Group Project

Adityan S

Contents

1 Oscillation **2**

1.1 First consider a particle with $m = 1kg$ in the potential $V_1(x) = 0.5\alpha x^2$ with $\alpha = 4kgsec^2$. Numerically calculate the time period of oscillation and check this against the expected value. Verify that the frequency does not depend on the amplitude of oscillation. 2

1.2 Next consider a potential $V_2(x) = e^{0.5x^2} - 1$. Numerically verify that for small amplitude oscillations you recover the same results as the simple harmonic oscillator. 3

2 Black Body Spectrum **5**

2.1 Plot the COBE data and see if it has a shape similar to the black body spectrum first explained by Planck. 5

2.2 Use these data to deduce the temperature T of the cosmic microwave background radiation. . . 5

2.3 Assess the accuracy of the fit by doing a Pearson's χ^2 (chi-squared) analysis. 6

3 Modules **8**

3.1 Module `mods` consists of all subroutines, variables and procedures that are used in this report. The codeblocks below are all part of the file `mods.f90` 8

3.2 Interfaces 8

3.3 Routines for Array Manipulation 8

3.3.1 Linspace Subroutine for creating an array(sequence) with the equal step value. 8

3.4 Routines for Numerical Integration 8

3.4.1 N-point Gauss Chebyshev Quadrature of 1st kind. 8

3.4.2 N-point Gauss Legendre Quadrature of 1st kind. 9

3.5 Routines for Optimization of Non Linear Equation 9

3.6 End of module `mods` 9

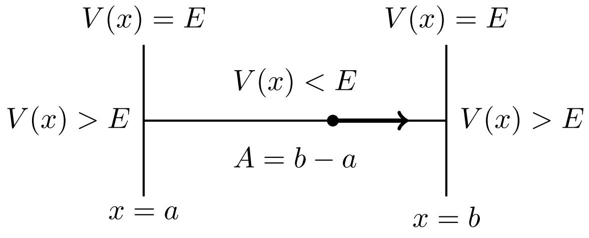
1 Oscillation

Consider the one-dimensional motion of a particle of mass m in a time independent potential $V(x)$.Since the energy E is conserved, one can integrate the equation of motion and obtain a solution in a closed form.

$$t - C = \sqrt{\frac{m}{2}} \int_{x_i}^{x} \frac{dx}{\sqrt{E - V(x)}}$$

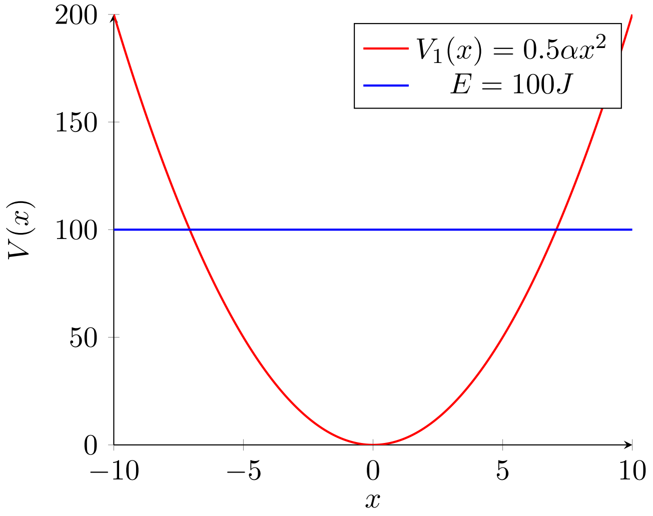
For the choice $C = 0$, the particle is at the position x_i at the time $t = 0$ and x refers to its position at any arbitrary time t. Consider a particular case where the particle is in bound motion between twopoints a and b where $V(x) = E$ for $x = a, b$ and $V(x) < E$ for $a < x < b$. The time period of theoscillation T is given by,

$$T = 2\sqrt{\frac{m}{2}} \int_{a}^{b} \frac{dx}{\sqrt{E - V(x)}}$$



1.1 First consider a particle with $m = 1kg$ in the potential $V_1(x) = 0.5\alpha x^2$ with $\alpha = 4kgsec^2$. Numerically calculate the time period of oscillation and check this against the expected value. Verify that the frequency does not depend on the amplitude of oscillation.

Particle with an Energy of 100J



From the above plot, we infer that the for a value of Energy E, the particle is bound for specific values of a and b (decided by E).

$$E = V_1(a) = V_1(b) = 2x^2$$

$$\implies a = -\sqrt{\frac{E}{2}} = -A, \quad b = +\sqrt{\frac{E}{2}} = +A$$

Hence Amplitude of oscillation is a function of Energy.

$$A(E) = b - a = \sqrt{\frac{E}{2}} \implies E = 2A^2$$

From the above equations, the analytic solution,

$$T = \frac{1}{A}\int_{-A}^{+A} \frac{dx}{\sqrt{1-(\frac{x}{A})^2}} = sin^{-1}\left(\frac{x}{A}\right)\Big|_{-A}^{+A} = \pi \quad (sec)$$

Using integral identities, T can be written as,

$$T = \int_{-1}^{+1} \frac{dx}{\sqrt{1-x^2}}$$

Let us solve the above numerically,

```
program occilation1
  use mods
  implicit none
  integer :: i
  integer, parameter :: N = 2
  real(dp), dimension(N) :: x, w
  call gauss_chebyshev_1(x, w)
  print *, sum(w)
end program occilation1
```

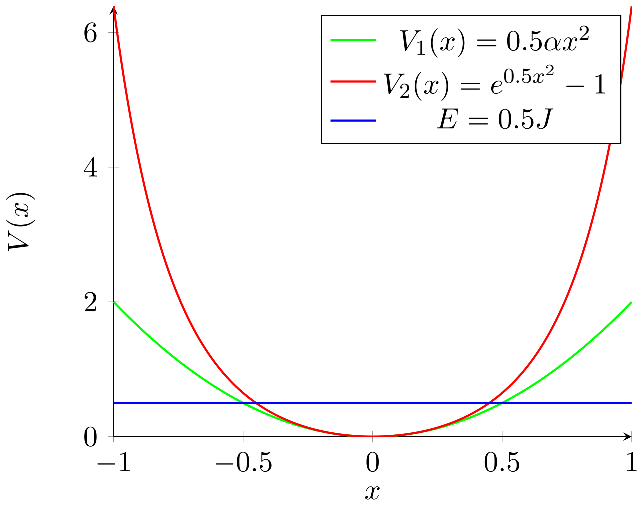
```
fpm run oscillation1
```

```
3.141592653589793
```

The above integral $T = 3.141592653589793 \approx \pi \neq T(E, A)$. Hence the timeperiod is independent of Amplitude.

1.2 Next consider a potential $V_2(x) = e^{0.5x^2} - 1$. Numerically verify that for small amplitude oscillations you recover the same results as the simple harmonic oscillator.

Particle with an Energy of 0.5J



From the above plot, we can infer that for particles of very small energies the given potential behaves as the simple harmonic occilator.

Let us verify the above statement numerically,

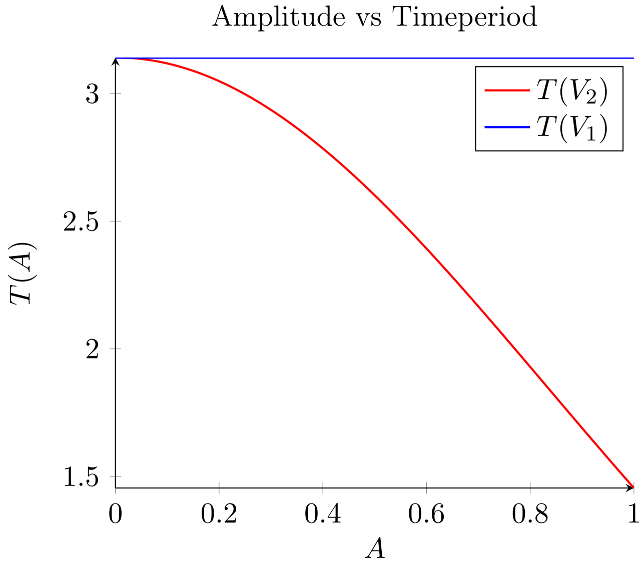
$$T = \frac{\sqrt{2}}{e^{A^2}}\int_{-A}^{A} \frac{dx}{\sqrt{1-e^{2(x^2-A^2)}}} = \frac{\sqrt{2}A}{e^{A^2}}\int_{-1}^{1} \frac{dx}{\sqrt{1-e^{2A^2(x^2-1)}}}$$

$$\implies A = \pm\sqrt{\frac{\ln{(E+1)}}{2}}$$

```
program occilation2
  use mods
  implicit none
  integer :: i
  integer, parameter :: N = 5000
  real(dp), dimension(100) :: A
  real(dp), dimension(N) :: x, w
  call gauss_legendre(x, w)
  call linspace(0.01_dp, 1.0_dp, A)
  open(1, file='./src/oscillation/o2T.csv', status='old')
  do i=1,100
     write (1,'(1f10.7, a, 1f10.7)') A(i), " ",
↪  sum((sqrt(2._dp)*A(i)/(e**(A(i)**2)))*w*(1/sqrt(1-e**(2*(A(i)**2)*((x**2)-1)))))
  end do
  write (*, *) "$A$ $T$"
  do i=1,10
     write (*,'(1f10.7, a, 1f10.7)') A(i), " ",
↪  sum((sqrt(2._dp)*A(i)/(e**(A(i)**2)))*w*(1/sqrt(1-e**(2*(A(i)**2)*((x**2)-1)))))
  end do
end program occilation2
```

```
fpm run oscillation2
```

A	T
0.01	3.1410088
0.02	3.1403022
0.03	3.1391248
0.04	3.1374772
0.05	3.13536
0.06	3.1327742
0.07	3.1297208
0.08	3.1262011
0.09	3.1222167
0.1	3.1177692



Hence for Oscillation of small amplitudes, we get the same results as simple harmonic oscillator.

2 Black Body Spectrum

Quantum mechanics began with Planck's fit to the spectrum of black body radiation:

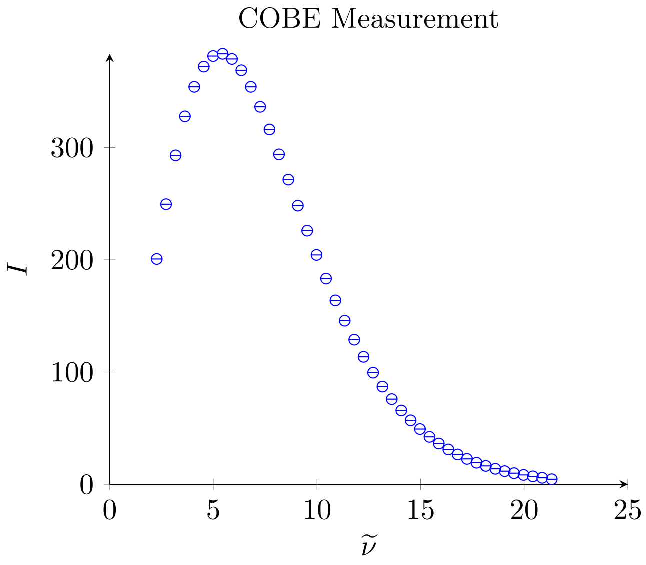
$$I(\nu, T) = \frac{2h\nu^3}{c^2}\frac{1}{e^{\frac{h\nu}{kT}} - 1}$$

Here $I(\nu, T)$ is the energy per unit time of radiation with frequency ν emitted per unit area of emitting surface, per unit solid angle, and per unit frequency by a black body at temperature T. The parameter h is Planck's constant, c is the speed of light in vacuum, and k is Boltzmann constant. The Cosmic Background Explorer (COBE) project measured the cosmic background radiation and obtained the results given in the file `cobe.dat`

```
cat ./src/cobe/cobe.dat
```

$\widetilde{\nu}$	$I(\nu, T)$	Error
2.27	200.723	14
2.72	249.508	19
3.18	293.024	25
3.63	327.77	23
...	...	...
20.42	7.087	88
20.87	5.801	155
21.33	4.523	282

2.1 Plot the COBE data and see if it has a shape similar to the black body spectrum first explained by Planck.


COBE Measurement

Yes, the given data has a shape similar to the black body spectrum first explained by Planck.

2.2 Use these data to deduce the temperature T of the cosmic microwave background radiation.

$$I = \frac{0.0014745\nu^3}{e^{b\nu} - 1} \quad in \quad \frac{MJy}{sr}$$

$$\nu = 30\widetilde{\nu} \quad in \quad GHz, \quad \widetilde{\nu} \quad in \quad \frac{1}{cm^{-1}}$$

$$T = 10^9\frac{h}{bk}, \quad b \quad in\frac{1}{GHz}, \quad T \quad in \quad K$$

Using Least Square Optimization

$$b = min\{\sum_{i=1}^{43} r_i(b)\} = min\{\sum_{i=1}^{43}(\frac{0.0014745\nu_i^3}{e^{b\nu_i}-1} - I_i)^2\}$$

Let the Initial Guess for b be $b = 0.048 \implies T \approx 1K$

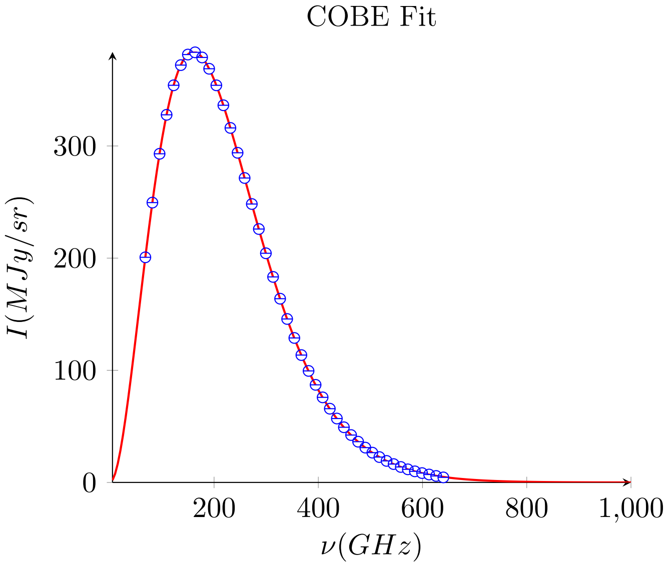
```
import numpy as np
from scipy.optimize import least_squares
import pandas as pd
import warnings
warnings.filterwarnings("ignore")

data = pd.read_csv("src/cobe/cobeplot.csv", header=None, delim_whitespace=True)
vbi = np.asarray(data[0])
Ii = np.asarray(data[1])
vi = 30 * vbi

def r(b, vi, Ii):
    return ((0.0014745*pow(vi,3))/(np.exp(b*vi)-1) - Ii)**2

b = least_squares(r, 0.048, args=(vi, Ii))
b.x
```

```
[0.01760962]
```

$$b = 0.01760962 \implies T \approx 2.73K$$

2.3 Assess the accuracy of the fit by doing a Pearson's χ^2 (chi-squared) analysis.

```
program cobe
  use mods
  implicit none

  real(dp) :: v(200), I(200), vi(43), vbi(43), Ii(43), erri(43), chisq
  integer :: m, j

  m = 43 ! No of data points
  open(99, file="./src/cobe/cobe.dat")
  open(98, file="./src/cobe/cobefit.csv")
  open(97, file="./src/cobe/cobeplotfreq.csv")

  do j=1, m
     read(99, *) vbi(j), Ii(j), erri(j)
  end do
  vi = 30._dp*vbi
  do j=1, m
     write(97, *) vi(j), Ii(j)
  end do
  call linspace(0._dp, 1000._dp, v)
  do j=1, 200
     I(j) = blackbody(v(j))
     write(98, "(2f20.14)") v(j), I(j)
  end do

  chisq = 0._dp
  do j=1,m
     chisq = chisq + (((Ii(j) - blackbody(vi(j)))**2._dp)/(blackbody(vi(j))))
  end do
  write(*, "(1f20.14)") chisq

  close(99)
  close(98)
  close(97)
contains
  real(dp) function blackbody(v) result(I)
    real(dp), intent(in) :: v
    I = (0.0014745_dp*(v**3._dp))/(exp(v*0.01760962_dp)-1._dp)
  end function blackbody
end program cobe
```



Hypothesis	Plank's Law Fit
Expected Values	$I(\nu_i)$
Observes Values	I_i
No of Observations	43

$$\chi^2 = \sum_{i=1}^{43} \frac{(I_i - I(\nu_i))^2}{I(\nu_i)}$$

```
fpm run cobe
```

```
0.04425666834533
```

$$\chi^2 = 0.04425666834533 < 1$$

Hence it is a good fit

3 Modules

3.1 Module `mods` consists of all subroutines, variables and procedures that are used in this report. The codeblocks below are all part of the file `mods.f90`

```fortran
module mods
  use, intrinsic :: iso_fortran_env, only: dp => real64
  use stdlib_quadrature, only: gauss_legendre
  implicit none

  real(dp), parameter :: pi=3.14159265358979323846_dp
  real(dp), parameter :: e= 2.7182818284590452353_dp
```

3.2 Interfaces

```fortran
contains
```

3.3 Routines for Array Manipulation

3.3.1 Linspace Subroutine for creating an array(sequence) with the equal step value.

```fortran
subroutine linspace(from, to, array)
  real(dp), intent(in) :: from, to
  real(dp), intent(out) :: array(:)
  real(dp) :: range
  integer :: n, i
  n = size(array)
  range = to - from
  if (n == 0) return
  if (n == 1) then
     array(1) = from
     return
  end if
  do i=1, n
     array(i) = from + range * (i - 1) / (n - 1)
  end do
end subroutine linspace
```

3.4 Routines for Numerical Integration

3.4.1 N-point Gauss Chebyshev Quadrature of 1st kind.

$$\int_{-1}^{+1} \frac{f(x)}{\sqrt{1-x^2}}\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

$$x_i = \cos\left(\frac{2i-1}{2n}\pi\right) \qquad w_i = \frac{\pi}{n}$$

```fortran
subroutine gauss_chebyshev_1(x, w)
  real(dp), intent(out) :: w(:), x(:)
  integer :: n, i
  n = size(w)
  do i=1, n
     x(i) = cos((pi*(2*real(i)-1))/(2*real(n)))
     w(i) = pi/real(n)
  end do
end subroutine gauss_chebyshev_1
```

3.4.2 N-point Gauss Legendre Quadrature of 1st kind.

$$\int_{-1}^{+1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i)$$

$$w_i = \frac{2}{\left(1 - x_i^2\right)\left[P_n'(x_i)\right]^2} \quad P_n(x_i) = 0$$

```
!use stdlib_quadrature, only: gauss_legendre
```

3.5 Routines for Optimization of Non Linear Equation

Using SCIPY

3.6 End of module mods

```
end module mods
```