

Aluno: VINICIUS GASPARINI

Submeter até: 23/08/2019 23:59hs

**Q1** Usando o método iterativo de Newton, com estimativa inicial  $p_1 = 1.967$ , para aproximar uma raiz da função  $f(x) = x^3 - 4x - 1$ , encontre  $p_5$ .

- a) 2.1160499      b) 2.1166099      c) 2.1168692      d) 2.116839      e) 2.116771      ~~f) 2.1149075~~

**Q2** Usando o método iterativo das Secantes, com estimativas iniciais  $p_1 = -0.926$  e  $p_2 = 0.247$ , para aproximar uma raiz da função  $f(x) = x^3 - 4x - 1$ , encontre  $p_5$ .

- a) -0.2527394      b) -0.2520228      c) -0.2529342      d) -0.2526835      ~~e) -0.2540063~~      f) -0.2525086

**Q3** Usando o método iterativo da Posição Falsa, com estimativas iniciais  $a_1 = -0.827$  e  $b_1 = 0.452$ , para aproximar uma raiz da função  $f(x) = x^3 - 4x - 1$ , encontre  $p_5$ .

- a) -0.2521953      b) -0.2524919      c) -0.2525821      d) -0.2524245      e) -0.2524123      ~~f) -0.2541016~~

### Implementação

```
def f(x):
    return x**3-4*x-1

def f_(x):
    return 3*(x**2)-4

def bolzano(a,b):
    if (f(a)*f(b)) < 0:
        return True
    return False

def newton(p,n):
    for i in range(n):
        p = p-(f(p)/f_(p))
    return p
```

```
def newton(p,n):
    for i in range(n):
        p = p-(f(p)/f_(p))
    return p

def secantes(pontos,n):
    for i in range(1,n-1):
        pontos.append((pontos[i-1]*f(pontos[i]) - pontos[i]* f(pontos[i-1]))
                        /(f(pontos[i]) - f(pontos[i-1])))
    return pontos[-1]
```

```

def posicao_falsa(p0,p1,n):
    if bolzano(p0,p1):
        p3=p0 - f(p0)*(p1-p0)/(f(p1)-f(p0))
        for i in range(n):
            if bolzano(p0,p3):
                p0=p3
            else:
                p1=p3
                p3=p0 - f(p0)*(p1-p0)/(f(p1)-f(p0))
        return p3
    else:
        print("Reprovado no teorema de Bolzano")

```

```

# Questao 1
p = float(input("Informe o p0: "))
n = int(input("Informe a quantidade de iterações: "))
ans = newton(p,n)
print("{:.8}".format(ans))

# Questao 2
p0 = float(input("Informe o p0: "))
p1 = float(input("Informe o p1: "))
n = int(input("Informe a quantidade de iterações: "))
pontos = [p0,p1]
ans = secantes(pontos,n)
print("{:.7}".format(ans))

# Questao 3
p0 = float(input("Informe o p0: "))
p1 = float(input("Informe o p1: "))
n = int(input("Informe a quantidade de iterações: "))
ans = posicao_falsa(p0,p1,n)
print("{:.7}".format(ans))

```