Luiz Augusto Dembicki Fernandes, GRR20202416

# Metodo de Newton

# a) Por derivada analitica

import numpy as np

def f(x):

    return (0.1134 \* (x \*\* 4)) - (1.809 \* (x \*\* 3)) + (9.3205 \* (x \*\* 2)) - (16.681 \* x) + 10.41

def fprime(x): # derivada analitica de f(x)

    return (0.1134 \* 4 \* (x \*\* 3)) - (1.809 \* 3 \* (x \*\* 2)) + (9.3205 \* 2 \* x) - 16.681

x0 = 10 # chute inicial

delta = 1

ni = 0

while delta > 1e-5:

    x1 = x0 - (f(x0) / fprime(x0))

    print(f"f(x0): {f(x0)}, fprime(x0): {fprime(x0)}, x: {x1}")

    delta = abs(x1 - x0)

    print(f"delta: {delta}")

    x0 = x1

    ni += 1

    if ni == 1000:

        print("Não foi encontrada solução que cooresponda a precisão requerida")

        break

print(f"x: {x1}")

f(x0): 100.64999999999995, fprime(x0): 80.62900000000006, x: 8.751689838643665

delta: 1.2483101613563345

f(x0): 30.952185375664104, fprime(x0): 34.84671353914261, x: 7.86345154492635

delta: 0.8882382937173157

...

f(x0): 2.868541518225772, fprime(x0): -2.756186971502377, x: 6.731212962437654

delta: 1.040764486547932

Não foi encontrada solução que cooresponda a precisão requerida

x: 6.731212962437654

# b) por derivada estimada analiticamente

def fprimeest(x, h):

    return (f(x + h) - f(x - h)) / (2 \* h)

x0 = 10 # chute inicial

delta = 1

ni = 0

while delta > 1e-5:

    x1 = x0 - (f(x0) / fprimeest(x0, 1e-5))

    print(f"f(x0): {f(x0)}, fprime(x0): {fprimeest(x0, 1e-5)}, x: {x1}")

    delta = abs(x1 - x0)

    print(f"delta: {delta}")

    x0 = x1

    ni += 1

    if ni == 1000:

        print("Não foi encontrada solução que cooresponda a precisão requerida")

        break

print(f"x: {x1}")

f(x0): 100.64999999999995, fprime(x0): 80.62899999004003, x: 8.751689838489463

delta: 1.2483101615105365

f(x0): 30.952185370290668, fprime(x0): 34.84671354385682, x: 7.863451545046515

delta: 0.8882382934429485

...

f(x0): 3.746056204201796, fprime(x0): -2.8388770310527884, x: 6.700064588378881

delta: 1.3195556423282566

Não foi encontrada solução que cooresponda a precisão requerida

x: 6.700064588378881

# c) metodo da secante

x0 = 10 # chute inicial

x1 = 5 # chute inicial

delta = 1

ni = 0

x2 = 5 # para entrar na função

while abs(f(x2)) > 1e-3:

    x2 = x0 - (((x1 - x0) \* f(x0)) / (f(x1) - f(x0)))

    print(f"f(x): {f(x2)}, x: {x2}")

    x0 = x1

    x1 = x2

    ni += 1

    if ni == 1000:

        print("Não foi encontrada solução que cooresponda a precisão requerida")

        break

print(f"x: {x2}")

f(x): 5.320502840905778, x: 4.751388418115924

f(x): 2.7308661547233335, x: 7.143308549176606

f(x): 76.17414778355257, x: 9.665675399155198

...

f(x): 6.069358605260735, x: 4.0172868208892965

f(x): 202514141.2814453, x: -201.59840636728597

Não foi encontrada solução que cooresponda a precisão requerida

x: -201.59840636728597

A equação não tem intercepta com eixo x. As estimativas não convergem e rodeiam os dois pontos de mínimos locais em x~1,396 e x~6,541.