Tarea Neville

Angel Caceres Licona

June 25, 2020

1 Haga un programa que obtenga las aproximaciones recursivas...

```
from sympy import symbols, init_printing, lambdify, horner, expand,
    pprint
import numpy as np
from numpy import zeros, diag

def polneville(x, y, x0):
    Q = np.zeros((len(x), len(x)), dtype=float)
    for k in range(0, len(x)):
        Q[k][0] = y[k]

    for i in range(1, len(x)):
        for j in range(1, i + 1):
            Q[i][j] = ((x0-x[i-j])*Q[i][j-1]-(x0-x[i])*Q[i-1][j-1])
                /(x[i]-x[i-j])

    return [Q[-1][-1], np.diag(Q, 0)]

datosx = np.array([8.1, 8.3, 8.6, 8.7], dtype=float)
datosy = np.array([16.94410, 17.56492, 18.50515, 18.82091], dtype=
    float)

interpolar = 8.4
valorN = polneville(datosx, datosy, interpolar)[0]
pprint('Para {0} obtenemos {1} con Neville. '.format(interpolar,
    valorN))
```

2 Use el método de Neville para obtener aproximaciones...

a)
Para el polinomio de grado 1 obtuve: $f(8.4) = 17.878330000000002$
Para el polinomio de grado 2 obtuve: $f(8.4) = 17.87713$
Para el polinomio de grado 3 obtuve: $f(8.4) = 17.877142500000005$

b)
Para el polinomio de grado 1 obtuve: $f(-\frac{1}{3}) = 0.014079166666666672$
Para el polinomio de grado 2 obtuve: $f(-\frac{1}{3}) = 0.01283680555555556$
Para el polinomio de grado 3 obtuve: $f(-\frac{1}{3}) = -0.04877314814814815$

3 Use el método de Neville para aproximar $\sqrt{3}$...

Para el polinomio de grado 4 obtuve: $f(\frac{1}{2}) = 1.7083338515625$

4 Use el método de Neville para aproximar $\sqrt{3}$ usando $f(x) = \sqrt{x}$...

Para el polinomio de grado 4 obtuve: $f(3) = 1.6906067646231164$