

Hw 5

Stefany Fabian Dubón*
 Bryn Mawr College
 (Dated: March 3rd, 2023)

Learned the computation and plotting of the error function of an integral using Trapezoidal integration and Gaussian quadrature and how to calculate the quantum uncertainty in the harmonic oscillator

1. INTRODUCTION

For this homework assignment, I practiced ways to evaluate integrals, as well as to how to compare their error as a function of N . One of this ways in to use the Trapezoidal rule, which was talked about in the last assignment. For this assignment specifically, the Gaussian quadrature method will be explained as well as to the calculation of both the trapezoidal integration and Gaussian quadrature error.

Gaussian quadrature is a numerical integration method that uses weighted sum of function values at specific points to approximate the integral of a function over a given interval. The correct rule for mapping the points to a general domain that runs from $x=a$ to $x=b$ is:

$$x'_k = \frac{1}{2}(b-a)x_k + \frac{1}{2}(b+a) \quad (1)$$

As for the weights, they don't change if the sample points is sliding up and down the x line, but if the width of the integration domain changes then the value of the integral will increase or decrease by a corresponding factor, which means the weights have to be re-scaled using

$$w'_k = \frac{1}{2}(b-a)w_k \quad (2)$$

After calculating the re-scaled positions and weights then the integral itself is given by

$$\int_a^b f(x) dx \approx \sum_{k=1}^n w'_k f(x'_k) \quad (3)$$

The basic idea behind Gaussian quadrature is to approximate the integral of a function $f(x)$ over an interval $[a, b]$ as a weighted sum of N points so that the integration rule is exact for all polynomial integrands up to and including polynomials of degree $2N - 1$. To get an integration rule accurate up to the highest possible degree of $2N - 1$, the sample point x_k should be chosen to coincide with the zeros of the N th Legendre Polynomial $P_N(x)$, if necessary re-scaled to the window of integration using Eq.(1), and the corresponding weights w_k are

Great explanation of Gaussian quadrature

$$w_k = \left[\frac{2}{(1-x^2)} \left(\frac{dP_N}{dx} \right)^{-2} \right] \quad (4)$$

To use the method Gaussian quadrature, given the values for x_k and w_k for a chosen N , all there is to do is to re-scale them if necessary using Eq. (1) and Eq(2) and then perform the sum in Eq. (3)

The error of Gaussian quadrature depends on the smoothness of the function $f(x)$ and the number of points used in the approximation. As mentioned earlier, Gaussian quadrature is exact for polynomials of degree up to $2n-1$, which means that for a given number of points, the error decreases as the function being integrated becomes smoother.

I also learned about the quantum uncertainty in the harmonic oscillator. For this exercise I learned that in units where all the constants are 1, the wavefunction of the n th energy level of the two-dimensional quantum harmonic oscillator, which is a spinless point particle in a quadratic potential well and is given by

$$\psi_n(x) = \frac{1}{\sqrt{2^n n! \sqrt{\pi}}} e^{-\frac{x^2}{2}} H_n(x) \quad (5)$$

for $n = 0 \dots \infty$, where $H_n(x)$ is the n th Hermite polynomial. Hermite polynomials satisfy a relation somewhat similar to that for the Fibonacci numbers but more complex:

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x) \quad (6)$$

Which I can rewrite to isolate H_n

$$H_n(x) = \frac{(H_{n+1}(x) + 2nH_{n-1}(x))}{2x} \quad (7)$$

Next, I used the recurrence relation again to substitute for $H_{n-1}(x)$

$$H_n(x) = \frac{H_{n+1}(x) + 2n(H_n(x) - 2xH_{n-1}(x))}{2x} \quad (8)$$

Once I simplify, I get

$$2xH_n(x) = H_{n+1}(x) + 2nH_n(x) - 4x(n-1)H_{n-1}(x) \quad (9)$$

*Electronic address: sfabiandub@brynmawr.edu

Finally, after rearranging the equation, I get

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x) \quad (10)$$

Which I will use in exercise 5.13

I also learned that the quantum uncertainty in the position of a particle in the n th level of a harmonic oscillator can be quantified by its root-to-mean-square position $\sqrt{\langle x^2 \rangle}$, where

$$\langle x^2 \rangle = \int_{-\infty}^{\infty} x^2 |\psi_n(x)|^2 dx \quad (11)$$

2. EXERCISE 5.3 CONTINUATION +24

For this exercise, using the same integral as last assignment:

$$E(x) = \int_0^x e^{-t^2} dt \quad (12)$$

I had to write a program to calculate the integral $E(x)$ for values of x from 0 to 3 in steps of 0.1, using the trapezoidal rule, and the Gaussian quadrature, and then make a graph of the error of the trapezoidal integration and Gaussian quadrature as a function of N . I started by defining the function to integrate (e^{-t^2}). I then defined the trapezoidal integration the same way I did in the last assignment. Then I defined the Gaussian quadrature, where I use "fixed-quad" a function from `scipy.integrate` to numerically integrate the integral from 0 to x using Gaussian quadrature with $n=5$, then multiply the result by the square root of π to get the value of $E(x)$. Next, I generated a sequence of values from 0 to 3 with a step of 0.1 for both methods, which I then use to calculate $E(x)$ using both the trapezoidal rule and Gaussian quadrature. Finally I find the absolute error of each method and plot them as a function of N

3. EXERCISE 5.13 +21

For part a) for this exercise I wrote a user-defined function $H(n, x)$ by using the first two Hermite polynomials are $H_0(x) = 1$ and $H_1(x) = 2x$ and Eq. (10) seen in the introduction. To do this, I created an if statement that returns 1 if $n=0$, it returns $2x$ if $n=1$, or if n is neither 0 nor 1, the function initializes the variables "Hn-minus-2" to 1 and "H-minus-1" to $2x$. I then created a for loop that iterates $i=2$ to n . For each value of "i", the function calculates the value of $H_n(x)$ using the Eq. (10), and once the loop is completed it returns the value of $H_n(x)$. To make the plot that shows the harmonic oscillator wavefunctions for $n=0, 1, 2$ and 3 all on the same graph, I incorporated the range of $x=-4$ to $x=4$ by creating an array of 1000 equally spaced points between -4 and 4, I

then put the labels and title. To make the actual plot I created a for loop that iterates over the values of n from 0 to 3 and calculates the corresponding values of $H_n(x)$ for each value of x in the range. It then plots the resulting wavefunctions on the same graph.

For part b) I did the same thing than part a, except that for the array I did the range between -10 to 10, and since I only needed a plot for the wavefunction for $n=30$, instead of doing a for loop for the values of n from 0 to 3, I just wrote a line that would calculate the corresponding value of $H_{30}(x)$, and then plot it

For part c) I defined the function that returns the value of the wave function ψ . I then defined another function that returns the integrand for the integral of $x^2 |\psi_n(x)|^2$ from $-\infty$ to ∞ since the Gaussian quadrature is only available for finite limits. I then created another function that will evaluate $x^2 |\psi_n(x)|^2$ using Gaussian quadrature with 100 points. Next, I created an uncertainty function that calculates the uncertainty of a particle in the harmonic oscillator potential for a given value of n . This uncertainty is calculated as the square root of the difference between the expectation value of x^2 and the square of the expectation value of x . Finally I calculated the uncertainty for $n=5$. by used the first two Hermite polynomials are $H_0(x) = 1$ and $H_1(x) = 2x$.

4. RESULTS

Nice plot!

For exercise 5.3 continuation, I got the following graph:

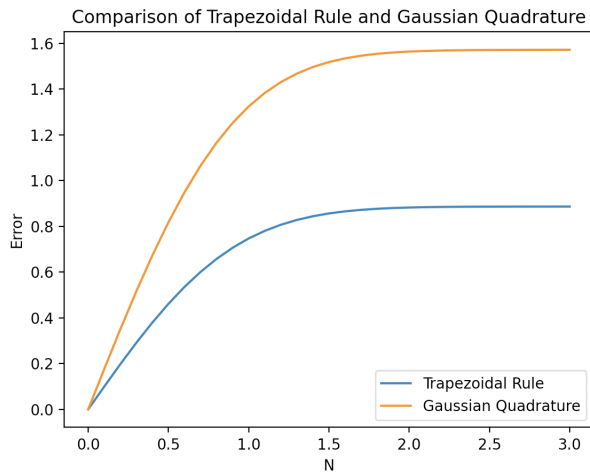


FIG. 1: Graph showing the Error of the Trapezoidal integration and Gaussian quadrature as a function of N for x values from 0 to 3 in steps of 0.1

For exercise 5.13, I got the following two graphs
For part a)

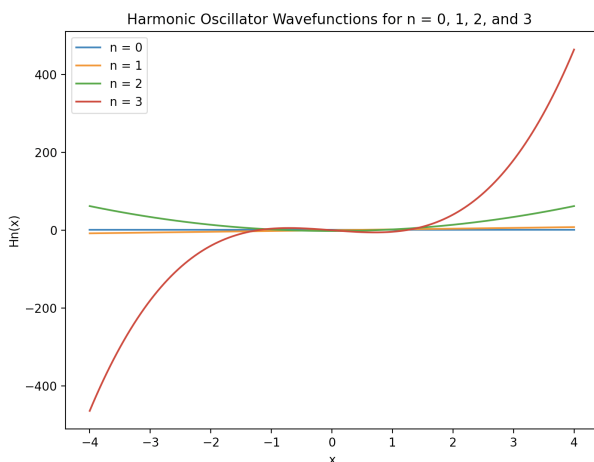


FIG. 2: Graph showing the harmonic oscillator wavefunctions for $n = 0, 1, 2$, and 3 in the range $x = -4$ to $x = 4$

For part b)

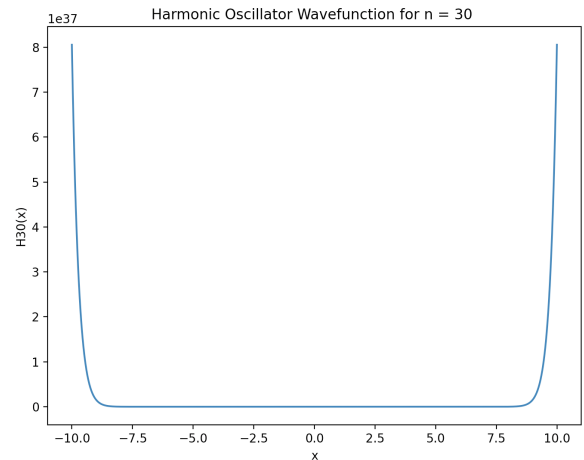


FIG. 3: Graph showing the harmonic oscillator wavefunction for $n = 30$ in the range $x = -10$ to $x = 10$

5. CONCLUSION

+5

Survey Questions For this assignment, it took me a long time mainly because I got ill and I found that exercise 5.13 was pretty hard and time consuming. I had some trouble on part c) since it kept saying that Gaussian quadrature is only available for finite limits, and I needed infinite limits. Which I then went back to the book and read some more, and I learned that when having infinite limits, we could use $x = \tan(z)$. I feel like I learned a lot on this assignment, on both exercise, mainly seeing the both the trapezoidal and Gaussian quadrature curves on the plot for exercise 5.3 continuation, made me realize the big difference they have, and now for future integration I will definitely use Gaussian quadrature. Once again, as I write more in overleaf, I get to learn more ways to use write symbols, and complex equations. Both programs were interesting to me, but I liked exercise 5.3 continuation since I could visibly see the difference between the two methods in the graph. Oh and I like flow charts better now. .

These plots aren't quite right. It looks like you did a change of variable in the integral to go from infinite limits to finite limits, which is right, but it looks like you used some trig to do that. There is an easier change of variable by saying $x=z/(1-z)$.

Computational Physics/Astrophysics, Winter 2023: Grading Rubrics ¹

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 25 points will be available per problem.

- +3 1. Does the program complete without crashing in a reasonable time frame? If yes, up to +3 points.
- +1 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? If yes, +1 points
- +2 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) Up to +2 points
- +4 4. Is the answer correct? Up to +4 points
- +2 5. Is the code readable? Up to +2 points
 - . 5.1. Are variables named reasonably?
 - . 5.2. Are the user-functions and imports used?
 - . 5.3. Are units explained (if necessary)?
 - . 5.4. Are algorithms found on the internet/book/etc. properly attributed?

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

- +2 6. Is the code well documented? +3points
- . 6.1. Is the code author named? [Please put your name at the top of your code](#)
 - . 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?
- +10 7. LaTeX writeup (up to 10 points)
- . Are key figures and numbers from the problem given? (3 points)
 - . Is a brief explanation of physical context given? (2 points)
 - . If relevant, are helpful analytic scalings or known solutions given? (1 point)
 - . Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (2 points)
 - . Are collaborators clearly acknowledged? (1 point)
 - . Are any outside references appropriately cited? (1 point)

[Really great explanation of all the methods in this problem.](#)

Note, even if (1), (2), (3), or (4) are not correct, one can still obtain many points via (5), (6), and (7).

Computational Physics/Astrophysics, Winter 2023:

Grading Rubrics ¹

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 25 points will be available per problem.

- +3 1. Does the program complete without crashing in a reasonable time frame? If yes, up to +3 points.
- +1 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? If yes, +1 points
- +2 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) Up to +2 points
- +1 4. Is the answer correct? Up to +4 points
- +2 5. Is the code readable? Up to +2 points
 - . 5.1. Are variables named reasonably?
 - . 5.2. Are the user-functions and imports used?
 - . 5.3. Are units explained (if necessary)?
 - . 5.4. Are algorithms found on the internet/book/etc. properly attributed?

It looks like something went wrong while calculating the wavefunctions. I suspect it has to do with how you did the change of variable. You can look at the solution set on moodle to see an example.

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

+2 6. Is the code well documented? +3points

Please put your name at the top of your code

- . 6.1. Is the code author named?
- . 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?

+10 7. LaTeX writeup (up to 10 points)

- . Are key figures and numbers from the problem given? (3 points)
- . Is a brief explanation of physical context given? (2 points)
- . If relevant, are helpful analytic scalings or known solutions given? (1 point)
- . Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (2 points)
- . Are collaborators clearly acknowledged? (1 point)
- . Are any outside references appropriately cited? (1 point)

Note, even if (1), (2), (3), or (4) are not correct, one can still obtain many points via (5), (6), and (7).