How to use our Schrödinger equation code

Nick Konz, Thomas Richards, Thomas Marshall

December 2, 2018

1 Needed Files

Make sure you have the following files:

1. `main.py`
2. `schrodingerutils.py`
3. `chebfft.py`
4. `cranknicholson.py`

2 Running the Code

The calling sequence is simple, once you're in a directory with the above python files, you can run the code with the following script arguments (in order):

1. `J` (number of positional support points, integer)
2. `dt` (time step, float)
3. `solver` (type of solver, string.) either `CN` or `CFFT`
4. `potential` (V(x), string), options are
   (a) `free`
   (b) `infwell`
   (c) `finwell`
   (d) `barrier`
   (e) `harmonic`
   (f) `hydrogen`
5. `psi0_name` (initial wavefunction, string); the options for each potential are:
   (a) `free`: `wavepacket`

(b) `infwell`: `stationarystate` stationary state, with default energy level $n = 3$. You can specify the energy level n by adding `-j n` at the end of the script call, for example,

```
python main.py 100 1.0 CN infwell stationarystate -j 5
```

calls Crank Nicholson with $J = 100$, $dt = 1.0$ on the infinite well potential with initial wave function of the $n = 5$ stationary state.

(c) `finwell`: `boundstate`

(d) `barrier`: `wavepacket`

(e) `harmonic`: `groundstate`

(f) `hydrogen`: `groundstate`

Another usage example is

```
python main.py 100 1.0 CN harmonic groundstate
```

which calls Crank Nicholson with $J = 100$, $dt = 1.0$, with harmonic oscillator potential, and initial wavefunction of the ground state.

Unfortunately, CFFT doesn't work for high J, or doesn't work at all for some potentials. Free particle didn't work for about $J = 45$ or higher, while others required even lower J. Or, of course, a lower Δt can be helpful. CN works for everything, however.