Palmer Edholm
Dr. Joe Koebbe
MATH 4610
November 8, 2021

Tasksheet 6

Task 1 We'll start by graphing the function to get an idea of how we should define our search interval.

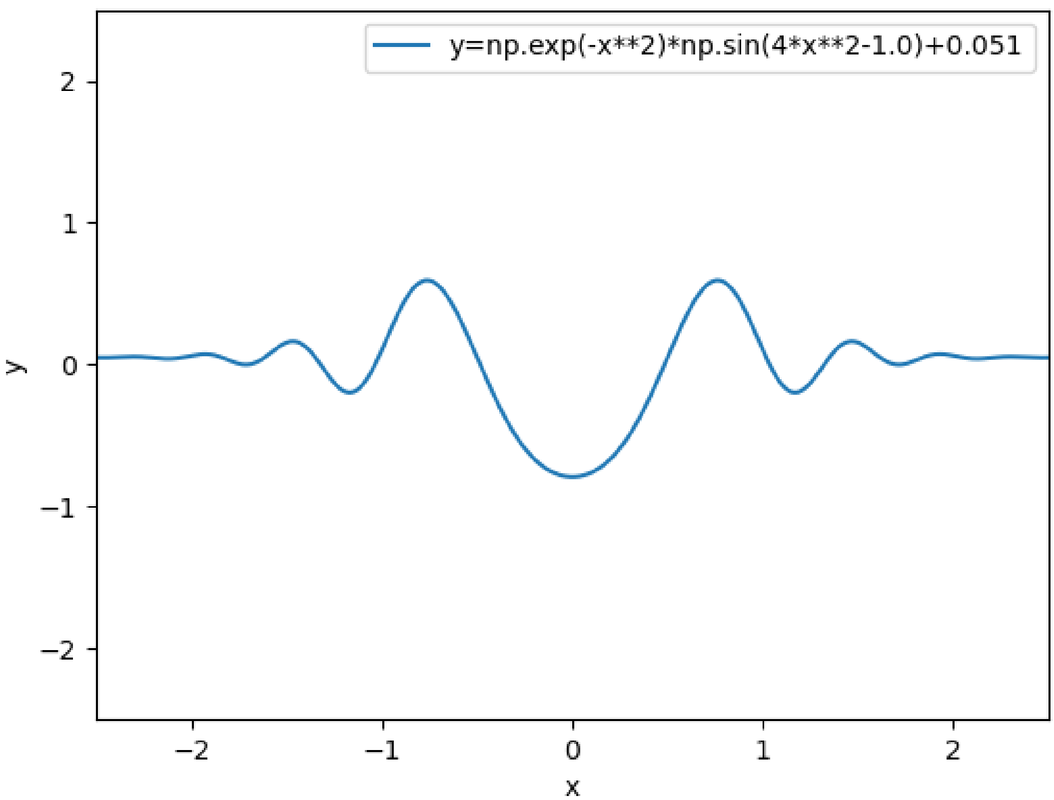

Figure 1: Graphed function

We can see that there is a root somewhere in the interval $[0.0, 0.75]$. The following code uses bisection with that interval to find the root contained therein:

```
import bisection
import numpy as np

f = lambda x: np.exp(-x**2)*np.sin(4*x**2-1.0)+0.051

df = lambda x: -2*x*np.exp(-x**2)*np.sin(4*x**2-1)+
               8*x*np.exp(-x**2)*np.cos(4*x**2-1)

print(f'Bisection: {bisection.bisection(0.0, 0.75, f, 0.0001)}')
```

which gives us the following result:

```
Bisection: 0.483673095703125
```

Now we can test all four methods with the same search interval and we'll choose $x_0 = 0.45$. The following code tests all four methods:

```python
import bisection
import fxd_pt_iter
import newton
import secant
import numpy as np

f = lambda x: np.exp(-x**2)*np.sin(4*x**2-1.0)+0.051

df = lambda x: -2*x*np.exp(-x**2)*np.sin(4*x**2-1)+\
               8*x*np.exp(-x**2)*np.cos(4*x**2-1)

print(f'Root Finding Algorithms Comparisons:\n'
      f'Bisection: {bisection.bisection(0.0,0.75,f,0.0001)}\n'
      f'Fixed Point Iteration:{fxd_pt_iter.fxd_pt_iter(0.45,\
                                0.0001,100)}\n'
      f'Secant Method: {secant.secant(0.0,0.75,f,0.0001,100)}\n'
      f'Newton\'s Method: {newton.newton(0.45,f,df,0.0001,100)}')
```

which gives us the following result:

```
Root Finding Algorithms Comparisons:
Bisection: 0.483673095703125
Fixed Point Iteration: 0.480527216850373
Secant Method: 0.48361069857905226
Newton's Method: 0.4836106985428372
```

Task 2 We run into a divide by zero error when using $x_0 = -5.0$ and $x_0 = 6.0$ because they are too far away from any root of the function. At both points, the function has converged and therefore the derivative is equal to zero.

Task 3 See my software manual entry.

Task 4 See my software manual entry for the hybrid secant-bisection method. See my alternative approach in my software manual entry to see how the result compares when finding the smallest root using the secant-bisection hybrid method to the result when using the newton-bisection hybrid method in task 3.

Task 5 See my software manual entry.

Task 6 In [1], I found a method called Laguerre's method that finds all of the roots of a polynomial $p(x)$ of any degree n. We start with an initial guess x_0 and, for $k = 0, 1, 2, \ldots$, we calculate

$$G = \frac{p'(x_k)}{p(x_k)}$$

which we use to calculate

$$H = G^2 - \frac{p''(x_k)}{p(x_k)}.$$

After which we can calculate

$$a = \frac{n}{G \pm \sqrt{(n-1)(nH - G^2)}}.$$

We then set $x_{k+1} = x_k - a$.

We repeat this until a is small enough or the maximum number of iterations has been reached. Once a root has been found, the linear factor can be removed from p and the polynomial is reduced by a degree of one. This is repeated until all roots are found.

References

[1] https://en.wikipedia.org/wiki/Laguerre