47 + 44.5 + 5 = 96.5

96.5/117

PHYS 304 AS4

Xiyue Shen[*]
Haverford College Department of Physics
(Dated: March 1, 2024)

Graded Work

1. EXERCISE 5.3 SIMPSON INTEGRAL

Given an integral,

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

✓

in part(a), we are supposed to form a code that calculates the integral. For the integral part, I employ the Simpson rule.

We have,

$$I(a,b) \approx \frac{1}{3}h[f(a)+f(b)+4\sum_{oddk} f(a+kh)+2\sum_{evenk} f(a+kh)] \quad (2)$$

✓

In the second part, we want to plot the integral value over the upper limit. I use the trick that Dan showed on Tuesday. From 0 to 1/2, I integrate over the original equation 1. For the second part, I defined a transformed version $G = \int \frac{e^{-z^2/(1-z)^2}}{(1-z)^2} dz$. Using this equation, we can make the integral toward infinity. Then, to have a more precise value, I double the N value, the number of sample points.

not necessary, but still works

state + explain final choice in N

Should converge to 0.88

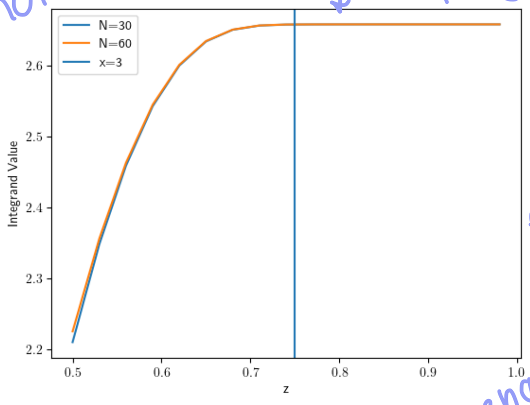

FIG. 1: [Simpson integral o er equation 1]

for 5.3?

In figure 1, I insert a vertical line labeled as $x = 3$, which is the upper limit set in the problem. However, in this plot, the x-axis is z, which is the variable for the transformed integral. The z value goes toward 1, which is infinity for the original function.

2. EXERCISE 5.9 HEAT CAPACITY

In Einstein's quantum model, degrees of freedom freeze out at the temperature below the oscillator frequency. The heat capacity drops like an exponential curve. The Einstein frequency is a fitting parameter [1], which is pretty accurate. However, it still fails to capture the phenomenon of some materials showing that the heat capacity is proportional to T^3. In 1912, Peter Debye found a better model and treated the quantum mechanism of the oscillation of atoms as a sound wave.

In this problem, we are given an equation of heat capacity based on Debye's theory. In part (a), I encoded a script to calculate the heat capacity given a temperature. Here, I employ the Gaussian quadrature method for the integral. First, I input the equation and collect all prefactors. I call them as CV_c. Then, I import the "gaussxw" file from the website https://public.websites.umich.edu/~mejn/cp/chapters.html. Then, I set my weight and space function by calling the "gaussxw" file. I set up two rounds of loops, one for temperature and one for sample points. The sample points loop sits inside the temperature loop. One crucial lesson from Dan's help hour is that the integral needs to be set equal to zero every time before the sample points loop, so that the new integral starts from zero instead of adding to the previous value.

In Figure 2, I use the Simpson method to do the calculation, while in Figure 3, I use the Gaussian Quadrature method. As we can tell, when the temperature is high enough, the heat capacity tends to flatten out, around 10^5. This is slightly high than expected in Debye's model. Since it's approximate, it's close enough. One way to examine the integral is to take the extreme values. When the temperature approaches zero Kelvin, the integral part is around 1; then, the prefactor dominates the heat capacity value, which should be a cubic proportionality. When the temperature approaches infinity, the temperature dependence is canceled out, and a flat line is shown on both plots.

should be closer to 10^3 or around 2500

Survey Questions

✓ +5

I took 5 hours for this week's homework. I learned an important lesson from Gaussian Quadrature. Make sure

Psuedo code?

*Electronic address: xshen2@brynmawr.edu; URL: Optionalhomepage

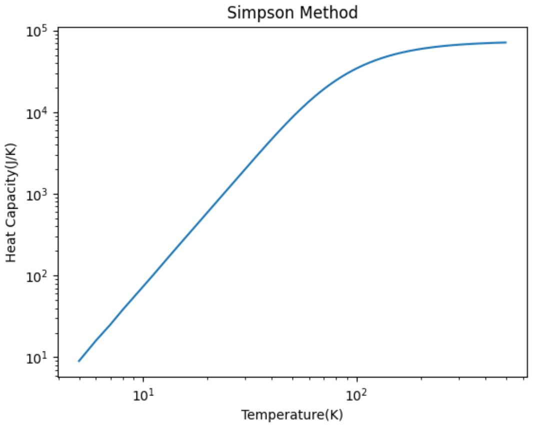

FIG. 2: [Heat capacity problem using Simpson method]

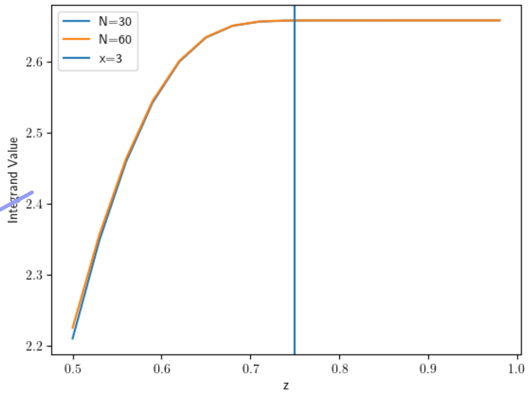

FIG. 3: [Heat capacity problem using Gaussian Quadrature method]

the integral equals zero every time before starting a new one (like a new loop of a new calculation). I like the heat capacity problem most! I think this problem set is about the right length.

Ungraded Work

I have downloaded the repository.

I have followed the class notes and understood every step.

I have completed the tutorial posted on Moodle.

[1] S. H. Simon, *The Oxford solid state basics* (Oxford University Press Oxford, Oxford, 2016), first edition, reprinted (with corrections) ed., ISBN 978-0-19-968076-4 0-19-968076-0 978-0-19-968077-1 0-19-968077-9, section: xv, 291 Seiten : Illustrationen, Diagramme.

5.3 47/56

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)

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)

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

Missing factor of $\frac{1}{3}$ in your odd/even sums −1

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? Name missing? -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) Remember psuedocode -5

0 . 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?

1 . Are key figures and numbers from the problem given? (+ 3 points) state + justify choice of N -2

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)

5.9 44.5/56

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)

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)

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

minor error defining V, should have V=0.001 -1

2 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

Need units for

. 5.3. Are units explained (if necessary)? V, P, θ_D, k_B -1

. 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? Name? -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) Remember psuedo code!

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

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

1.5 . Are key figures and numbers from the problem given? (+ 3 points) define all variables in the problem -1.5

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)

Missing Cv eq

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