

```

1  #!/usr/bin/env python
2  """
3  Program computes Euler's method algorithm for an
4  inputted function and graphs the resulting data
5  along with original function
6  """
7  import matplotlib
8  import matplotlib.pyplot as plt
9  import numpy as np
10 from sympy import *
11
12 x, y = symbols("x y")
13
14 __author__ = "Devon Rojas"
15 __date__ = "December 10, 2018"
16 __maintainer__ = "Devon Rojas"
17 __email__ = "devonmrojas@gmail.com"
18 __status__ = "Prototype"
19
20 def euler(x0, y0, h, xn):
21     """
22     Computes Euler's Method
23
24     Parameters:
25     -----
26     x0 : int
27         Initial x-value
28
29     y0 : int
30         Initial y-value
31
32     h : int
33         Interval step size
34
35     xn : int
36         Ending x-value
37
38     Returns:
39     -----
40     void
41

```

```

42     """
43     x_arr = []
44     y_arr = []
45     err_arr = []
46     ### Euler algorithm
47     n = int((xn - x0)/h + 1)
48     x_arr.append(x0)
49     y_arr.append(y0)
50     err_arr.append(y0)
51     print('\nX\tY\tError\n-----')
52     for i in range(0,n):
53         res = eq.subs([(x, x0), (y, y0)])
54         y_temp = y0 + h * res
55         x_temp = x0 + h
56         print('%.2f\t%.2f\t%.3f' % (x0, y0, res-y0))
57         if x_temp <= xn:
58             x0 = x_temp
59             y0 = y_temp
60             x_arr.append(x0)
61             y_arr.append(y0)
62             err_arr.append(res-y0)
63         else:
64             break
65     ### Graph resulting euler data & original function
66     graph(x_arr, y_arr, n, err_arr)
67
68 def graph(x, y, n, err):
69     """
70     Helper function to graph data
71
72     Parameters:
73     -----
74     ax : Axes
75         The axes to draw
76
77     x : array
78         The x data
79
80     y : array
81         The y data
82

```

```

83     Returns:
84     -----
85     ax : list
86         List of artists added
87     """
88     ax = plt.plot(x, y, marker="o", label="Euler")
89
90     ### Plot original function over same interval
91     _x = np.linspace(0, xn, 100)
92     _y = []
93     for i in _x:
94         res = eq.subs('x', i)
95         _y.append(res)
96     ax = plt.plot(_x, _y, label="Original")
97     ax = plt.plot(x, err, label="Error")
98     plt.legend()
99     return ax
100
101     ### Parse input into function
102     eq = sympify(input("Enter an equation: "))
103
104     x0 = int(input("Enter initial x value: "))
105     y0 = int(input("Enter initial y value: "))
106     h = float(input("Enter interval: "))
107     xn = int(input("Enter ending value: "))
108
109     euler(x0, y0, h, xn)
110
111     ### Display plot graph
112     plt.grid()
113     plt.show()
114

```