Computational Physics VT2020 - Assignment 2

Ólafur Siemsen Sigurðarson

April 7, 2020

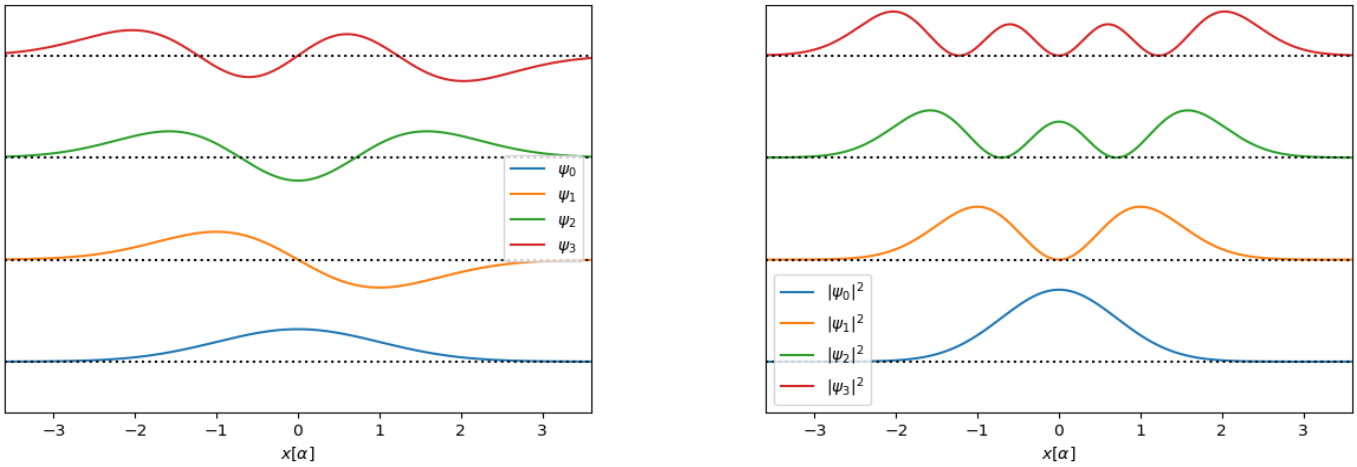

Figure 1: Wavefunction (left) and square modulus (right) for the first several energy eigenvalues.

Herein we consider the perennial favorite, the simple harmonic oscillator. Starting with Schrödinger's equation:

$$H\Psi = E\Psi \tag{1}$$

Where the Hamiltonian for a harmonic oscillator of mass m is

$$H = -\frac{\hbar}{2m}\frac{\partial^2}{\partial\chi^2} + \frac{1}{2}m\omega^2\chi^2. \tag{2}$$

This standard expression uses $\omega = \sqrt{k/m}$ with the harmonic potential $V(\chi) = k\chi^2$. By scaling the spatial coordinate by $\alpha = \sqrt{m\omega/\hbar}$ this becomes

$$H = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + \frac{1}{2}x^2; \qquad x = \sqrt{\frac{m\omega}{\hbar}}\chi. \tag{3}$$

This is a handy scaling where the energy of the eigenfunctions is

$$H\psi_n = E_n\psi_n; \qquad E_n = n + \frac{1}{2}; \qquad \langle\psi_i|\psi_j\rangle = \delta_{ij} \tag{4}$$

in units of $\hbar\omega$. In a proper treatment of this problem the basis of eigenfunctions is infinite but to make computation actually possible we'll need to truncate it in some way. Here we will truncate it by limiting the spatial extent of the wavefunctions – we assume that $\psi(x) = 0$ whenever $x \notin [-a, a]$ for some a. This will produce artifacts in the higher energy wavefunctions whose density is spread out.
Computation also requires a discretization of space. So we consider the spatial coordinates $\{x_1, x_2, \ldots, x_N\}$ where $-x_1 = x_N = a$. The Hamiltonian's second derivative can then be stated in finite differences as

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{x_n} = -\frac{1}{12h^2}\left[f_{n-2} - 16f_{n-1} + 30f_n - 16f_{n+1} + f_{n+2}\right] + \mathcal{O}(h^4). \tag{5}$$

This means that the total Hamiltonian can be stated in a matrix form as

$$H = -\frac{1}{24h^2}\begin{bmatrix} 30 & -16 & 1 & 0 & 0 & 0 & \ldots \\ -16 & 30 & -16 & 1 & 0 & 0 & \ldots \\ 1 & -16 & 30 & -16 & 1 & 0 & \ldots \\ & & & \ddots & & & \\ & \ldots & 0 & 1 & -16 & 30 & -16 \\ & \ldots & 0 & 0 & 1 & -16 & 30 \end{bmatrix} + \frac{1}{2}\mathrm{diag}(\vec{V}) + \mathcal{O}(h^4) \tag{6}$$

where $\mathrm{diag}(\vec{V})$ is a diagonal matrix with the entries of $\{V(x_1), V(x_2), \ldots, V(x_N)\}$. The above matrix's eigenvalues and eigenvectors (in this case eigen*functions*) can then be found with whatever tool is handy.

Handy Tools

As with so many things, there are many ways to compute eigenvalues of a matrix. Which way is used depends on the exact character of the matrix (as well as the details of hardware and software available). Unless stated otherwise, the LAPACK routine heevd is used in the presented calculations; albeit through an interface in Python. The heevd routine exploits the structure of Hermitian matrices and implements a divide-and-conquer algorithm where the problem is broken into smaller problems whose eigenvalues can be found separately and from which the original matrix's eigenvalues can be synthesized.
An iterative method is another way to accomplish the task. We note the fundamental character of eigenpairs that

$$A^n \vec{u} = A^n \sum a_i \vec{v}_i = \sum_i a_i \lambda_i^n \vec{v}_i. \tag{7}$$

This means that eigenvalues of large magnitude blow up compared to the others under repeated application of A. Since we are more interested in the lowest eigenvalue a small modification yields the inverse shifted power method:

$$\vec{b}_{k+1} = \frac{(A-\mu I)^{-1}\vec{b}_k}{\left\|(A-\mu I)^{-1}\vec{b}_k\right\|}. \tag{8}$$

Iteration causes $\vec{b}_k$ to converge to the eigenvector $\vec{v}_i$ whose eigenvalue is $\min|\lambda_i - \mu|$. The quality of the initial guess μ dictates the number of steps needed to converge satisfactorily to an eigenvector. Another consideration is that inverting the matrix as above is often not the fastest way to solve the formula. Instead $(A-\mu I)$ can be factorized and a solver applied according to As characteristics. A naive LU-factorization method is implemented and tested against the LAPACK standard. The performance was predictably abysmal: it took approximately 220 s to calculate 2000 eigenvalues to 6 digits using $0.6 + k$, $k = 0, 1, 2, \ldots$ as an initial guess. Compare this to the 3 s that heevd took to complete the same task (with more precision). As an additional comparison the professional effort of SciPy's sparse.linalg.eigsh routine that uses a Lanczos iterative algorithm took 44 s to match heevd. It's important to note that iterative algorithms are appropriate when only a limited subset of eigenpairs are desired, so the above is a somewhat misguided application.

Cut Short

Let us further explore the effects of truncation on the eigenfunctions, using figure 2.

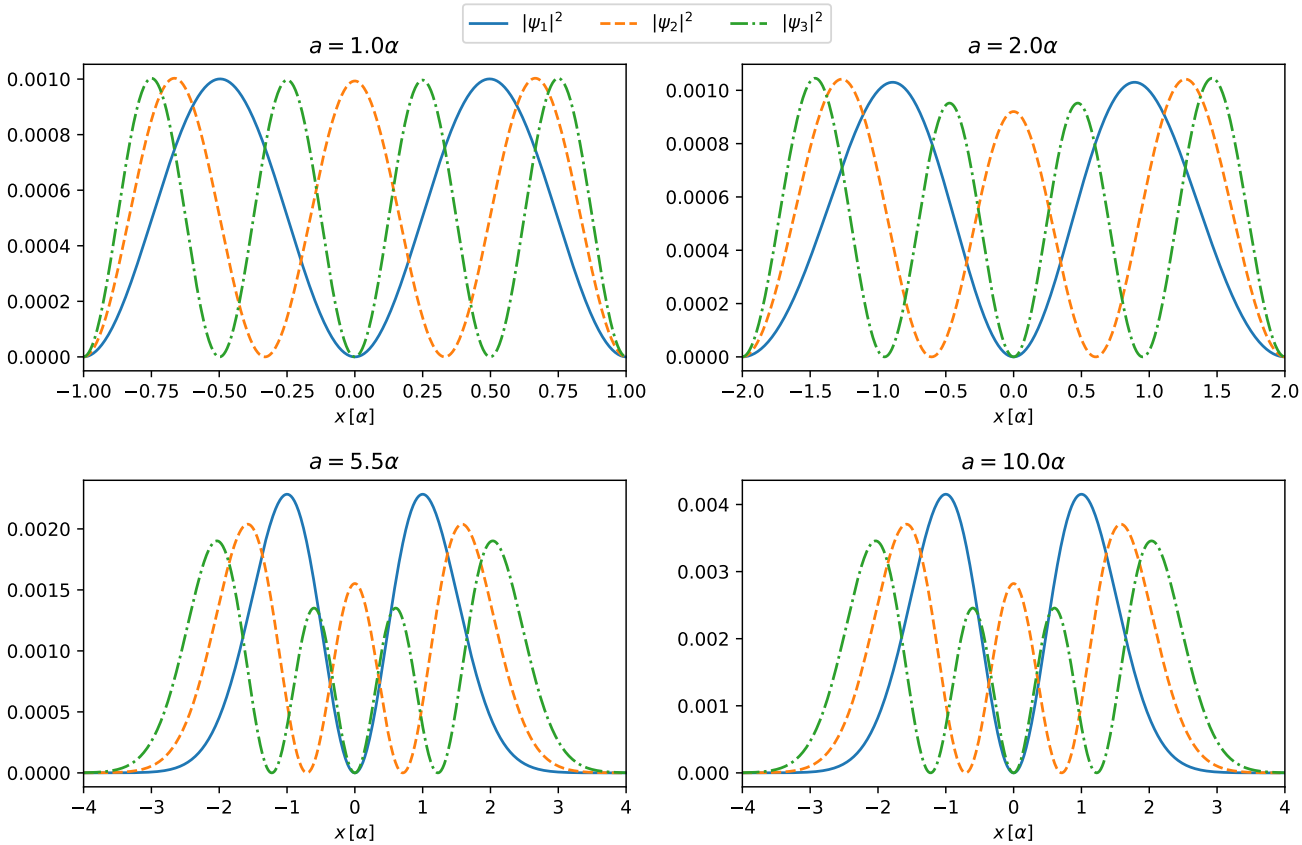

Figure 2: Square modulus of the wavefunction for different truncation lengths a. Note the varying scaling on the vertical and horizontal. Very small a yields a probability density that looks like that of the infinite square well, more so for higher energy eigenfunctions. An eigenfunction stabilizes once the truncation falls outside the bulk of its probability density.

As one would expect, the effect of having a very small truncation is like that of imposing an infinite square well potential in addition to the harmonic one. This will distort the eigenfunction (and corresponding eigenvalue) for as long as the eigenfunction extends outside the length a. Note that it doesn't simply restrain the wavefunction but changes fundamental characteristics. This effectively "spoils" the eigenpairs above a certain n for a given a. The behavior can be seen in more detail in figure 3 where the dependence of eigenvalue energy on its principle quantum number can be seen. Initially the expected behavior of $E_n \sim n$ can be seen but once the square potential is interfering sufficiently with the eigenfunctions the energy takes on the behavior of $E_n \sim n^2$ as per the infinite square well's spectrum. Also, the incline is smaller for larger values of a which corresponds to the infinite square well's $E_n \sim n^2/a$ for a well of width a. There is a another wrinkle to the story of E_n. As n gets close to N the curve flattens out due to truncation in terms of number of eigenpairs that can be computed. This means that choosing N so that this flattening happens shortly after the transition to n^2 behavior of E_n would be most economical.[1]

There is yet another kind of truncation which occurs, as can be seen in equation (5). This comes in the form of some leading term in h. This puts a constraint on the implementation of our solution. If we were to focus overly on the $\mathcal{O}(h^4)$ term this might lead us to select a unduly small h, which would slow down the calculations (as eigenpair calculation is generally $\sim h^{-3}$). This is doubly unwise since such fine resolution is only really relevant on quickly varying, higher energy functions where the spatial truncation mentioned above might produce inaccuracy anyway.

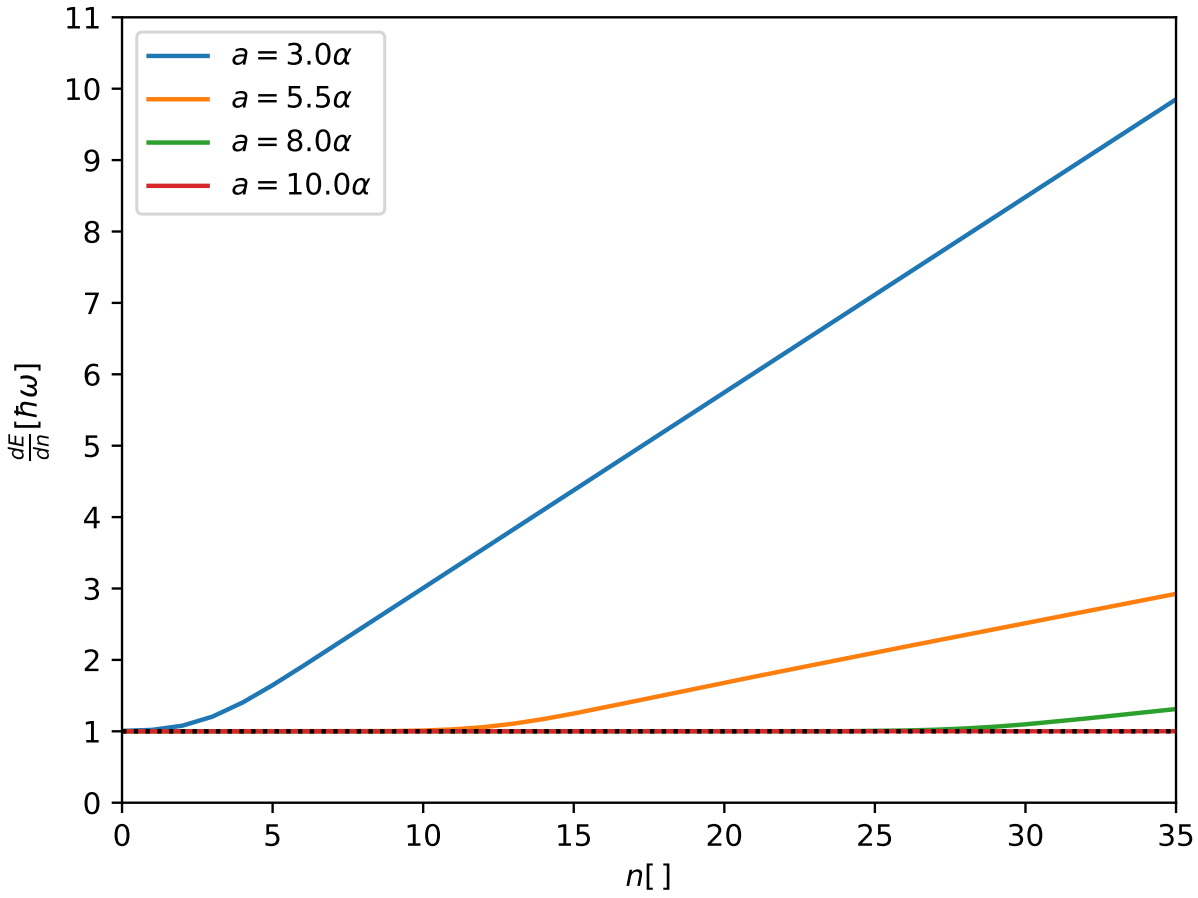

Figure 3: $\frac{\mathrm{d}E_n}{\mathrm{d}n}$ of the first n eigenstates of the harmonic oscillator given spacial truncation at a cut-off distance a. The trajectories starts out flat, like one would expect of the harmonic energy levels where $E_n \sim n$. At a certain point they deviate to grow like n as would be expected of a square well potential where $E_n \sim n^2$. This branching happens earlier for smaller values of a.

The bump

Let's now consider a slight change to the potential, where we add a new term to the potential above, creating a new Hamiltonian:

$$H = -\frac{1}{2}\frac{\partial^2}{\partial x^2} + \frac{1}{2}x^2 + C_1 e^{-C_2 x^2} \tag{9}$$

This new potential has a Gaussian bump in the middle of the harmonic well. This causes probability densities to be pushed away from the center and therefore uphill, increasing energies generally for eigenfunctions that have some presence at $x = 0$. The effect of this bump on the first three wavefunctions can be seen in figure 4.

[1] The author chose to ignore this out of reluctance to take his own advice and used $N = 2000$ for greater peace of mind.

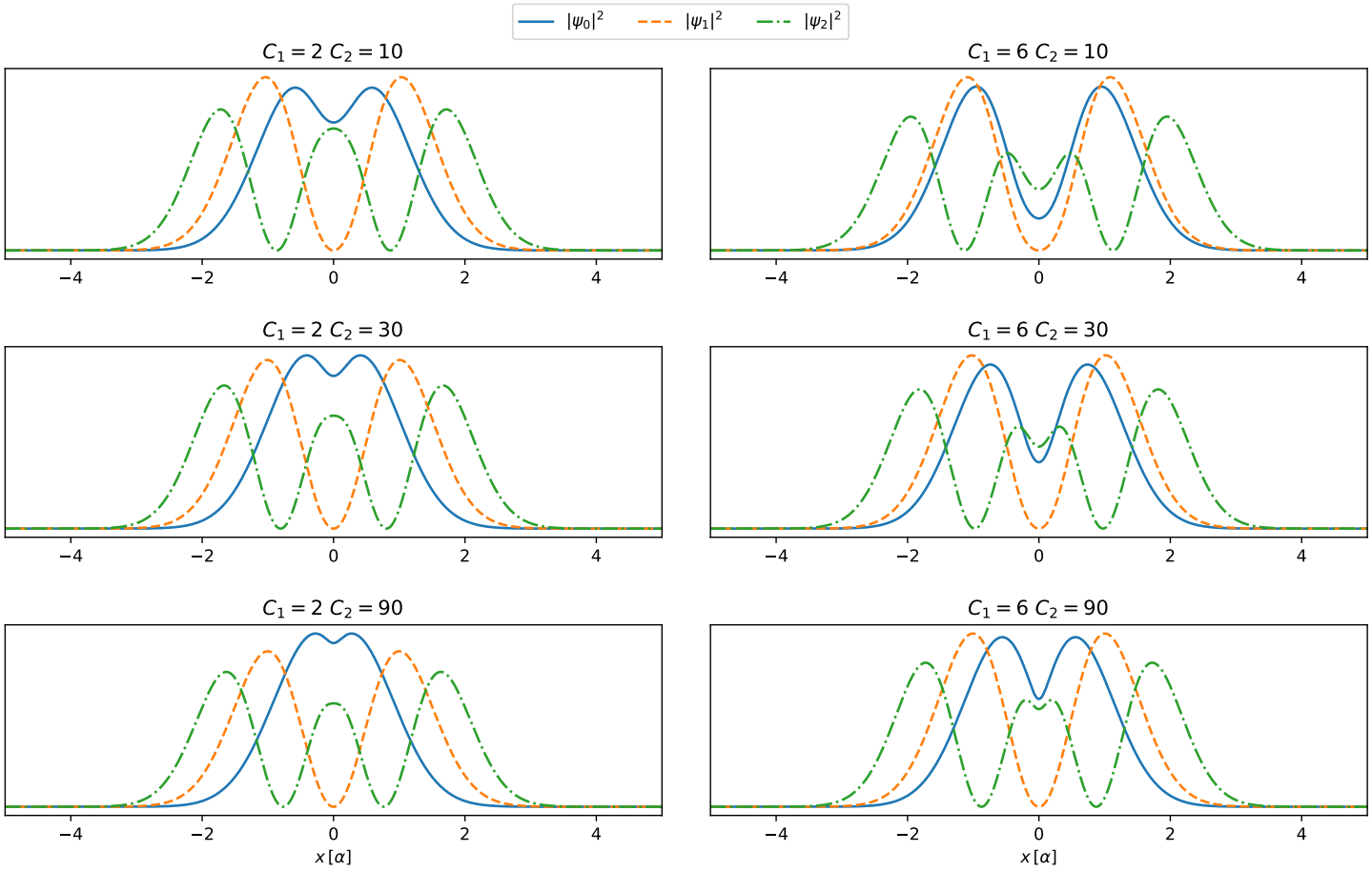

Figure 4: How an addition of $C_1 e^{-C_2 x^2}$ to the potential affects the shape of the first 3 eigenfunctions of the Hamiltonian. The first and third eigenfunctions are repelled from the bottom of the harmonic well, while the second eigenfunction that has a minimum in $x = 0$ is largely unaffected.

Figure 5 shows how introducing the Gaussian bump mixes together the eigenstates of the original harmonic oscillator. There are two thing of note here. Firstly that the mixing is biased towards an increase in energy as would be predicted from the probability density being pushed from the bottom of the well. Second, the wavefunction doesn't mix with neighbouring wavefunctions but skips over one neighbour. This is because the potential is symmetric which means that its eigenfunctions will be either odd or even and since the bump is also symmetric this divide is maintained and the two distinct solution sets aren't mixed as the bump grows.

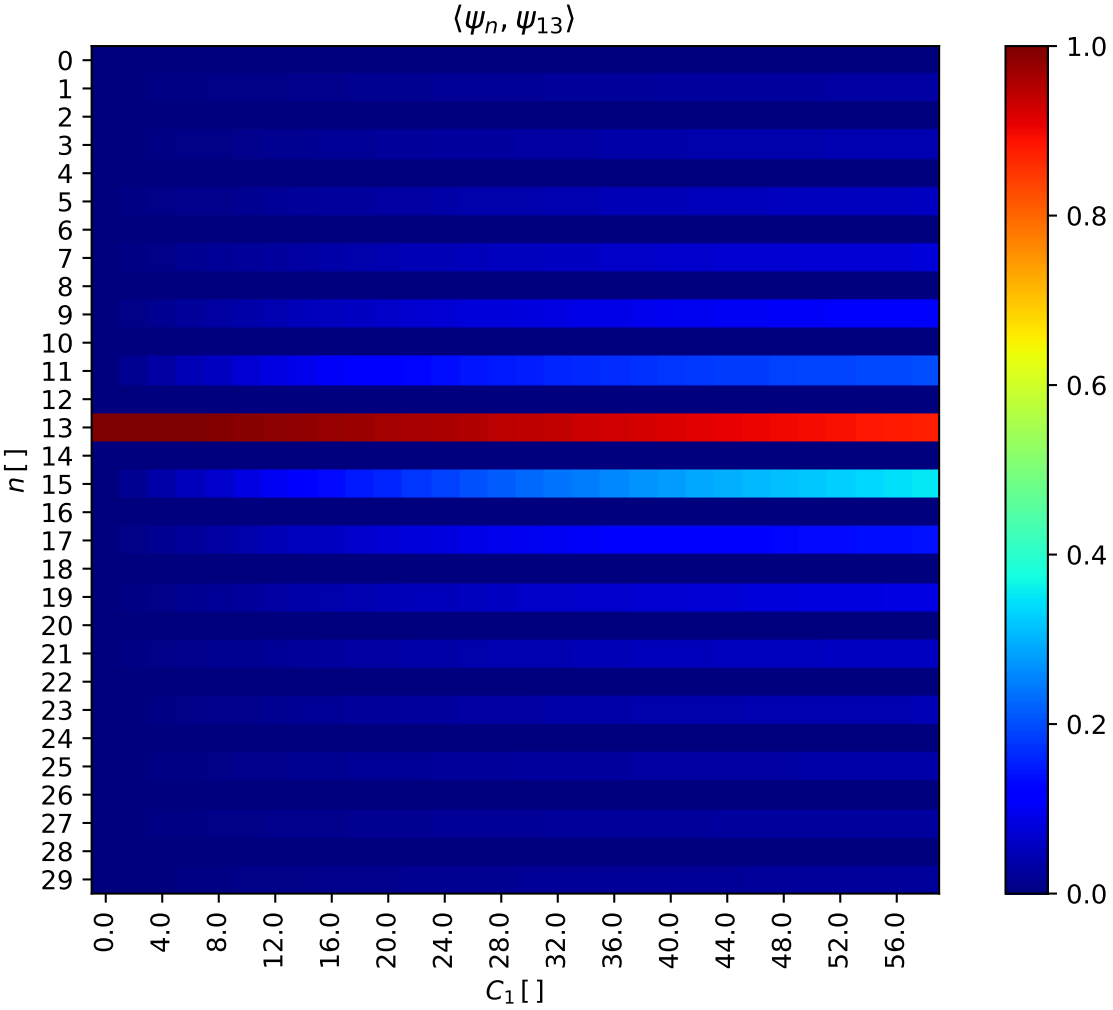

Figure 5: Inner product of ψ_{13} with other eigenfunctions. This shows how the addition of $C_1 e^{-C_2 x^2}$ to the potential with $C_2 = 20$ mixes neighbouring eigenfunctions as C_1 increases. The eigenfunctions are mixed with their next neighbour of the same parity and mixed more with eigenfunctions of higher energies, as is expected.