Project 1

Quantum Mechanics

October 30, 2022

C Seitz

Part 1

(A) We were given the Hamiltonian:

$$-t(\phi_{n,i+1} + \phi_{n,i-1}) + (2t + V_i)\phi_{n,i} = \epsilon_n \phi_{n,i}$$

which gives us a relationship between $\phi_{n,i}$ and the neighboring elements $\phi_{n,i-1}$ and $\phi_{n,i+1}$. The explicit matrix form is

$$\hat{H}_0 \phi_n = \begin{pmatrix} 2t+V_1 & -t & 0 & \dots \\ -t & 2t+V_2 & -t & \dots \\ 0 & -t & 2t+V_3 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \phi_{n,1} \\ \phi_{n,2} \\ \phi_{n,3} \\ \vdots \end{pmatrix} = \epsilon_n \begin{pmatrix} \phi_{n,1} \\ \phi_{n,2} \\ \phi_{n,3} \\ \vdots \end{pmatrix} \tag{1}$$

The full matrix $\hat{H}_0$ is shown in Figure 1a.

(B) From (1) we can see that the diagonal elements represent the discretized potential V_n (plus a constant $2t$ where $t = \frac{\hbar^2}{2ma^2}$). The off-diagonal elements are just constants with dimension of energy over length squared. The matrix of normalized eigenvectors of $\hat{H}_0$, which we will call T, is shown in Figure 1b.

(C) To show that the eigenvectors form an orthonormal set, We can define a matrix T such that each column of T is one eigenvector $\vec{\phi}_n$ of $\hat{H}_0$. If the eigenvectors are indeed orthonormal, then

$$T^T T = I$$

This product is shown in Figure 1c, and we can see that the eigenvectors are orthonormal.

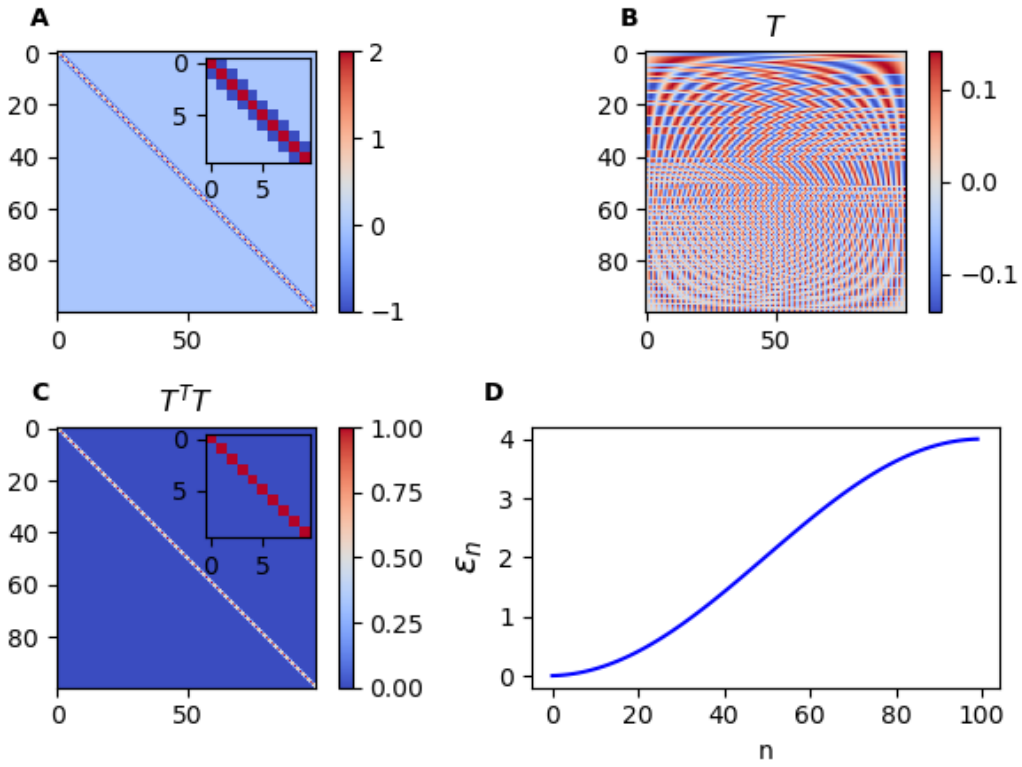

Figure 1: (A) The Hamiltonian H_0 (B) Eigenvectors as columns of a matrix (sorted by ascending eigenvalue) (C) Eigenvalue spectrum sorted in ascending order

(D) The sorted eigenvalues are shown in Figure 1d.

(E) Three example probability distributions are shown in Figure 2

(F) The standard quantum mechanics problem this corresponds to is the free particle. Schrodinger's equation for a free particle reads

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2} = E\psi$$

or

$$\frac{\partial^2\psi}{\partial x^2} = -k^2\psi$$

for $k = \frac{\sqrt{2mE}}{\hbar}$. So clearly the energy eigenvalues are $E_k = \hbar k^2/2m$. Notice that k is a continuous parameter and therefore there is a continuum of solutions to Schrodingers equation. One solution to the above equation is

$$\psi(x) = Ae^{ikx}$$

We would expect that the energy eigenvalues in Figure 1d would vary

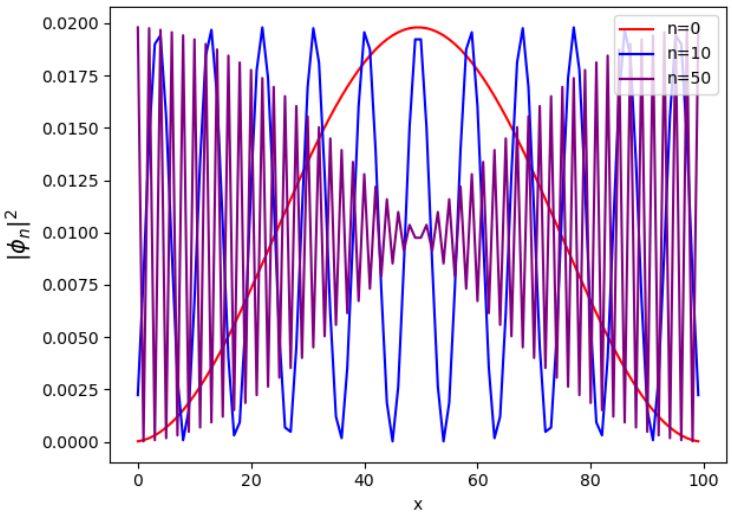

Figure 2: Energy eigenkets in the position representation for $n = 0, 10, 50$

quadratically in n; however, the curve has a more sigmoidal shape. Around $n = 50$, we can see that the eigenvalues are increasing more linearly. Moreover, the eigenvalue curve plateaus as $n \to 100$ because we have chosen a finite sampling frequency a, and higher energy solutions cannot be resolved.

(G) Consider the probability density shown in Figure 2 where $n = 50$. It is no coincidence that this value of n is exactly half of the number of samples $N = 100$. The Nyquist theorem says that to resolve a signal of frequency f we need a sampling rate of at least $2f$. Put another way, the highest frequency we can resolve, given some sampling rate, is half the sampling rate. Half the sampling rate is the folding frequency, which occurs at $n = 50$. Beyond the folding frequency, aliasing occurs and beats are observed.

(H) The unitary transformation that transforms $\hat{H}_0$ to the $|\phi_n\rangle$ basis is simply

$$\hat{H}_0' = U_0 H_0 U_0^{-1} = T^{-1} H_0 T$$

$\hat{H}_0'$ is shown in Figure 3a, and is diagonal.

(I) The values along the diagonal of $\hat{H}_0'$ are the energy eigenvalues

(J) The energy eigenvalues are shown in Figure 3b

(K) The energy eigenvalues are the same as they were before the change of basis. All we have done is changed our representation, so they should be.

(L) Three representative probability distributions are shown in Figure 3d.

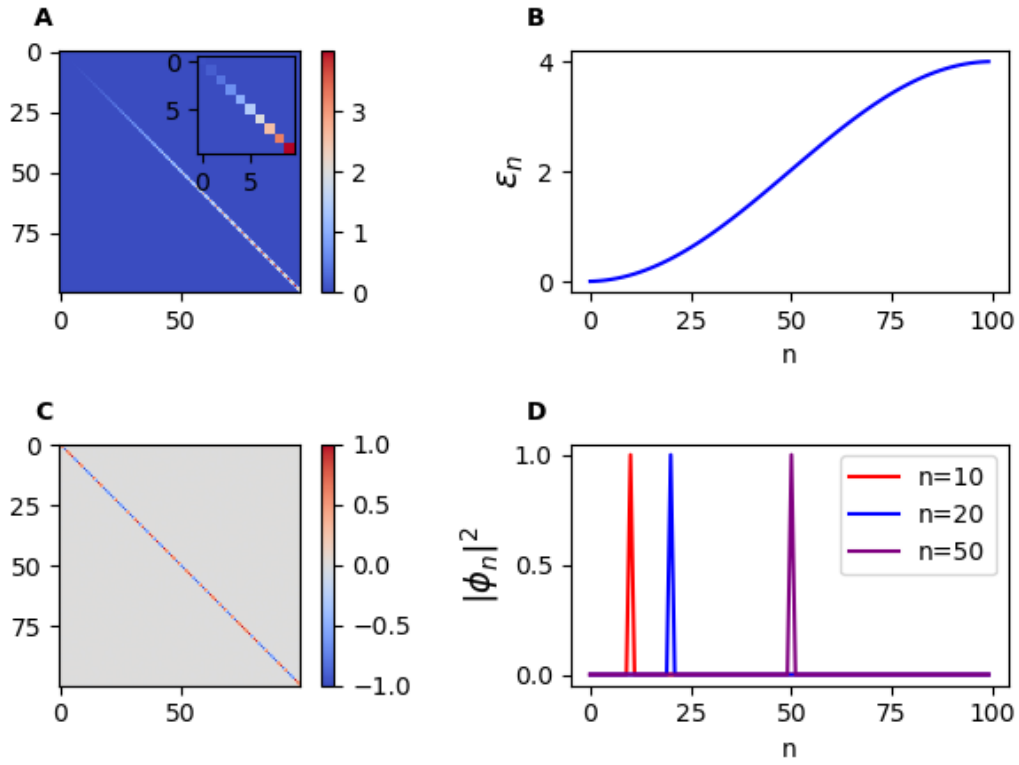

Figure 3: (A) The Hamiltonian $\hat{H}_0'$ after unitary transformation with $U_0 = T^{-1}$ (B) Eigenvalue spectrum sorted in ascending order (C) Eigenvectors as columns of a matrix (sorted by ascending eigenvalue) (D) Probability densities for a few eigenvectors in the energy basis

These are delta functions because we have changed to the energy basis.

Part 2

(M) The Hamiltonian matrix is shown in Figure 4a.

(N) $\hat{H}$ differs from $\hat{H}_0$ from zero to the 29th element and the 69th element to the 100th element along the diagonal. This is because we have set $V = V_L$ for $0 \leq x \leq 29a$ and $V = V_R$ for $69a \leq x \leq 100$. The matrix $\hat{H}$ is shown in Figure 4a, its sorted eigenvectors are shown in Figure 4b, and their corresponding eigenvalues, sorted in ascending order, are shown in Figure 4d.

(O) The energy eigenvalues for this Hamiltonian are shown in Figure 4d.

(P) Probability distributions for $n = 1, 25, 26, 35, 39, 41, 55$ are shown in Figure 5.

(Q) For $n = 0$ a particle is most likely to be in the region where $V = 0$,

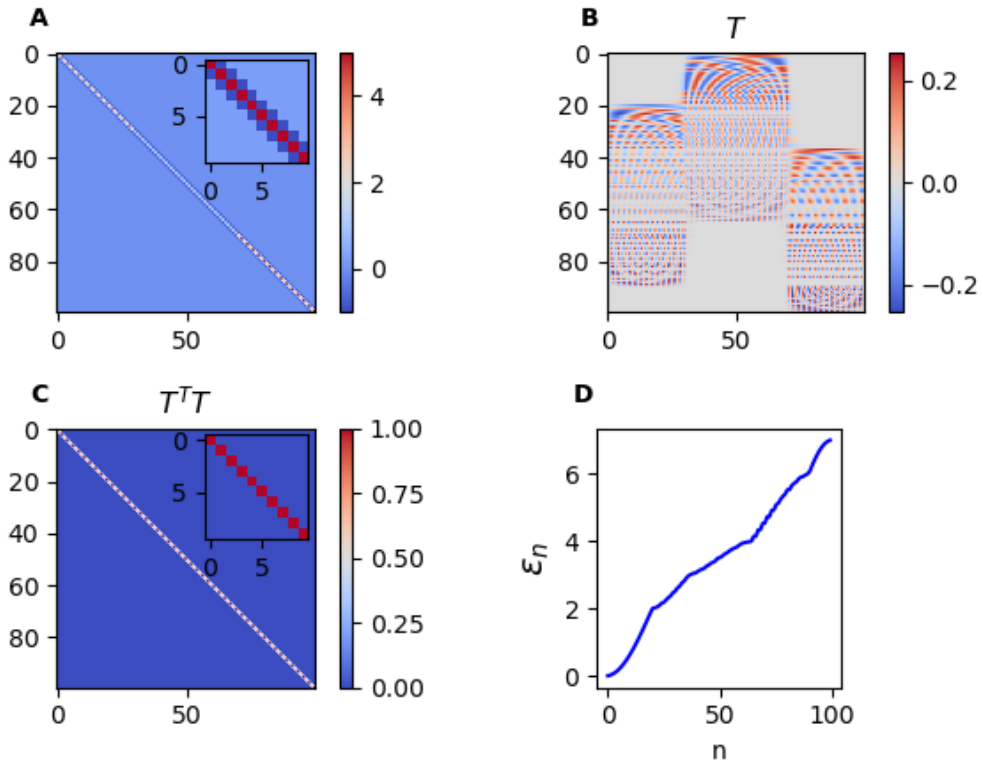

Figure 4: (A) The Hamiltonian H (B) Eigenvalue spectrum sorted in ascending order (C) Eigenvectors as columns of a matrix (sorted by ascending eigenvalue)

which makes sense because this is the ground state. As we increase the energy for $n = 24, 25, 34$, we see that the particle is no longer bound to the potential well ($E > V_L$), but it doesn't have enough energy to be found from $69a \leq x \leq 100$ where $V = V_R$ ($E < V_R$). So we see decaying exponentials there. Furthermore, for $n = 38, 40, 54, 55$ we see sinusoidal solutions in both regions $0 \leq x \leq 29a$ and $69a \leq x \leq 100$. Clearly the energy is then high enough for the particle to be found there ($E > V_R$).

(R) There are kinks in the energy eigenvalue plot because the nonzero potential modifies the Hilbert space with respect to the free particle Hilbert space. So we shouldn't expect a smoothly varying function.

(S) The matrix after unitary transformation is shown in Figure 6a.

(T) Before we changed basis, we had an eigenvalue equation for H. That means that after the change, if we hit certain superpositions of energy eigenkets of H_0 with this matrix, we satisfy the eigenvalue equation. So the matrix elements represent the components of that superposition that will satisfy the eigenvalue equation.

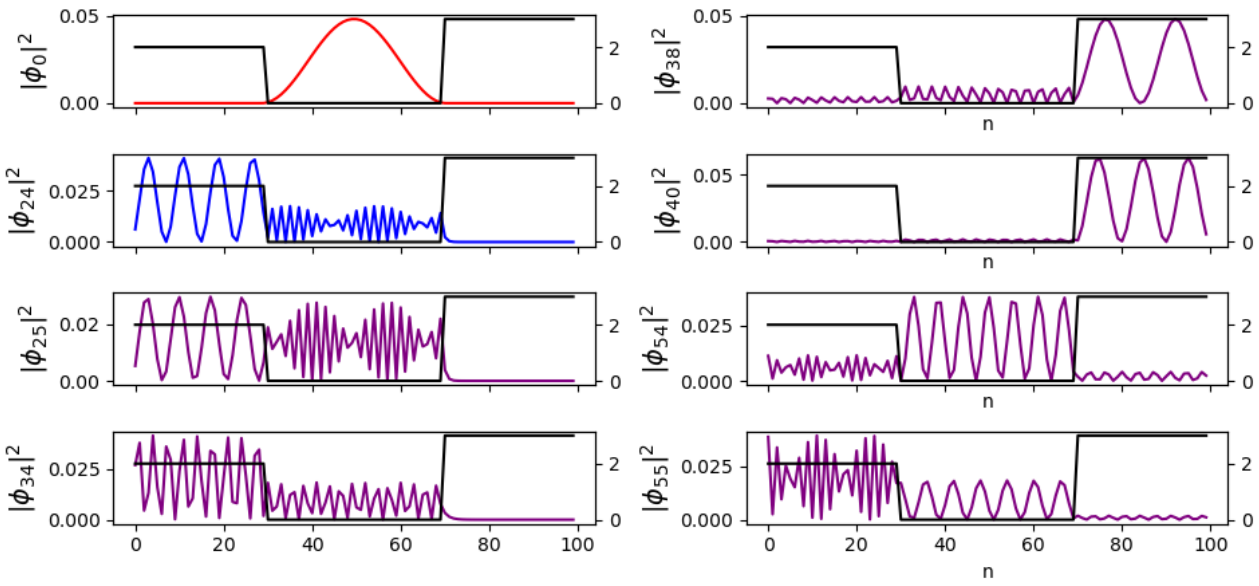

Figure 5: Probability densities for a few eigenkets of H (see main text for description of the figure)

(U) The eigenvalue plot for $U_0 H U_0^{-1}$ is the same as for H, as they should be. Again, we have changed our representation but nothing physical has changed.

(V) The probability distributions $|\langle\phi_{0,m}|\phi_n\rangle|^2$ for $n = 1, 25, 26, 35, 39, 41, 55$ are shown in Figure 7.

(W) Let $|\phi_n\rangle$ be an orthonormal set of energy eigenkets of $\hat{H}$ and $|\phi_m\rangle$ be an orthonormal set of eigenkets of $\hat{H}_0$. Then $\sum_m |\phi_m\rangle\langle\phi_m|\phi_n\rangle$ is the representation of $|\phi_n\rangle$ in the $|\phi_m\rangle$ basis. It follows that $|\langle\phi_m|\phi_n\rangle|^2$ is the norm squared of that representation. Ultimately, we see spikes because for certain m, because $\langle\phi_n|\phi_n\rangle$ has greater magnitude. One interpretation of this is that we couple together special solutions from the family of solutions to the free particle problem.

(X) The matrix of values $\langle\phi_m|\phi_n\rangle$ is shown in Figure 6c. Each column of this matrix is an eigenvector $|\phi_n\rangle$ in the $|\phi_m\rangle$ basis. We can see that the kets $|\phi_n\rangle$ are superpositions of the plane wave solutions to the free particle problem. This makes sense, because using Fourier analysis, we should be able to construct arbitrary wavefunctions using a basis consisting of fundamental harmonics. Again, by introducing a nonzero potential, we have effectively coupled together special solutions from the family of solutions to the free particle problem.

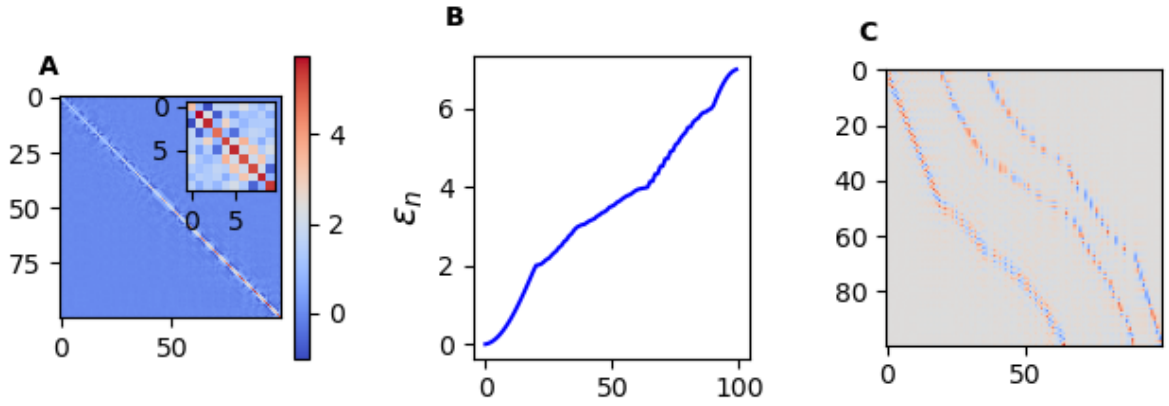

Figure 6: (A) The Hamiltonain after unitary transformation with U_0 (B) Eigenvalue spectrum sorted in ascending order (C) Eigenvectors as columns of a matrix (sorted by ascending eigenvalue)

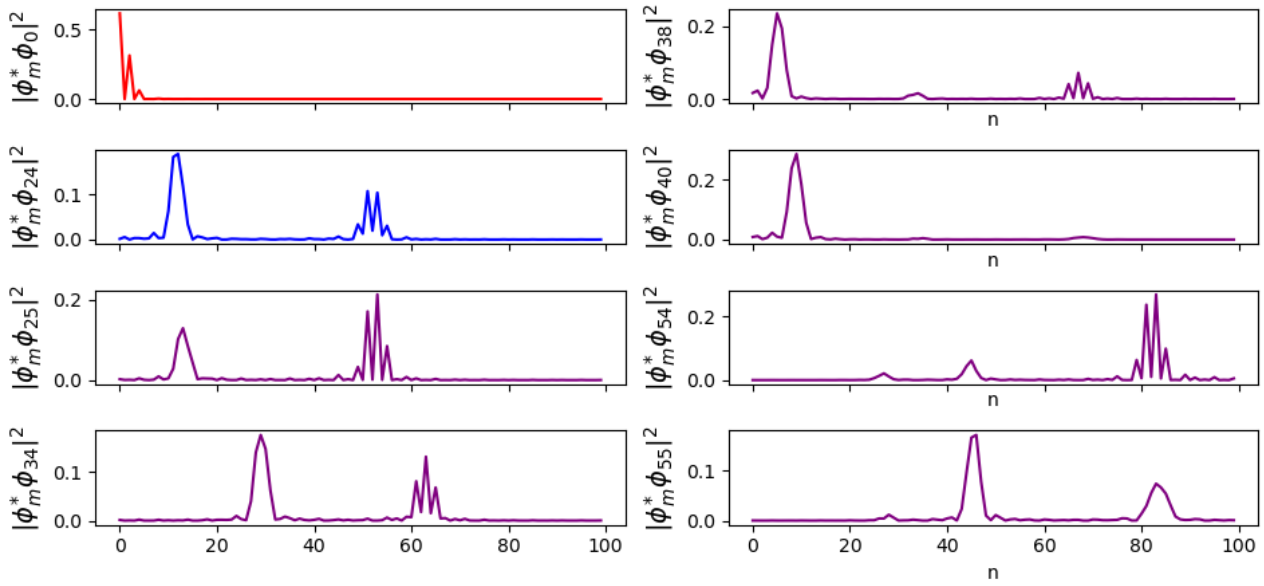

Figure 7: Coupling energy eigenkets by introducing a nonzero potential (see main text for description of the figure)

```
import numpy as np
import matplotlib.pyplot as plt
from numpy import linalg as LA

##########################################
# Part 1
##########################################

##########################################
# Define the Hamiltonian H0
##########################################

N = 100 #Number of samples
V = np.zeros((N,)) #Potential
H0 = np.zeros((N,N)) #Hamiltonian matrix H0
H0 += np.diag(2 + V,k=0) #main diagonal
H0 += np.diag(-1*np.ones((N-1,)),k=1) #upper diagonal
H0 += np.diag(-1*np.ones((N-1,)),k=-1) #lower diagonal

##########################################
# Find eigenvectors and eigenvalues of H0
##########################################

vals, vecs = LA.eig(H0)
idx = np.argsort(vals)
vecs = vecs[:,idx]
vals = vals[idx]

##########################################
# Show H0
##########################################

fig, ax = plt.subplots(2,2) #Figure 1
x = ax[0,0].imshow(H0,cmap='coolwarm')
axin1 = ax[0,0].inset_axes([0.575, 0.575, 0.4, 0.4])
y = axin1.imshow(H0[:10,:10],cmap='coolwarm')
plt.colorbar(x,ax=ax[0,0])

##########################################
# Show eigenvectors form an orthonormal set
##########################################

x = ax[0,1].imshow(vecs.T, cmap='coolwarm')
plt.colorbar(x,ax=ax[0,1])
ax[0,1].set_title(r'$T$')
X = vecs.T @ vecs
x = ax[1,0].imshow(X, cmap='coolwarm')
axin1 = ax[1,0].inset_axes([0.575, 0.575, 0.4, 0.4])
```

```
y = axin1.imshow(X[:10,:10],cmap='coolwarm')
plt.colorbar(x,ax=ax[1,0])
ax[1,0].set_title(r'$T^{T}T$')

##########################################
# Plot the eigenvalues as a function of n
##########################################


ax[1,1].plot(vals, color='blue')
ax[1,1].set_xlabel('n')
ax[1,1].set_ylabel(r'$\epsilon_{n}$',fontsize=14)
plt.tight_layout()
ax[0,0].text(-0.1, 1.1, 'A', transform=ax[0,0].transAxes,
           size=10, weight='bold')
ax[0,1].text(-0.1, 1.1, 'B', transform=ax[0,1].transAxes,
           size=10, weight='bold')
ax[1,0].text(-0.1, 1.1, 'C', transform=ax[1,0].transAxes,
           size=10, weight='bold')
ax[1,1].text(-0.1, 1.1, 'D', transform=ax[1,1].transAxes,
           size=10, weight='bold')
plt.show()

##########################################
# Plot a few representative prob dists
##########################################

fig, ax = plt.subplots() #Figure 2
pdf1 = vecs[:,0]**2
pdf2 = vecs[:,10]**2
pdf3 = vecs[:,50]**2

ax.plot(pdf1, color='red',label='n=0')
ax.plot(pdf2, color='blue',label='n=10')
ax.plot(pdf3, color='purple',label='n=50')
ax.set_xlabel('x')
ax.set_ylabel(r'$|\phi_{n}|^{2}$',fontsize=14)
ax.legend(loc='upper right')
plt.show()

##########################################
# Find projection operator to energy basis
##########################################

U0 = LA.inv(vecs)

##########################################
# Diagonalize H0
##########################################
```

```
Hd = U0 @ H0 @ LA.inv(U0)

fig, ax = plt.subplots(2,2) #Figure 3
x = ax[0,0].imshow(Hd,cmap='coolwarm')
axin1 = ax[0,0].inset_axes([0.575, 0.575, 0.4, 0.4])
y = axin1.imshow(Hd[:10,:10],cmap='coolwarm')
plt.colorbar(x,ax=ax[0,0])


##########################################
# Find eigenvectors and eigenvalues of H
##########################################

dvals, dvecs = LA.eig(Hd)
idx = np.argsort(dvals)
dvecs = dvecs[:,idx]
dvals = dvals[idx]

##########################################
# Plot the eigenvalues as a function of n
##########################################

x = ax[1,0].imshow(dvecs,cmap='coolwarm')
ax[0,1].plot(dvals, color='blue')
ax[0,1].set_xlabel('n')
ax[0,1].set_ylabel(r'$\epsilon_{n}$',fontsize=14)
plt.colorbar(x,ax=ax[1,0])


##########################################
# Plot a few representative prob dists
##########################################

pdf1 = dvecs[:,10]**2
pdf2 = dvecs[:,20]**2
pdf3 = dvecs[:,50]**2

ax[1,1].plot(pdf1, color='red',label='n=10')
ax[1,1].plot(pdf2, color='blue',label='n=20')
ax[1,1].plot(pdf3, color='purple',label='n=50')
ax[1,1].set_xlabel('n')
ax[1,1].set_ylabel(r'$|\phi_{n}|^{2}$',fontsize=14)
ax[1,1].legend()
ax[0,0].text(-0.1, 1.1, 'A', transform=ax[0,0].transAxes,
           size=10, weight='bold')
ax[0,1].text(-0.1, 1.1, 'B', transform=ax[0,1].transAxes,
           size=10, weight='bold')
ax[1,0].text(-0.1, 1.1, 'C', transform=ax[1,0].transAxes,
```

```
           size=10, weight='bold')
ax[1,1].text(-0.1, 1.1, 'D', transform=ax[1,1].transAxes,
           size=10, weight='bold')
plt.show()

##########################################
# Part 2
##########################################

##########################################
# Define the Hamiltonian H
##########################################

N = 100
Vl = 2
Vr = 3
t = 1
V = np.zeros((N,))
V[:30] = Vl; V[70:] = Vr
H = np.zeros((N,N))
H += np.diag(2*t + V,k=0) #main diagonal
H += np.diag(-t*np.ones((N-1,)),k=1) #upper diagonal
H += np.diag(-t*np.ones((N-1,)),k=-1) #lower diagonal


##########################################
# Find eigenvectors and eigenvalues of H
##########################################

vals, vecs = LA.eig(H)
idx = np.argsort(vals)
vecs = vecs[:,idx]
vals = vals[idx]

##########################################
# Show the Hamiltonian matrix
##########################################

fig, ax = plt.subplots(2,2) #Figure 4
x = ax[0,0].imshow(H,cmap='coolwarm')
axin1 = ax[0,0].inset_axes([0.575, 0.575, 0.4, 0.4])
y = axin1.imshow(H[:10,:10],cmap='coolwarm')
plt.colorbar(x,ax=ax[0,0])


##########################################
# Show eigenvectors form an orthonormal set
##########################################
```

```
x = ax[0,1].imshow(vecs.T, cmap='coolwarm')
plt.colorbar(x,ax=ax[0,1])
ax[0,1].set_title(r'$T$')
X = vecs.T @ vecs
x = ax[1,0].imshow(X, cmap='coolwarm')
axin1 = ax[1,0].inset_axes([0.575, 0.575, 0.4, 0.4])
y = axin1.imshow(X[:10,:10],cmap='coolwarm')
plt.colorbar(x,ax=ax[1,0])
ax[1,0].set_title(r'$T^{T}T$')
##########################################
# Plot the eigenvalues as a function of n
##########################################

ax[1,1].plot(vals, color='blue')
ax[1,1].set_xlabel('n')
ax[1,1].set_ylabel(r'$\epsilon_{n}$',fontsize=14)
ax[1,1].set_aspect(15)
ax[0,0].text(-0.1, 1.1, 'A', transform=ax[0,0].transAxes,
          size=10, weight='bold')
ax[0,1].text(-0.1, 1.1, 'B', transform=ax[0,1].transAxes,
          size=10, weight='bold')
ax[1,0].text(-0.1, 1.1, 'C', transform=ax[1,0].transAxes,
          size=10, weight='bold')
ax[1,1].text(-0.1, 1.1, 'D', transform=ax[1,1].transAxes,
          size=10, weight='bold')
plt.show()

##########################################
# Plot a few representative prob dists
##########################################

pdf1 = vecs[:,0]**2
pdf2 = vecs[:,24]**2
pdf3 = vecs[:,25]**2
pdf4 = vecs[:,34]**2
pdf5 = vecs[:,38]**2
pdf6 = vecs[:,40]**2
pdf7 = vecs[:,54]**2
pdf8 = vecs[:,55]**2

fig, ax = plt.subplots(4,2,sharex=True,figsize=(12,4)) #Figure 5
ax[0,0].plot(pdf1, color='red',label='m=0')
ax1 = ax[0,0].twinx()
ax1.plot(V,color='black',label='V(x)')
ax[1,0].plot(pdf2, color='blue',label='m=24')
ax1 = ax[1,0].twinx()
ax1.plot(V,color='black',label='V(x)')
ax[2,0].plot(pdf3, color='purple',label='m=25')
ax1 = ax[2,0].twinx()
```

```
ax1.plot(V,color='black',label='V(x)')
ax[3,0].plot(pdf4, color='purple',label='m=34')
ax1 = ax[3,0].twinx()
ax1.plot(V,color='black',label='V(x)')
ax[0,1].plot(pdf5, color='purple',label='m=38')
ax1 = ax[0,1].twinx()
ax1.plot(V,color='black',label='V(x)')
ax[1,1].plot(pdf6, color='purple',label='m=40')
ax1 = ax[1,1].twinx()
ax1.plot(V,color='black',label='V(x)')
ax[2,1].plot(pdf7, color='purple',label='m=54')
ax1 = ax[2,1].twinx()
ax1.plot(V,color='black',label='V(x)')
ax[3,1].plot(pdf8, color='purple',label='m=55')
ax1 = ax[3,1].twinx()
ax1.plot(V,color='black',label='V(x)')
ax[0,1].set_xlabel('n')
ax[1,1].set_xlabel('n')
ax[2,1].set_xlabel('n')
ax[3,1].set_xlabel('n')
ax[0,0].set_ylabel(r'$|\phi_{0}|^{2}$',fontsize=14)
ax[0,1].set_ylabel(r'$|\phi_{38}|^{2}$',fontsize=14)
ax[1,0].set_ylabel(r'$|\phi_{24}|^{2}$',fontsize=14)
ax[1,1].set_ylabel(r'$|\phi_{40}|^{2}$',fontsize=14)
ax[2,0].set_ylabel(r'$|\phi_{25}|^{2}$',fontsize=14)
ax[2,1].set_ylabel(r'$|\phi_{54}|^{2}$',fontsize=14)
ax[3,0].set_ylabel(r'$|\phi_{34}|^{2}$',fontsize=14)
ax[3,1].set_ylabel(r'$|\phi_{55}|^{2}$',fontsize=14)

plt.tight_layout()
plt.show()


##########################################
# Apply U0 to H
##########################################

H_ = U0 @ H @ LA.inv(U0)

##########################################
# Show the result
##########################################

fig, ax = plt.subplots(1,3) #Figure 6
x = ax[0].imshow(H_,cmap='coolwarm')
axin1 = ax[0].inset_axes([0.575, 0.575, 0.4, 0.4])
y = axin1.imshow(H_[:10,:10],cmap='coolwarm')
plt.colorbar(x,ax=ax[0])
```

```
##########################################
# Find eigenvectors and eigenvalues of H
##########################################

vals, vecs = LA.eig(H_)
idx = np.argsort(vals)
vecs = vecs[:,idx]
vals = vals[idx]

##########################################
# Plot the eigenvalues as a function of n
##########################################

ax[1].plot(vals, color='blue')
ax[1].set_xlabel('n')
ax[1].set_ylabel(r'$\epsilon_{n}$',fontsize=14)
ax[2].imshow(vecs,cmap='coolwarm')
ax[0].text(-0.1, 1.1, 'A', transform=ax[0].transAxes,
           size=10, weight='bold')
ax[1].text(-0.1, 1.1, 'B', transform=ax[1].transAxes,
           size=10, weight='bold')
ax[2].text(-0.1, 1.1, 'C', transform=ax[2].transAxes,
           size=10, weight='bold')
plt.show()

##########################################
# Plot a few representative prob dists
##########################################

pdf1 = vecs[:,0]**2
pdf2 = vecs[:,24]**2
pdf3 = vecs[:,25]**2
pdf4 = vecs[:,34]**2
pdf5 = vecs[:,38]**2
pdf6 = vecs[:,40]**2
pdf7 = vecs[:,54]**2
pdf8 = vecs[:,55]**2

fig, ax = plt.subplots(4,2,sharex=True,figsize=(12,4)) #Figure 7
ax[0,0].plot(pdf1, color='red',label='m=0')
ax[1,0].plot(pdf2, color='blue',label='m=24')
ax[2,0].plot(pdf3, color='purple',label='m=25')
ax[3,0].plot(pdf4, color='purple',label='m=34')
ax[0,1].plot(pdf5, color='purple',label='m=38')
ax[1,1].plot(pdf6, color='purple',label='m=40')
ax[2,1].plot(pdf7, color='purple',label='m=54')
ax[3,1].plot(pdf8, color='purple',label='m=55')
ax[0,1].set_xlabel('n')
```

```
ax[1,1].set_xlabel('n')
ax[2,1].set_xlabel('n')
ax[3,1].set_xlabel('n')
ax[0,0].set_ylabel(r'$|\phi_{m}^{*}\phi_{0}|^{2}$',fontsize=14)
ax[0,1].set_ylabel(r'$|\phi_{m}^{*}\phi_{38}|^{2}$',fontsize=14)
ax[1,0].set_ylabel(r'$|\phi_{m}^{*}\phi_{24}|^{2}$',fontsize=14)
ax[1,1].set_ylabel(r'$|\phi_{m}^{*}\phi_{40}|^{2}$',fontsize=14)
ax[2,0].set_ylabel(r'$|\phi_{m}^{*}\phi_{25}|^{2}$',fontsize=14)
ax[2,1].set_ylabel(r'$|\phi_{m}^{*}\phi_{54}|^{2}$',fontsize=14)
ax[3,0].set_ylabel(r'$|\phi_{m}^{*}\phi_{34}|^{2}$',fontsize=14)
ax[3,1].set_ylabel(r'$|\phi_{m}^{*}\phi_{55}|^{2}$',fontsize=14)


plt.tight_layout()
plt.show()
```