PHYS512 A2

Vasily Piccone

September 2022

1 Question 1

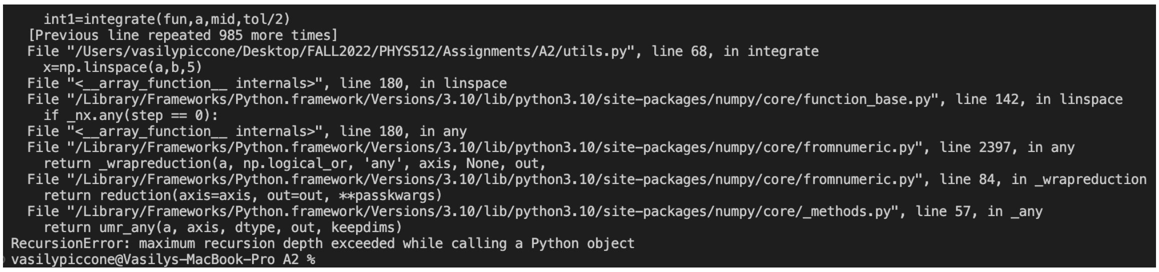
The goal of this exercise is to demonstrate the functionality of the integrator written in class (see the interpigrate slides) and the scipy.integrate.quad() function. Thus, a formal derivation is outside of the scope of the question and was acquired from problem 2.7 of David Griffith's "Introduction to Electrodynamics" textbook. We are to use the aforementioned integration tools to compute the electric field of an infinitesimally thin spherical shell of radius R.

Our integrand is the following:

$$dE = \frac{2\pi R^2\sigma}{4\pi\epsilon_0}\int_0^\pi \frac{[z - R\cos(\theta)]\sin(\theta)}{[R^2 + z^2 - 2Rz\cos(\theta)]^{\frac{3}{2}}}d\theta \tag{1}$$

Analytically, this yields the Coulomb's law outside the sphere (where $z > R$) and zero inside the shell (where $z \leq R$). Using the scipy integrator, we get the exact same result as Coulomb's law. (Please note that in the code $\sigma = 2\epsilon_0$ as the units are irrelevant in the context of this question. Similarly, when the Coulomb field was plotted, the units were removed.)

There is a singularity in the integral when $\theta = 0$, and $z = R$. For this reason, the integrator we made in class fails.



```
    int1=integrate(fun,a,mid,tol/2)
  [Previous line repeated 985 more times]
  File "/Users/vasilypiccone/Desktop/FALL2022/PHYS512/Assignments/A2/utils.py", line 68, in integrate
    x=np.linspace(a,b,5)
  File "<__array_function__ internals>", line 180, in linspace
  File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/numpy/core/function_base.py", line 142, in linspace
    if _nx.any(step == 0):
  File "<__array_function__ internals>", line 180, in any
  File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/numpy/core/fromnumeric.py", line 2397, in any
    return _wrapreduction(a, np.logical_or, 'any', axis, None, out,
  File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/numpy/core/fromnumeric.py", line 84, in _wrapreduction
    return reduction(axis=axis, out=out, **passkwargs)
  File "/Library/Frameworks/Python.framework/Versions/3.10/lib/python3.10/site-packages/numpy/core/_methods.py", line 57, in _any
    return umr_any(a, axis, dtype, out, keepdims)
RecursionError: maximum recursion depth exceeded while calling a Python object
vasilypiccone@Vasilys-MacBook-Pro A2 %
```

Figure 1: Terminal Output when the integrator from class is used.

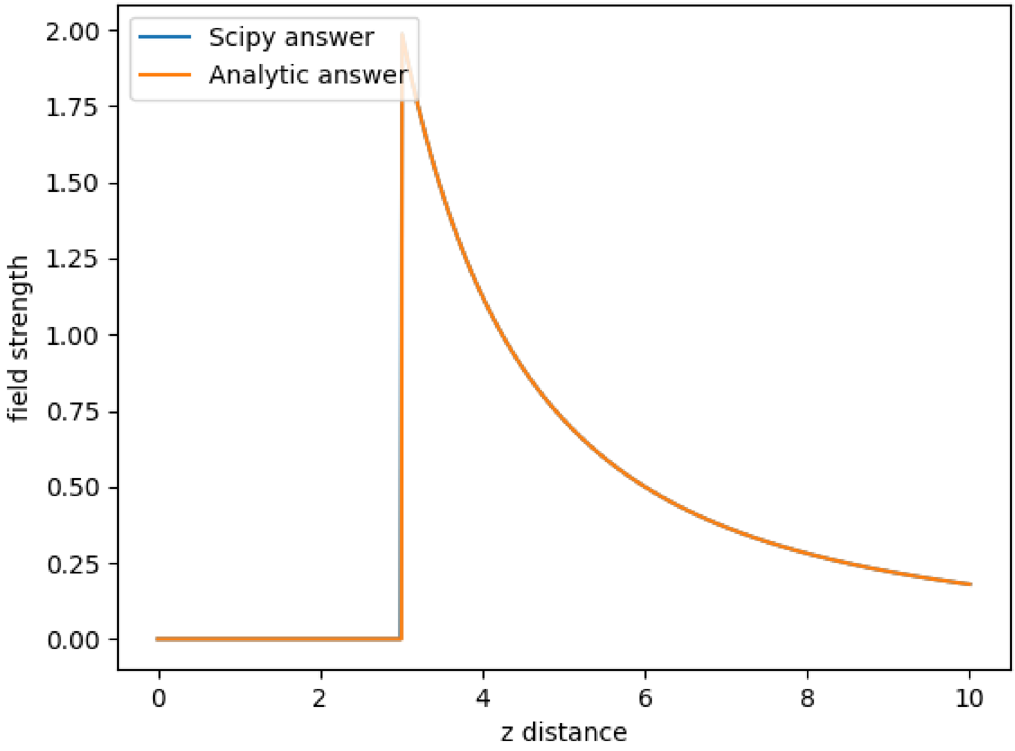
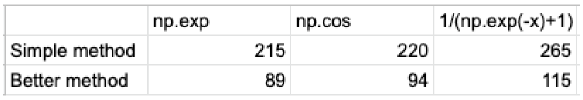


Figure 2: Graph of the scipy answer and the analytic answer.

2 Question 2

The table below shows a comparison of the different integration methods between 3 different functions.



	np.exp	np.cos	1/(np.exp(-x)+1)
Simple method	215	220	265
Better method	89	94	115

Figure 3: A table comparing the number of function calls between the simple and improved integration methods.

The simple integrator has to call the function at 5 points every time the method is run. The advanced integrator makes only 2 additional function calls every time the method is called.

3 Question 3

Please refer to the code in this submission.