52 + 51.5 + 5 = 108.5

$\boxed{108.5/117}$

Phys 304: Assignment 4

Mary Smith*

Haverford College Department of Physics

(Dated: March 2, 2024)

+5

This problem set took me about 5 hours to complete. I learned a lot of interesting things about the different plots you can make with matplotlib and how to apply those plots. I also learned about when to use empty lists when plotting a function. This problem set was the perfect length. My collaborators for this problem set were Joey Carol and Luke Smithberg.

1. PROBLEM 1: DIFFRACTION LIMIT OF A TELESCOPE

The ability for precise measurements to be made with a telescope is limited by the diffraction of light. When observing far away stars, the light can essentially be treated as coming from a point source at infinity. Telescopes take in light with wavelength λ through a central aperture, which is then focused in the focal plane. This produces a circular diffraction pattern caused by the bending of light as it comes in through the aperture. The intensity of light is given by

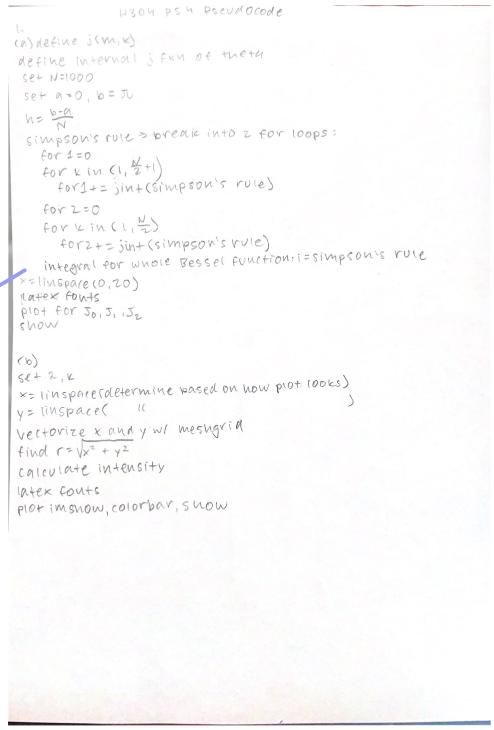
$$I(r) = (\frac{J_1(kr)}{kr})^2 \tag{1}$$

what is λ?

where r is the distance in the focal plane from the center of the diffraction pattern, $k = 2\pi/\lambda$, and $J_1(x)$ is a Bessel function. Bessel functions are given by

$$J_m(x) = \frac{1}{\pi}\int_0^{\pi} \cos m\theta - x\sin\theta d\theta \tag{2}$$

where m is a nonnegative integer and $x \geq$? . The pseudocode for this section can be found in Figure 1.



```
H304 PS4 Pseudocode
1.
(a) define j(m,x)
 define internal j fxn of theta
 set N=1000
 set a=0, b=π
 h= (b-a)/N
 simpson's rule -> break into 2 for loops:
   for1=0
   for k in (1, N/2+1)
     for1+= jint(simpson's rule)
   for2=0
   for k in (1, N/2)
     for2+= jint(simpson's rule)
   integral for whole Bessel function: i=simpson's rule
x=linspace(0,20)
latex fonts
plot for J0, J1, J2
show

(b)
set λ, k
x= linspace(determine based on how plot looks)
y= linspace(        "                        )
vectorize x and y w/ meshgrid
find r=√(x² + y²)
calculate intensity
latex fonts
plot imshow, colorbar, show
```

FIG. 1: [The pseudocode for Problem 1, the diffraction limit of a telescope.]

1.1. Part (a): Finding the Bessel function

The Bessel Function, shown in Eq. 2 was approximated using Simpson's rule. To do so, I created 2 for loops to sum over the summations given in the rule, then applied those sums back into Simpson's rule. This gives an approximation of an integral, which I applied to the Bessel Function. Then, I plotted the Bessel function for several values of m and plotted those in Figure 2.

1.2. Part(b): Intensity of the diffraction

To make a density plot of the intensity of the circular diffraction pattern of a point light source, the process highlighted in Figure 1 was followed. I created arrays for both x and y, then vectorized the x and y arrays. These were then used to calculate the distance from the center of the diffraction pattern r, and that was used to create the intensity function. Lastly, I plotted the intensity in Figure 3.

2. PROBLEM 2: HEAT CAPACITY OF A SOLID

define

Debye's theory of solids gives the heat capacity of a solid at temperature T as

$$C_V = 9V\rho k_B(\frac{T}{\theta_D})^3\int_0^{\frac{\theta_D}{T}} \frac{x^4e^x}{(e^x-1)^2}dx, \tag{3}$$

where V is the solid's volume, ρ is the number density

*Electronic address: masmith@haverford.edu

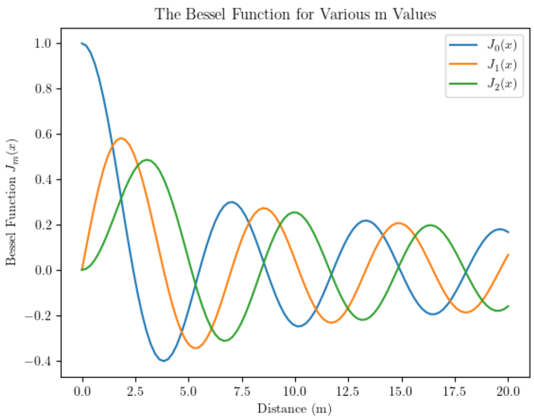


FIG. 2: [The Bessel functions for several values of m, plotted against distance.]

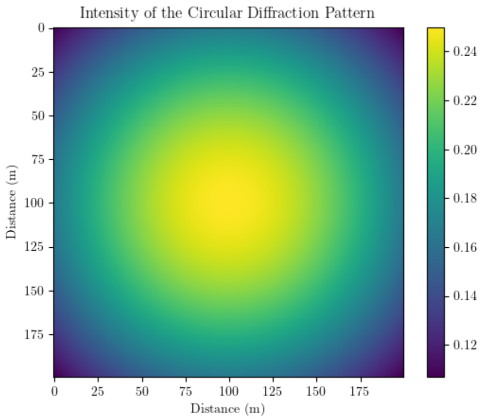


FIG. 3: [The intensity of the circular diffraction pattern of a point light source.]

include numerical values for all variables

of atoms, k_B is the Boltzmann constant, and θ_D is the Debye temperature. This Debye temperature is a property of solids that depends on their density and the speed of sound. The pseudocode for this section can be found in Figure 4.

2.1. Part (a): Calculating heat capacity

To create a function for the heat capacity at constant volume C_V for the given conditions provided, I followed the process highlighted in the pseudocode in Figure 4. I defined an internal integral function of x, then used the Guassian quadrature function provided by Mark Newman in the textbook to approximate a solution to that integral. I then plugged the result of this integral into Equation 3. This generates heat capacity values to be used in the next section.

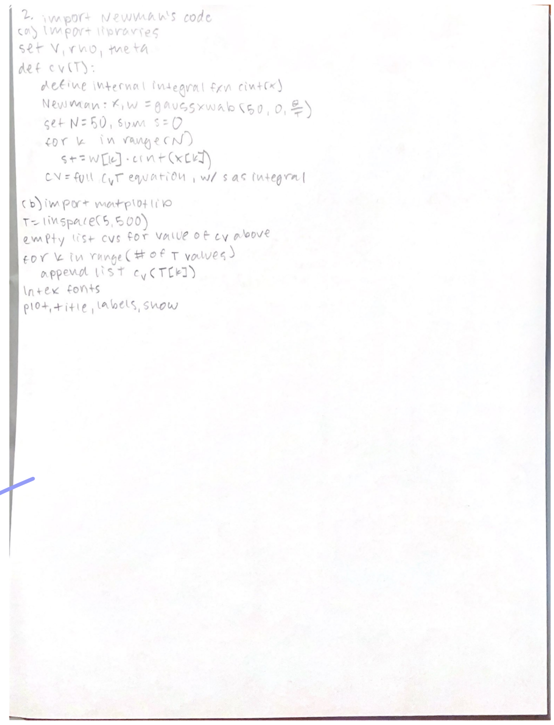


FIG. 4: [The pseudocode for Problem 2, find the heat capacity of a solid.]

2.2. Part (b): Plotting heat capacity

To create a plot of the heat capacity as a function of temperature from $T = 5\text{K}$ to $T = 500\text{K}$, I followed the processes outlined in Figure 4. I created an array of values for T and an empty list for the values of heat capacity. Then I appended this list in the range of the number of T values as the heat capacity at each temperature. From here, a plot of heat capacity values was generated in Figure 5.

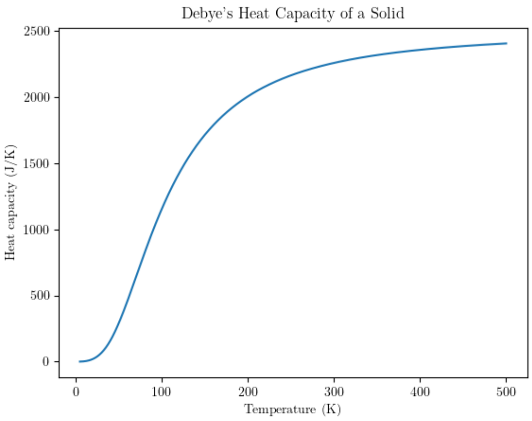


FIG. 5: [The heat capacity of a solid from temperatures 5-500K.]

5.4 52/56

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)

2 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points) ← doublecheck conversion of λ to meters -1

3 6. Is the output answer correct? (+4 points). slight errors in density plot due to errors above -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? Missing 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)

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?

2 . Are key figures and numbers from the problem given? (+ 3 points) Numerically define λ -1

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 51.5/56

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)

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? Missing 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)

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?

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

0. Is a brief explanation of physical context given? (+2 points) No context given, define Heat capacity and explain trends in graph -2

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)