

$$54 + 55 + 5 = 114$$

$$114/117$$

Phys 304: Homework 5

Mary Smith*
Haverford College Department of Physics
(Dated: March 8, 2024)

Great write-up!

[The homework took me about 10 hours this week. I learned about the nature of Gaussian quadrature and to be careful when using it that it is being used with a polynomial. The most interesting problem was the quantum uncertainty in the harmonic oscillator because it was so challenging. The problem set was just the right length. My collaborator for this problem set was Luke Smithberg.]

1. PROBLEM 1: SIMPSON INTEGRATION

Simpson's rule is a modified and much more accurate version of the trapezoidal approximation of a function. It gives an approximation to the area under two adjacent slices of a function. The application to the area under a function involves the height of the slices and values of the function at evenly spaced values. In formula version, this is given by

$$I(a, b) = \frac{1}{3} [f(a) + f(b) + 4 \sum_{k=1}^{N/2} f(a + (2k-1)h) + 2 \sum_{k=1}^{N/2-1} f(a + 2kh)] \quad (1)$$

where N is the number of slices, $f(x)$ is the function being integrated over, a and b are the bounds of integration, h is $h = \frac{b-a}{N}$, and k is the index of summation.

1.1. Coding Simpson's rule

In order to create the code to use Simpson's rule to approximate the integral of a function, the process outlined in Figure 1 was followed. In this method, we broke the sums into two different for loops and evaluated the function over the for loops. These were then evaluated as the total approximation of the integral. The integral $\int_0^{10} x^2 dx$ was evaluated using the approximation, resulting in a value of 333.333.

2. PROBLEM 2: QUANTUM UNCERTAINTY IN THE HARMONIC OSCILLATOR

A spinless point particle in a quadratic potential well can be modelled as a one-dimensional quantum harmonic oscillator. In units where all the constants are 1, the wavefunction of the n th energy level of the one-dimensional quantum harmonic oscillator is given by

*Electronic address: masmith@haverford.edu

```
def f:
  set N, a, b:
  calc h:
  def I, (f, N, a, b, h):
    for i = 0
    for i in range(1, N/2):
      for i += f(a + (2i-1)h)
    for i = 0
    for i in range(1, N/2-1):
      for i += f(a + 2ih)
  I = 1/3 * h * (f(a) + f(b) + 4 * for1 + 2 * for2)
  return I
```

FIG. 1: [The pseudocode to approximation an integral using Simpson's rule.]

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

for $n = 0, 1, 2, \dots, \infty$, where $H_n(x)$ is the n th Hermite polynomial. The Hermite polynomial is given by

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x), \quad (3)$$

where the first two Hermite polynomials are $H_0(x) = 1$ and $H_1(x) = 2x$. The pseudocode for this problem can be found in Figure 2.

2.1. Part (a): Plot of harmonic oscillator wavefunctions for varying n

A user defined function $H(n, x)$ was created to calculate the Hermite polynomial in Eq. 3 for given x and any integer $n \geq 0$, using recursion. The Hermite polynomial was then plugged into the wavefunction in Eq. 2 to create a user defined function $\psi(n, x)$. This was then used to create a plot that shows the harmonic oscillator wavefunctions for $n=0, 1, 2$, and 3 on the same graph in the range $x=-4$ to $x=4$. This graph can be seen in Figure 3. The figure shows how the number of nodes of a particle in a potential well varies with the number n .

5.13 H304 HW5 Pseudocode

```

(a)
Hn+1(x) = 2xHn(x) - 2nHn-1(x) ⇒ H0(x) = 1, H1(x) = 2x
Hn(x) = 2xHn-1(x) - 2(n-1)Hn-2(x) ← Hermite poly
define Hn(x):
  if n < 0:
    pass
  elif:
    values given
  else:
    h = Hermite poly
  define wavefunction pscn(x):
    if n < 0:
      pass
    else:
      p = Hn(x)
  set domain of x
  plot Hn(x) for n values given
  titles, labels, legend, show

(c)
set domain of x given
plot H30(x)
titles, labels, legend, show

(c)
copy in Newman's code
define root-mean-square function:
  : integral function
  Newman's Gaussian quadrature function of sufficient range
  ⇒ sufficiently large bounds
  for loop from Newman
  square root of result
  print for n=5

```

FIG. 2: [The pseudocode for the quantum uncertainty in the harmonic oscillator.]

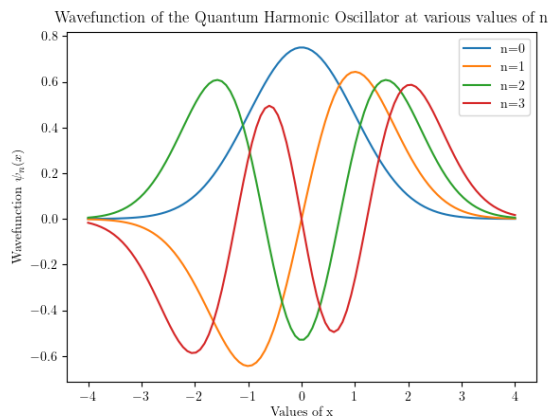


FIG. 3: [A plot showing the harmonic oscillator wavefunctions at various given values of n, in range x=-4 to x=4.]

2.2. Part (c): Plot of the harmonic oscillator wavefunction at value of n

The user defined function for the wavefunction of the harmonic oscillator derived in Sec. 2.1 was used to create

a plot for the harmonic oscillator wavefunction at n=30 over the range x=-10 to x=10. The method for this is highlighted in Figure 2. The plot for this wavefunction is included in Figure 4. The plot shows an interesting behavior for an increased value for n.

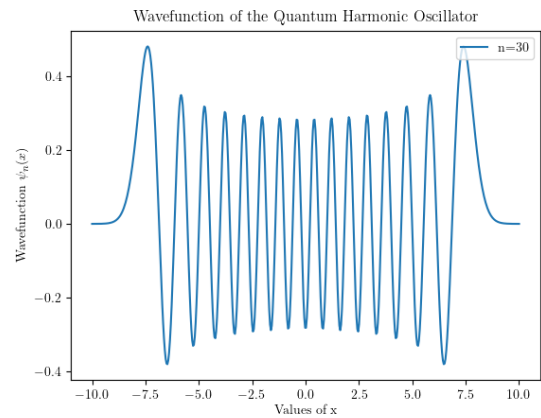


FIG. 4: [A plot showing the harmonic oscillator wavefunction at n=30, in range x=-10 to x=10.]

2.3. Part (c): The quantum uncertainty

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

$$\langle x^2 \rangle = \int_{-\infty}^{\infty} x^2 |\psi_n(x)|^2 dx. \quad (4)$$

A program was written to evaluate this integral using Mark Newman's Gaussian quadrature on 100 points, and then calculate the uncertainty for a given value of x. The program was used to calculate the uncertainty for n=5, about 2.3. However, I encountered something very interesting when running the Gaussian quadrature equation. Because I was using a very large domain in order to accurately model an integration over infinity, I was adding up a lot of very small numbers because of the nature of the integrand. The integrand is not a polynomial, so care must be taken with the domain and the number of points. In order to get an uncertainty close to what was expected, I had to integrate from x=-10 to x=10. If I integrated over a large domain like x=-1000 to x=1000, my uncertainty was 0 because the integrand was not a polynomial.

Q.1

54/56

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)
- 1 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points) *printed answers must have a description such as "the integral of x^2 from 0 to 10, is ..."*
- 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?

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

- . 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?

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) *compare your result with a known solution*
- 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)

5.13

55/56

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)
- 1 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points) *printed answers must have a description such as "the uncertainty is ..."*
- 3 3. Does the code follow the problem specifications (i.e. numerical method; output requested etc.) (+3 points) *-1*
- 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?

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

- . 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?

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)