1D Eigensolver

Daniel Huantes

June 2022

1 What

The Schodinger equation is often written in the form of an operator $\hat{H}$, the Hamiltonian operator, acting on one of n eigenstates $|\psi\rangle$ in an eigenvalue equation. The Hamiltonian corresponds to the potential that the particle finds itself in, and the mod squared of the wave function gives the probability that the particle will be found in some state

$$\hat{H}\,|\psi_n\rangle = E\,|\psi_n\rangle \tag{1}$$

there are normally an infinite number of eigenstates, and these eigenstates form a complete set, which can be used to find stationary states for a system, and to construct arbitrary wave packets.

2 Why

Using discrete eigensolvers we can find the eigenvectors and eigenvalues of our hamiltonian. In analytic situations this can be done by constructing ladder operators, but using numerical methods we can treat arbitrary potentials.

So we have the tools to describe arbitrary wavepackets evolutions in arbitrary potentials, with one caveat. Because our solutions are discrete, we have a finite domain that they are defined under. Because we desire wavefunctions normalized in our domain, that leaves the probability the particle will be within our bounds to be unity. This means that the wavefunction is zero at any position ourside of our domain. This has the effect of modifying our "arbitrary" potential to always be immersed in a infinite well.

3 How

We do it via the code uploaded here, under `1D_eigensolver.py`

4 Future

The python code is mostly a proof of concept, that can hopefully be used in more efficient C++ code for beam propagation using perturbative methods. It could also potentially be modified to be a python module so it can be utilized to find eigenvectors. One thing the program does not currently do is find the decomposition of arbitrary wavepackets into $a(n)$ coefficients, so arbitrary normalized wavepackets can be represented as linear combinations of eigenstates of a given potential. This would also be the way to introduce time dependence, but at this time it feels like that is clear, and is actually relatively uniform for many of the methods