Final project

Josip Dujmenović
Fizički odsjek, Prirodoslovno-matematički fakultet, Bijenička 32, Zagreb

1. Trial-and-error search

1.1. Problem - quantum states in a box potential

We are looking at the following transcendental equation:

$$\sqrt{V_0 - E_B}\; tan\sqrt{V_0 - E_B} = \sqrt{E_B} \quad (1)$$

Where V_0 has values of 10, 20 and 30 and E_B is the parameter that we are looking for. We can rewrite equation (1) as:

$$\sqrt{V_0 - E_B}\; tan\sqrt{V_0 - E_B} - \sqrt{E_B} = 0 \quad (2)$$

Now we are looking for values E_B for which equation (2) is zero, this is easily done using Bisection and Newton-Raphson algorithm. Because we have square roots in the equation, we must have $E_B \leq V_0$ and because we are dealing with *tan* function, which goes to $\pm$ infinity for the argument of $\pi/2 + n\pi$, we must have that $E_B \neq V_0 - (\frac{\pi}{2} + n\pi)^2$ for n = 0, 1, 2, ...

Before running the algorithms, equation (2) is ploted for multiple values of E_B using Jupyter notebook so that it is easier to know where one should look for the solution.

If we set equation (2) as equal to f, then we can find it's derivative with respect to E_B, which will be useful for Newton-Raphson algorithm:

$$\frac{df}{dE_B} = -\frac{tan(\sqrt{V_0 - E_B})}{2\sqrt{V_0 - E_B}} - \frac{1}{2\sqrt{E_B}} - \frac{1}{2cos^2(\sqrt{V_0 - E_B})} \quad (3)$$

Using the algorithms in C++ we get:

V_0	E_B N.-R.	E_B B.	Time N.-R. [μs]	Time B. [μs]
10	0.00401926	0.00401929	11	4
10	8.59279	8.59279	19	3
20	6.10847	6.10847	2	4
20	18.3605	18.3605	2	75
30	14.6661	14.6661	2	5
30	28.2411	28.2411	2	4

We can see that for a bigger V_0, bigger E_B is possible. If we want to find out more about the system, we could see if for even bigger V_0 more than two solutions could be found. Both algorithms get same results, time taken to execute each algorithm is different every time and in general, there is no noticeable difference. Newton-Raphson algorithm is more sophisticated algorithm, but both algorithms have their own problems, that is why it is useful to plot the function before looking for a solution.

2. Two masses on a string

To find the solutions to all nine variables, Newton-Raphson algorithm will be used. We are dealing with nine equations, so we can't really graph the problem. We are solving the following equation:

$$\Delta x = -F^{'-1} f \tag{4}$$

Since we are calculating inverse matrix, we have to make sure that guessed solution doesn't produce singular F matrix. Three point formula is used to carry out numerical differentiation. Some forums suggest solving equation (4) using LU decomposition, since calculating an inverse can be demanding. Using C++, two solutions can be found:

X	Solution 1	Solution 2
$sin\theta_1$	0.761003	-0.761003
$sin\theta_2$	0.264954	-0.264954
$sin\theta_3$	0.835706	-0.835706
$cos\theta_1$	0.648749	0.648749
$cos\theta_2$	0.964261	0.964261
$cos\theta_3$	0.549177	0.549177
T_1	17.1602	-17.1602
T_2	11.5453	-11.5453
T_3	20.2715	-20.2715

If we are looking at the solution of the angels, we can see that one solution is the negative of the other. Solution 1 is physical and a correct solution. Solution 2 is not physical because the tensions are negative, which they shouldn't be for a static situation. In both situations absolute value of tensions is similar to the weights of the spheres. If the majority of the initial angles are set to negative values, then the computer finds the non physical solution. Newton-Raphson algorithm takes a while to execute, so maybe some other algorithms should be considered, or further optimization should be carried out.