

# Utilizando openMP no Método de Newton-Raphson para encontrar as raízes de determinadas funções

Mariana Bastos dos Santos

Agosto 2019

## 1 Introdução

Para a solução de problemas computacionais é necessário utilizar métodos numéricos. Esses métodos utilizam aproximação e diversas iterações para tentar convergir para a solução e por isso, tendem a ser mais genéricos e computacionalmente mais custosos. Sendo assim, esse trabalho propõe o uso do Método de Newton-Raphson para encontrar as raízes de duas funções e utiliza openMP para paralelizar o processo.

## 2 Conceitos Fundamentais

### 2.1 Método de Newton-Raphson

[1] O Método de Newton-Raphson é utilizado para encontrar aproximadamente as raízes de uma função de forma iterativa. Para o cálculo, utiliza a função  $f(x)$  e sua derivada  $f'(x)$ , em pontos arbitrários  $x$ . A fórmula de Newton-Raphson (equação 1) consiste geometricamente em estender a linha tangente (calculada pela derivada  $f'(x)$ ) no ponto atual  $x_i$  até que ela cruze zero, então definindo o próximo palpite  $x_{i+1}$ , para a abscissa daquele cruzamento zero.

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (1)$$

## 2.2 openMP

O openMP permite a programação paralela. O programa é executado em diferentes processadores em um único computador, compartilhando a mesma memória. O programa então é compilado com comandos que dizem quais partes podem ser paralelizadas.

## 3 Implementação

Neste trabalho foram implementados os métodos de Newton-Raphson e Descida de Gradiente. Para implementação foi utilizada a linguagem C++.

Para implementação do Método Newton-Raphson foi implementado o algoritmo 1 em que itera os valores da função  $f(x)$  e de sua derivada  $S_i$  até encontrar o valor mais próximo da raiz. A variável  $\beta$  se refere a taxa de aprendizado ( $0 < \beta < 1$ ). O critério de parada foi estabelecido para até 70 iterações do algoritmo, caso não encontre o valor aproximado da raiz.

---

**Algorithm 1** Calcular o Método de Newton-Raphson

---

```
max  $\leftarrow$  70
 $x_0 \leftarrow 0$ 
 $f_0 \leftarrow f(x_0)$ 
while  $f_i > 0$  do
   $S_i \leftarrow \frac{df}{dx}(x_i)$ 
   $x_{i+1} \leftarrow x_i + \beta * (-f_i) * \frac{1}{S_i}$ 
   $f_{i+1} \leftarrow f(x_{i+1})$ 
  if  $i == max$  then
    break
  end if
end while
```

---

O openMP foi implementado antes do looping em C++ acrescentando a linha de código `#pragma omp parallel for`.

## 4 Experimentos e Resultados

Os algoritmos foram testados para as equações 2 e 3, mudando a taxa de aprendizado  $\beta$  de 0,1 até 1,0, tanto para o Método de Newton-Raphson

quanto para a Descida de Gradiente.

$$f(x) = x^2, \text{ sendo } x_0 = 2 \quad (2)$$

$$f(x) = x^3 - 2x^2 + 2, \text{ sendo } x_0 = 2 \quad (3)$$

Tabela 1: Resultados do Método de Newton-Raphson para as equações 2 e 3

Método de Newton-Raphson											
Função 2						Função 3					
<i>i</i>	<i>f(i)</i>	<i>x<sub>i</sub></i>	<i>β</i>	<i>x<sub>0</sub></i>	max	<i>i</i>	<i>f(i)</i>	<i>x<sub>i</sub></i>	<i>β</i>	<i>x<sub>0</sub></i>	max
64	0,01	0,0073	0,1	2	70	69	0,09	-0,823	0,1	2	70
30	0,01	0,08	0,2	2	70	69	0,01	-0,837	0,2	2	70
20	0,01	0,082	0,3	2	70	9	2	-0,039	0,3	2	70
14	0,01	0,088	0,4	2	70	69	1,02	0,98	0,4	2	70
12	0,01	0,073	0,5	2	70	52	0,82	1,346	0,5	2	70
9	0,01	0,081	0,6	2	70	2	0,81	1,337	0,6	2	70
7	0,01	0,093	0,7	2	70	40	1,74	-0,332	0,7	2	70
6	0,01	0,086	0,8	2	70	4	0,81	1,339	0,8	2	70
5	0,01	0,105	0,9	2	70	15	0,82	1,361	0,9	2	70

Observando a Tabela 4 é possível afirmar que a taxa de aprendizado 0,8 é a melhor para a Função 2, pois utiliza menor número de iterações *i* para alcançar o resultado. Em Relação a Função 3 aplicada ao mesmo método, observa-se que a taxa de aprendizado 0,6 é a mais adequada, pois com 2 iterações obtem valor aproximado da raiz da Função. É possível observar, ainda na Função 3 que as taxas 0,1, 0,2 e 0,4 não encontraram o valor da raiz, o algoritmo executou o critério de parada (o algoritmo 1 para quando o número de iterações atinge o valor máximo de 70). Porém, o openMP não apresentou melhor desempenho em para o cálculo de função simples. Seu funcionamento pode ser testado com funções mais complexas para poder verificar diferença em seus resultados.

## 5 Conclusão

Esse trabalho propôs a implementação dos algoritmos do Método de Newton-Raphson, com o objetivo de encontrar as raízes da função 2. Nos resultados observados na Seção 4 foi possível avaliar a influência da taxa de aprendizado no número de iterações executados pelo Algoritmo 1 e no valor da função. Portanto, é preciso analisar a aproximação dos valores de  $f(i)$  a zero, pois deve-se ter o equilíbrio tanto do número de iterações quanto do valor da função em relação a zero. O openMP não apresentou ganho.

## Referências

- [1] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.