Joshua Cook
November 13, 2014

1 Problem

Eigenproblems abound in the modeling of nature.

Density Functional Theory models yield single particle-like Schrödinger equations with a nonlinear potential term that accounts for all the many-body interactions.

Electronic Structure Theory generates Schrodinger like equations.

2 Goal

We seek a method for an efficient computational solution to the time-independent Schrödinger equation.

2.1 the Time-Independent Schrödinger Equation

Given, $\hat{H}$ is the Hamiltonian operator (or Hamiltonian matrix) for a discrete system), ψ is the wavefunction, and E, discrete energy states, where

$$\hat{H} = T + V(x) = -\frac{\hbar^2}{2m}\Delta + V(x) \tag{1}$$

for Δ, the Laplacian.

We have

$$\hat{H}\psi_j(r) = E_j\psi_j(r), \quad j \in \mathbb{N} \tag{2}$$

For numerical calculations, we will typically take H to be a matrix of discrete values describing the position and momentum of particles in the system.

Note that this is an Eigenproblem of the form

$$A\mathbf{x} = \lambda\mathbf{x} \tag{3}$$

The Imaginary Time Propagation Method

The imaginary time propagation method (ITP) relies on solving the time-dependent Schrödinger equation:

$$i\hbar\frac{\partial\psi(r,t)}{\partial t} = \hat{H}\psi(r,t) \tag{4}$$

in imaginary time where .

Wick Rotation

We perform a Wick Rotation (setting $t = -i\tau$) to transform eq. 4 into a simple diffusion equation

$$\frac{\partial \psi(r,\tau)}{\partial \tau} = -\frac{\hat{H}}{\hbar}\psi(r,\tau) \tag{5}$$

2.2 Solution to the Diffusion Equation

The formal solution to eqn. eq. 5 is given by

$$\psi(r,\tau) = \exp(-\hat{H}\tau/\hbar)\psi(r,0) \tag{6}$$

We expand the initial state $\psi(r,0)$ in terms of the eigenfunctions $\phi_j(r)$ the correspond to the eigenvalues E_j for

$$\hat{H}\phi_j(r) = E_j\phi_j(r) \tag{7}$$

The time evolution starting from the initial state $\psi(r,0)$ can now be written as

$$\psi(r,\tau) = e^{-\hat{H}\tau/\hbar}\psi(r,0) = e^{-\hat{H}\tau/\hbar}\sum_{j=0}^{\infty} a_j\psi_j(r) = \sum_{j=0}^{\infty} a_j e^{E_j\tau/\hbar}\phi_j(r) \tag{8}$$

2.3 Imaginary Time Propagation as Iterative Solution

As $\tau \to \infty$, $\psi(r,\tau)$ becomes proportional to $\phi_0(r)$. In other words, iterated ψ functions will converge on the eigenfunction for the base state of the time-**independent** equation (eq. 6). Here we are solving an eigenproblem through an interative approximation of a differential equation.

iterative differential equation methods

Olver finite differences

3 Implementation

```
import numpy as np
import numpy.linalg as la
import math, time
import matplotlib.pyplot as plt
from sys import argv
import datetime
%matplotlib inline


k = 100

eps = 10E-6
times = np.array([[0.,0.]])
temp_times = times
H = np.random.rand(k+200,k+200)
H = H.T.dot(H)
file = datetime.datetime.now().strftime("%Y%m%d%H%M%S")


for i in range(k):
        # print i
        i = i+2
        n = i
        err = 1
        conv = 1
        int_H = H[0:n,0:n]

        start = time.clock()

        phi0 = np.random.rand(n)
        # print la.eig(H)[1].T
        CayleyN = (np.identity(n)-0.5*int_H)
        CayleyP = (np.identity(n)+0.5*int_H)

        while(conv > eps):
                phi1 = la.solve(CayleyP,CayleyN.dot(phi0))
                mu = math.sqrt(phi1.dot(phi1))
                phi1 = phi1/mu
                conv = math.sqrt((np.abs(phi1)-np.abs(phi0)).dot(np.abs(phi1)-np.abs(phi0)))
                # err = math.sqrt(2)*math.sqrt(abs(phi1.dot(int_H.dot(int_H)).dot(phi1)- (phi1.dot(int_
                # print err
                phi0 = phi1

        end = time.clock()
        delta_t = end-start
        temp_times[0][0] = i
```

```
        temp_times[0][1] = delta_t
        times = np.concatenate((times,temp_times),axis=0)
        np.savetxt(file,times,fmt='%.4e')


plt.plot(times[:k,0],times[:k,1])

plt.show()
```

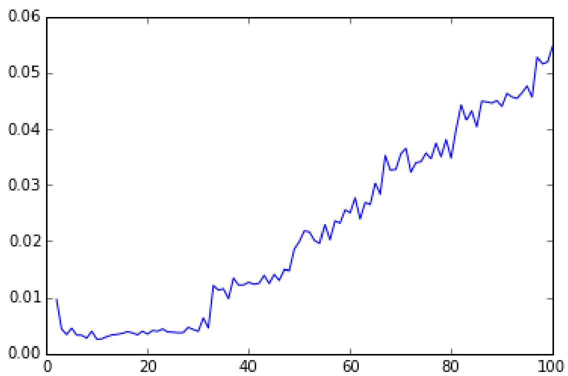

Figure 1: png

```
np.set_printoptions(precision=6)
phi0
```

```
array([-0.098619, -0.100255, -0.095033, -0.095798, -0.09463 , -0.096591,
       -0.096496, -0.100948, -0.103793, -0.105997, -0.102556, -0.101054,
       -0.106374, -0.104128, -0.096394, -0.101002, -0.098431, -0.101118,
       -0.09653 , -0.100673, -0.09977 , -0.096231, -0.097664, -0.095953,
       -0.10621 , -0.101699, -0.100809, -0.095755, -0.093848, -0.097243,
       -0.097106, -0.097369, -0.10216 , -0.1036  , -0.094454, -0.099593,
       -0.100047, -0.102212, -0.094055, -0.095699, -0.098182, -0.10711 ,
       -0.097889, -0.097309, -0.099708, -0.09835 , -0.100899, -0.100339,
       -0.0983  , -0.090089, -0.110011, -0.097674, -0.098536, -0.097574,
       -0.099521, -0.098617, -0.100734, -0.099086, -0.099396, -0.100646,
       -0.100855, -0.105439, -0.101658, -0.101174, -0.095862, -0.100688,
       -0.09901 , -0.101402, -0.098972, -0.103596, -0.09954 , -0.097495,
       -0.102617, -0.10404 , -0.10443 , -0.102043, -0.097404, -0.096225,
       -0.101167, -0.100604, -0.097121, -0.100775, -0.101373, -0.095651,
       -0.10295 , -0.098441, -0.095474, -0.094655, -0.097631, -0.093334,
       -0.09568 , -0.098619, -0.103043, -0.091929, -0.096411, -0.10155 ,
       -0.096962, -0.093571, -0.103762, -0.102534, -0.109668])
```

```
la.eig(int_H)[1][:,0]
```

```
array([ 0.098598,  0.100265,  0.095025,  0.095775,  0.09464 ,  0.096599,
        0.096508,  0.100957,  0.103805,  0.106031,  0.102552,  0.101034,
        0.10636 ,  0.104147,  0.096384,  0.101009,  0.098411,  0.101118,
        0.096546,  0.100696,  0.09975 ,  0.096243,  0.097646,  0.095947,
        0.106207,  0.101702,  0.100812,  0.095777,  0.093859,  0.097235,
        0.097111,  0.097374,  0.102151,  0.103593,  0.094456,  0.099608,
        0.100065,  0.102219,  0.094074,  0.095701,  0.098194,  0.107129,
        0.097895,  0.097316,  0.099701,  0.098345,  0.100888,  0.100346,
        0.098301,  0.090111,  0.110021,  0.097654,  0.09853 ,  0.097568,
        0.099519,  0.09863 ,  0.100716,  0.099087,  0.099384,  0.100641,
        0.100856,  0.105444,  0.101621,  0.101194,  0.095854,  0.100674,
        0.09899 ,  0.101403,  0.098962,  0.103598,  0.099548,  0.097491,
        0.102601,  0.104041,  0.104411,  0.102041,  0.097384,  0.096235,
        0.101188,  0.100597,  0.097123,  0.100782,  0.101379,  0.095635,
        0.102954,  0.098423,  0.095493,  0.094661,  0.097597,  0.093326,
        0.095669,  0.098604,  0.10304 ,  0.091931,  0.096429,  0.101558,
        0.096972,  0.093569,  0.103769,  0.102539,  0.109665])
```

```
la.eig(int_H)[0]
```

```
array([  7.574560e+03,   5.851004e+01,   5.632109e+01,   5.459495e+01,
         5.362252e+01,   5.296774e+01,   5.150103e+01,   4.943875e+01,
         4.818028e+01,   4.691145e+01,   4.621418e+01,   4.521559e+01,
         4.484616e+01,   4.425040e+01,   4.365566e+01,   4.232450e+01,
         4.043672e+01,   3.973590e+01,   3.936313e+01,   3.874096e+01,
         3.814460e+01,   3.781824e+01,   3.655263e+01,   3.594899e+01,
         3.502903e+01,   3.480710e+01,   3.395651e+01,   5.005400e+00,
         5.314336e+00,   3.341442e+01,   3.285543e+01,   3.288507e+01,
         5.539514e+00,   6.032592e+00,   3.221875e+01,   3.176203e+01,
         3.166426e+01,   3.041883e+01,   6.391642e+00,   6.543347e+00,
         6.718073e+00,   2.977022e+01,   2.926405e+01,   2.893288e+01,
         7.281683e+00,   2.836177e+01,   7.416408e+00,   7.637716e+00,
```

```
2.810322e+01,   2.790089e+01,   2.732124e+01,   8.033226e+00,
8.130448e+00,   2.670814e+01,   8.462147e+00,   8.385890e+00,
2.586330e+01,   9.370232e+00,   9.638478e+00,   2.547537e+01,
2.487394e+01,   1.001520e+01,   2.481070e+01,   1.029646e+01,
1.064633e+01,   2.368296e+01,   2.346100e+01,   1.108152e+01,
2.316317e+01,   1.124323e+01,   1.136374e+01,   2.257901e+01,
2.201280e+01,   1.193988e+01,   1.167881e+01,   1.239554e+01,
2.160414e+01,   2.108255e+01,   2.124395e+01,   1.266829e+01,
1.309715e+01,   1.340048e+01,   1.361722e+01,   1.411554e+01,
2.033496e+01,   2.044325e+01,   1.944888e+01,   1.735240e+01,
1.514861e+01,   1.548841e+01,   1.873003e+01,   1.841149e+01,
1.631191e+01,   1.453414e+01,   1.463134e+01,   1.655187e+01,
1.588873e+01,   1.820149e+01,   1.898044e+01,   1.811454e+01,
1.584247e+01])
```

Finding an Excited Eigenstate

Given our basic