DERIVATIVES : 15.0 POINTS

For Newton's method, we will also need a way to find the derivative of a polynomial; that is, we will need to find $f'(x)$, where $f'(x)$ is the derivative of $f(x)$.

Recall that the derivative of a polynomial $f(x) = ax^b$ is $f'(x) = abx^{b-1}$, unless $b = 0$, in which case $f'(x) = 0$.

To compute the derivative of a polynomial function with many terms, you just do the same thing to every term individually. For example, if $f(x) = x^4 + 3x^3 + 17.5x^2 - 13.39$, then $f'(x) = 4x^3 + 9x^2 + 35x$.

Implement the `computeDeriv` function. This function computes the derivative of a polynomial function. It takes in a list of numbers `poly` and returns the derivative, which is also a polynomial represented by a list of floats.

Example Usage:

```
# - 13.39 + 17.5x^2 + 3x^3 + x^4
>>> poly = [-13.39, 0.0, 17.5, 3.0, 1.0]
>>> print computeDeriv(poly)
[0.0, 35.0, 9.0, 4.0] # 35x + 9x^2 + 4x^3
```

You should implement this function on your own machine, in the file `ps3_newton.py`. Test your code well in Idle, and when you are convinced it is correct, cut and paste your definition into this tutor window.

Here are some example Test Cases to test your code with - feel free to make up your own additional test cases. Be sure to test these on your own machine - and that you get the same output! - before running your code on this webpage!

```
1  def computeDeriv(poly):
2      '''
3      Computes and returns the derivative of a polynomial function as a list of
4      floats. If the derivative is 0, returns [0.0].
5
6      poly: list of numbers, length &gt; 0
7      returns: list of numbers (floats)
8      '''
9      # FILL IN YOUR CODE HERE...
10
```

Unsubmitted

Check Save *You have used 0 of 30 submissions*

Show Discussion

New Post