55:5:60

60/61

Homework 4 Part 1: Evaluating the Error Function

Nina Martinez Diers*
Bryn Mawr College Department of Physics
(Dated: March 8, 2024)

Great write-up!

An algorithm was written to evaluate the error function using Simpson's method of integration.

1. INTRODUCTION

The error function,

$$E(x) = \int_0^x e^{-t^2}\, dt \tag{1}$$

represents the probability that a random variable with a normal distribution will fall within the range $[-x, x]$.[1, 2] There are no known methods to solve this integral analytically, so it has to be solved numerically. This is a good application of computational methods because the process of evaluating the error function by hand is tedious, while the computer is able to calculate the integral much more quickly and with higher precision than is realistic otherwise. The error function has applications in several physical contexts, including quantum mechanics and thermodynamics.

To evaluate the function, I chose to use Simpson's rule to approximate the the integral. By implementing Simpson's rule in an algorithm to solve the integral, I was able to evaluate the Eq. 1 over the range $0 \leq x \leq 3$ in increments of $h = 0.1$.

2. EXPERIMENT

Simpson's method of integration takes 3 evenly spaced points in a function and fits them to a parabolic curve to evaluate the integral of the function over that range.[1] It is a better method to approximate the integral of a function than the trapezoid method, because there are more degrees of freedom to get a closer fit to the curve than with the trapezoid method.[1] The equation for Simpson's method that was implemented for the algorithm is below:[1]

$$\int_a^b f(x)\, dx = \frac{h}{3}\Bigg[f(a) + f(b) + 4 \sum_{\substack{k_{odd} \\ 1\ldots N-1}} f(a+kh) + 2 \sum_{\substack{k_{even} \\ 1\ldots N-2}} f(a+kh)\Bigg] \tag{2}$$

The error associated with using the approximation of Simpson's rule is given by:[1]

$$\epsilon = \frac{h^4}{180}[f'''(a) - f'''(b)] \tag{3}$$

In order to implement these equations, I defined functions to calculate the integrand of the error function, the third derivative of the integrand of the error function, and the Simpson's rule evaluation of the error function.

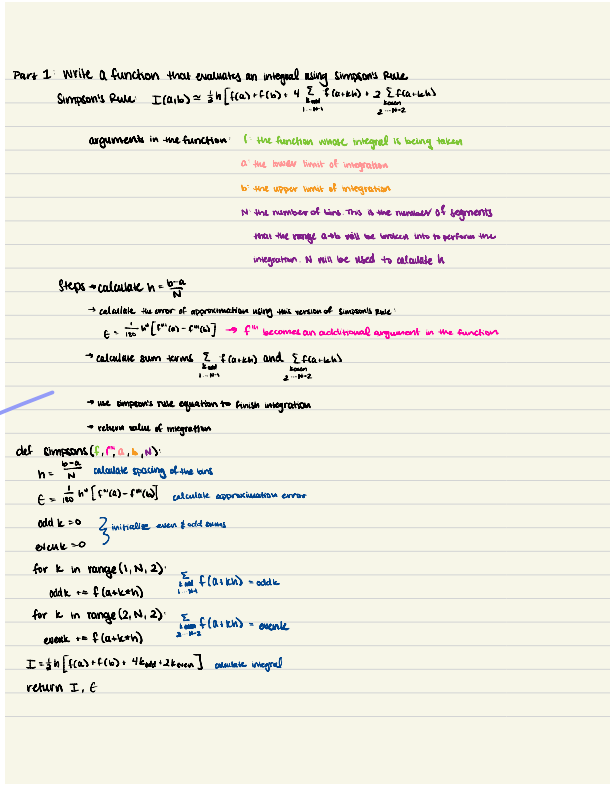

FIG. 1: Pseudocode for a general version of Simpson's rule including the approximation error calculation

3. RESULTS

The value of the error function increases, leveling off at 1. This is consistent with expectations because the integrand is a normalized Gaussian. Because the total probability is one represented by the Gaussian is equal to one, we expect the error function to increase to approach one. The approximation error with using $N = 30$ is much

*Electronic address: nmartinezd@brynmawr.edu; URL: Optionalhomepage

FIG. 2: Pseudocode for calculating the integrand of the error function, calling the Simpson's rule and graphing.

less than 0.1, which is low enough not to impact the shape of the error function curve, so we know that this number of bins is appropriate for the function.

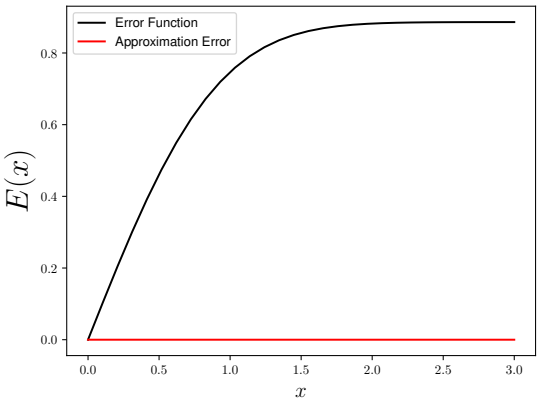

FIG. 3: The Error Function. The error function begins at zero, increasing logarithmically to converge at 1.

4. CONCLUSIONS

The algorithms for the Simpson's Rule evaluation of the error functions were successful in producing an accurate function and visualization of the data. The algorithm written would be easily used to evaluate the integral of different functions and to different levels of precision.

[1] M. Newman, *Computational Physics* (2013), revised and expanded ed., ISBN 978-1-4801-4551-1.
[2] *Error function*, https://en.wikipedia.org/wiki/Error_function.

Appendix A: Declaration of Collaborators

I collaborated with Woody on this assignment.

Appendix B: Survey Question

This assignment took me a much more reasonable amount of time, I was able to complete it in one day with 2 hours of writing pseudocode, 2.5 hours coding, and 2 hours of writing the report. The most interesting thing I learned about this assignment was the theory behind Simpson's rule and especially calculating the approximation error.

5.3 55/56 NICE WORK!

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

4 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

2 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points)

3 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) (+3 points)

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

4 6. Is the output answer correct? (+4 points).

3 7. Is the code readable? (+3 points)

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

. 5.3. Are units explained (if necessary)?

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

2 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named? please comment your name at the top of your code -1

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

9. Write-up (up to 28 points)

5 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓ . Is a clear, legible LaTeX type-set write up handed in?

3 . Are key figures and numbers from the problem given? (+ 3 points)

4 . Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3 . Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

2 . Is a brief explanation of physical context given? (+2 points)

1 . If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3 . Is the algorithm used explicitly stated and justified? (+3 points)

2 . When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2. Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)
1. Are collaborators clearly acknowledged? (+1 point)
2. Are any outside references appropriately cited? (+2 point)