

Project 1

Clayton Seitz
October 27, 2022

1 PART I

Here, we are trying to solve for the solutions to Schrodinger's eigenvalue equation:

$$\hat{H}_0 \phi_n = \epsilon_n \phi_n$$

By discretizing ϕ_n , each ϕ_n becomes a finite dimensional vector and we can write \hat{H} explicitly as a matrix. That matrix satisfies

$$\sum_j \langle i | \hat{H}_0 | j \rangle \vec{\phi}_{n,j} = \epsilon_n \vec{\phi}_n$$

where $\langle i | \hat{H}_0 | j \rangle$ is the matrix element $[H_0]_{ij}$. It was shown the Schrodinger's wave equation could be expressed in discrete form, as

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

which gives us a relationship between $\phi_{n,i}$ and the neighboring elements $\phi_{n,i-1}$ and $\phi_{n,i+1}$. The eigenvalues equation can then be written as a matrix multiplication

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

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

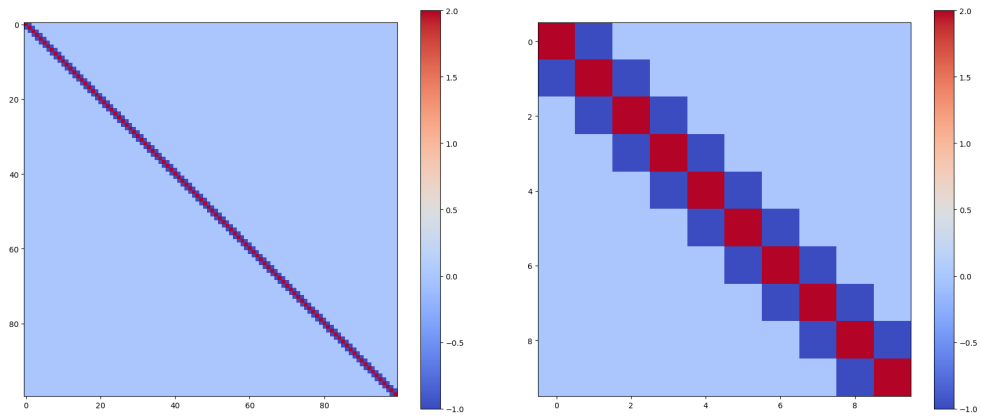


Figure 1.1:

$$T^T T = I_{N \times N}$$

These are the solutions to a free particle. The analytical eigenvalues of that problem should be

1.1 SOURCE CODE

```
1 import numpy as np
2
3 def incmatrix(genl1, genl2):
4     m = len(genl1)
5     n = len(genl2)
```

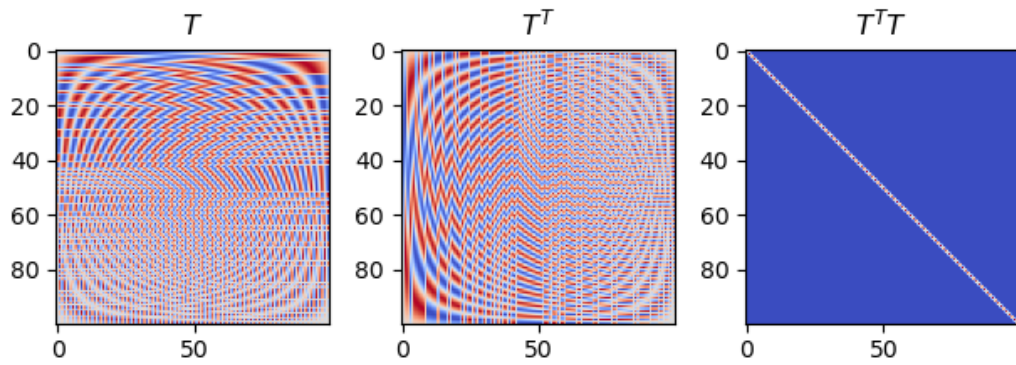


Figure 1.2:

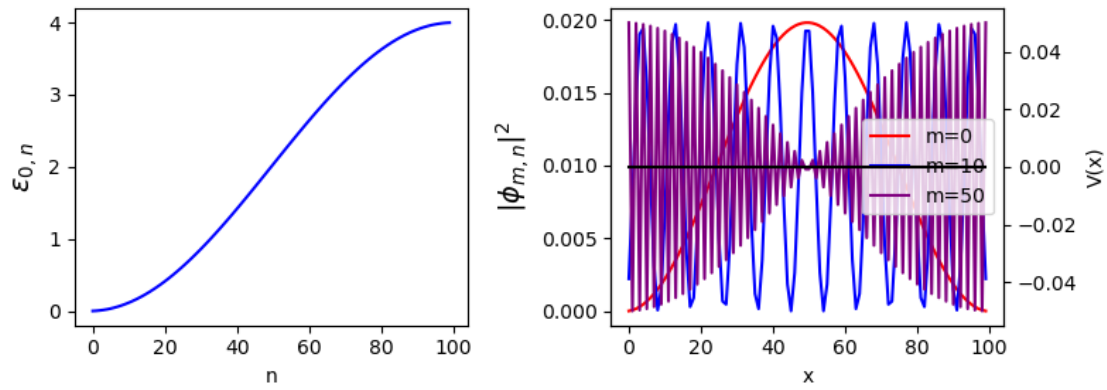


Figure 1.3:

```

6   M = None #to become the incidence matrix
7   VT = np.zeros((n*m,1), int) #dummy variable
8
9   #compute the bitwise xor matrix
10  M1 = bitxormatrix(genl1)
11  M2 = np.triu(bitxormatrix(genl2),1)
12
13  for i in range(m-1):
14      for j in range(i+1, m):
15          [r,c] = np.where(M2 == M1[i,j])
16          for k in range(len(r)):
17              VT[(i)*n + r[k]] = 1;
18              VT[(i)*n + c[k]] = 1;
19              VT[(j)*n + r[k]] = 1;
20              VT[(j)*n + c[k]] = 1;
21
22      if M is None:
23          M = np.copy(VT)
24      else:
25          M = np.concatenate((M, VT), 1)
26
27      VT = np.zeros((n*m,1), int)
28
29  return M

```

2 PART II