+1) = x_n - \frac{f(x_n)}{f'(x_n)} \tag{1}$$

B. Descida de Gradiente

Método de otimização iterativo para encontrar o mínimo de uma função iterando a Equação (2) enquanto $f'(x_n) \approx 0$.

$$x_(n+1) = x_n - f'(x_n) \tag{2}$$

III. METODOLOGIA

Foram implementadas 3 classes no total. A Figura 1 apresenta o diagrama de classes que foram implementadas nesse trabalho. A primeira classe foi a classe Function, que possui os seguintes atributos:

- *x0*: valor de x inicial.
- *beta*: taxa de aprendizado.
- *epsilon*: taxa de erro aceitável.

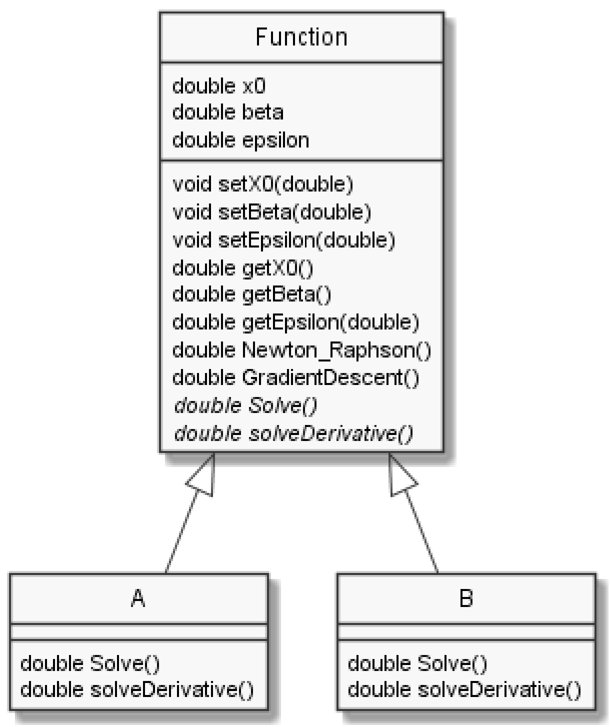

Figura 1: Diagrama UML das classes implementadas neste trabalho.

O Algoritmo 1 apresenta um pseudo-código do método Newton_Raphson() da classe Function. O Algoritmo 2 apresenta um pseudo-código do método GradientDescent() da classe Function.

Também foram implementadas as classes A e B, sendo derivadas da classe Function, herdando todos os seus métodos

Algorithm 1: Function.Newton_Raphson()

f = f(x0)
while *abs(f) ¿= epsilon* **do**
| s = f'(x0) x0 = x0 - (beta*f/s) f = f(x0)
end
return x0

Algorithm 2: Function.GradientDescent()

s = f'(x0)
while *abs(s) ¿= epsilon* **do**
| x0 = x0 - (beta*s) s = f'(x0)
end
return x0

e atributos. A classe A representa a função dada pela Equação (3). A classe B representa a função dada pela Equação (4).

$$f(x) = x^2 \tag{3}$$

$$f(x) = x^3 - 2x^2 + 2 \tag{4}$$

IV. EXPERIMENTOS E RESULTADOS

Para avaliar os métodos implementados neste trabalho, foi realizado um experimento onde utilizou-se os métodos para resolver as funções representadas pelas classes A e B descritas na Seção III.

Foi definido o valor de $x0 = 2$, e o valor de β foi variado entre 0.1 até 1.0. Para ϵ foi considerado o valor de 0.001. Os resultados para o método Newton-Raphson são mostrados na Figura 2. Foi possível encontrar a raiz das duas funções para todas as taxas de aprendizado utilizadas. A Figura 3 apresenta os resultados obtidos pelo método Descida de Gradiente para encontrar o mínimo da função representada pela classe A. É possível observar que quando $\beta > 0.9$, não foi possível encontrar o mínimo da função. A Figura 4 apresenta os resultados obtidos pelo método Descida de Gradiente para encontrar o mínimo da função representada pela classe B. Quando $\beta >= 0.6$, não foi possível encontrar o mínimo da função.

Figura 2: Resultados do método Newton-Raphson

V. CONCLUSÃO

Neste trabalho foi realizada a implementação dos métodos numéricos Newton-Raphson e Descida de Gradiente para encontrar a raiz e o mínimo de uma função respectivamente.

Foram realizados experimentos, onde para cada método foi variado a taxa de aprendizado β de 0.1 até 1.0 para duas funções diferentes A e B. Os resultados obtidos mostraram que com o método Newton-Raphson, foi possível encontrar a raiz das duas funções com todas as taxas de aprendizado utilizadas. Utilizando o método Descida de Gradiente, foi possível encontrar o mínimo da função A para todos os valores de $\beta <= 0.9$ e da função B para os valores de $\beta <= 0.5$.

REFERÊNCIAS

[1] J. W. Herivel. Newton's first solution to the problem of kepler motion. *The British Journal for the History of Science*, 2(4):350–354, 1965.

```
Descida de Gradiente Funcao A - Beta = 0.1
x = 0.000415384
Descida de Gradiente Funcao A - Beta = 0.2
x = 0.000338533
Descida de Gradiente Funcao A - Beta = 0.3
x = 0.000209715
Descida de Gradiente Funcao A - Beta = 0.4
x = 0.000128
Descida de Gradiente Funcao A - Beta = 0.5
x = 0
Descida de Gradiente Funcao A - Beta = 0.6
x = 0.000128
Descida de Gradiente Funcao A - Beta = 0.7
x = 0.000209715
Descida de Gradiente Funcao A - Beta = 0.8
x = -0.000338533
Descida de Gradiente Funcao A - Beta = 0.9
x = 0.000415384
```

Figura 3: Resultados do método Descida de Gradiente aplicado na função A.

```
Descida de Gradiente Funcao B - Beta = 0.1
x = 1.33349
Descida de Gradiente Funcao B - Beta = 0.2
x = 1.33326
Descida de Gradiente Funcao B - Beta = 0.3
x = 1.33325
Descida de Gradiente Funcao B - Beta = 0.4
x = 1.33309
Descida de Gradiente Funcao B - Beta = 0.5
x = 0
```

Figura 4: Resultados do método Descida de Gradiente aplicado na função B.