

EigenTuner 0.95 Documentation

Teemu Isojärvi

August 12, 2021

1 Introduction

This application has been made for the purpose of solving the quantum inverse scattering problem partially, producing the shape of some 1D potential well where the first N eigenenergies $E_1 \dots E_N$ have some values pre-defined by the user. The original inverse scattering method of Gel'fand, Levitan and Marchenko ([1, 2]) takes as an input the whole spectrum of eigenvalues E_n , required to not asymptotically increase faster than $E_n \propto n^2$ ([3]), and produces a function $V(x)$ resulting in that spectrum when used as the potential energy function of the well. The partial problem considering only a few of the lowest eigenenergies obviously has a larger amount of solutions than the full problem.

Recently, the inverse scattering method has been applied for the designing of semiconductor quantum wells having some given spectrum of bound state energies ([4]). The semiconductor wells can be modelled as 1D potential wells and have applications in the production of THz infrared radiation detectors ([5]) and possibly in the design of qubits for quantum computing. A problem with the original inverse scattering method is that it involves integral equations, not very familiar for people not specialized in mathematical physics, and difficult to visualize intuitively. Therefore, this application uses a more intuitive calculation scheme that modifies some initial 1D potential energy $V(x)$ repeatedly with perturbation terms constructed to move the eigenvalues to the direction of the correct target values. The easiest way is to start with a finite square well of some depth V_0 and length L , and to modify the shape of its bottom to shift the eigenvalues of the Hamiltonian to correct positions.

The method applied in this program is also described in a preprint article available in ResearchGate ([6]). A somewhat similar calculation method, called Inverse Perturbation Analysis, was proposed by Kosman and Hinze in 1970s for the fine-tuning of molecular potential energy curves on the basis of spectroscopic data ([7, 8]). Approaches classified as "inverse methods" are also used for the tuning of molecular structures to produce chemical compounds with useful properties, but that problem is somewhat different as the set of possible molecules is discrete rather than a continuum ([8]).

2 Theory

In all calculations, the energy and x-coordinate values have been nondimensionalized consistently with the Schrödinger equation

$$-\frac{1}{2} \frac{d^2 \psi(x)}{dx^2} + V(x) \psi(x) = E \psi(x), \quad (1)$$

where $\hbar = 1$ and $m = 1$. If wanted, the energy and distance units can be interpreted to be Hartree and Bohr, as in atomic units, even though Coulomb interaction with its quantities e and ϵ_0 doesn't explicitly appear anywhere.

We begin with a finite square well, centered at $x = 0$ and having potential energy

$$V(x) = \begin{cases} 0, & |x| \leq \frac{L}{2} \\ V_0, & |x| > \frac{L}{2} \end{cases}. \quad (2)$$

The energy eigenvalues and eigenfunctions of this system are, at the limit $V_0 \rightarrow \infty$,

$$E_n = \frac{\hbar^2 n^2}{8mL^2} = \frac{\pi^2 \hbar^2 n^2}{2mL^2}, \quad (3)$$

and

$$\psi_n(x) = \begin{cases} C_n \cos\left(\frac{n\pi x}{L}\right), & n \text{ odd} \\ C_n \sin\left(\frac{n\pi x}{L}\right), & n \text{ even} \end{cases}. \quad (4)$$

Now, to purposefully shift the spectral lines from their original locations, we'd need perturbation terms that have more effect on one eigenvalue E_n than on others. To do this, consider the equation for the first order energy correction in time-independent perturbation theory:

$$E'_n = \langle \psi_n | \hat{V}' | \psi_n \rangle = \int_{-\infty}^{\infty} |\psi_n(x)|^2 V'(x) dx. \quad (5)$$

Clearly, if the function $V'(x)$ overlaps much better with one probability density $|\psi_n(x)|^2$ than with others, it is good for this purpose. Two good choices of $V'(x)$ for shifting a particular energy eigenvalue are to make it proportional to $|\psi_n(x)|^2$ or $|\psi_n(x)|^4$, with a multiplier that gives it dimensions of energy. Putting something proportional to wave function probability densities in a potential energy function can seem a bit strange, but it produces good results for this numerical scheme.

To move the first N eigenvalues, E_i , of a 1D potential well to some target values $E_{i,target}$, a proper form of perturbation would be a linear combination

$$V'(x) = \sum_{k=1}^N C_k |\psi_k(x)|^2, \quad (6)$$

or

$$V'(x) = \sum_{k=1}^N C_k |\psi_k(x)|^4. \quad (7)$$

Using the first of these two, and assuming the effect of the perturbation on the eigenvalues is what the linear approximation of first order perturbation theory predicts, the correct values of the multipliers C_k could be solved from a linear system of equations. For instance, in the case $N = 4$, the linear system would be

$$\begin{bmatrix} I_{11} & I_{12} & I_{13} & I_{14} \\ I_{21} & I_{22} & I_{23} & I_{24} \\ I_{31} & I_{32} & I_{33} & I_{34} \\ I_{41} & I_{42} & I_{43} & I_{44} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix} = \begin{bmatrix} E_{1,target} - E_1 \\ E_{2,target} - E_2 \\ E_{3,target} - E_3 \\ E_{4,target} - E_4 \end{bmatrix}. \quad (8)$$

Here the quantities I_{mn} are defined, in the case of perturbation term (6), as

$$I_{mn} = \int_{-\infty}^{\infty} |\psi_m(x)|^2 |\psi_n(x)|^2 dx, \quad (9)$$

and the matrix is symmetric with $I_{mn} = I_{nm}$. The symmetry disappears if the added potential energy term of form (7) is used instead.

To not attempt to move the spectral lines too much at once and have too nonlinear a response, the multipliers C_i solved from (8) are all multiplied with some same number F before forming the sum (6). The number F should be between 0 and 1 in value. After calculating the new energies E_n and probability densities $|\psi_k(x)|^2$ for the perturbed system, another round of the same process is done by solving a similar linear system again to form a new perturbation term of type (6). This is repeated iteratively with the total number of iterations being large if F is small. So, this process is much like Newton's method for solving the zeros of a polynomial, where a linear approximation is used on every iteration to get closer to the actual location of the zero.

To find the first few eigenenergies and corresponding eigenfunctions for a 1D potential well, the program uses the *shooting method* and 4th order Runge-Kutta integration. The linear system of (8) is solved with a Gauss-Jordan elimination algorithm taken from the book "Numerical Recipes in C" ([9]). As the square matrix of the system is small in any practical application, there is no need to use a more optimized solution routine such as those in LAPACK.

The new version v0.95 also has the possibility to tune the square well by converting its bottom to a piecewise constant function similar to those used for approximating Riemann integrals. There the interval $[-L/2, L/2]$ is divided to equally long sub-intervals with endpoints $-L/2 = x_0 < x_1 < x_2 < \dots < x_{n-1} < x_n = L/2$ where $x_j - x_{j-1}$ doesn't depend on $j \in \mathbb{N}$. The perturbation terms are piecewise defined functions

$$f_m(x) = \begin{cases} 1, & \text{when } x_{m-1} \leq x \leq x_m \\ 0, & \text{when } x < x_{m-1} \text{ or } x > x_m \end{cases}, \quad (10)$$

and a linear combination of them, $V'(x) = D_1 f_1(x) + D_2 f_2(x) + \dots + D_n f_n(x)$, has a graph that looks like a histogram. In the calculation performed by this program, the number n of subdivisions is chosen so that each half of the initial square well contains $2k$ intervals $[x_{j-1}, x_j]$ when the interval at the middle of the square well is counted too and k is the number of eigenstates to be tuned. The potential energy function $V(x)$ is made symmetric, $V(x) = V(-x)$, and the multipliers D_m of the odd-numbered PTs $f_m(x)$ are tuned on odd-numbered iterations and even-numbered ones on even-numbered iterations. This doubling of the number of subintervals is done

to have more freedom in the shape of the final QPW. This form of a perturbation has not been intentionally chosen so that each PT function would overlap much better with one probability density $|\psi_j(x)|^2$ than with others, but it still produces good results. In the article [4], the potential energy profiles obtained with the inverse scattering method are converted to a similar step-like approximation because that is easier to accomplish when engineering semiconductor quantum wells in practice.

3 Using the application

The program is written in C++ and should be compiled in Linux as "g++ -O3 -ffast-math eigentuner.cpp". This source code file is for using the perturbation (6) and the other one, "eigentuner_sqr_density.cpp" for using (7). When running the program, it asks for the potential well length L , depth V_0 , a lower bound estimate for the lowest eigenenergy E_{lb} , the multiplier F , the number of iterations n_{it} and the target energy values $E_{1,target} \dots E_{N,target}$. It will also ask whether to produce a potential energy output file on every iteration or only the last one. The output text file has x values in one column and $V(x)$ values in another. After last iteration, the program also produces an output file containing x -coordinate values and $|\psi_m(x)|^2$ values in different columns.

The application in its current form does not ask for the RK4 numerical integration step length and the number of discrete grid points. Those have to be changed by modifying the source code if necessary.

The computational results of several example problems were checked for consistency by taking the final potential energy function produced as an output and doing a shooting method calculation by hand in Wolfram Mathematica 11.2.0 to see whether the energy eigenvalues will be the same. There was not much discrepancy between the energy values from the two sources.

4 Notable issues in the current version

Sometimes the shooting method for finding energy eigenvalues misses the correct energy value and returns the next even or odd eigenenergy instead. This would usually also make the next iteration step move the eigenvalues in a completely unpurposeful way. The current version is able to detect and attempt to correct this by retrying that particular iteration with a different value of F or a different set of histogram-like perturbation terms. Having both attempts fail on some iteration is very unlikely unless the calculation consists of a very large number (>80) of iterations.

None of the target eigenvalues $E_{n,target}$ given to the program should be negative numbers, and during the calculation none of the energies should temporarily drop below -10.0, as the shooting method algorithm does not scan for eigenvalues lower than that value. It's also not recommended to set all target energies lower than the values in the original unperturbed system. It's better if about half of the energies have to moved upwards and the other half about the same amount downwards during the calculation. This should not be a problem, because the energy spacings between eigenvalues have much more practical significance than the absolute energy values and their median.

One feature to be later added to the program is to produce a new directory for the output files of every run. The name of the output directory would contain the date and time when the calculation began. There may also be an option to form the quantum well by starting with a harmonic oscillator potential energy function $V(x) \propto x^2$ (or some $V(x)$ function imported from an external data file) instead of a finite square well, and using the same iterative method.

On lines 193, 204 and 205 of "eigentuner.cpp" and the equivalent place in other source files, it has been chosen how large values of $|x|$ beyond $|x| = L/2$ are considered when recording the new probability densities $|\psi_n(x)|^2$ after each iteration. This limit should be set higher if the well depth is really small and the exponential "tails" of the wave function extend far beyond the original square well boundary $|x| = L/2$.

If moving the spectral lines of some particular potential well to correct positions causes trouble, the easiest solution is to try different combinations of values of F and n_{it} , and possibly different kind of perturbation terms as well. Also, anyone using this code for actual research purposes would most likely tune the final potential energy profile $V(x)$ by hand with one or two more iterations (as shown in Ref. [6]) to get a more precise result and to set the shooting method energy step and x -step sizes to be appropriate for the particular system and its length scales.

5 Example problems

This section describes three different example problems where the calculation converged properly.

5.1 Example with PT (6)

In this example, the parameters were $L = 1.0$, $V_0 = 200.0$, $E_{lb} = 2.0$, $F = 0.5$, $n_{it} = 10$, $E_{1,target} = 6.0$, $E_{2,target} = 9.0$ and $E_{3,target} = 25.0$. The resulting energy eigenvalues after last iteration were $E_1 = 5.988$, $E_2 = 8.996$ and $E_3 = 25.016$. The final potential energy function and wave function probability densities P_1 to P_3 are shown as a graph in Figure 1. All graphs in this document have been plotted in Ubuntu Linux with Grace-5.1.25. This system could be used as a model of a "two-level quantum system" because the first excitation energy is much smaller than the second one, and excitation to the third energy eigenstate isn't likely at low temperatures.

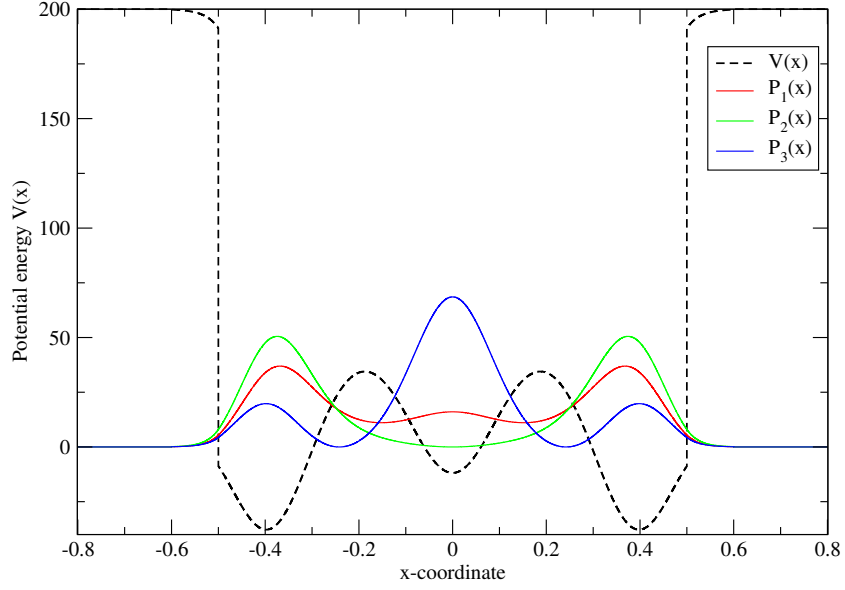


Fig. 1: The resulting function $V(x)$ and probability densities $P_1(x)$ to $P_3(x)$ in the first example calculation, where the three lowest eigenenergies are tuned to be close to values 6.0, 9.0 and 25.0.

5.2 Example 1 with PT (7)

Here the parameters were $L = 2.0$, $V_0 = 50.0$, $E_{lb} = 0.3$, $F = 0.5$, $n_{it} = 10$, $E_{1,target} = 3.0$, $E_{2,target} = 4.0$, $E_{3,target} = 8.0$ and $E_{4,target} = 10.0$. The energies after 10 iterations were $E_1 = 2.9936$, $E_2 = 3.9962$, $E_3 = 7.9946$ and $E_4 = 10.0052$. The converged function $V(x)$ and functions $P_1(x)$ to $P_4(x)$ are in Figure 2.

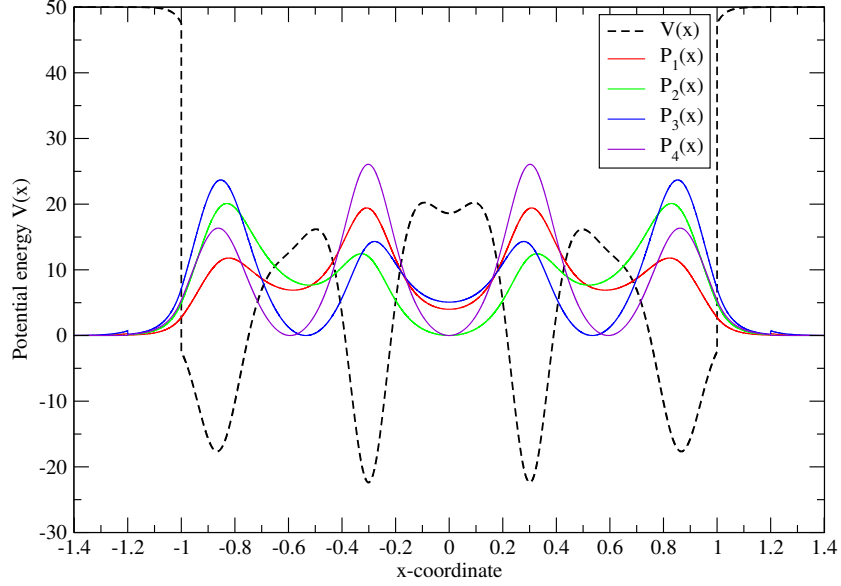


Fig. 2: The resulting function $V(x)$ and probability densities $P_1(x)$ to $P_4(x)$ in the example calculation of Subsection 5.2, where first four energy eigenvalues are tuned to be near values 3.0, 4.0, 8.0 and 10.0. The function $V(x)$ is clearly less similar to simple sine and cosine functions than in the problem where perturbation (6) was used.

5.3 Example 2 with PT (7)

The parameter set in this example was $L = 2.0$, $V_0 = 50.0$, $E_{lb} = 0.3$, $F = 0.5$, $n_{it} = 10$, $E_{1,target} = 4.0$, $E_{2,target} = 10.0$, $E_{3,target} = 11.0$ and $E_{4,target} = 20.0$. The energies after last iteration were $E_1 = 3.995$, $E_2 = 9.989$, $E_3 = 10.9952$ and $E_4 = 19.9988$. The results are shown in Figure 3. The Figure 4, in turn, shows how the function $V(x)$ converges to the final result during the 10 iterations. The peculiarity of this example system is that the second excitation energy is much lower than the first or the third.

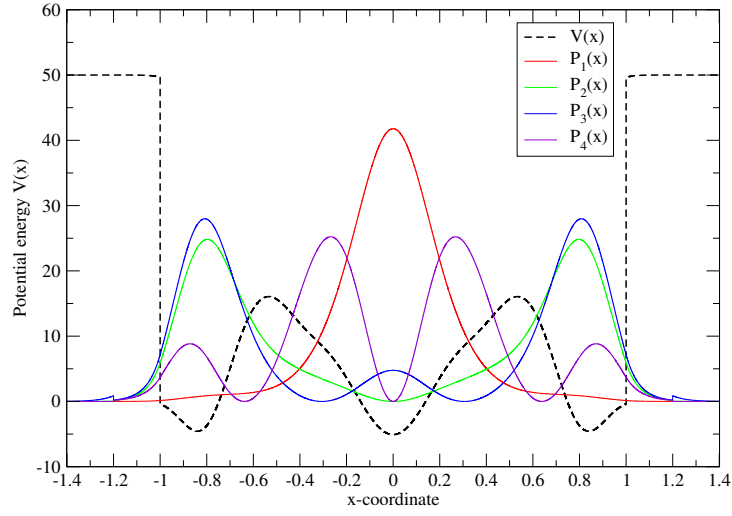


Fig. 3: The resulting function $V(x)$ and probability densities $P_1(x)$ to $P_4(x)$ in the example calculation of Subsection 5.3.

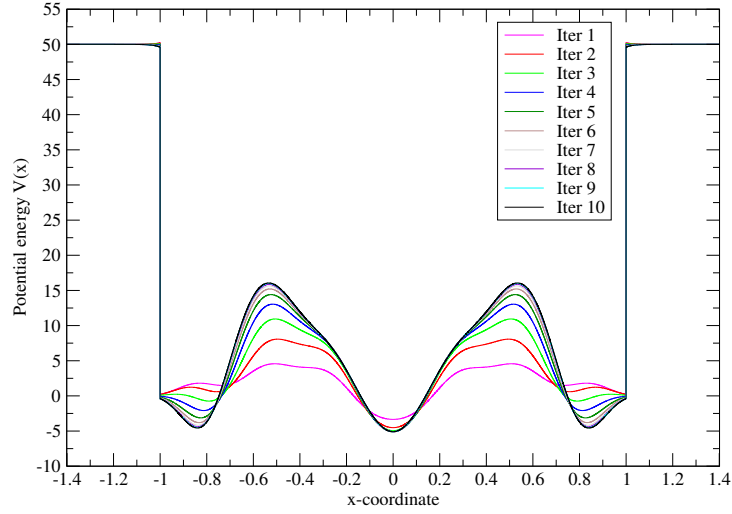


Fig. 4: The change in the potential energy $V(x)$ during the 10 iterations of third example calculation.

5.4 Example 3 with PT (7)

In this example, as many as 6 energies were moved to values $E_{1,target} = 2.0$, $E_{2,target} = 4.0$, $E_{3,target} = 6.0$, $E_{4,target} = 20.0$, $E_{5,target} = 25.0$ and $E_{6,target} = 30.0$. This can already be a bit tricky to do, because it's likely that on a single run the shooting method misses some eigenvalue and wrong results are obtained. Therefore different values of parameters E_{lb} , F and n_{it} have to be tested until the calculation goes right.

The parameters used in this example, in addition to the target energies, were $L = 2.0$, $V_0 = 50.0$, $E_{lb} = 0.8$, $F = 0.65$ and $n_{it} = 6$. The actual energy values after the last iteration were $E_1 = 1.9856$, $E_2 = 3.9856$, $E_3 = 6.0016$, $E_4 = 19.9888$, $E_5 = 24.9904$ and $E_6 = 30.0096$. When the data set for the function $V(x)$ after last iteration was imported to Mathematica and the shooting method done by hand as a consistency check, the E_n values from the two sources differed by something like 0.3% max. Even this small difference can probably be easily removed by improving the shooting method algorithm in later versions. The Figure 5 shows the final potential energy function $V(x)$ from the calculation, as well as the target energies as horizontal lines.

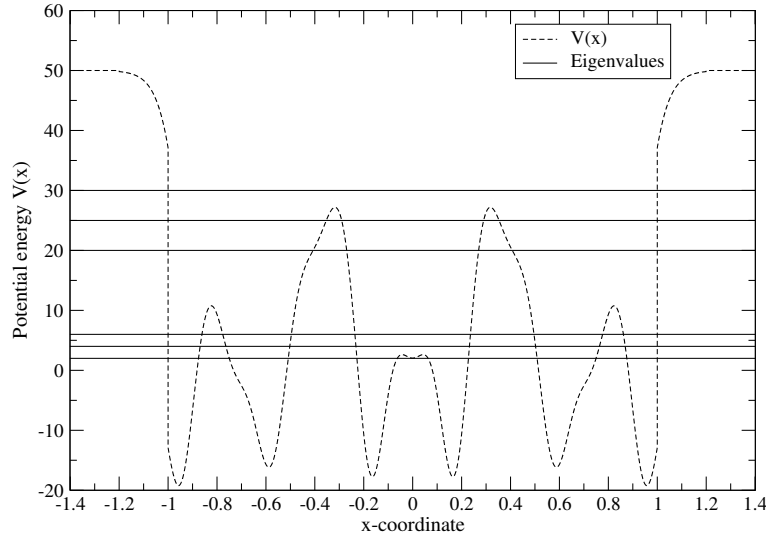


Fig. 5: The potential energy function $V(x)$ after the 6th iteration of a run where six eigenvalues were tuned to pre-defined values. The target energy "spectral lines" are shown as solid horizontal lines in the graph. The lowest 6 energy eigenvalues calculated for this system differ less than 1% from the target values of 2.0, 4.0, 6.0, 20.0, 25.0 and 30.0.

5.5 Example 1 with PT (10)

This time the goal was to modify a finite square well with $L = 1$ and $V_0 = 30$ to a one where the first three energies are $E_1 = 6.0$, $E_2 = 18.0$ and $E_3 = 21.0$. The perturbation used was the

step-like function constructed as a linear combination of the functions f_m in Eq. (10). Other parameters of the calculation were $E_{lb} = 1.0$, $F = 0.1$ and $n_{it} = 60$. The original eigenvalues of the unperturbed well were $E_1 = 3.096$, $E_2 = 12.058$ and $E_3 = 25.240$.

A video file showing the change of $V(x)$ and eigenenergies during the 60 iterations in the file "step_like_animation.avi". The current values of the E_m are drawn as horizontal black lines in each frame and the target values $E_{m,target}$ are drawn as horizontal red lines. The final energies after the tuning were $E_1 = 5.994$, $E_2 = 17.974$ and $E_3 = 21.008$.

Moving the eigenvalues to correct target positions with this step-like PT seems to require a quite large perturbation, producing a potential barrier higher than V_0 in the middle of the well. This possibly reflects the fact how the tuning of eigenvalues is more difficult when the PT has not been chosen to overlap more with one probability density than the others, as in (6) and (7).

References

- [1] I. Gel'fand and B. Levitan, *On the determination of a differential equation from its spectral function*, Izv. Akad. Nauk SSSR, Ser. Math. **15**, 309-60 (1951).
- [2] D. Abramov, *Quantum mechanical inverse problem on a finite interval as an initial-value problem*, Inverse Problems **7**, 493-497 (1991).
- [3] Dániel Schumayer, Brandon P. van Zyl, and David A. W. Hutchinson, *Quantum mechanical potentials related to the prime numbers and Riemann zeros*, Phys. Rev. E **78**, 056215 (2018).
- [4] Igor Tralle and Klaudiusz Majchrowski, *"Smart Design" of Quantum Wells and Double-Quantum Wells Structures*, World Journal of Condensed Matter Physics **4**, 24-32 (2014).
- [5] Igor Tralle, Grzegorz Haldas, Klaudiusz Majchrowski, and Andrzej Kolek, *Design of Quantum Double-Barrier Tunnel Structures for THz Detection*, IEEE Journal of Selected Topics in Quantum Electronics **20** (6), November/December 2014.
- [6] Teemu Isojärvi, *On the Designing of Quantum Potential Wells with Given Spectral Properties*, Article preprint, <http://dx.doi.org/10.13140/RG.2.2.28663.14240>.
- [7] Warren M. Kosman and Jürgen Hinze, *Inverse Perturbation Analysis: Improving the Accuracy of Potential Energy Curves*, Journal of Molecular Spectroscopy **56**, 93-103 (1975).
- [8] Thomas Weymuth and Markus Reiher, *Inverse Quantum Chemistry: Concepts and Strategies for Rational Compound Design*, International Journal of Quantum Chemistry 2014, 114, 823-837.
- [9] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing 3rd Ed.* (Cambridge University Press, 2007).