Tutorial VI

November 28

Excercise 1 [*2D wavefunction*] In quantum mechanics a wavefunction of a particle in 2 dimensions is given by a function $\psi(x,y) \in \mathbb{C}$. Due to finite memory, on a computer it is only possible to store such a function in an approximate way by using a large but finite set of variables One can e.g. consider a rectangular region $\mathcal{R}$ and discretize therein the variables x and y. Obviously, given a point $(x_0, y_0) \in \mathcal{R}$, we will have that $\psi(x_0, y_0) \in \mathbb{C}$. The easiest solution to store a complex number in a program is to store its real and its imaginary parts.
Consider, now, the region

$$\mathcal{R} = \{(x,y) \in \mathbb{R}^2 \mid 0 \leqslant x \leqslant L \wedge 0 \leqslant y \leqslant L\}.$$

discretize this region with a $N \times N$ lattice. The coordinates of each point will be given by:

$$\hat{x} = \left(n_x + \frac{1}{2}\right)\frac{L}{N} \quad \text{with} \quad n_x \in [0, N-1] \qquad \hat{y} = \left(n_y + \frac{1}{2}\right)\frac{L}{N} \quad \text{with} \quad n_y \in [0, N-1]\,,$$

where both n_x and n_y are integer numbers. Then, given a wavefunction ψ, one can calculate its value at each lattice point $(\hat{x}, \hat{y})$ and store the result in a 3 dimensional array where the first index corresponds to n_x, the second to n_y, and the third to real and imaginary parts. In a program one will have a `const int N` variable to create the array `double wavefunction[N][N][2]`.

(i) Given then the wavefunction

$$\psi(x,y) = \sin\Big(\frac{\pi}{L}\,x\Big)\sin\Big(\frac{\pi}{L}\,y\Big)e^{-\imath\omega t_0}\,,$$

where

$$\omega \equiv \frac{\hbar\pi^2}{m\,L^2}$$

and m is the mass of the particle, build the array `wavefunction` and fill it with a `C` function, i.e.

```
void generate_wave_function(int N, double L, double[][N][2] psi)
```

Fix $L = 0.2301$ fm, $m = 3.727$ GeV, $t_0 = 1/(6\pi)$ s and $N = 10$.

(ii) Write a `C` function that normalizes the wave function from (i). Again, write a dedicated `C` function, i.e.

```
void normalize_wave_function(int N, double L, double[][N][2] psi)
```

Note: In the continuum the procedure would be to calculate the square norm of ψ via

$$\mathcal{N}^2 = \iint_{\mathcal{R}} \psi^*(x,y)\psi(x,y)dx\,dy$$

and then to multiply the wavefunction by $1/\mathcal{N}$. When the region $\mathcal{R}$ has been discretized, the above integral (that corresponds to the volume below the function $\psi^*\psi$) can be approximated by a double sum, namely

$$\mathcal{N}^2 \approx \sum_{n_x=0}^{N-1}\sum_{n_y=0}^{N-1} \psi^*(\hat{x},\hat{y})\,\psi(\hat{x},\hat{y})\,\mathcal{A}$$

where $\mathcal{A}$ is the *volume* element corresponding to $dx\,dy$. The idea is to approximate the function $|\psi(x,y)|^2$ as constant in each of the smallest sub-region of $\mathcal{R}$, hence to sum the volumes of all the cuboids built in this way. In Figure 1 you can see a sketch of this idea.

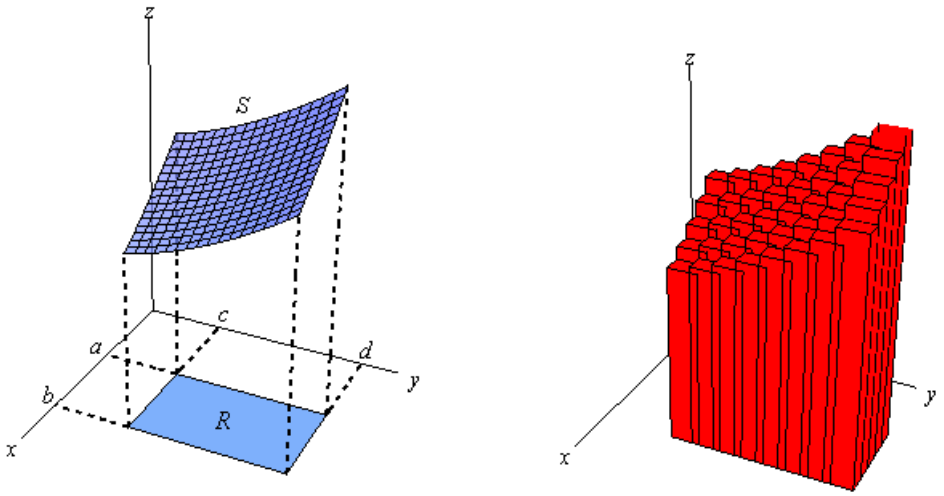

Figure 1: Approximation of a bidimensional integral.

(*iii*) Write a function that computes the probability to find a particle between x_0 and x_1 and y_0 and y_1. Use the following function prototype.

```
double probability(int N, double L, double[][N][2] psi,
                   double x0, double x1, double y0, double y1)
```

Again, in the continuous, this corresponds to calculate the integral

$$P = \iint_{[x_0,x_1]\times[y_0,y_1]} \psi^*(x,y)\psi(x,y)dx\,dy\ .$$

How can you approximate this integral? Deduce the easiest discrete approximation for P and implement it.

Excercise 2 [*Superindex*] In the previous problem you have been asked to use a 3 dimensional array, and this is neither so confortable nor often the most performing choice. Repeat the previous exercise using a superindex instead of 3 different indeces in the array `wavefunction`. Your functions will become

```
void generate_wave_function(int N, double L, double *psi)

void normalize_wave_function(int N, double L, double *psi)

double probability(int N, double L, double *psi,
                   double x0, double x1, double y0, double y1)
```

where now you have 1 dimensional array which has $2N^2$ elements. Write a suitable superindex function to transform the three indices $\hat{x}$, $\hat{y}$, $\Re/\Im$ to a single index[1].

Test your code by calling your functions in the same program with different array sizes N.

[1] See lecture slides set 5, page 30 in the pdf version.