Application for computing excited state wave functions

Teemu Isojärvi (teemu.isojarvi@outlook.com)

January 31, 2026

1 Introduction

This is an improved version of my earlier GitHub code 'meanstate' ([1]), which allows for calculation of excited state wave functions of quantum systems without having to first solve all the lower energy eigenstates of $\hat{H}$ starting from the ground state. The method is designed to converge towards an eigenstate of $\hat{H}$ with energy closest to some given target value E_t. This can potentially save computation time, if only an excited state wave function and energy eigenvalue is needed without having to know the lower part of the spectrum. However, in an excited state calculation the spatial resolution needs to be higher than for the ground state, as the wave function varies more rapidly as a function of position (in 1D this means that there are more nodes in higher energy eigenstates). Instead of the R language, which the earlier program was made with, this one is made with C++ using the Google Canvas AI application as help when writing the code.

The idea for this numerical method was inspired by seeing a similar one for finite dimensional matrix eigevalue problems, allowing the finding of an eigenvector when given an estimate for the associated eigenvalue. ([2])

The application can solve eigenstates of the Hamiltonian for 1D or 2D single-particle systems with time-independent potential energy function $V(x)$ or $V(x, y)$.

2 Theory

To find the ground state $|0\rangle$ of a quantum system with Hamiltonian $\hat{H}$, the usual method is to start with a trial state vector $|\psi(0)\rangle$, with unknown spectral decomposition

$$|\psi(0)\rangle = C_1|1\rangle + C_2|2\rangle + C_3|3\rangle + \dots, \tag{1}$$

where $|k\rangle$ are the eigenstates of the Hamiltonian and C_k are complex-valued multipliers. The time evolution of this state is

$$|\psi(t)\rangle = C_1 e^{-iE_1 t}|1\rangle + e^{-iE_2 t}C_2|2\rangle + e^{-iE_3 t}C_3|3\rangle + \dots, \tag{2}$$

when $\hat{H}$ is not a function of time and constants are set so that $\hbar = 1$. Changind to an imaginary time coordinate s, defined as $t = -is$, the state vector $|\psi(s)\rangle$ converges towards something with only a $|1\rangle$ component in the limit $s \to \infty$, as far as the trial vector $|\psi(0)\rangle$ has been chosen to be not orthogonal with $|1\rangle$.

To calculate the first excited state $\left|2\right\rangle$, this is done again with an initial state $\left|\psi(0)\right\rangle$ such that $\left\langle 2\right|\psi(0)\rangle \neq 0$ and simulating time propagation towards $s \rightarrow \infty$ while removing the $\left|1\right\rangle$ component from $\left|\psi(s)\right\rangle$ with a projection operator on every time step. This means that for an excited state calculation, approximations for all the lower energy eigenfunctions of $\hat{H}$ have to be computed first so that they can be eliminated from the state vector during imaginary time propagation.

The application presented here is based on forming an operator

$$\hat{A}(T) = \frac{1}{T}\int_0^T \hat{U}(\tau)d\tau = \frac{1}{T}\int_0^T \exp\left(-i\hat{H}\tau\right)d\tau, \tag{3}$$

which is the average of evolution operators $\hat{U}(t)$ for all time intervals t with $0 \leq t \leq T$.

If this operator is applied on a trial state vector $\left|\psi\right\rangle = C_1\left|1\right\rangle + C_2\left|2\right\rangle + \dots$, the result is

$$\hat{A}\left|\psi\right\rangle = \frac{1}{T}\left[\frac{iC_1}{E_1}\left(e^{-iE_1t}-1\right)\left|1\right\rangle + \frac{iC_2}{E_2}\left(e^{-iE_2t}-1\right)\left|2\right\rangle + \dots\right], \tag{4}$$

which is the average state vector on the evolution time interval of length T.

Operating on the vector $\left|\psi\right\rangle$ with $\hat{A}(T)$ n times, multipliers of E_k^{-n} accumulate in front of each term in the spectral decomposition. This means that unless the initial state vector $\left|\psi\right\rangle$ or the time interval T has been chosen really badly, the result should converge towards something that contains only the eigenstate of the Hamiltonian where energy is closest to value 0.

Forming the average evolution operator from a shifted Hamiltonian,

$$\hat{A}(T) = \frac{1}{T}\int_0^T \exp\left[-i(\hat{H}-E_t)\tau\right)d\tau, \tag{5}$$

where E_t is a trial energy value, the result should converge towards the eigenstate of $\hat{H}$ with energy closest to value E_t.

The C++ program here uses the Crank-Nicolson time stepping ([3]), and Alternating Direction Implicit method ([4]) for 2D, to calculate the evolution of a given initial wave function and to find the average wave function on the time interval of length T. This is done for many iterations to try to approach the energy eigenvalue closest to the target energy E_t.

In the source code, the iteration lenths T, time step δt in the implicit method time propagation, spatial step size δx (and δy in 2D), as well as the number of iterations N_{it} are defined as global constants. Also the lenghts of the domain in x and y directions, L_x and L_y, need be defined. In 2D calculations, adaptive grid refinement is usually needed to make the computation faster, and the resolution is multiplied with some factor M several times during the simulations. This means that δx, δy and δt are divided by M and the iteration numbers when doing that are defined in an array with name `'Refine_points'`. 2D grid refinement is done with bicubic interpolation. ([5])

In the 2D version, there is also a Python script named `'plot.py'` for plotting the square modulus of resulting wave function, $|\psi(x,y)|^2$, and potential energy $V(x,y)$, which are produced as output files. The names of the data files, number of spatial grid points and domain lengths have to be correctly defined in the 'plot.py' source file if they are changed from previous values. The xy coordinate system is defined so that $-L_x/2 \leq x \leq L_x/2$ and $-L_y/2 \leq y \leq L_y/2$.

3 Examples

The examples demonstrate how to use the program to find a high excited state with many nodes for an anharmonic potential well $V(x) \propto x^4$, for a 2D rectangular potential well and for a 2D supercircle shaped potantial well with finite potential step at the rectangular or supercircular boundary. Units with $\hbar = m = 1$, where m is the particle mass, are used in all examples.

3.1 Excited state of quartic 1D potential well

In this 1D example, the system is a 1D anharmonic potential well with units $\hbar = m = 1$. The potential energy is

$$V(x) = 0.25x^4. \tag{6}$$

The simulation had 400 iterations with time interval of about $T = 0.25$ on each iteration. The target energy was $E_t = 12.5$, which is close to the value $E_8 = 12.738$ calculated with the shooting method for 7th excited state of this system. An optimal value of T for convergence should be close to $T = \frac{\pi}{E}$, where E is the energy of the eigenstate that is sought as a result. The spatial domain had length $L = 20$ and there were 1200 grid points. The time step was $\delta t = 0.01$. The resolution (in either space or time stepping) was not refined at any point in the calculation, because in 1D the initial values of δt and δx can be made small enough for good accuracy without making the process too slow. The initial wave function was a multiple of a sine function and a Gaussian.

The energy seemed to converge to about $E = 12.733$, which is close to the shooting method result. Squared modulus wave function $|\psi(x)|^2$ is shown in Figure 1, and it clearly has 7 nodes as supposed for the 7th excited state of a 1D system.

The source code prints both the energy convergence and the squared wave function values to standard output, and the latter needs to be transferred to a separate text file to be plotted with Grace or other similar application.

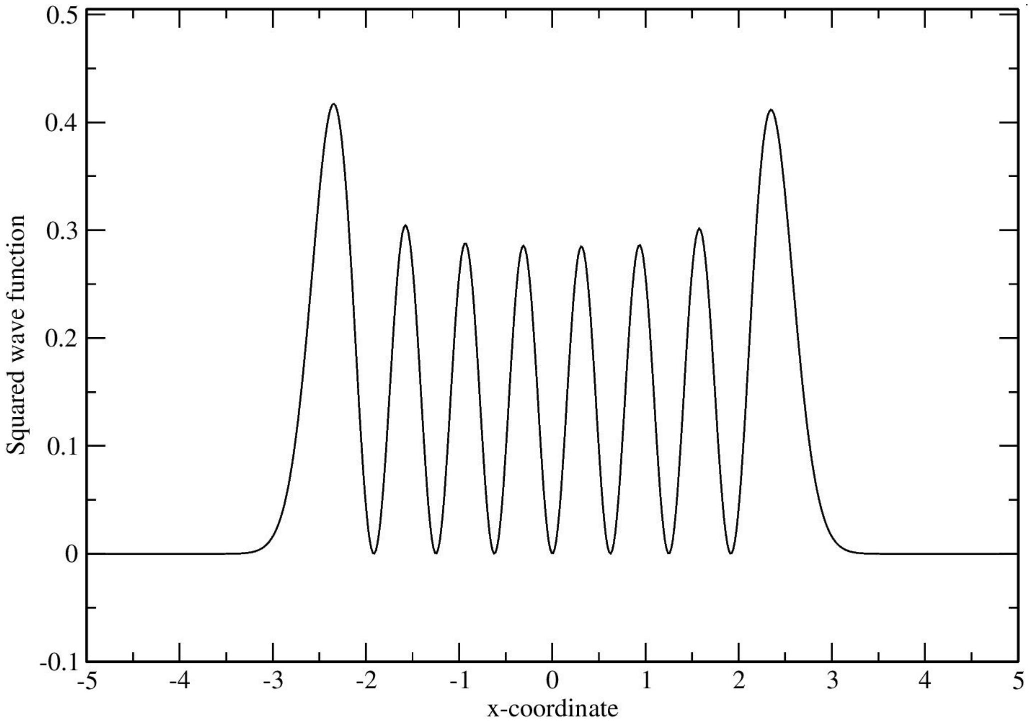

Fig. 1: The absolute square wave function $|\psi(x)|^2$ computed for a 1D potential well with $V(x) = \frac{1}{4}x^4$.

3.2 Rectangular 2D finite potential well

In this example, the $V(x)$ is 0 inside a rectangle of length 1.3 in the x-direction and length 1 in the y-direction. The reason for choosing different side lengths was to avoid complications from having subspaces of degenerate eigenstates with same energy eigenvalue. The potential step at the boundary of the rectangle is $V_0 = 2 \times 10^5$ to have a result close to that of a hard-walled 2D box ($V_0 \to \infty$). The trial energy was $E_t = 46.0$ to attempt to converge towards the eigenstate of $\hat{H}$ equivalent to that of a hard-walled box with quantum numbers $n_x = 2$ and $n_y = 3$. In the $V_0 \to \infty$ case the exact energy eigenvalue of that state is about 46.05. The time length of one iteration was set to $T \approx 0.068$ and the number of iterations was $N_{it} = 1000$. Initially, the grid contains 100 points in both x and y directions and the time step is $\delta t = 0.004$. Spatial resolution is doubled at iterations 800, 940, 980 and 992 (with cubic interpolation) and the time step divided by 1.5 at same iterations. The initial wave function is a sum of two Gaussian functions constructed to not be likely to be completely orthogonal with any eigenstate of $\hat{H}$.

The energy convergence is shown in the output file `'rectangle_stdout.txt'` and the data files for plotting the wave function and potential energy with the Python script are named `'psi_2d_data.csv'` and `'potential_2d_data.csv'`.

In the result, the energy eigenvalue seems to converge to something near $E_{23} = 45.8$.

The $|\psi(x,y)|^2$ and $V(x,y)$ plots are shown in Figure 2. There it is apparent the the solution has two nodes in the x-direction and one node in the y-direction, corresponding to the state with quantum numbers $n_x = 3$ and $n_y = 2$.

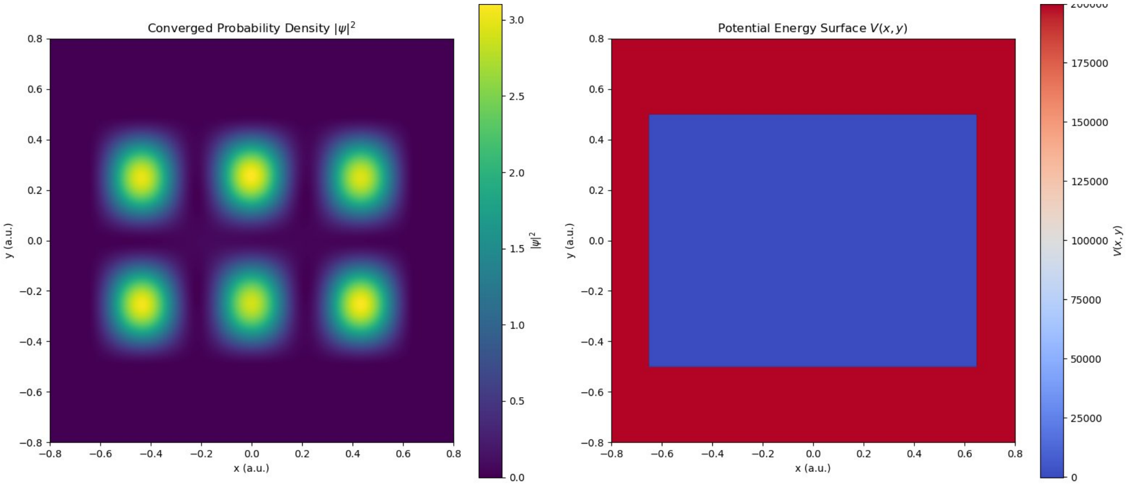

Fig. 2: 2D plots produced by the Python script from the results for an excited state of a rectangular potential well.

3.3 Supercircular 2D finite potential well

A supercircle is a 2D curve defined with the equation

$$|x|^q + |y|^q = r^q, \tag{7}$$

where r is something equivalent to the radius in the circular $q = 2$ case. The absolute values are needed because q is not necessarily an integer divisible by 2. When $2 < q < \infty$, this curve has a shape that is something between a circle $q = 2$ and a square $q \to \infty$.

3.3.1 Supercircle with q=3.0

In this example calculation, the potential energy $V(x, y)$ has value zero inside a supercircle with $q = 3$ and value 2×10^4 outside it. This is similar to the system in my publication [6] The radius parameter was set to $r = 1.063999$ to have a supercircle of area $A = 4.0$. When the exponent q is known, the value of r for a supercircle of area 4 can be calculated with

```
double supercircleRadius(double q)
{
    double gamma1 = std::tgamma(1.0 + 1.0 / q);
    double gamma2 = std::tgamma(1.0 + 2.0 / q);

    return std::sqrt(gamma2 / (gamma1 * gamma1));
}
```

To find an approximation to the eigenstate of $\hat{H}$ corresponding to the second excited state of the circular $q = 2$ equivalent, the target energy was set to $E_t = 11.0$. The time interval T in each iteration has value 0.286. Tnitially there are 120 grid points in both x and y directions and the time step has lenght 0.02. The resolution is adaptively refined three times during the calculation, doubling the amount of grid points in x and y directions and dividing the time step by 1.5. The total number of iterations is 600. Initial wave function was again set to be a sum of two Gaussians.

The energy eigenvalue converges to something around $E = 10.98$, as seen in the standard output. The $|\psi(x, y)|^2$ and $V(x, y)$ plotted with the Python script are shown in Figure 3.

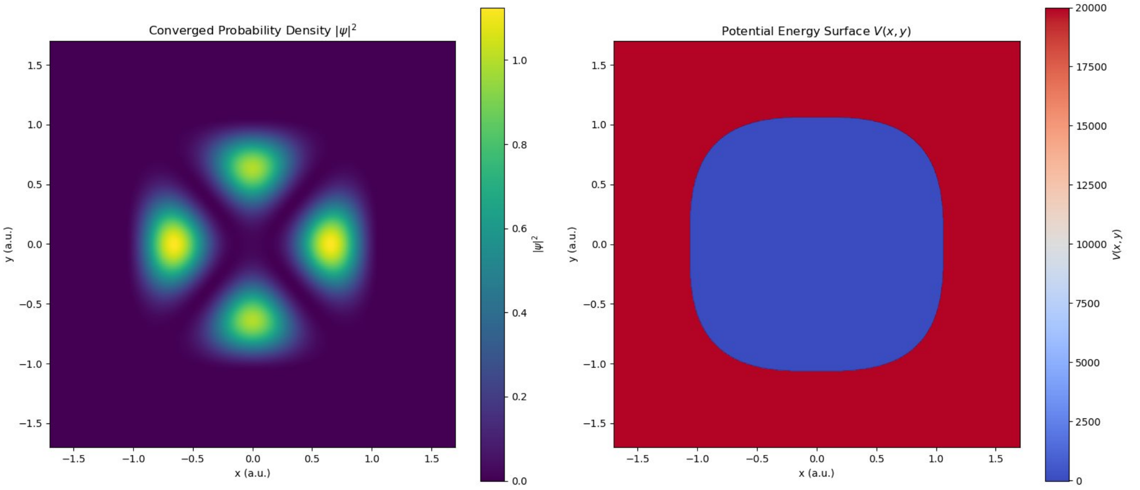

Fig. 3: A 2D plot of the approximate second excited state of particle in a supercircular potential well with shape parameter $q = 3$, area $A = 4$ and potential step of $V_0 = 20000$ at the boundary.

3.3.2 Supercircle with q=0.8

This same type of calculation was also made for a supercircle with $q = 0.8$ and $r = 1.609$ to have area 4.0. The potential step at the boundary was again $V_0 = 20000$. The iteration length was about $T = 0.286$ and the initial spatial and temporal step sizes δx, δy and δt were the same as in the $q = 3.0$ example, and so were the three adaptive refinements during the calculation.

The target energy was set to values $E_t = 1.5$, $E_t = 6.0$, $E_t = 10.0$ and $E_t = 12.5$ in four different simulations, while keeping all other parameters fixed. The estimated energy eigenvalues at the end of these four cases were, to three significant figures, about $E = 2.60$, $E = 6.46$, $E = 10.1$ and $E = 12.7$. The included source code file `supercircle_A4.0_q0.8_V20000.cpp` is one in which the target energy has been set to $E_t = 12.5$.

Plots of the probability density functions are shown in Figure 4.

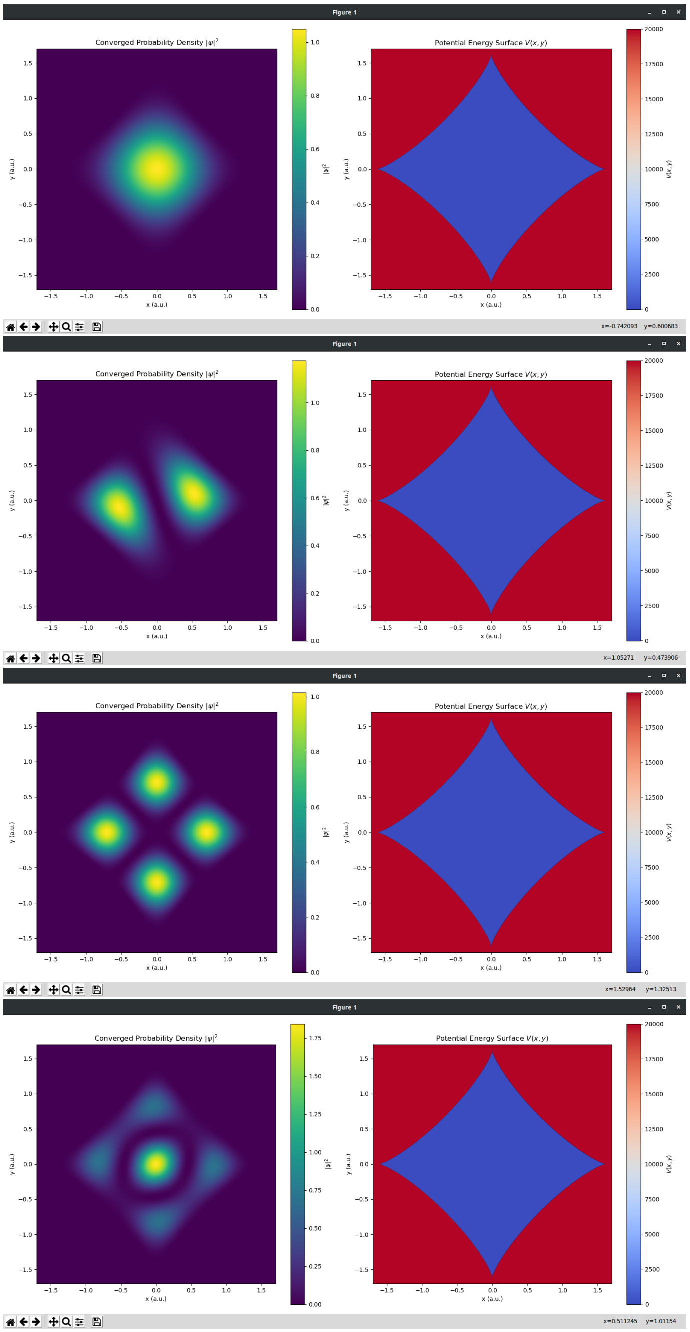
Figure 1
Converged Probability Density $|\psi|^2$
Potential Energy Surface $V(x, y)$
x (a.u.)
y (a.u.)
$|\psi|^2$
$V(x, y)$
x=-0.742093 y=0.600683
Figure 1
Converged Probability Density $|\psi|^2$
Potential Energy Surface $V(x, y)$
x=1.05271 y=0.473906
Figure 1
Converged Probability Density $|\psi|^2$
Potential Energy Surface $V(x, y)$
x=1.52964 y=1.32513
Figure 1
Converged Probability Density $|\psi|^2$
Potential Energy Surface $V(x, y)$
x=0.511245 y=1.01154

Fig. 4: 2D plots of the probability densities from attempting to find energy eigenstates near energy values $E = 1.5$, $E = 6.0$, $E = 10.0$ and $E = 12.5$ for a particle confined in a supercircle with $q = 0.8$ and area $A = 4.0$.

The top image in Figure 4, from the $E_t = 1.5$ calculation, appears to be the ground state of the system, and the third ($E_t = 10.0$) and fourth ($E_t = 12.5$) images seem to be excited states. The second image is asymmetric enough to look like it is probably a superposition of two eigenstates of $\hat{H}$ with almost same energy. As can be seen from the potential energy plot, the shape of a $q = 0.8$ supercircle is a slightly deformed square with concave sides (the case with $q = 1.0$ is exactly a square that has been rotated 45 degrees from parallel with coordinate axes). While a square shaped 2D box has two degenerate energy levels with quantum numbers $(n_x, n_y) = (1, 2)$ and $(n_x, n_y) = (2, 1)$, the slightly deformed square here is likely to have equivalent eigenfunctions with only a small splitting between the energy eigenvalues. Alternatively, there actually are two states with same energy, and that in Figure 4 is just one linear combination of them. This example demonstrates that the computational method I describe here is more difficult to apply unless there is a large enough separation between neighboring energy levels.

3.4 Particle in equilateral triangle

A potential well having the shape of an equilateral triangle with area 4, and an energy step of $V_0 = 20000$ when crossing the boundary, can be defined by putting these lines to the source code after the definition of `const double E_trial`:

```
struct Point {
    double x, y;
};

const double targetArea = 4.0;
const double s = std::sqrt(16.0 / std::sqrt(3.0));
const double R = s / std::sqrt(3.0);

Point v1 = {0, R-0.5};
Point v2 = {R * std::cos(210.0 * M_PI / 180.0), R * std::sin(210.0 * M_PI / 180.0)-0.5};
Point v3 = {R * std::cos(330.0 * M_PI / 180.0), R * std::sin(330.0 * M_PI / 180.0)-0.5};

double calculateArea(Point p1, Point p2, Point p3) {
    return std::abs((p1.x * (p2.y - p3.y) + p2.x * (p3.y - p1.y) + p3.x * (p1.y - p2.y)) / 2.0);
}

bool pointTest(Point v1, Point v2, Point v3, Point p) {
    double totalArea = calculateArea(v1, v2, v3);

    double a1 = calculateArea(p, v1, v2);
    double a2 = calculateArea(p, v2, v3);
    double a3 = calculateArea(p, v3, v1);

    const double EPSILON = 1e-9;
    return std::abs(totalArea - (a1 + a2 + a3)) < EPSILON;
}

double potential(double x, double y) {
    if(pointTest(v1,v2,v3,{x,y})) return -E_trial;
    else return 20000.0;
}
```

I did not upload an example source code about this, but graphs of the probablity densities for this system in the $V_0 \to \infty$ limit can be found from [7]. Most of the energy eigenvalues are degenerate in a potential well of this shape, which makes this problem more complicated when using the computational method described here.

3.5 2D quartic well

This example is to demonstrate, in 2D, how a system without discontinuities in $V(x,y)$ makes the eigenstate computation possible with a smaller number of iterations.

The potential energy function was defined

$$V(x) = x^4 + y^4. \tag{8}$$

The source code is in the file `quartic_2d.cpp`. Parameters were $E_t = 6.0$, $T = 0.5235$, $\delta t = 0.02$, domain side length 6.0 and initially 120 grid points in both x and y directions. The number of iterations was 100 and adaptive refinement was done at iterations 75, 85 and 93. The initial condition for $\psi(x,y)$ was again a sum of two Gaussian functions, but not exactly the same as in other examples.

The energy eigenvalue seemed to converge to about $E = 5.365$. The probability density and $V(x,y)$ are shown in Figure 5. When plotting this, the line

```
contour2 = ax2.pcolormesh(x_coords, y_coords, potential_energy, shading='auto',
cmap='coolwarm')
```

was changed to

```
contour2 = ax2.pcolormesh(x_coords, y_coords, potential_energy, shading='auto',
cmap='coolwarm',vmin=0,vmax=50)
```

to visualize the function $V(x,y)$ better. The eigenfunctions and energies for this system are not as simply to calculate in exact form as for a particle in 2D rectangular box or for the 2D harmonic oscillator, but the $|\psi(x,y)|^2$ in Fig. 5 looks (by symmetry) like equivalent to the state with both x and y direction quantum numbers one step above ground state in those simpler systems.

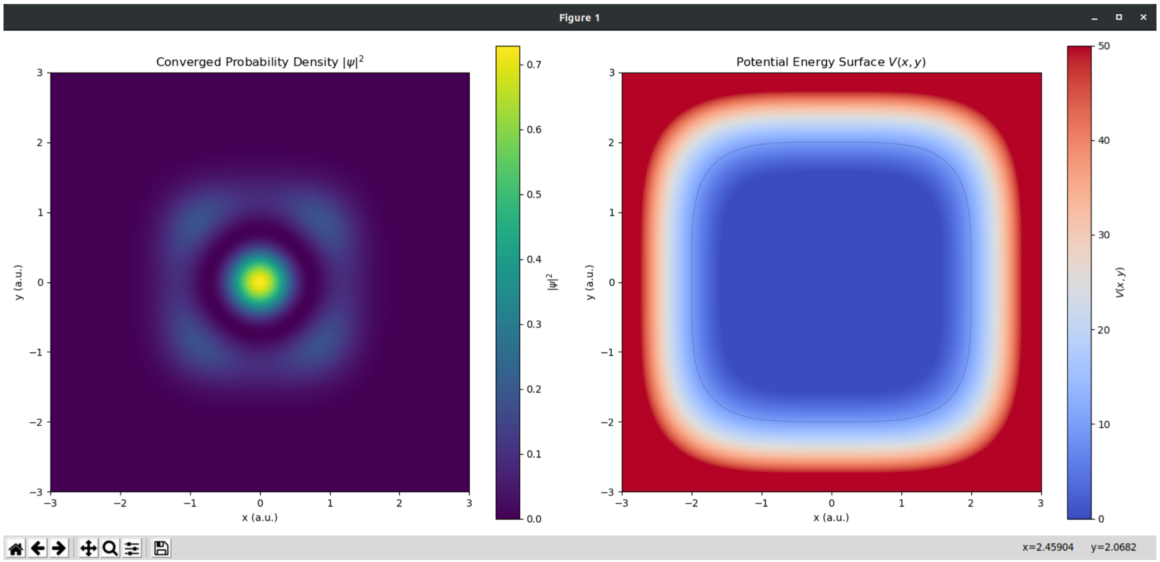

Fig. 5: 2D plots of the probability density and potential energy when an eigenstate of $\hat{H}$ with energy near $E = 6.0$ was looked for in a system with $V(x,y) = x^4 + y^4$.

4 Additional comments

The usual way to find a numerical approximation to an isolated eigenstate of $\hat{H}$ several steps above ground state seems to be ([8]) the calculation of imaginary time evolution with the square of shifted Hamiltonian $(\hat{H} - E_t)^2$. This should converge towards a state nearest the target energy E_t, but a squared Hamiltonian contains 4th order spatial derivatives which seems to make it impossible to reduce the numerical calculation of time evolution to solution of only tridiagonal linear systems even with operator splitting. Fourth order differential equations more commonly appear in classical continuum mechanics problems with engineering applications.

The time averaging used in my C++ code has some similarity to how a measurement makes a superposition of eigenstates collapse to one with a definite value of the measured observable. Any measurement process spans a nonzero time interval, and therefore also has something to do with a time average.

The calculation of eigenfunctions with this program is clearly faster if there is a good initial approximation for the wave function and energy value. When finding an approximation for an excited wave function with the usual sequential calculation using imaginary propagation and starting from the ground state, numerical error accumulates while moving to higher energy states. The type of time-averaged state calculation as in this program can also be used for obtaining more correct significant figures to those results, without depending on the accuracy with which the lower energy eigenstates have been computed.

Using this iterative scheme for systems of more degrees of freedom quickly becomes time and memory space consuming when there are many position coordinates. It is probably difficult to do this for anything more than a 2-dimensional helium atom or a hydrogen molecular ion without Born-Oppenheimer approximation but with rotation ignored (4 degs of freedom in both examples), as the linear systems in the time stepping quickly contain very large matrices with increasing number of space coordinates. This situation may change if quantum computing allows for better scaling of computation time as a function of the number of spatial grid points when solving diffusion-type problems such as time-dependent Schrödinger.([9])

One development idea is whether finding an eigenstate for a sequence of Hamiltonians $\hat{H}(t_i)$ by adiabatic passage ([10]) can be made faster by combining it with the method presented here to make $|\psi(t)\rangle$ remain an instantaneous eigenstate of the explicitly time dependent Hamiltonian. This would mean starting with an initial state that is a good approximation to some eigenstate $|n\rangle$ and applying the $\hat{A}(T)$ operator repeatedly while changing the potential energy function V a bit on every iteration. I did not include the source code where I tried this, but it seemed to be more efficient with systems where there are no abrupt changes in potential energy, such as the step of $\Delta V = 20000$ at the boundary of the rectangle and supercircle potential wells described here.

References

[1] https://github.com/TeemuIsojarvi/meanstate (Retrieved on 22.1.2026).

[2] I. Sudiarta and H. Susanto *New Power Method for Solving Eigenvalue Problems*, 10.48550/arXiv.2211.06303 (2022).

[3] 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).

[4] V. A. Vasil'ev and P. S. Chernov, *Time-dependent simulations of two-dimensional quantum waveguides of arbitrary shape*, 2015 Days on Diffraction (DD), St. Petersburg, Russia, 2015, pp. 1-4, doi: 10.1109/DD.2015.7354893.

[5] C. C. Lalescu, *Two hierarchies of spline interpolations. Practical algorithms for multivariate higher order splines*, https://arxiv.org/abs/0905.3564 .

[6] T. Isojärvi, *Quantum mechanics of particles trapped in a Lamé circle or Lamé sphere shaped potential well*, Rev. Mex. Fis. **67** (2) 206-218 (2021).

[7] W. Gaddah, *A Lie group approach to the Schrödinger equation for a particle in an equilateral triangular infinite well*, Eur. J. Phys. **34** 1175 (2013).

[8] G. H. Booth and G. K.-L. Chan, *Communication: Excited states, dynamic correlation functions and spectral properties from full configuration interaction quantum Monte Carlo*, J. Chem. Phys. **137**, 191102 (2012).

[9] P. J. Ollitrault, A. Miessen, and I. Tavernelli, *Molecular Quantum Dynamics: A Quantum Computing Perspective*, Accounts of Chemical Research **54** (23), 4229-4238 (2021).

[10] A. Benseny and K. Mølmer, *Adiabatic theorem revisited: The unexpectedly good performance of adiabatic passage*, Phys. Rev. A **103** (6) 062215 (2021).