- Markdown
-
- Python
- IPython
- NumPy
- SciPy
- Matplotlib
- SymPy
- pandas
-
- About

Code

CellToolbar

In [1]:

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
import itertools
import copy

#basis of subspace of hilbert space with number of ones conserved
def basis_sub(total, ones):
    output = []
    for indices in itertools.combinations(range(total), ones):
        vector = [0] * total
        for index in indices:
            vector[index] = 1
        output.append(vector)
    return output

#XXX hamiltonian with closed boundary conditions and onsite random disorder
```

```
def hamiltonian(basis, delta, h):
    m = len(basis)
    n = len(basis[0])
    H = np.zeros((m,m))
    defects = np.random.uniform(-h, h, m)
    for i in range(m):
        for j in range(n):
            if basis[i][j] == 1:
                H[i][i] += defects[i]/2
            else:
                H[i][i] -= defects[i]/2
        for j in range(n-1):
            if basis[i][j] == basis[i][j+1]:
                H[i][i] += delta/4
            else:
                H[i][i] -= delta/4
                swapped = copy.copy(basis[i])
                swapped[j], swapped[j+1] = swapped[j+1], swapped[j]
                H[i][basis.index(swapped)] += .5
        if basis[i][-1] == basis[i][0]:
                H[i][i] += delta/4
        else:
            H[i][i] -= delta/4
            swapped = copy.copy(basis[i])
            swapped[-1], swapped[0] = swapped[0], swapped[-1]
            H[i][basis.index(swapped)] += .5
    return H
```

In [2]:

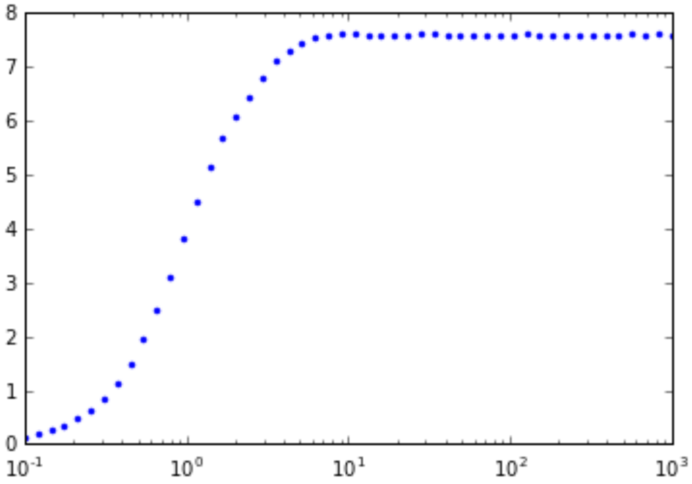
```
states = 10
disorders = 10

site_basis = basis_sub(14,7)
```

```
shannon_avg = np.zeros(50)
times = np.logspace(-1, 3, 50)
for alpha in range(disorders):
    print alpha
    H = hamiltonian(site_basis, 1., 2.5)
    eig = np.linalg.eigh(H)

    for beta in range(states):
        initial_index = np.random.randint(3432)
        initial_state = eig[1][initial_index]
        Wk = np.empty((50, 3432))
        for i in range(3432):
            C = eig[1][i]
            for l in range(50):
                t = times[l]
                Wk[l, i] = abs(np.sum(C*initial_state * np.exp(-1j*eig[0]*t)))**2 + 10**-20

        shannon = -np.sum(Wk*np.log(Wk), axis = 1)
        shannon_avg += shannon / (states*disorders)
```

```
0
1
2
3
4
5
6
7
8
9
```

In [4]:

```
xxxxxxxxxx


plt.semilogx(times, shannon_avg, '.')
plt.show()
```



In [ ]: