Università degli Studi di Trento – Dipartimento di Fisica

Computational Physics 1

Recitation class, March 31, 2021

Quantum scattering

Indice

1 Tunneling in one-dimensional barriers

1.1 The rectangular barrier

Implement the Numerov algorithm and use it to calculate the transmission coefficient of a quantum particle of mass m through a one-dimensional rectangular barrier, defined as

$$V(x) = \begin{cases} V_0 & \text{for } -\frac{a}{2} < x < \frac{a}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

with $V_0 > 0$ and use it to reproduce the following figure, taken from the Wikipedia.

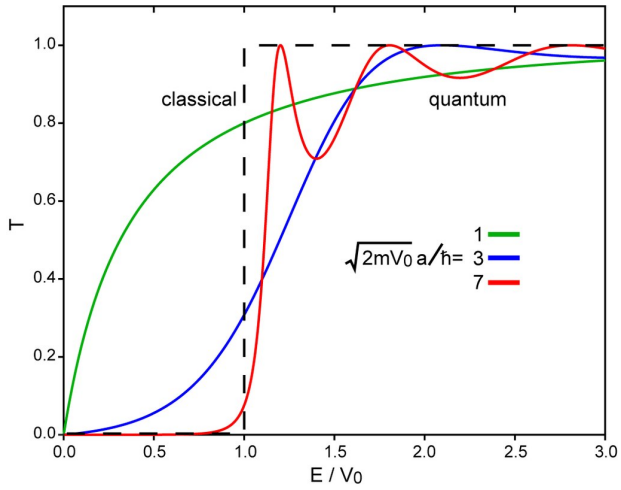

Figura 1: The transmission coefficient of a rectangular barrier

1.2 The Gaussian barrier

Compute, under the same conditions as the previous section, the transmission coefficient of the Gaussian barrier in one dimension

$$V(x) = V_0 \exp\left(-\frac{x^2}{2a^2}\right). \tag{2}$$

1.3 An asymmetric barrier

Calculate the transmission and reflection coefficients in the case of an asymmetric barrier

$$V(x) = \begin{cases} V_L & \text{for } -\frac{a}{2} < x < 0 \\ V_R & \text{for } 0 < x < \frac{a}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

in the cases where $[V_L = V_0/2; V_R = V_0]$ and $[V_L = V_0; V_R = V_0/2]$. Verify that they turn out equal.

1.4 Transmission coefficients of potential wells

Repeat the calculation for the rectangular and Gaussian potential wells in one dimension, defined as in Eqs. (1) and (2), respectively, with $V_0 < 0$.
What happens in the classical ($\hbar \to 0$) limit?

2 Three-dimensional scattering

2.1 Hard-sphere scattering

Consider particles of mass m scattering off a hard sphere of radius a. Derive expressions of the *classical* differential and total scattering cross sections. Verify that the first is a constant and the last is πa^2, as expected.

In the case of quantum particles, calculate the phase shifts numerically and verify that they are given by the following expression

$$\sin^2 \delta_l = \frac{j_l(ka)^2}{j_l(ka)^2 + n_l(ka)^2}. \tag{4}$$

- What is the value of the total scattering cross section for $ka \to 0$?
- (Not easy but worth it) What is the classical limit of the quantum cross section?
  - How is this so?*

2.2 H–Kr scattering

It has been proposed that the interaction between atomic hydrogen and krypton can be approximated with a Lennard-Jones potential having

$$\begin{aligned} \varepsilon/k_\mathrm{B} &= 68.5 \text{ K} \\ \sigma &= 3.18 \text{ Å}. \end{aligned}$$

Check this assumption by calculating the total scattering cross section between H and Kr, for kinetic energies in the range 0.05 meV to 5 meV and comparing the results with the experimental values reported by J. P. Toennies and coworkers in figure 3 of J. Chem. Phys., **71** (1979), 614.

2.3 Notes

In order to answer this set of problems, it will be necessary to compute the spherical Bessel functions and the Legendre polynomials. To this end, please use the functions provided by the https://www.gnu.org/software/gsl/ (GSL). In particular the functions needed are

- `gsl_sf_bessel_jl(l,x)` and `gsl_sf_bessel_yl(l,x)` for the evaluation of $j_l(x)$ and $n_l(x)$, respectively;
- `gsl_sf_legendre_Pl(l,x)` for the Legendre polynomials $P_l(x)$.

In order to use this library, one has to include the function definitions at the beginning of the program with

*Did I really have to ask?

```
#include <gsl/gsl_sf.h>
```

and compile it including the reference to the library, that is

```
gcc -O2 -o program program.c -lgsl -lgslcblas -lm
```

You can check that GSL has been correctly installed by compiling the following code

```
#include <stdio.h>
#include <gsl/gsl_sf.h>

int main()
{
  printf("j_0(1) = %g\n",gsl_sf_bessel_jl(0, 1.0));
  printf("j_1(2) = %g\n",gsl_sf_bessel_jl(1, 2.0));
  printf("j_2(3) = %g\n",gsl_sf_bessel_jl(2, 3.0));

  printf("n_0(1) = %g\n",gsl_sf_bessel_yl(0, 1.0));
  printf("n_1(2) = %g\n",gsl_sf_bessel_yl(1, 2.0));
  printf("n_2(3) = %g\n",gsl_sf_bessel_yl(2, 3.0));

  printf("P_0(0.3) = %g\n",gsl_sf_legendre_Pl(0,0.3));
  printf("P_1(0.3) = %g\n",gsl_sf_legendre_Pl(1,0.3));
  printf("P_2(0.3) = %g\n",gsl_sf_legendre_Pl(2,0.3));
}
```

whose output should be

```
j_0(1) = 0.841471
j_1(2) = 0.435398
j_2(3) = 0.298637
n_0(1) = -0.540302
n_1(2) = -0.350612
n_2(3) = -0.267038
P_0(0.3) = 1
P_1(0.3) = 0.3
P_2(0.3) = -0.365
```