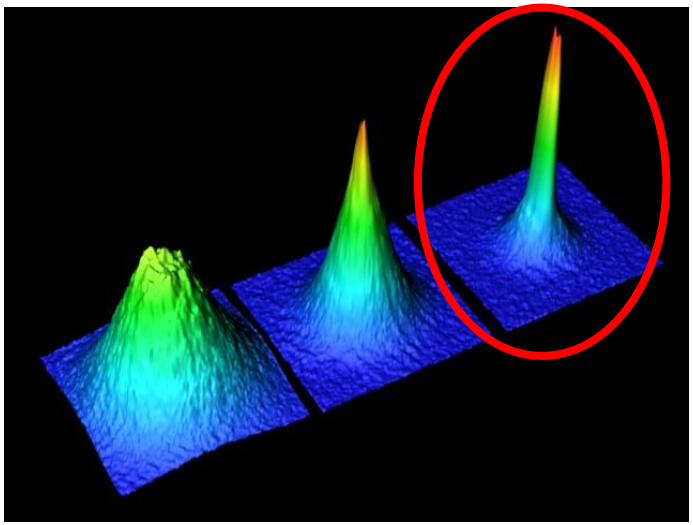
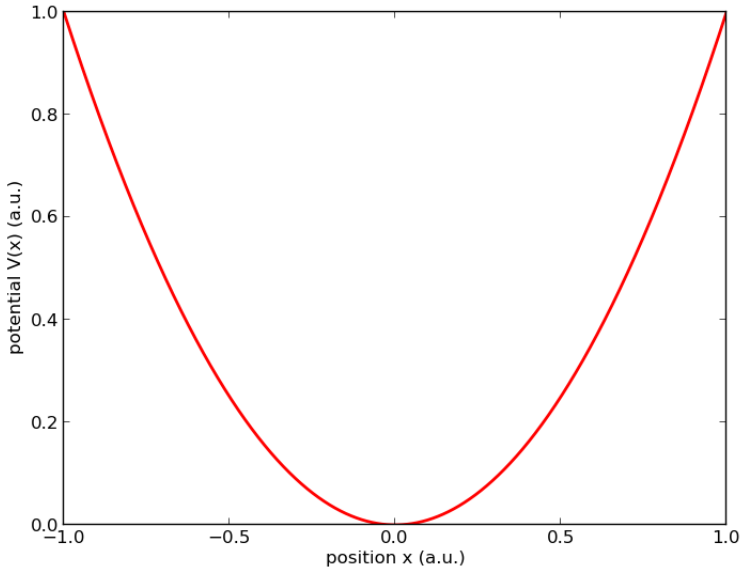
Numerical solving Gross-Pitaevskii-equation

Journal club 9-21-2012

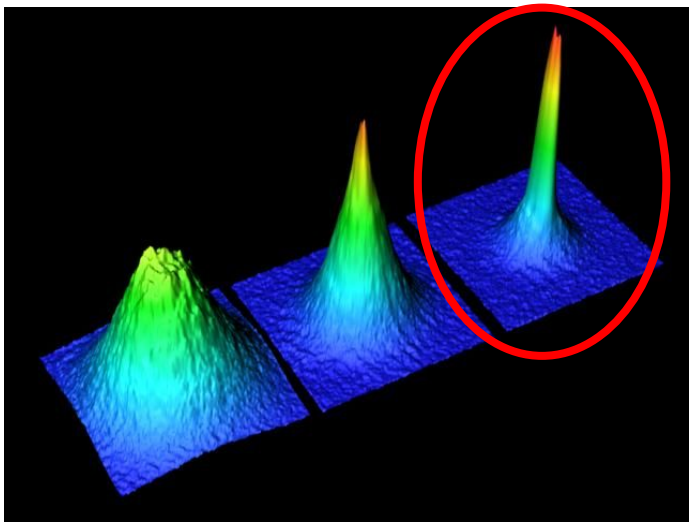
Kris

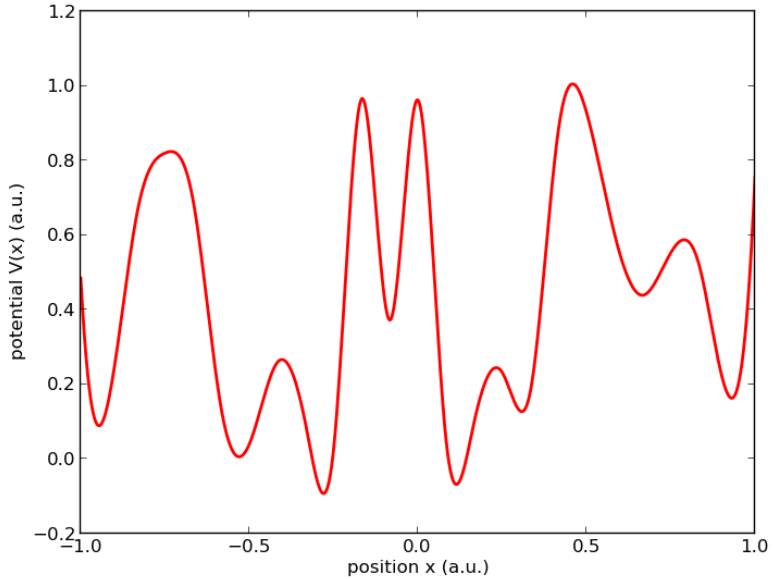
Goal:





Goal:




1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
potential V(x) (a.u.)
−1.0
−0.5
0.0
0.5
1.0
position x (a.u.)

Goal:

- Find ground state wave function
  of time-independent Schrödinger equation

- 1D for simplicity

- arbitrary potentials

- easy to implement and use

- include atom-atom s-wave interactions in mean field
  -> solve GPE

Harmonic oscillator: analytic

$$\hat{H} = \hat{T} + \hat{V} = \frac{\hat{p}}{2m} + V(x)$$

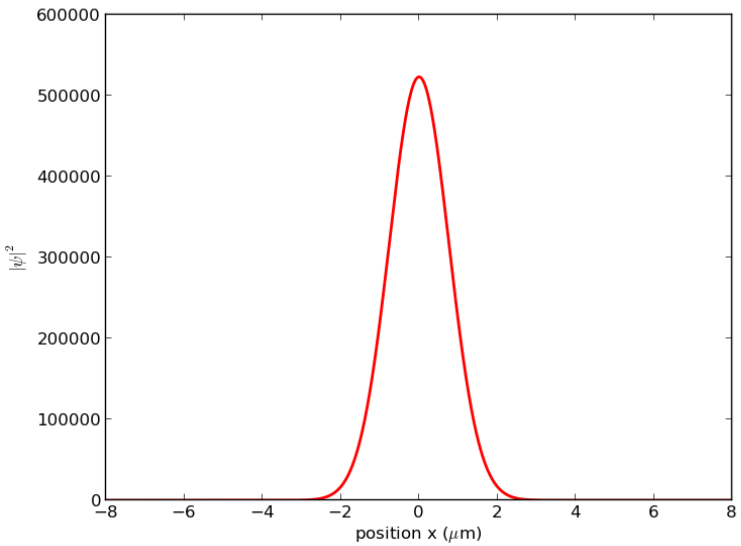
Harmonic oscillator: analytic

$$\hat{H} = \hat{T} + \hat{V} = \frac{\hat{p}}{2m} + \frac{1}{2} m\omega^2 \hat{x}^2$$

Harmonic oscillator: analytic

$$\hat{H} = \hat{T} + \hat{V} = \frac{\hat{p}}{2m} + \frac{1}{2} m\omega^2 \hat{x}^2$$

$$\psi_n(x) = \frac{1}{\sqrt{2^n\, n!}} \cdot \left(\frac{m\omega}{\pi\hbar}\right)^{1/4} \cdot e^{-\frac{m\omega x^2}{2\hbar}} \cdot H_n\left(\sqrt{\frac{m\omega}{\hbar}} x\right), \qquad n = 0, 1, 2, \ldots.$$



Method I: Direct diagonalization

$$\hat{H}\Psi(x) = E\Psi(x)$$

- Find suitable basis $|n\rangle$

- Write down Hamiltonian in matrix form

$$H_{m,n} = \langle m|\hat{H}|n\rangle$$

- solve the eigenvalue problem

Method I: Direct diagonalization

$$H = T + V = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega^2 x^2$$

Method I: Direct diagonalization

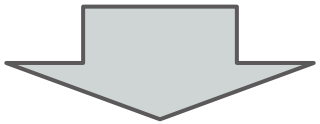
$$H = T + V = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega^2 x^2$$

$$V(x) = \sum_b c_b \, e^{ik_b x} \qquad k_b = \frac{2\pi b}{L}$$

Method I: Direct diagonalization

$$H = T + V = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega^2 x^2$$

$$V(x) = \sum_b c_b \, e^{ik_b x} \qquad k_b = \frac{2\pi b}{L}$$



$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \sum_b c_b \, e^{ik_b x}$$

Method I: Direct diagonalization

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \sum_b c_b \, e^{ik_b x}$$

- Find suitable basis

$$|n\rangle = \frac{1}{\sqrt{L}} e^{ik_n x}$$

Method I: Direct diagonalization

$$H = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \sum_b c_b\, e^{ik_b x} \qquad |n\rangle = \frac{1}{\sqrt{L}} e^{ik_n x}$$

- Write down Hamiltonian in matrix form

$$H_{m,n} = \langle m|\hat{H}|n\rangle$$

$$T_{m,n} = \frac{\hbar^2 k_n^2}{2m}\delta_{m,n} \qquad V_{m,n} = c_b\, \delta_{m,n+b}$$

Method I: Direct diagonalization

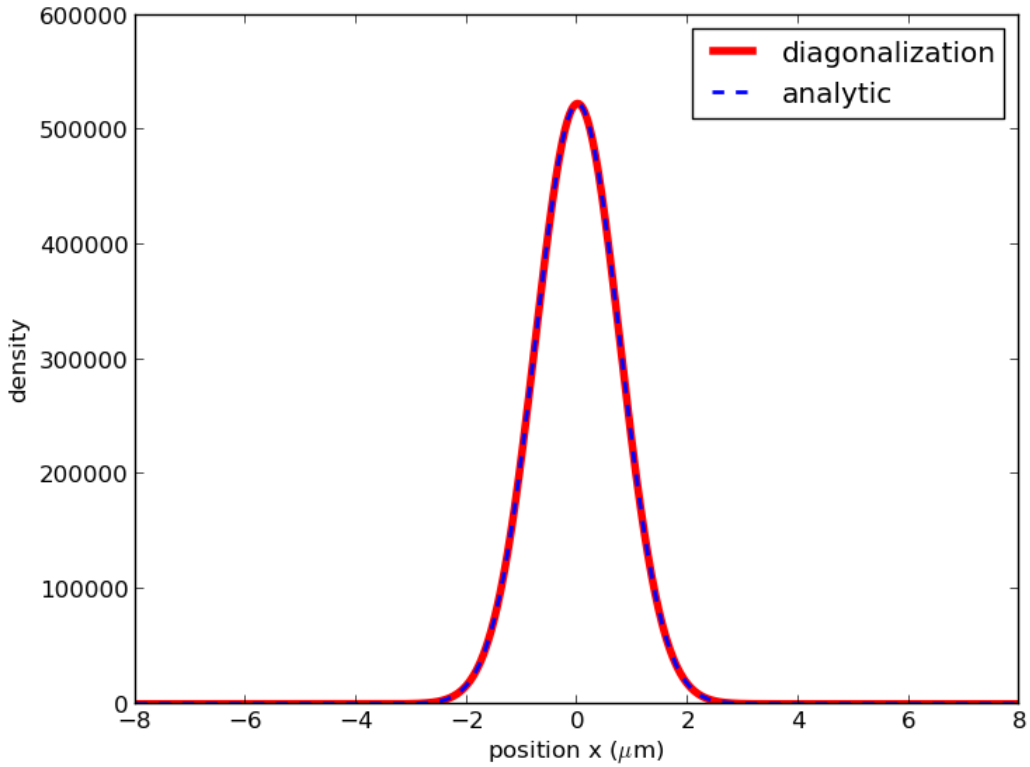
```
x,dx = linspace(-8e-6,8e-6,N,endpoint=False,retstep=True)
k = 2*pi*ft.fftshift(ft.fftfreq(len(x), dx))

T = diag(k**2*hbar**2/(2.*m)+1j*0)

potential = .5*m*omega**2*x**2
potential_ft = concatenate((zeros(N/2),ft.fftshift(fft(potential))/N,zeros(N/2)))
V = array([roll(potential_ft,i) for i in xrange(N)])[:,N-1:]

eva,eve= eig(T+V)
```

Method I: Direct diagonalization



Imaginary time propagation

- find approximation of the ground state of a Hamiltonian

Imaginary time propagation

- find approximation of the ground state of a Hamiltonian

- assume general wave function $|\psi\rangle = \sum_k c_k |k\rangle$
  with $\hat{H}|k\rangle = \epsilon_k |k\rangle$ and $\epsilon_0 = 0$ for simplicity

Imaginary time propagation

- find approximation of the ground state of a Hamiltonian

- assume general wave function $|\psi\rangle = \sum_k c_k |k\rangle$
  with $\hat{H}|k\rangle = \epsilon_k |k\rangle$ and $\epsilon_0 = 0$ for simplicity

- apply the time propagator $e^{i\Delta_t \hat{H}/\hbar}$ in imaginary time
  i.e. $\Delta_t = -i\tau$

$$|\psi(\tau)\rangle = e^{-\hat{H}\tau/\hbar} \sum_k c_k |k\rangle = \sum_k e^{-\epsilon_k \tau} c_k |k\rangle$$

- not unitary -> norm not preserved

Split-Step Technique

- In general $\hat{H} = \hat{T} + \hat{V}$

Split-Step Technique

- In general $\hat{H} = \hat{T} + \hat{V}$ and $[\hat{T}, \hat{V}] \neq 0$

Split-Step Technique

- In general $\hat{H} = \hat{T} + \hat{V}$ and $[\hat{T}, \hat{V}] \neq 0$

$$e^{-i\Delta t\hat{H}/\hbar} = e^{-i\Delta t(\hat{T}+\hat{V})/\hbar} \neq e^{-i\Delta t\hat{T}/\hbar}e^{-i\Delta t\hat{V}/\hbar}.$$

Split-Step Technique

- In general $\hat{H} = \hat{T} + \hat{V}$ and $[\hat{T}, \hat{V}] \neq 0$

$$e^{-i\Delta t\hat{H}/\hbar} = e^{-i\Delta t(\hat{T}+\hat{V})/\hbar} \neq e^{-i\Delta t\hat{T}/\hbar}e^{-i\Delta t\hat{V}/\hbar}.$$

$$e^{-\frac{i}{\hbar}\hat{H}\Delta t} = e^{\widetilde{T}+\widetilde{V}} = e^{\widetilde{T}} \cdot e^{\widetilde{V}} \cdot e^{-\frac{1}{2}[\widetilde{T},\widetilde{V}]+\mathcal{O}(\Delta t^3)}$$

Split-Step Technique

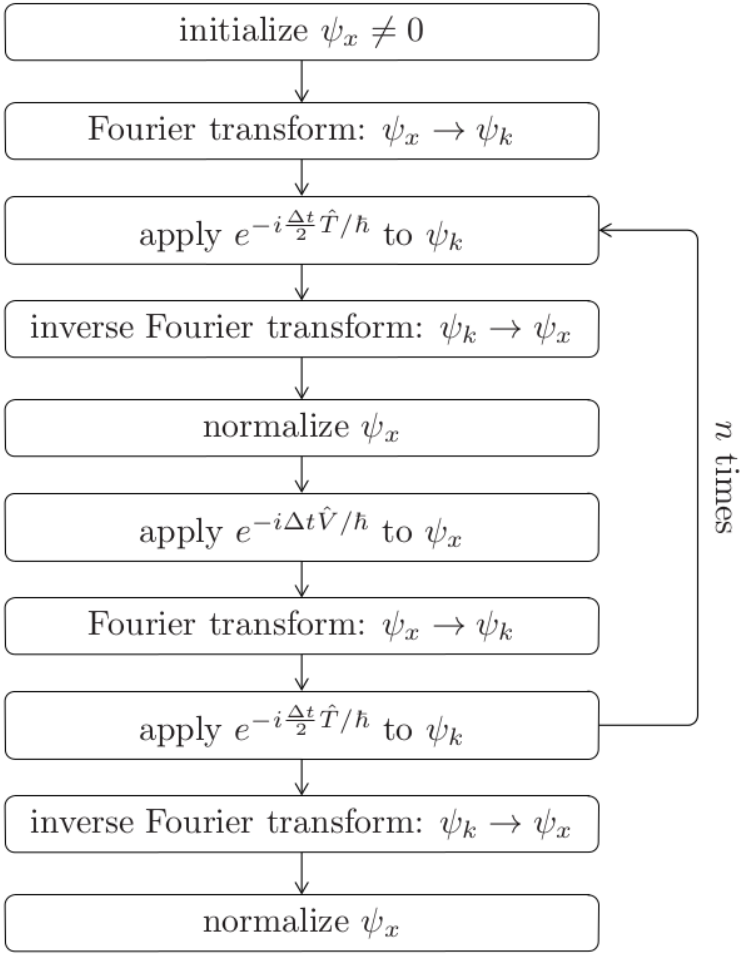
- In general $\hat{H} = \hat{T} + \hat{V}$ and $[\hat{T}, \hat{V}] \neq 0$

$$e^{-i\Delta t\hat{H}/\hbar} = e^{-i\Delta t(\hat{T}+\hat{V})/\hbar} \neq e^{-i\Delta t\hat{T}/\hbar}e^{-i\Delta t\hat{V}/\hbar}.$$

$$e^{-\frac{i}{\hbar}\hat{H}\Delta t} = e^{\widetilde{T}+\widetilde{V}} = e^{\widetilde{T}} \cdot e^{\widetilde{V}} \cdot e^{-\frac{1}{2}[\widetilde{T},\widetilde{V}]+\mathcal{O}(\Delta t^3)}$$

$$\begin{aligned}
e^{-\frac{i}{\hbar}H\Delta t} &= e^{\widetilde{T}/2+\widetilde{V}+\widetilde{T}/2} \\
&= e^{\widetilde{T}/2} \cdot e^{\widetilde{V}+\widetilde{T}/2} \cdot e^{-\frac{1}{2}[\widetilde{T}/2,\widetilde{T}/2+\widetilde{V}]+O(\Delta t^3)} = e^{\widetilde{T}/2} \cdot e^{\widetilde{V}+\widetilde{T}/2} \cdot e^{-\frac{1}{2}[\widetilde{T}/2,\widetilde{V}]+O(\Delta t^3)} \\
&= e^{\widetilde{T}/2} \cdot e^{\widetilde{V}} \cdot e^{\widetilde{T}/2} \cdot e^{-\frac{1}{2}[\widetilde{V},\widetilde{T}/2]+O(\Delta t^3)} \cdot e^{-\frac{1}{2}[\widetilde{T}/2,\widetilde{V}]+O(\Delta t^3)} \\
&= e^{\widetilde{T}/2} \cdot e^{\widetilde{V}} \cdot e^{\widetilde{T}/2} \cdot e^{O(\Delta t^3)}
\end{aligned}$$

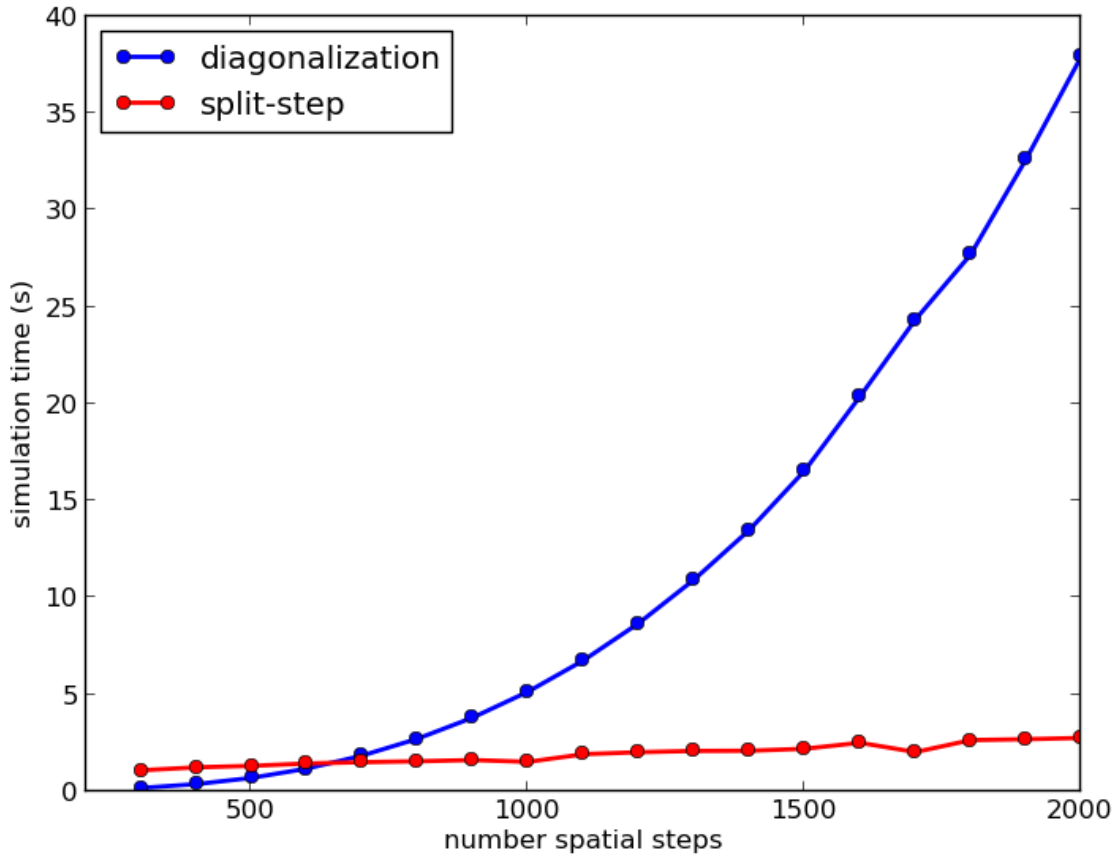
Split-Step Technique



Imaginary-time split-step Technique

```
class ssft():
    def __init__(self, lx,steps = 1001):
        self.p = parameters.tp_parameters()
        self.lx = lx
        self.steps = steps
        self.x,self.dx = scipy.linspace(-self.lx/2.,self.lx/2.,self.steps,endpoint=False,retstep=True)
        self.dk =  2*pi/self.lx
        self.k = 2*pi*scipy.fftpack.fftfreq(len(self.x), self.dx) ## k-grid
    def flat_psi(self):
        return self.Normalize(scipy.ones(self.x.shape))
    def rand_psi(self):
        return self.Normalize(scipy.rand(self.x.shape[0]))
    def Normalize(self,psi):
        return (1./((scipy.absolute(psi)**2).sum()*self.dx))**.5*psi
    def Normalize_k(self,psi_k):
        return (1./((scipy.absolute(psi_k)**2).sum()*self.dk))**.5*psi_k
    def ssft(self, potential, dt=-.1j*50e-6,max_its = 10000, start_psi = None, precision= 1e-8):
        self.dt = dt
        self.T = hbar**2*self.k**2/(2*self.p.m)
        self.exp_T =  scipy.exp(-1j*self.dt/(2.*hbar)*self.T)
        self.exp_2T =  scipy.exp(-1j*self.dt/(hbar)*self.T)
        if start_psi == None:
            psi = self.rand_psi()
        else:
            psi = self.Normalize(start_psi)
        psi_old = scipy.array(psi)
        exp_prefactor = -1j*self.dt/hbar
        psik = scipy.fftpack.fft(psi)
        psik *= self.exp_T
        psi = scipy.fftpack.ifft(psik)
        for i in scipy.arange(max_its):
            if scipy.iscomplex(self.dt):
                psi = self.Normalize(psi)
            psi *= scipy.exp(exp_prefactor*potential)
            psik = scipy.fftpack.fft(psi)
            psik *= self.exp_2T
            psi = scipy.fftpack.ifft(psik)
            if precision != None:
                if ((scipy.absolute(psi-psi_old)/scipy.absolute(psi)).sum() < precision)*(i>50):
                    break
                else:
                    psi_old = psi
        if scipy.iscomplex(self.dt):
            psi = self.Normalize(psi)
        psi *= scipy.exp(-1j*self.dt/hbar*potential)
        psik = scipy.fftpack.fft(psi)
        psik *= self.exp_T
        psi = scipy.fftpack.ifft(psik)
        return self.Normalize(psi)
```

Split-step vs. diagonalization



Interactions

Gross-Pitaevskii equation:

$$\left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x) + \frac{4\pi\hbar^2 a}{m}|\psi(x)|^2\right]\psi(x) = \mu\,\psi(x)$$

Interactions

Gross-Pitaevskii equation:

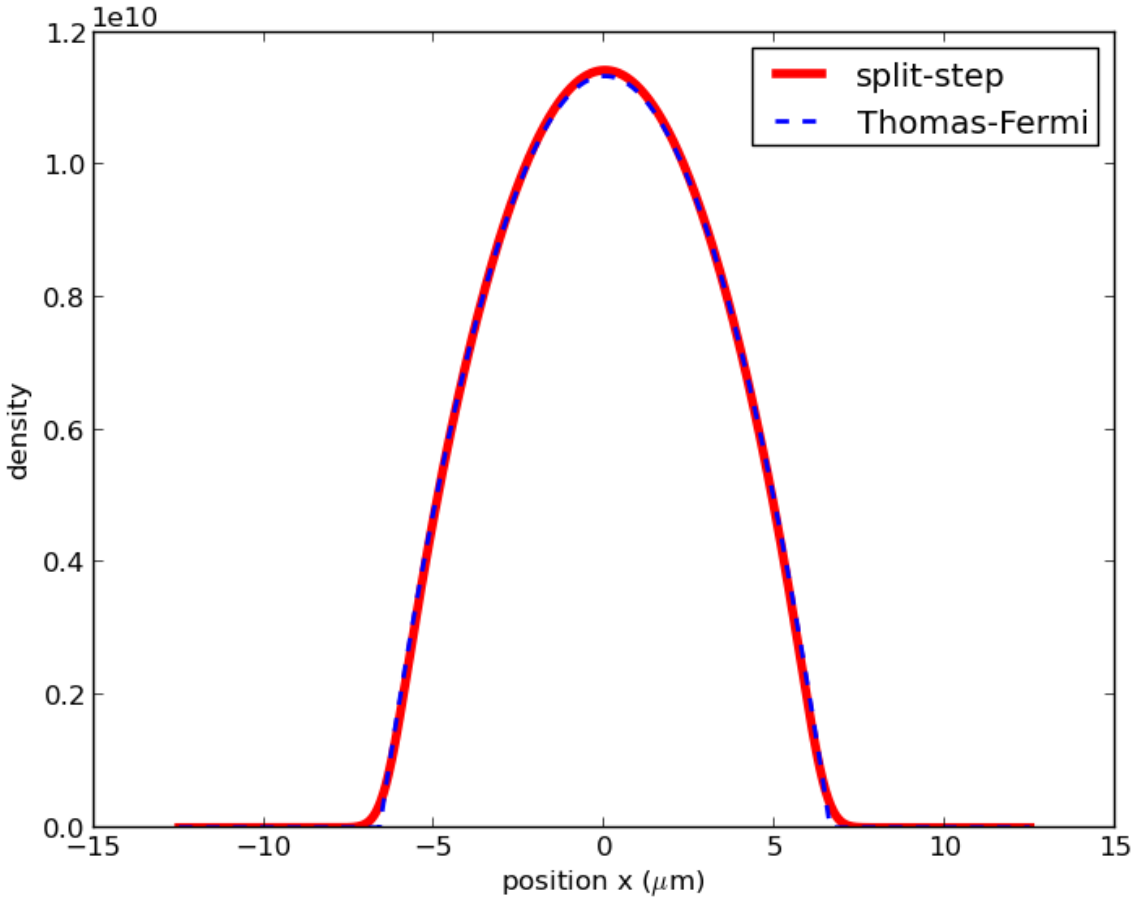
$$\left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x) + \frac{4\pi\hbar^2 a}{m}|\psi(x)|^2\right]\psi(x) = \mu\,\psi(x)$$

Thomas-Fermi approximation:

$$\left[V(x) + U_0|\psi(x)|^2\right]\psi(x) = \mu\psi(x)$$

$$n(x) = |\psi(x)|^2 = \frac{[\mu - V(x)]}{U_0}$$

Split-step


1e10
1.2
1.0
0.8
0.6
0.4
0.2
0.0
density
split-step
Thomas-Fermi
−15
−10
−5
0
5
10
15
position x (μm)

Problems, pitfalls, etc...

- Programming language

Problems, pitfalls, etc...

- Programming language

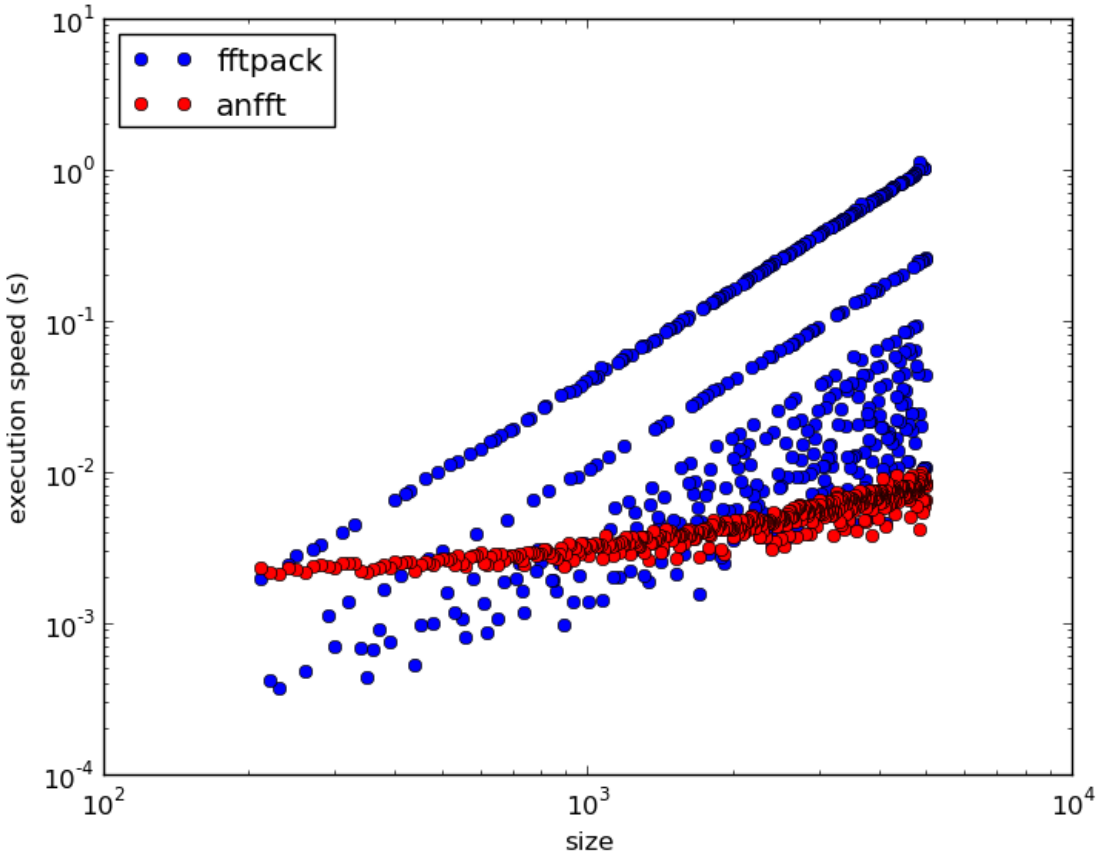
- FFT

Problems, pitfalls, etc...

- Programming language

- FFT
    - speed

Comparison: FFTPACK vs. FFTW



Problems, pitfalls, etc...

- Programming language

- FFT
  - speed
  - alignment

Problems, pitfalls, etc...

- Programming language

- FFT
  - speed
  - alignment

- scattering length

Scattering length in 1d & 2d

$$\frac{4\pi\hbar^2 a}{m}|\psi|^2$$

Scattering length in 1d & 2d

$$\frac{4\pi\hbar^2 a}{m}|\psi|^2$$

$$\begin{aligned} a_{1\mathrm{d}} &= \left(\frac{15Na}{\bar{a}}\right)^{2/5}\left(\frac{\hbar\bar{\omega}}{2}\right)^{3/2}\frac{\sqrt{2m}}{3\omega_x\hbar 2\pi N} \\ a_{2\mathrm{d}} &= \frac{5^{4/5}\left(\frac{aN}{\bar{a}}\right)^{2/5}\bar{\omega}^2}{2\cdot 3^{1/5}N\pi\omega_x\omega_y}, \end{aligned}$$