Project 2: Solving quantum systems numerically

Key Topic: Numerical integration and Monte Carlo methods

1 Introduction

Consider a time-dependent 1D quantum-mechanical system, consisting of a single particle initially well-localised in space. We can model this particle's wavefunction as a wavepacket, i.e. a Gaussian in both frequency and position space. In position space, at some fixed point in time, this is

$$\Psi(z) = \frac{1}{\pi^{1/4}} e^{ia(z)} e^{-z^2/2}, \tag{1}$$

where z is a spatial position in some arbitrary units, and $a(z)$ is some (real-valued) phase function that depends on the position.

Now imagine designing an experiment to measure the position of this particle. From quantum mechanics, you know that the probability of measuring the particle's position to be somewhere between z_1 and z_2 is given by the square of the absolute value of the wavefunction, integrated between z_1 and z_2. This particular wavefunction cannot be integrated analytically however, so you can't find a closed-form expression for the probability of finding the particle in the range $z \in [z_1, z_2]$. In this assignment, you will learn how to calculate this probability numerically. This is much more powerful than the analytical approach, as it allows you to obtain a solution regardless of the form of the wavefunction.

2 Newton-Coates integration

a) Write a general routine to perform numerical integration on a user-supplied 1D function using the extended trapezoidal rule. Design your program so that it automatically refines the result to a user-specified relative accuracy ϵ. Here 'user-supplied' means that you should design your integration function so that one of its arguments is a pointer to another function, which will be the integrand. This other function should take a single argument (the value of the integration variable) and return the value of the integrand at that value of the integration variable.

b) Write another routine to do the same as a), but this time using the extended version of Simpson's Rule. Note the trick that relates this to evaluations of the Trapezoidal Rule, discussed in class and in Numerical Recipes!

c) Use both your integrators from a) and b) to compute the probability of finding the particle between $z = 0$ and $z = 2$, i.e.

$$P \equiv \int_0^2 |\Psi(z)|^2 \, \mathrm{d}z, \tag{2}$$

to a relative accuracy of $\epsilon = 10^{-6}$. Compare the number of function evaluations required in each case.

3 Monte Carlo integration

Write a Monte Carlo integration routine that can integrate a user-supplied function using importance sampling, and automatically refines its result to a user-specified relative accuracy ϵ. Perform the integral Eq. 2 twice more, this time using your Monte Carlo integrator with

a) A flat sampling pdf (i.e. no importance sampling)

b) A sampling pdf that drops off from a maximum at $x = 0$ to a (non-zero) minimum at $x = 2$, e.g.

$$\mathrm{pdf}(x) = Ax + B, \qquad \text{with } A = -0.48, B = 0.98. \tag{3}$$

Note that you are free to hardcode the probability distributions for importance sampling directly into your integrator.

Compare the number of function evaluations required in a) and b) to obtain different relative accuracies (say $\epsilon = 10^{-3}$, $\epsilon = 10^{-4}$, $\epsilon = 10^{-5}$ and $\epsilon = 10^{-6}$). If you have trouble achieving one or more of these accuracies, discuss why. Give estimates for how many more steps you expect your algorithm would take to reach this accuracy, based on its performance so far and the results you found in Q1.

Try adding the option to perform adaptive importance sampling, and then compare the performance to your results in b).