Computational Physics, HW 4

Ben Johnston

October 2023

1 Problem 1

The first part of this problem involved writing a function that calculates the heat capacity, C_V as a function of temperature, T for $1000cm^3$ of solid aluminium. The equation used to determine this relationship comes from Debye's theory of solids and is given by:

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

where V is the volume of the solid, $\rho = 6.022 \times 10^{28}\text{m}^{-3}$ is the number density of atoms, k_B is the Boltzmann constant, $\theta_D = 428\text{K}$ is the Debye temperature. The script written uses Gaussian quadrature to evaluate the integral with $N = 50$ sample points. *Figure 1* below shows the output of the script:

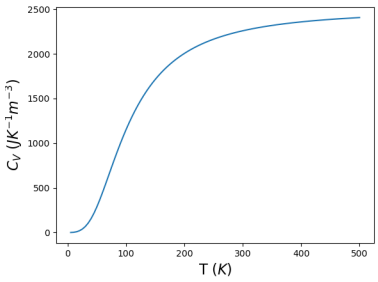

Figure 1: Heat capacity of aluminium as a function of temperature

The graph clearly shows that as temperature increases, so does the heat capacity, the shape of which follows the expected trend.

The aim of the second part of this question was to test the convergence by evaluating the integral at $N = 10, 20, 30, 40, 50, 60, 70$ sample points. In order to calculate this the difference for equivalent points for subsequent values of N was determined, following which the differences were summed up and the absolute value of this sum was taken. A log-linear graph was then plotted of $\log(diff)$ vs N as seen in *Figure 2*:

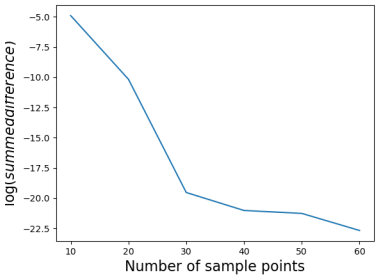

Figure 2: log(diff) as a function of N

At a value of $N = 70$ Python encountered a RunTimeWarning of divide by zero, meaning that the summed difference between N=60 and N=70 was zero, hence the calculation had converged at $N = 60$ as both $N = 60$ and $N = 70$ have exactly the same values.

2 Problem 2

This problem involved determining the period of oscillation for an anharmonic oscillator, whose period varies with amplitude and cannot (usually) be calculated analytically. An anharmonic oscillator corresponds to an oscillator whose potential, $V(x)$ does not depend on x^2. If the particle has mass, m and position, x, then the total energy of the system is given by:

$$E = \frac{1}{2}\left(\frac{dx}{dt}\right)^2 + V(x) \tag{2}$$

which is a nonlinear equation for x and t as the total energy must be constant. We assume here that the potential is symmetric around $x = 0$ and the oscillator has an amplitude of a. Therefore, at $t = 0$, $x = a$ and $\frac{dx}{dt} = 0$ and so the above equation reduces to:

$$E = V(a) \tag{3}$$

The first part of this question was to rearrange eq.(2) to show that the period of oscillation of an anharmonic oscillator is given by:

$$T = \sqrt{8m}\int_0^a \frac{dx}{\sqrt{V(a) - V(x)}} \tag{4}$$

The steps for this are as follows:

$$E = \frac{1}{2}\left(\frac{dx}{dt}\right)^2 + V(x)$$

$$\sqrt{E - V(x)} = \sqrt{\frac{m}{2}}\frac{dx}{dt}$$

$$dt = \sqrt{\frac{m}{2}}\frac{dx}{\sqrt{E - V(x)}}$$

$$\frac{T}{4} = \sqrt{\frac{m}{2}}\int_0^a \frac{dx}{\sqrt{V(a) - V(x)}}$$

$$T = 4\sqrt{\frac{m}{2}}\int_0^a \frac{dx}{\sqrt{V(a) - V(x)}}$$

$$T = \sqrt{8m}\int_0^a \frac{dx}{\sqrt{V(a) - V(x)}}$$

As required. We now suppose the potential is given by $V(x) = x^4$ and that the mass of the particle is $m = 1$. A Python script was written that calculates the period of the oscillator for a given amplitude using Gaussian quadrature with 20 points in the region $0 \leq a \leq 2$. The output produced can be seen in *Figure 3* below:

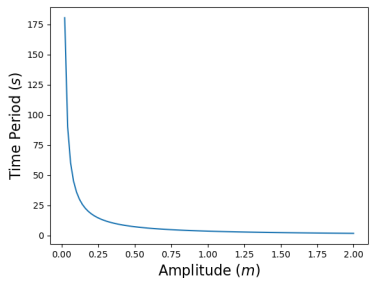

Figure 3: Time period as a function of amplitude

It is clear from the plot that as the amplitude increases, the oscillator gets faster and that the period diverges as the amplitude goes to zero. A possible reason for this is that as the potential $\sim x^4$ in this case (instead of the usual x^2 dependence for a simple harmonic oscillator), as the amplitude increases the steeper sides of the x^4 potential starts to play a significant role in increasing the frequency of oscillation. In other words the non-linearity here causes the oscillator's restorative force to depend on the displacement, x. Therefore, as the amplitude increases the restorative force increases, thus reducing the time period of the oscillation. The same reasoning can be used to explain why the time period diverges towards zero amplitude as in when the amplitude tends to zero, the restorative force also tends to zero which leads to the time period diverging.

3 Problem 3

This question relates to quantum uncertainty in the harmonic oscillator. In units where all the constants are 1, the wavefunction of the nth energy level of a 1D quantum harmonic oscillator is given by:

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

for $n = 0, ..., \infty$, where $H_n(x)$ is the nth Hermite polynomial, which are given by the formula:

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

with the first 2 polynomials being $H_0(x) = 1$ and $H_1(x) = 2x$

The first part of the problem required a Python script to be written that calculates $H_n(x)$ for a given x and any integer $n \geq 0$, which was then to be used to plot the harmonic oscillator wavefunctions for $n = 0$, 1, 2 and 3 on the same graph in the range $-4 \leq x \leq 4$. The output of this script can be seen in *Figure 4*:

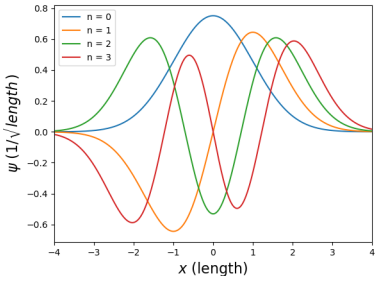

Figure 4: Harmonic oscillator wavefunctions for n=0,1,2,3

It can be seen here that the wavefunctions all look as expected for the first 4 levels of a harmonic oscillator and so it can be deduced that the functions to generate the Hermite polynomials and the wavefunctions are working correctly.

Then, the second part of the question was asking for a separate plot of the wavefunction for $n = 30$ in the range $-10 \leq x \leq 10$, the output of which can be seen below:

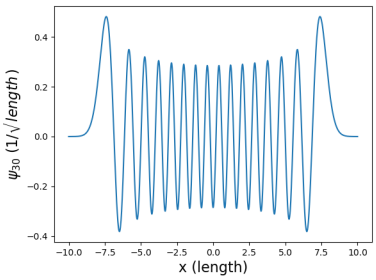

Figure 5: Wavefunction for n=30

There was a caveat to this question in which it said if the program was taking too long to run then it is possible that the calculation was being performed incorrectly. It can be confirmed that the script *Problem3b.py* does not take very long to run (around 4 seconds) and so it can be inferred that the calculation has been set up correctly.

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

$$\langle x^2 \rangle = \int_{-\infty}^{\infty} x^2 |\psi_n(x)|^2 dx \tag{7}$$

The third part of this problem involved writing a script that evaluates this integral using Gaussian quadrature on 100 sample points for a given value of n, which for this case was $n = 5$. In order to evaluate this integral, the following transformation was undertaken in order to evaluate the integral between $-1 \leq x \leq 1$ instead of $-\infty \leq x \leq \infty$:

$$x = \frac{z}{1 - z^2} \quad , \quad dx = \frac{1 + z^2}{(1 - z^2)^2} dz \tag{8}$$

which would yield:

$$\int_{-\infty}^{\infty} f(x)dx = \int_{-1}^{1} \frac{1+z^2}{(1-z^2)^2} f\left(\frac{z}{1-z^2}\right) dz \tag{9}$$

Once this transformation was undertaken, the integral was performed, which yielded an uncertainty in the position of the particle to be:

$$\sqrt{\langle x^2 \rangle} = 2.345207873785824$$

It was stated in the question that the answer determined should be in the vicinity of $\sqrt{\langle x^2 \rangle} = 2.3$, which this answer is, therefore the calculation seems to have been performed correctly.

The fourth and final part of this question was to repeat the above integral calculation, however now using Gauss-Hermite quadrature. This type of quadrature is used for approximating the value of integrals of the form:

$$\int_{-\infty}^{\infty} e^{-x^2} f(x)dx \tag{10}$$

As this type of quadrature is based on a different function class, the wavefunction needed to be altered to:

$$\psi_n(x) = \frac{1}{\sqrt{2^n n! \sqrt{\pi}}} H_n(x) \tag{11}$$

Using the same equation for $\langle x^2 \rangle$ as in eq.(7) the integral was calculated, from which the uncertainty in the particle's position was determined to be:

$$\sqrt{\langle x^2 \rangle} = 2.3452078799117193$$

In order to check whether the result determined here is an exact integral (i.e. zero approximation error) a plot of the calculated uncertainty in position value vs number of integration points was calculated as seen in *Figure 5* below:

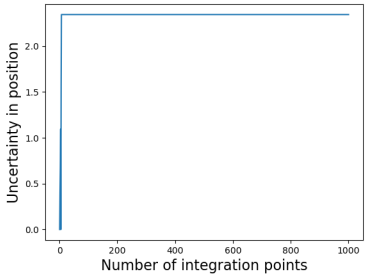

Figure 6: Uncertainty in position as a function of number of integration points

From the above graph it can be seen that at around 10 integration points, the value of the uncertainty in position remains seemingly constant as the number of integration points increases. This indicates that it is possible that there may be a zero error approximation here. However, when inspecting the data more closely there were discrepancies in the values at around 10^{-12}, which can be attributed to the fact that there may be round off errors in the calculation. From this it can be seen that both Gaussian quadrature and Gauss-Hermite quadrature can calculate integrals accurately, however for this function especially (due to the Hermite polynomials) Gauss-Hermite generates an accurate value with a number of integration points around one order of magnitude smaller than the corresponding calculation d