Solving 3D Time Independent Schrodinger Equation Using Numerical Method

Wong Wai Kui, Chris*

(Dated: December 23, 2021)

The goal of this project is to solve 3D time independent Schrodinger equations using the numerical method base on my 1D and 2D solver. The finite difference method is used to find out the eigenvalues and eigenvectors from the time independent Schrodinger equation for an arbitrary potential energy function. For each case, the potential function, eigenvector and eigenvalue will be drawn. Different potential functions will be used to in this project, for example 3D infinite square wall potential, 3D simple harmonic oscillator potential, 3D circular well, 3D torus potential, hydrogen atom potential. The main language of the code will be written using Python3. The main python library used in this project are Numpy, Scipy, Matplotlib.

I. INTRODUCTION

1D time independent Schrodinger equation is a well-known problem, which is an eigenvalue and eigenvector problem. It can be solved analytical and numerically. In numerical method, one of the methods is using the finite difference method.[1]

$$-\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2}+V\left(x\right)\psi=E\psi \tag{1}$$

For finite difference scheme, the approximation we used for the term $\frac{\partial^2\psi}{\partial x^2}$ is three-point stencil. It will turn the term $\frac{\partial^2\psi}{\partial x^2}$ form continuous wave function to discrete wave function.

$$\frac{\partial^2\psi}{\partial x^2}=\frac{\psi_{i+1}-2\psi_i+\psi_{i-1}}{\Delta x^2} \tag{2}$$

After that, the 1D time independent Schrodinger equation can be rewritten as

$$\begin{bmatrix} \frac{1}{\Delta y^2}+L^2mV_1 & -\frac{1}{2\Delta y^2} & 0 & 0\dots \\ -\frac{1}{2\Delta y^2} & \frac{1}{\Delta y^2}+L^2mV_2 & -\frac{1}{2\Delta y^2} & 0\dots \\ \dots & \dots & \dots & -\frac{1}{2\Delta y^2} \\ \dots 0 & 0 & -\frac{1}{2\Delta y^2} & \frac{1}{\Delta y^2}+L^2mV_{N-1} \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \dots \\ \psi_{N-1} \end{bmatrix} = L^2mE \begin{bmatrix} \psi_1 \\ \psi_2 \\ \dots \\ \psi_{N-1} \end{bmatrix} \tag{3}$$

Where the kinetic energy operator is

$$-\frac{1}{2\Delta y^2}\begin{bmatrix} -2 & 1 & 0 & \dots & \dots & 0 \\ 1 & -2 & 1 & \dots & \dots & \vdots \\ 0 & 1 & -2 & 1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 1 \\ 0 & \dots & \dots & 0 & 1 & -2 \end{bmatrix} \tag{4}$$

And the potential energy operator is

$$L^2m\begin{bmatrix} V_1 & 0 & 0 & \dots & \dots & 0 \\ 0 & V_2 & 0 & \dots & \dots & \vdots \\ 0 & 0 & V_3 & 0 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 0 \\ 0 & \dots & \dots & 0 & 0 & V_{N-1} \end{bmatrix} \tag{5}$$

The problem now become a matrix problem, and it can be solved using computer.[2]

II. 2D TIME INDEPENDENT SCHRODINGER EQUATION

The 2D time independent Schrodinger equation can be written as

$$-\frac{\hbar^2}{2m}\left(\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial y^2}\right)+V\left(x,y\right)\psi=E\psi \tag{6}$$

The state representation in 1D can be written as

$$\begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{N-1} \end{bmatrix} \tag{7}$$

However, the state representation in 1D is different from the state representation in 2D, is a grid.

* kuchrisuk@gmail.com

$$\begin{bmatrix} \psi_{11} & \psi_{12} & \cdots & \psi_{1N} \\ \psi_{21} & \psi_{22} & \cdots & \psi_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{N1} & \psi_{N2} & \cdots & \psi_{NN} \end{bmatrix} \tag{8}$$

It is necessary for us to turn the $N \times N$ matrix into a vector with length N^2 so that Scipy can read this. To do so,

```
resharp()
```

function is used.

$$\begin{bmatrix} \psi_{11} \\ \vdots \\ \psi_{1N} \\ \psi_{21} \\ \vdots \\ \psi_{2N} \\ \vdots \\ \psi_{NN} \end{bmatrix} \tag{9}$$

To Deal with the terms of $\frac{\partial^2 \psi}{\partial x^2}$ and $\frac{\partial^2 \psi}{\partial y^2}$, one of the methods is to trade the problem as a discrete Laplacian on a regular grid.

Let D is matrix where

$$D = \begin{bmatrix} -2 & 1 & 0 & \dots & \dots & 0 \\ 1 & -2 & 1 & \dots & \dots & \vdots \\ 0 & 1 & -2 & 1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 1 \\ 0 & \dots & \dots & 0 & 1 & -2 \end{bmatrix} \tag{10}$$

$$L = D_{xx} \oplus D_{yy} \tag{11}$$

$$= D_{xx} \otimes I + I \otimes D_{yy} \tag{12}$$

Where L is the Laplacian, I is the identity matrix. The Laplacian is the Kronecker sum of two sparse matrices D_{xx} and D_{yy}. The relation between equation 11 and equation 12 can be shown.[3]

Assume that

$$D\Psi = \frac{\partial^2 \psi}{\partial x^2} \tag{13}$$

To find the term $\frac{\partial^2 \psi}{\partial y^2}$, we can say x and y are swapped that Ψ^T represent ψ. Therefore,

$$\frac{\partial^2 \psi}{\partial y^2} = D\Psi^T \tag{14}$$

Since matrix D is symmetric,

$$D = D^T \tag{15}$$

Therefore,

$$D\Psi^T = (\Psi D)^T \tag{16}$$

In the end,

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = D_{xx} \oplus D_{yy} \tag{17}$$

$$= D_{xx} \otimes I + I \otimes D_{yy} \tag{18}$$

The term $D_{xx} \otimes I$ can be written as

$$D_{xx} \otimes I = \frac{1}{\Delta x^2} \begin{bmatrix} -2I & I & 0 & \dots & \dots & 0 \\ I & -2I & I & \dots & \dots & \vdots \\ 0 & I & -2I & I & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & I \\ 0 & \dots & \dots & 0 & I & -2I \end{bmatrix} \tag{19}$$

Where $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{bmatrix}$. The term $I \otimes D_{yy}$ can be written as

$$I \otimes D_{yy} = \begin{bmatrix} D & 0 & 0 & \dots & \dots & 0 \\ 0 & D & 0 & \dots & \dots & \vdots \\ 0 & 0 & D & 0 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 0 \\ 0 & \dots & \dots & 0 & 0 & D \end{bmatrix} \tag{20}$$

The potential energy is also a grid, we can write down the matrix form of this

$$VI = \begin{bmatrix} V_{11} & 0 & 0 & \dots & \dots & 0 \\ 0 & V_{12} & 0 & \dots & \dots & \vdots \\ 0 & 0 & V_{13} & 0 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & & 0 \\ \vdots & \vdots & \vdots & & \ddots & 0 \\ 0 & \dots & \dots & 0 & 0 & V_{NN} \end{bmatrix} \tag{21}$$

Using the above equation, the 2D time independent Schrodinger equation become

$$\left[-\frac{\hbar^2}{2m}(D \oplus D) + VI\right]\psi = E\psi \tag{22}$$

The dimension of the E is $N^2 \times N^2$ and N^2 for ψ. For $\hbar$=1,

$$\left[-\frac{1}{2}(D \oplus D) + m\Delta x^2 VI\right]\psi = m\Delta x^2 E\psi \tag{23}$$

III. 3D TIME INDEPENDENT SCHRODINGER EQUATION

The 3D time independent Schrodinger equation can be written as

$$-\frac{\hbar^2}{2m}\left(\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} + \frac{\partial^2\psi}{\partial z^2}\right) + V(x, y, z)\psi = E\psi \tag{24}$$

It is not difficult to see that the 3D equation can be rewritten as

$$\left[-\frac{1}{2}(D \oplus D \oplus D) + m\Delta x^2 VI\right]\psi = m\Delta x^2 E\psi \tag{25}$$

where

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} + \frac{\partial^2\psi}{\partial z^2} = D_{xx} \oplus D_{yy} \oplus D_{zz} \tag{26}$$

$$= (D_{xx} \otimes I + I \otimes D_{yy}) \oplus D_{zz} \tag{27}$$

$$= D_{xx} \otimes I \otimes I + I \otimes D_{yy} \otimes I + I \otimes I \otimes D_{zz} \tag{28}$$

A. Explanation of the python coding

Meshgrid

The concept of this code is to create a 3D mesh grid with finite size N. N is the number of elements of the length of the 3D mesh grid. The larger the N is, the longer the time spend, the more accurate we have. To do so, we can write the code as

```
N = 50
L = 8
X,Y,Z= np.meshgrid(np.linspace(-L/2,L/2,N,
       dtype=float),
np.linspace(-L/2,L/2,N, dtype=float),
np.linspace(-L/2,L/2,N, dtype=float))
```

Here, the L is the dimensionless quantity so that the potential function can be easily defined.

This mesh grid will give the whole function space with no property and no information. In 3D, we will acquire an eigenvector for different positions X, Y, Z.

Therefore, it will be a 4D plot. The information of eigenvector and probability density will be shown using the colour mapping, which will be used to fill into the empty mesh grid and give the mesh grid a property.

3D State Representation

The state representation in 3D is a rank-3 tensor containing $N \times N \times N$ elements. Therefore, we turn the cube matrix into a column matrix with length N^3.

Energy Operator

The kinetic energy operator is $-\frac{1}{2}(D \oplus D \oplus D)$. To define the energy operator, we need to create a matrix. First, we use

```
diag = np.ones([N])
diags = np.array([diag, -2*diag, diag])
```

to create an array that contains N element of 1 only and the create a must-array. After that, we put this muti-array to the diagonal of the $N \times N$ matrix

```
D = sparse.spdiags(diags, np.array([-1, 0,1]),
       N, N)
```

We can also define the identity matrix using Numpy.

```
I = np.identity(N)
```

Calculating Kronecker product or Kronecker sum is not difficult using Scipy.

```
T = -1/2 * (np.kron(np.kron(D,I),I)
```

```
+ np.kron(np.kron(I,D),I)
+ np.kron(np.kron(I,I),D))
```

Instead of making a long statement, we can use

```
D1 = sparse.kronsum(D,D)
D2 = sparse.kronsum(D1,D)
```

calculating the Kronecker product spend most of the time in this program, the calculation time is proportional to the number of elements in the matrix. In the case of 3D with $N = 50$, there will be 125000 elements. For $N = 100$, there will be 1000000 elements. The time will be exponentially increased. If possible, I will try to involve GPU calculation for this part to accelerate the time.

The potential energy operator VI now have N^3 elements in the diagonal of the matrix. In python, we can define the potential term as

```
U = sparse.diags(V.reshape(N**3),(0))
```

In the end, the Hamiltonian matrix can be define as

```
H = T+U
```

Solving for eigenvector and eigenvalue

To solve the eigenvalue problem, we can use iterative routine eigsh from scipy.sparse.linalg. We only care about the lowest eigenvalues, which the 'SM' is chosen. The k is the number of the row we want to calculate, which is the number of states.[4]

```
eigenvalues , eigenvectors = eigsh(H, k=6,
    which='SM')
def get_e(n):
  return eigenvectors.T[n].reshape((N,N,N))
```

B. Computing Setup

In this project, a laptop with Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz, 16GB ram is used, no GPU acceleration is used. The operating system is 64-bit Window 10.

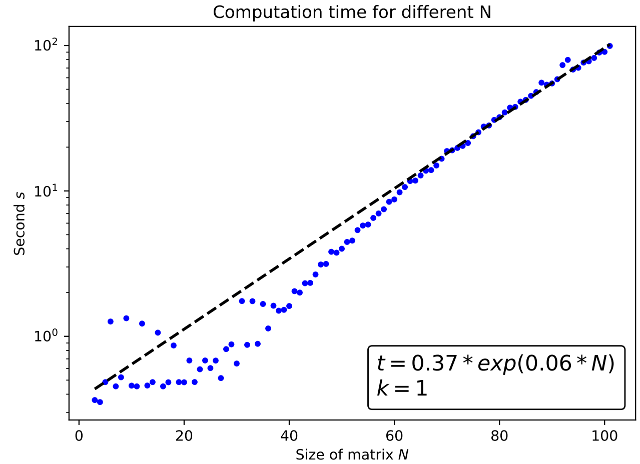

FIG. 1. Computing time for different N, $k = 1$. For $N = 50$, $t = 7.43s$. For $N = 100$, $t = 149.26s$.

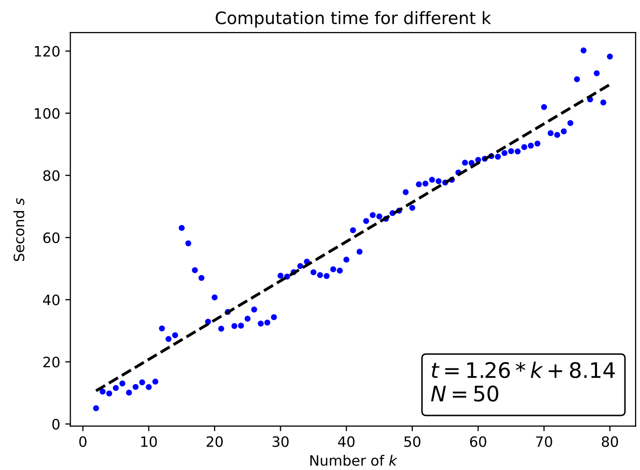

FIG. 2. Computing time for different k, $k = 50$. For $k = 50$, $t = 71.14s$. For $k = 100$, $t = 134.14s$.

From Figures 1 and 2, the computation time exponentially increases as the function of the size of matrix N whereas the computation time linearly increases as the function of the number of k.

IV. RESULT

A. 3D infinite square wall potential

The 1st case I did is the 3D infinite square wall potential. The wavefunction is confined in $N \times N \times N$ 3D space.[5][6]

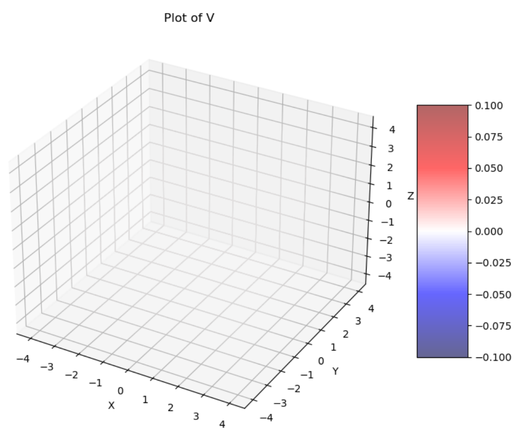

FIG. 3. Plot of the 3D infinite square wall potential.

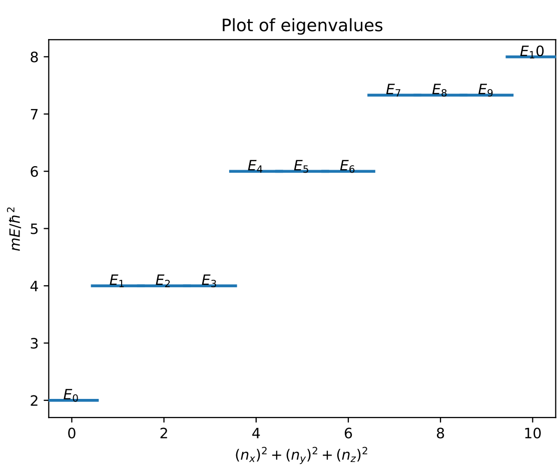

FIG. 4. Plot of the eigenvalues from ground state to 9th state.

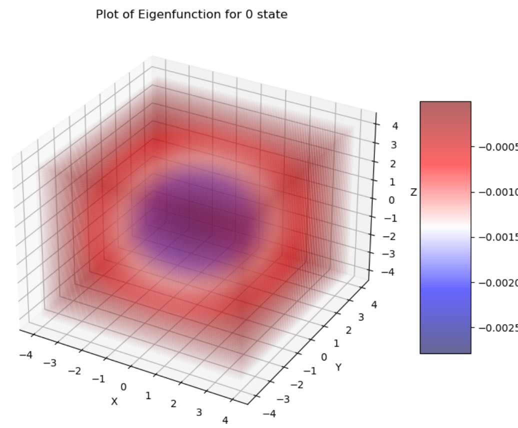

FIG. 5. Plot of the eigenfunctions for ground state.

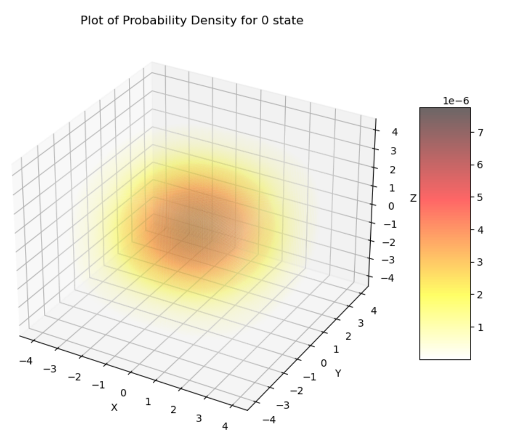

FIG. 6. Plot of the probability density for ground state.

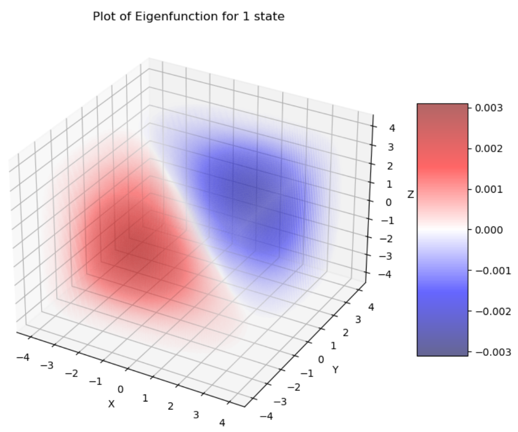

FIG. 7. Plot of the eigenfunctions for 1st state.

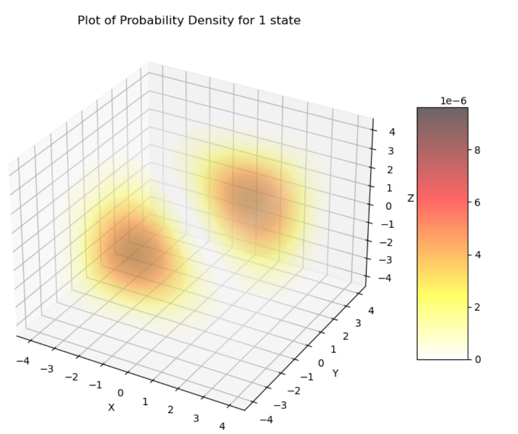

FIG. 8. Plot of the probability density for 1st state.

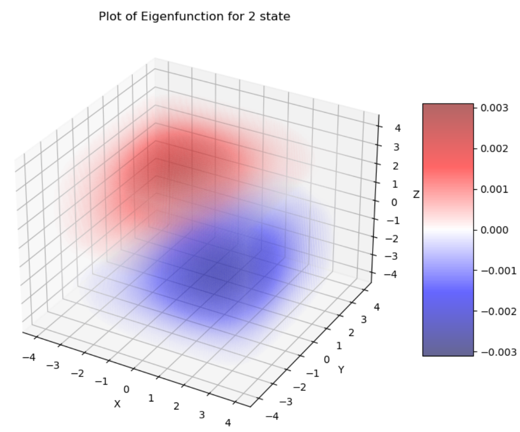

FIG. 9. Plot of the eigenfunctions for 2nd state.

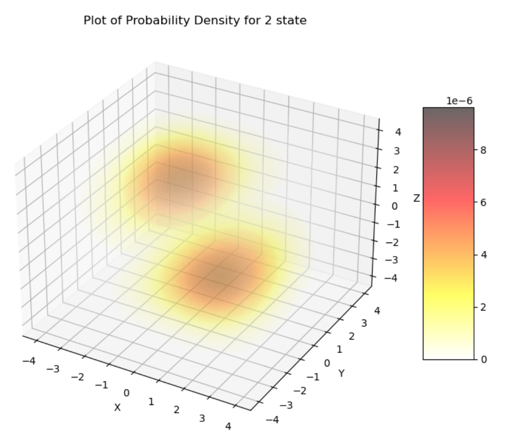

FIG. 10. Plot of the probability density for 2nd state.

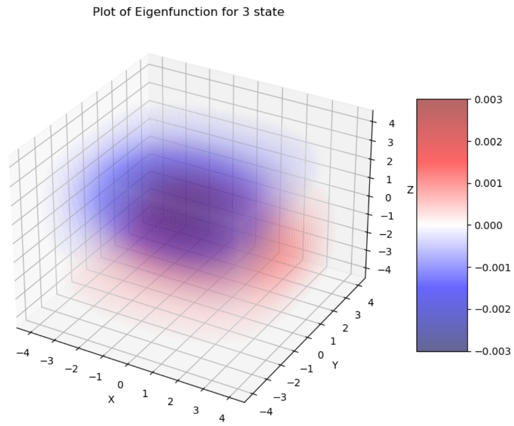

FIG. 11. Plot of the eigenfunctions for 3rd state.

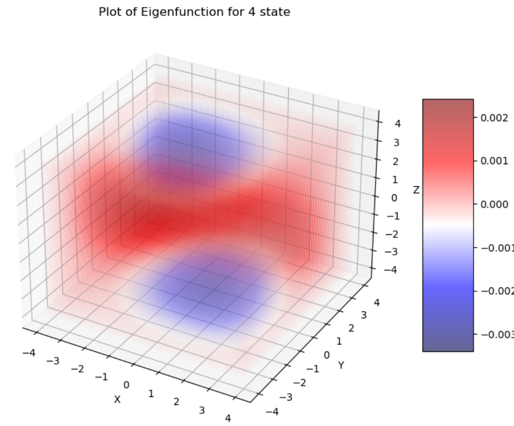

FIG. 13. Plot of the eigenfunctions for 4th state.

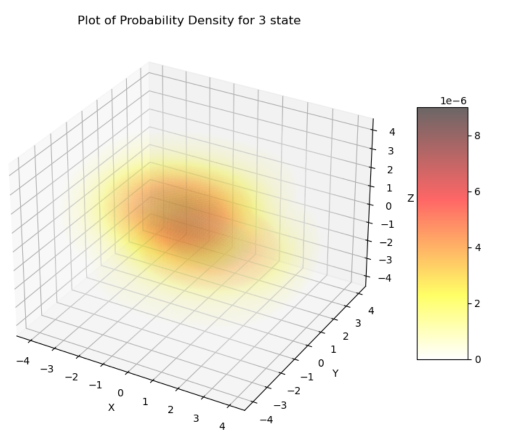

FIG. 12. Plot of the probability density for 3rd state.

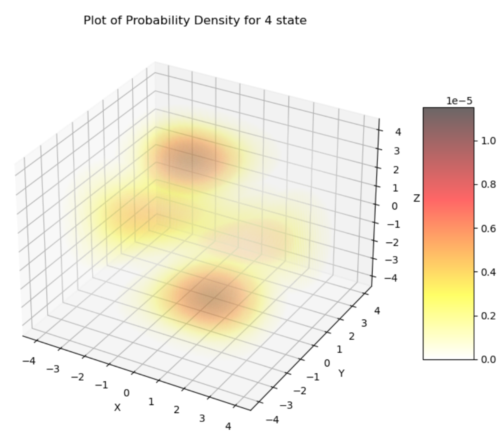

FIG. 14. Plot of the probability density for 4th state.

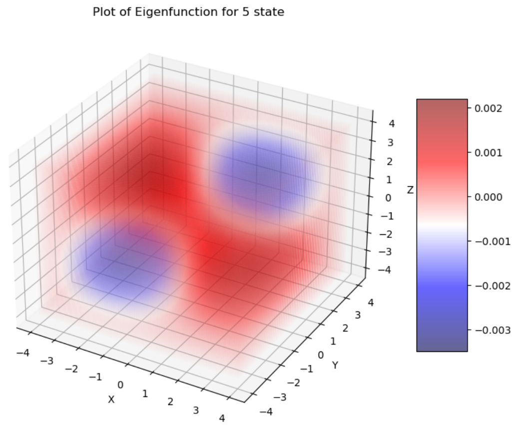

FIG. 15. Plot of the eigenfunctions for 5th state.

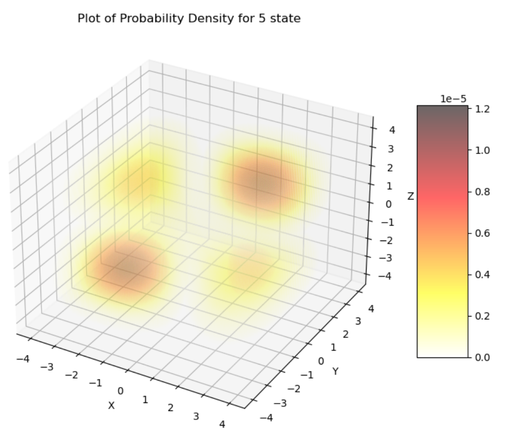

FIG. 16. Plot of the probability density for 5th state.

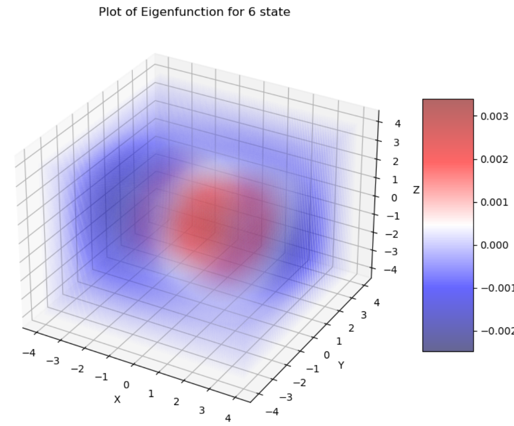

FIG. 17. Plot of the eigenfunctions for 6th state.

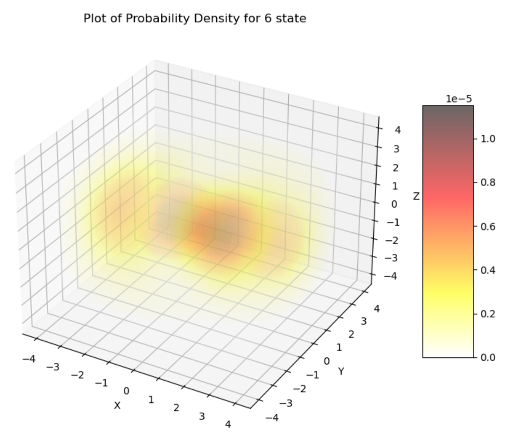

FIG. 18. Plot of the probability density for 6th state.

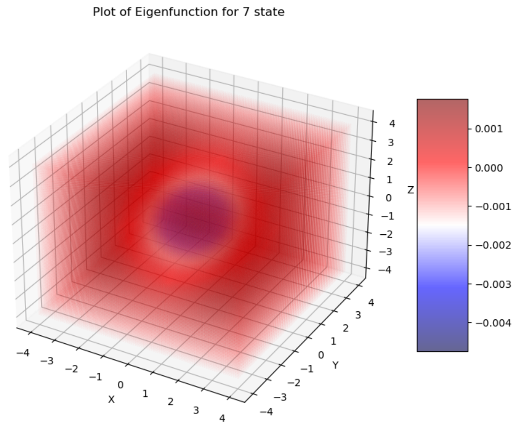

FIG. 19. Plot of the eigenfunctions for 7th state.

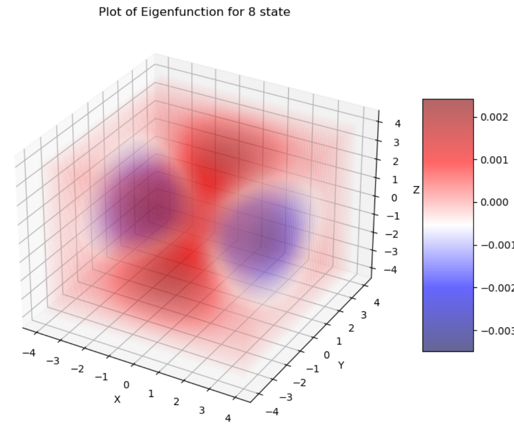

FIG. 21. Plot of the eigenfunctions for 8th state.

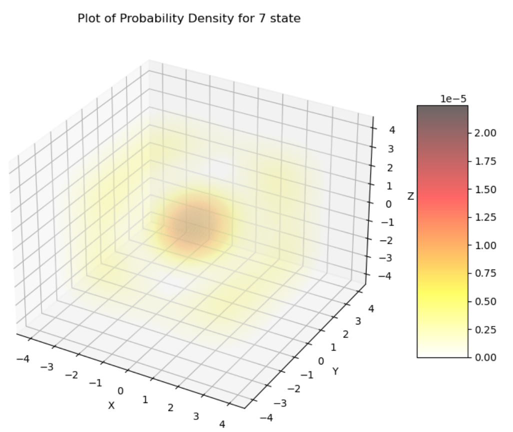

FIG. 20. Plot of the probability density for 7th state.

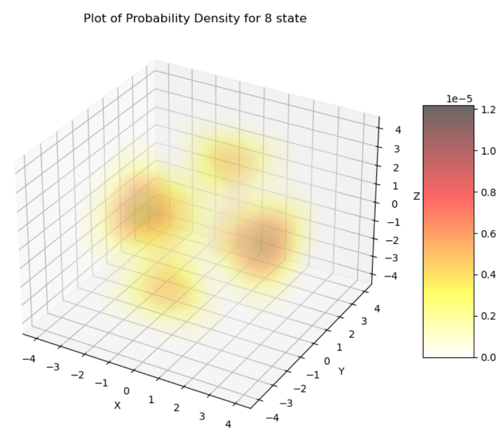

FIG. 22. Plot of the probability density for 8th state.

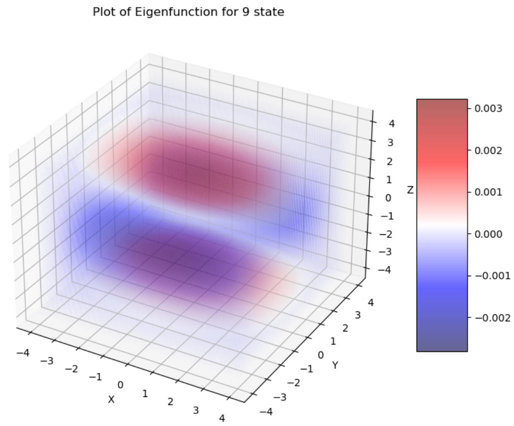

FIG. 23. Plot of the eigenfunctions for 9th state.

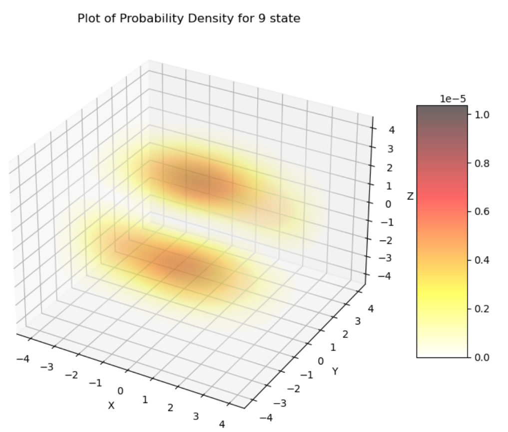

FIG. 24. Plot of the probability density for 9th state.

B. 3D simple harmonic oscillator potential

The 2nd case is the 3D simple harmonic oscillator potential. The wavefunction is confined in

$$V = \frac{0.0001}{2}\left(X^2 + Y^2 + Z^2\right) \tag{29}$$

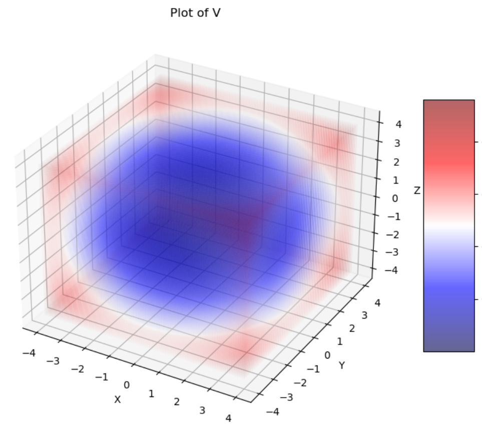

FIG. 25. Plot of the 3D infinite square wall potential.

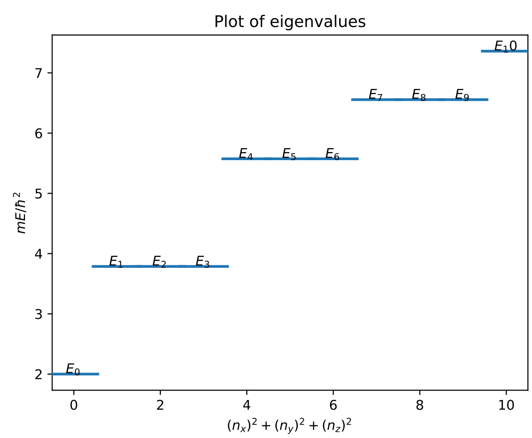

FIG. 26. Plot of the eigenvalues from ground state to 9th state.

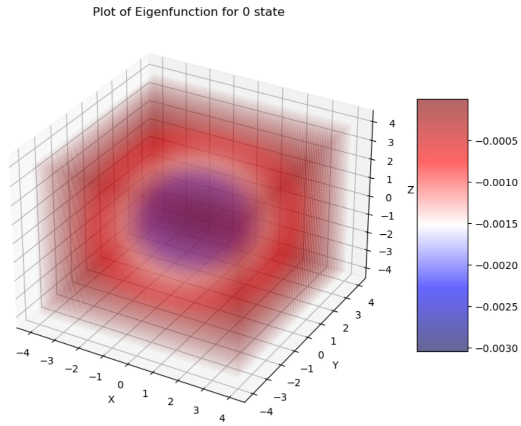

FIG. 27. Plot of the eigenfunctions for ground state.

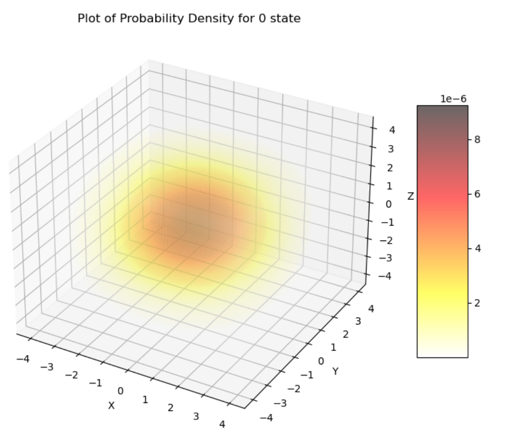

FIG. 28. Plot of the probability density for ground state.

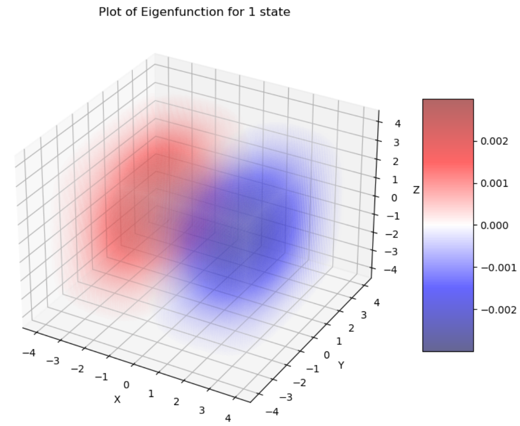

FIG. 29. Plot of the eigenfunctions for 1st state.

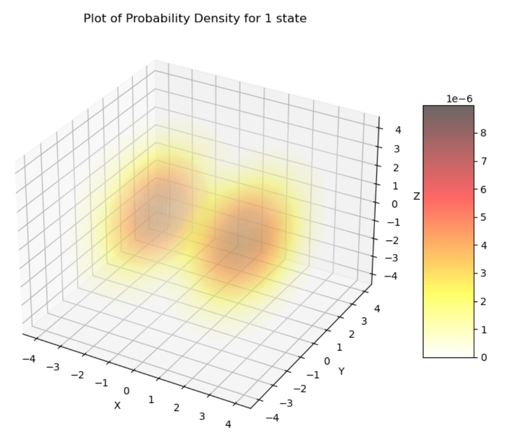

FIG. 30. Plot of the probability density for 1st state.

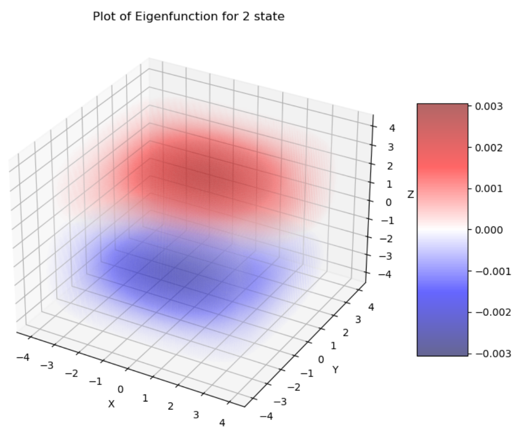

FIG. 31. Plot of the eigenfunctions for 2nd state.

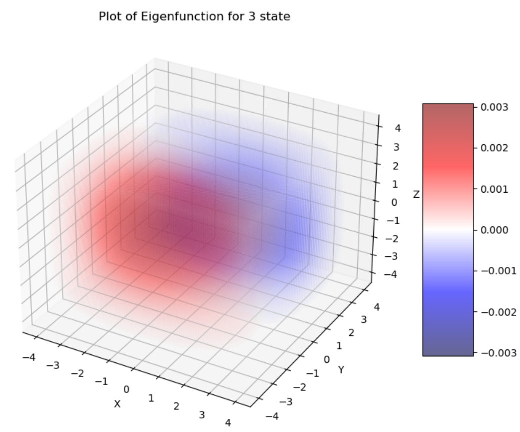

FIG. 33. Plot of the eigenfunctions for 3rd state.

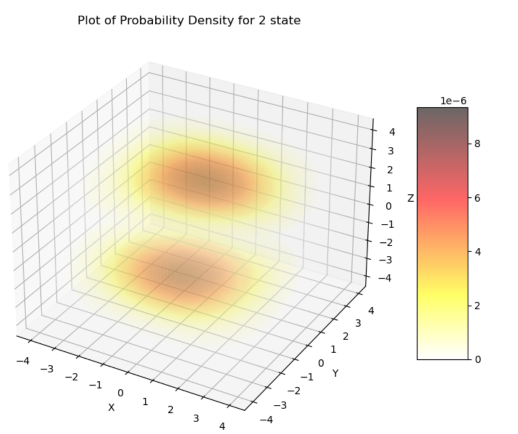

FIG. 32. Plot of the probability density for 2nd state.

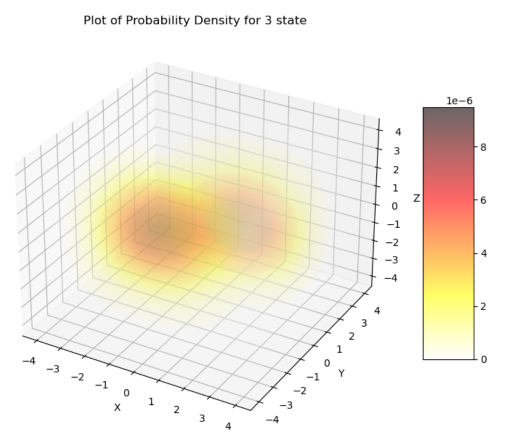

FIG. 34. Plot of the probability density for 3rd state.

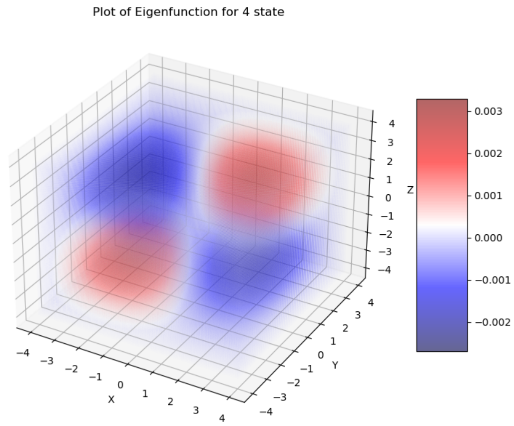

FIG. 35. Plot of the eigenfunctions for 4th state.

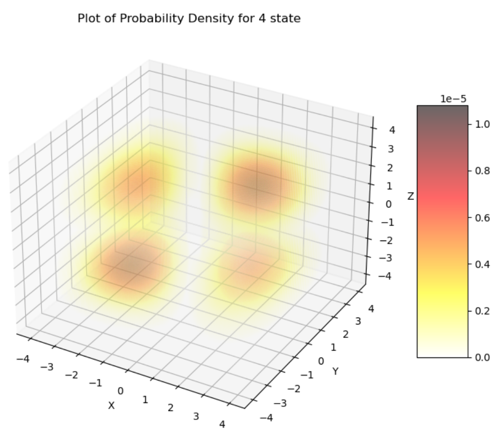

FIG. 36. Plot of the probability density for 4th state.

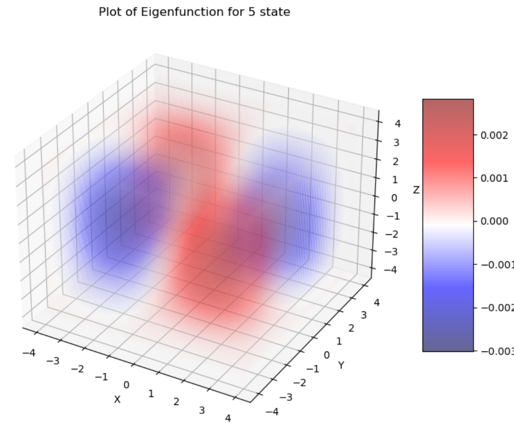

FIG. 37. Plot of the eigenfunctions for 5th state.

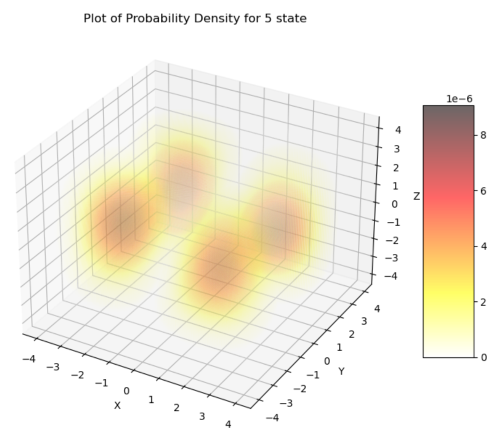

FIG. 38. Plot of the probability density for 5th state.

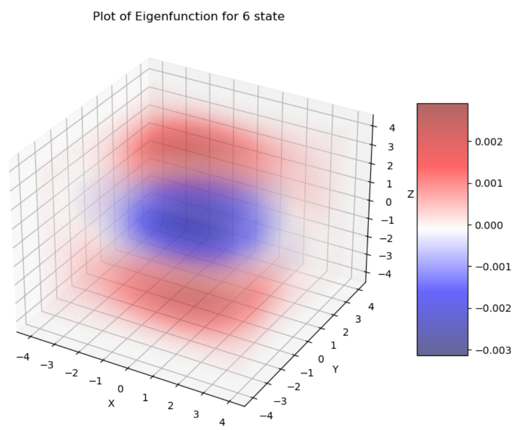

FIG. 39. Plot of the eigenfunctions for 6th state.

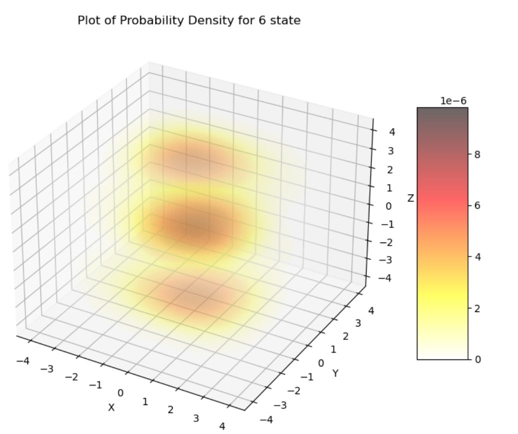

FIG. 40. Plot of the probability density for 6th state.

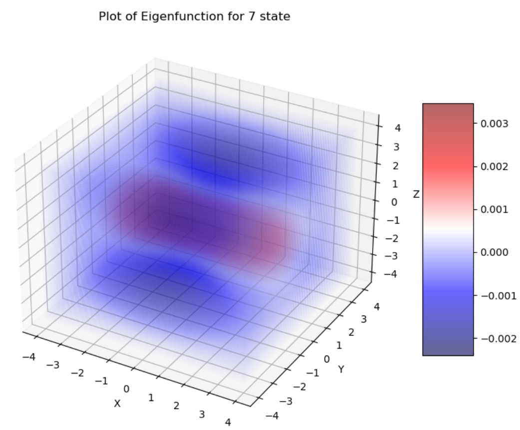

FIG. 41. Plot of the eigenfunctions for 7th state.

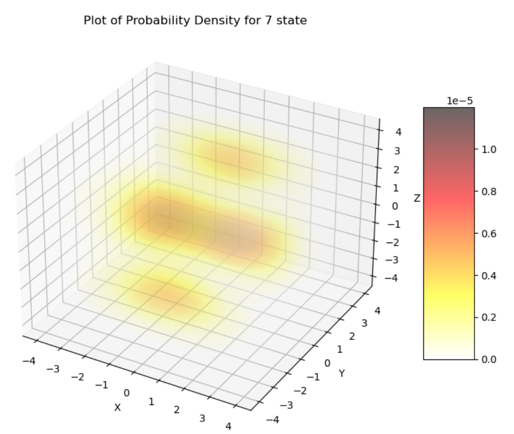

FIG. 42. Plot of the probability density for 7th state.

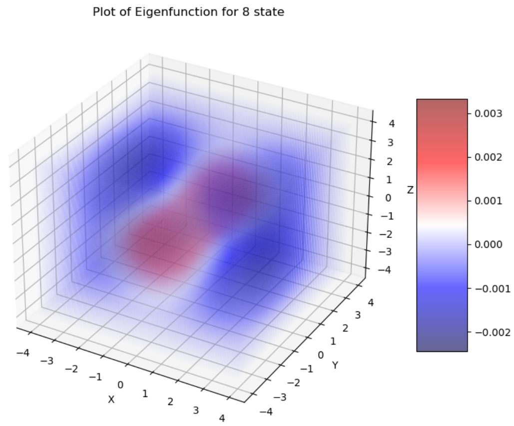

FIG. 43. Plot of the eigenfunctions for 8th state.

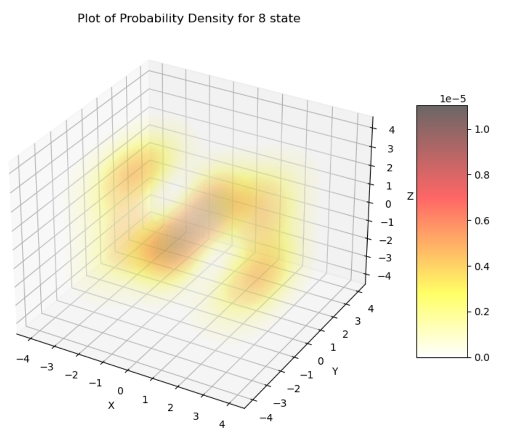

FIG. 44. Plot of the probability density for 8th state.

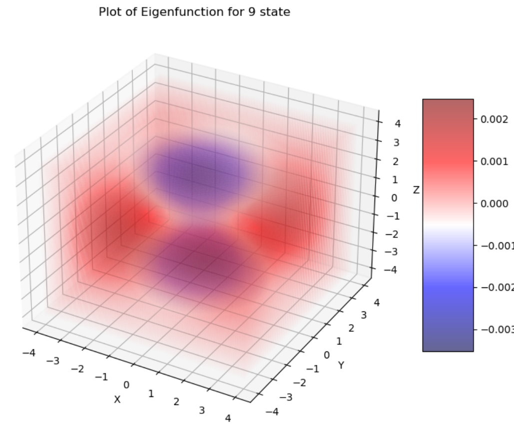

FIG. 45. Plot of the eigenfunctions for 9th state.

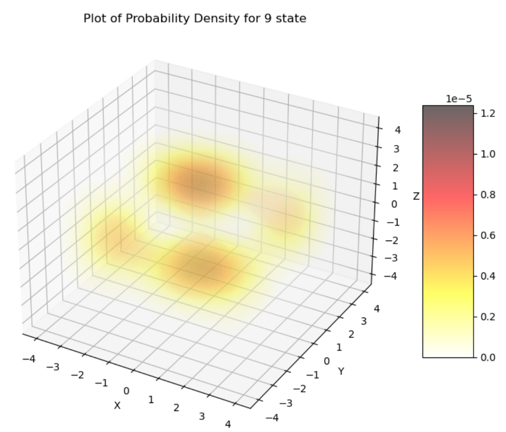

FIG. 46. Plot of the probability density for 9th state.

C. 3D circular well potential

The 3rd case is the 3D circular well potential. The wavefunction is confined in

$$V = 200 - \left(410 \times \pi \times \sqrt{\left(\frac{X}{L}\right)^2 + \left(\frac{Y}{L}\right)^2 + \left(\frac{Z}{L}\right)^2}\right) \tag{30}$$

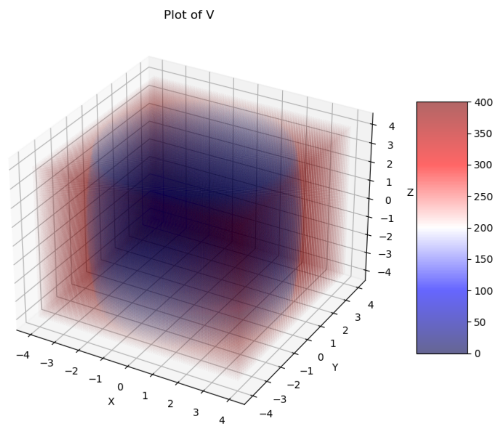

FIG. 47. Plot of the 3D circular well potential.

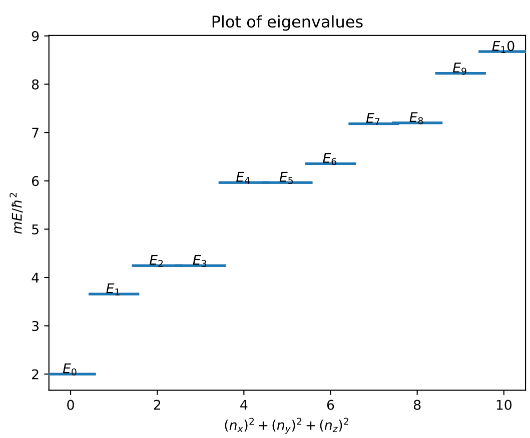

FIG. 48. Plot of the eigenvalues from ground state to 9th state.

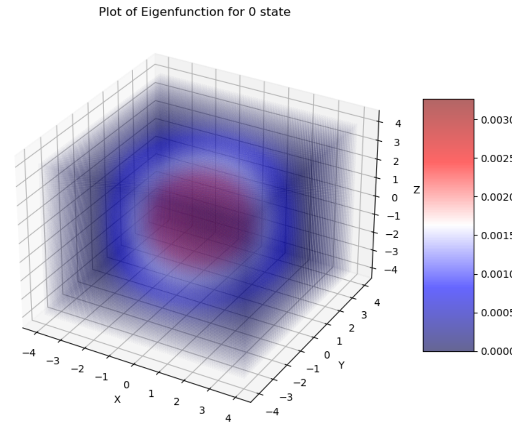

FIG. 49. Plot of the eigenfunctions for ground state.

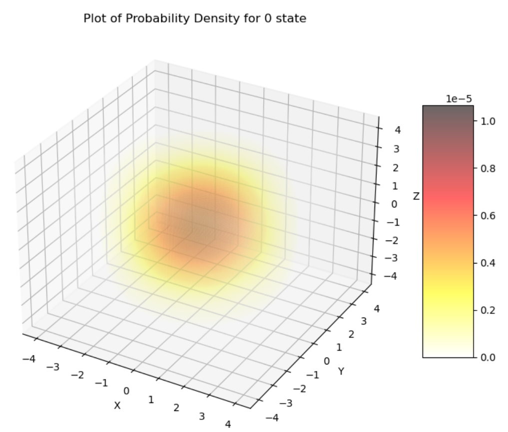

FIG. 50. Plot of the probability density for ground state.

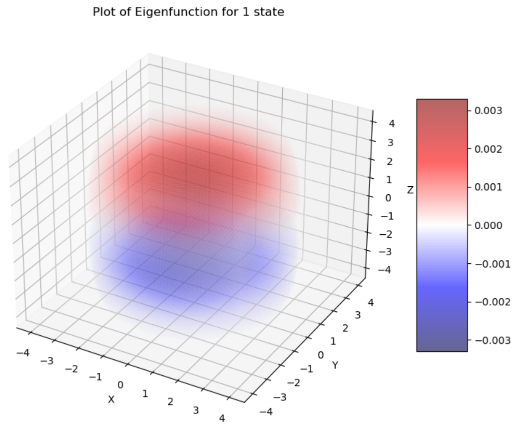

FIG. 51. Plot of the eigenfunctions for 1st state.

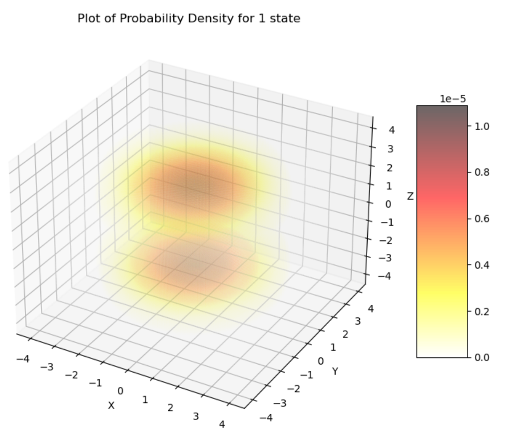

FIG. 52. Plot of the probability density for 1st state.

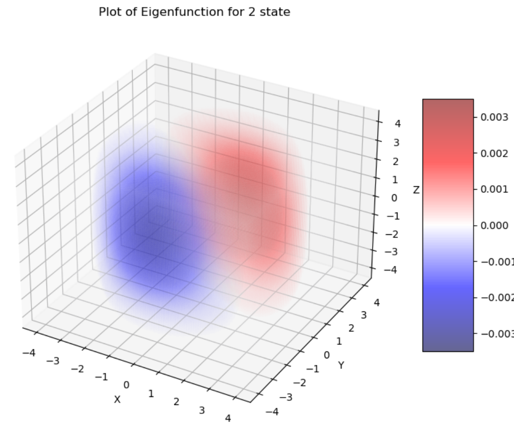

FIG. 53. Plot of the eigenfunctions for 2nd state.

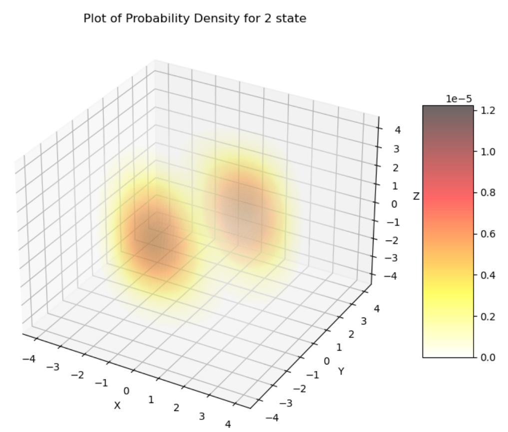

FIG. 54. Plot of the probability density for 2nd state.

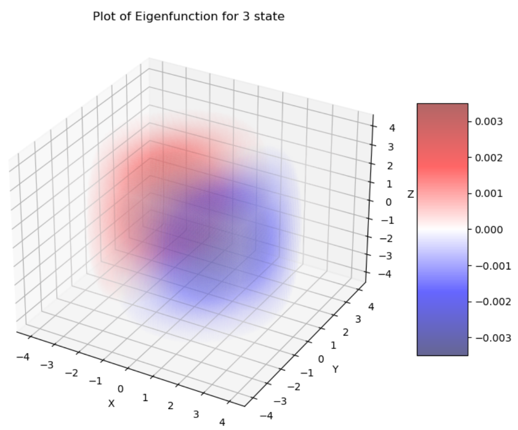

FIG. 55. Plot of the eigenfunctions for 3rd state.

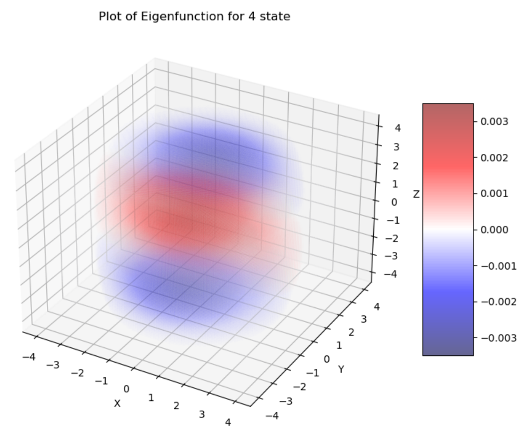

FIG. 57. Plot of the eigenfunctions for 4th state.

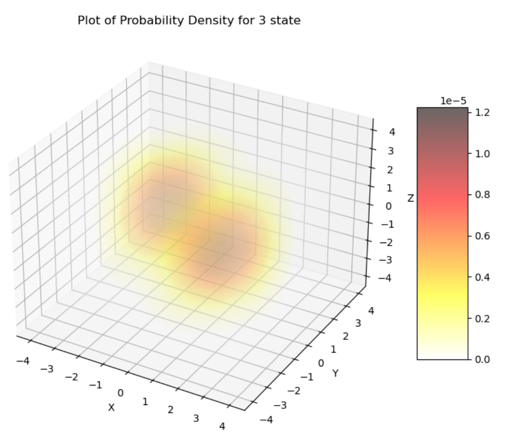

FIG. 56. Plot of the probability density for 3rd state.

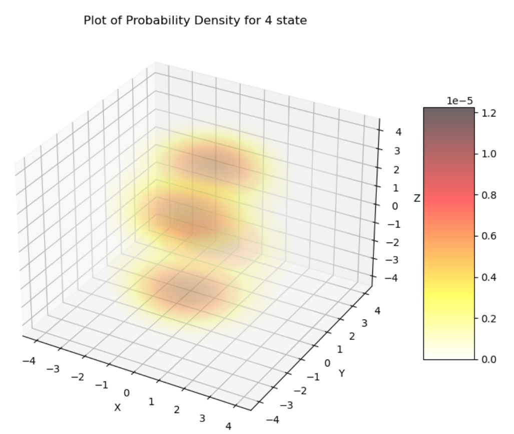

FIG. 58. Plot of the probability density for 4th state.

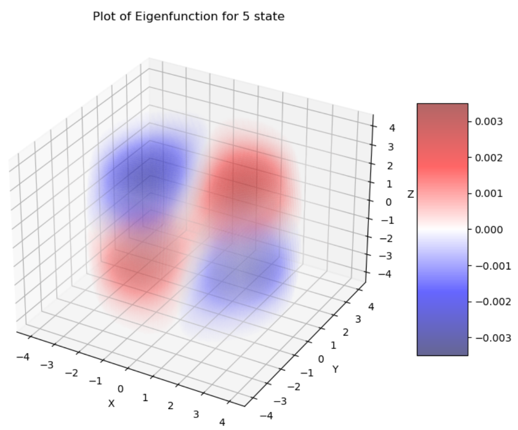

FIG. 59. Plot of the eigenfunctions for 5th state.

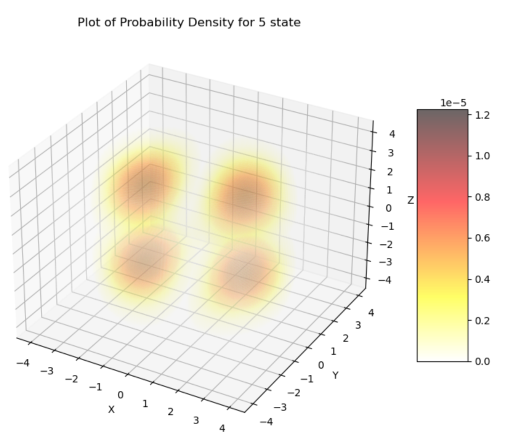

FIG. 60. Plot of the probability density for 5th state.

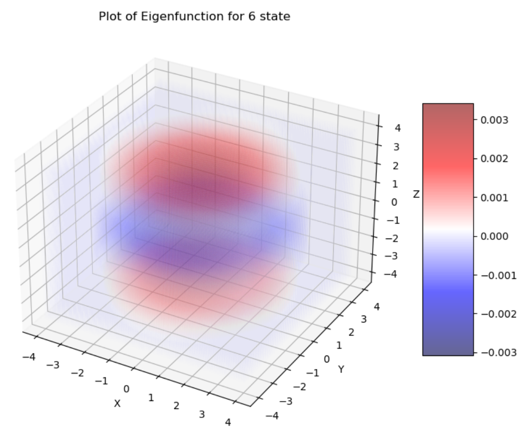

FIG. 61. Plot of the eigenfunctions for 6th state.

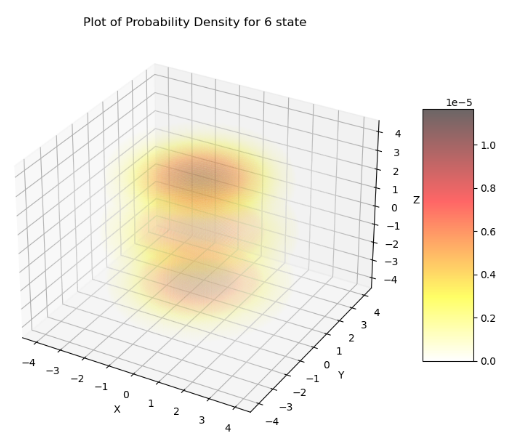

FIG. 62. Plot of the probability density for 6th state.

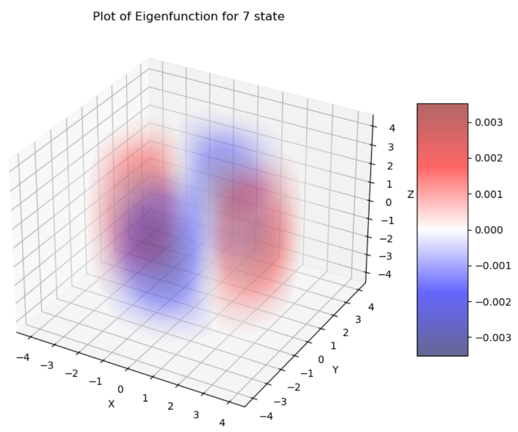

FIG. 63. Plot of the eigenfunctions for 7th state.

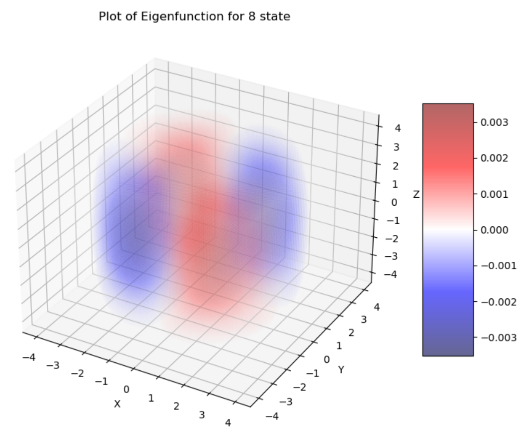

FIG. 65. Plot of the eigenfunctions for 8th state.

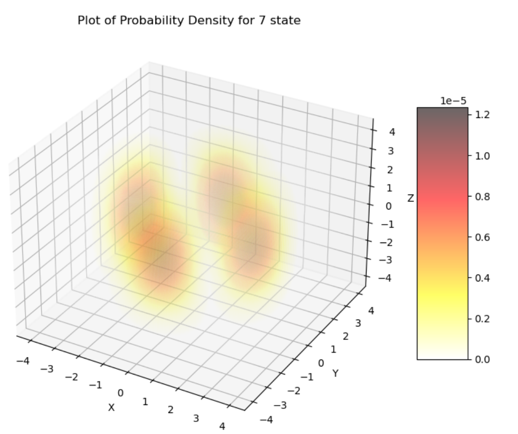

FIG. 64. Plot of the probability density for 7th state.

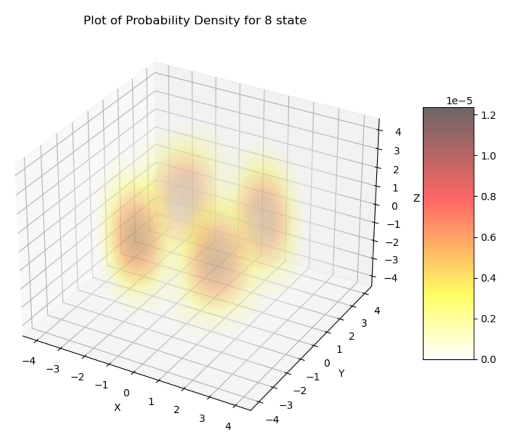

FIG. 66. Plot of the probability density for 8th state.

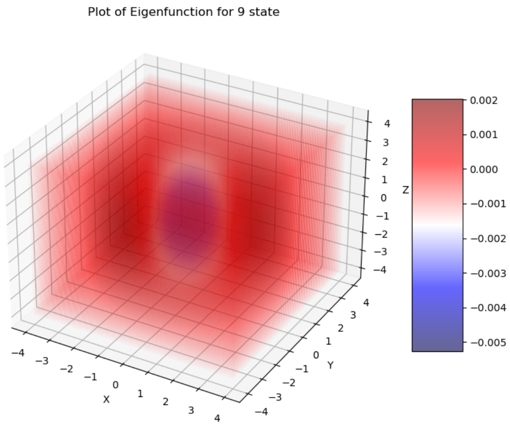

FIG. 67. Plot of the eigenfunctions for 9th state.

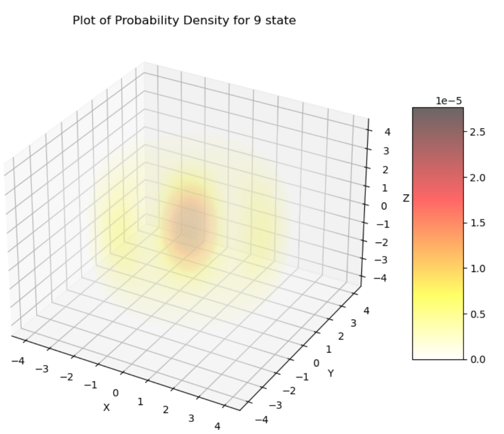

FIG. 68. Plot of the probability density for 9th state.

D. 3D torus potential

The 4th case is the 3D torus potential. The wavefunction is confined in

$$V = \left(\sqrt{\left(\frac{x}{L}\right)^2 + \left(\frac{y}{L}\right)^2} - \frac{R}{L}\right)^2 + \left(\frac{z}{L}\right)^2 - \left(\frac{r}{L}\right)^2 \quad (31)$$

where $r = 0.1L$, $R = 0.4L$

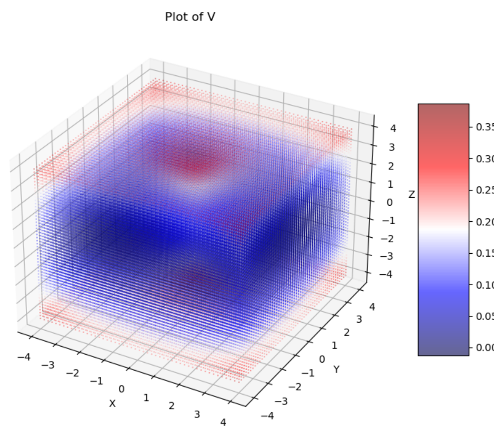

FIG. 69. Plot of the 3D circular well potential.

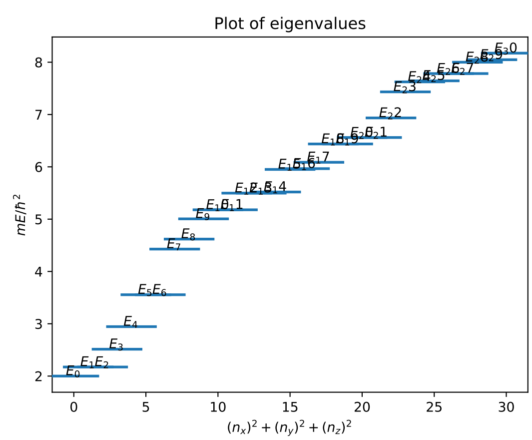

FIG. 70. Plot of the eigenvalues from ground state to 30th state.

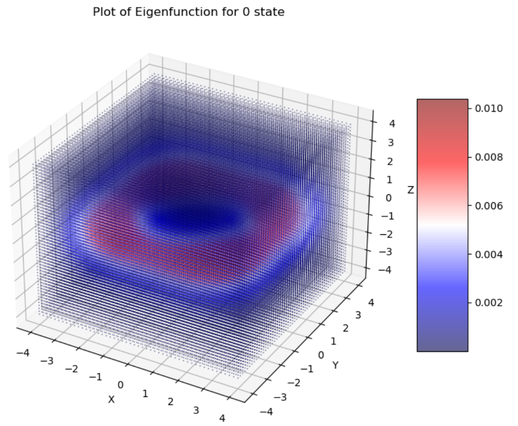

FIG. 71. Plot of the eigenfunctions for ground state.

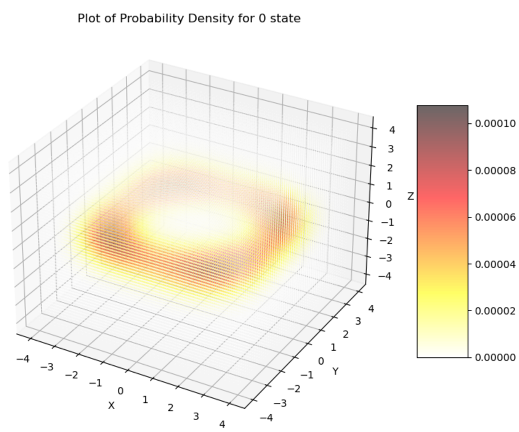

FIG. 72. Plot of the probability density for ground state.

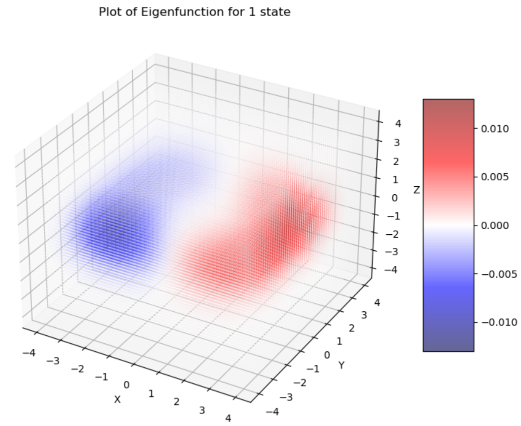

FIG. 73. Plot of the eigenfunctions for 1st state.

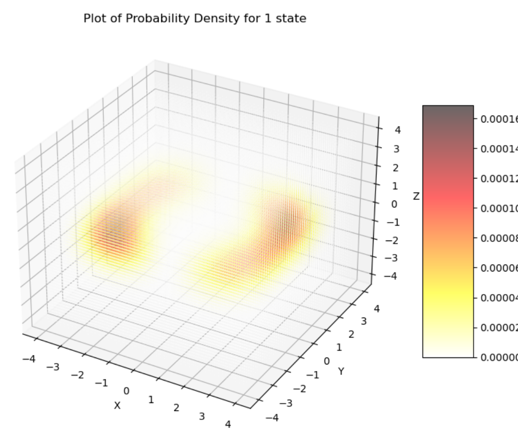

FIG. 74. Plot of the probability density for 1st state.

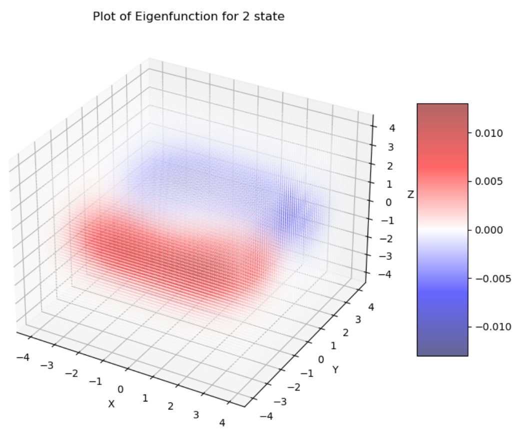

FIG. 75. Plot of the eigenfunctions for 2nd state.

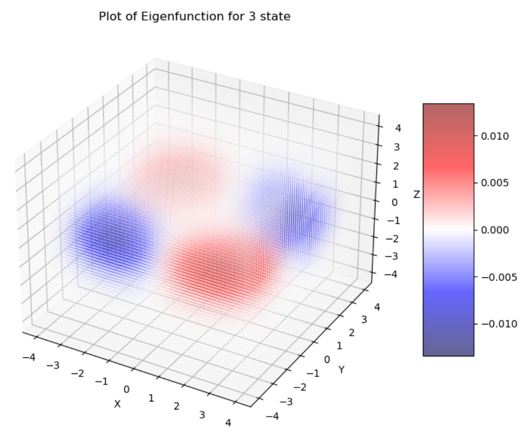

FIG. 77. Plot of the eigenfunctions for 3rd state.

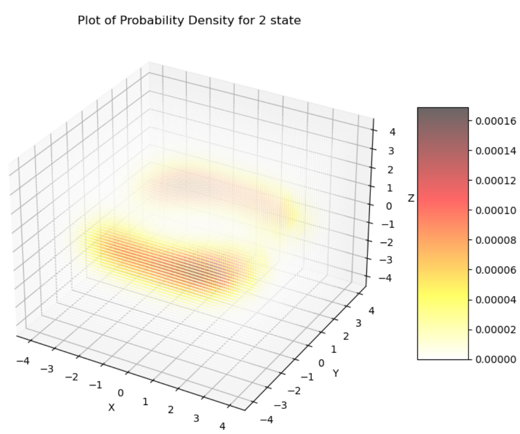

FIG. 76. Plot of the probability density for 2nd state.

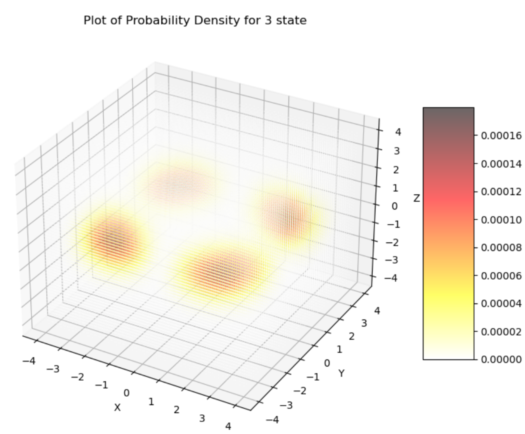

FIG. 78. Plot of the probability density for 3rd state.

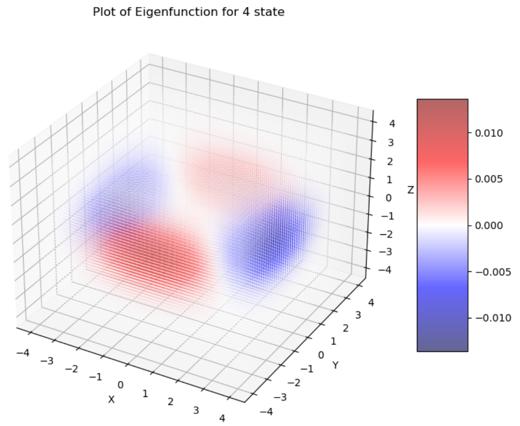

FIG. 79. Plot of the eigenfunctions for 4th state.

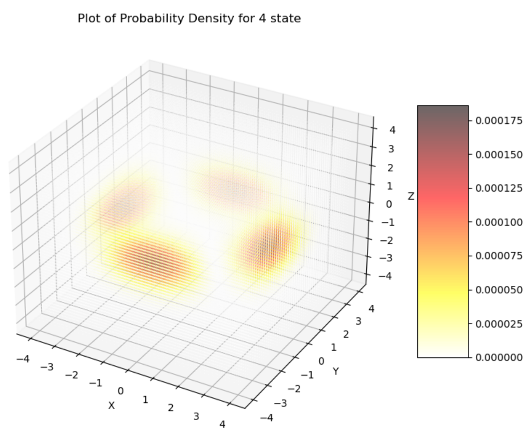

FIG. 80. Plot of the probability density for 4th state.

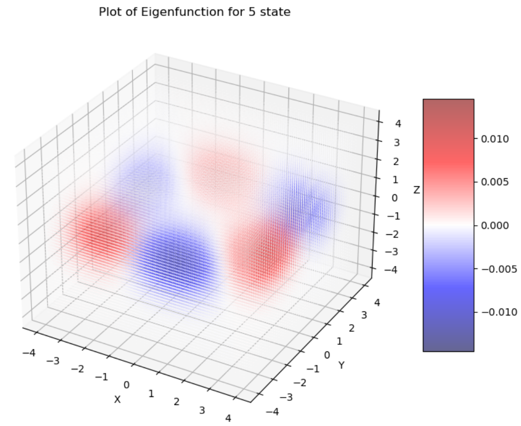

FIG. 81. Plot of the eigenfunctions for 5th state.

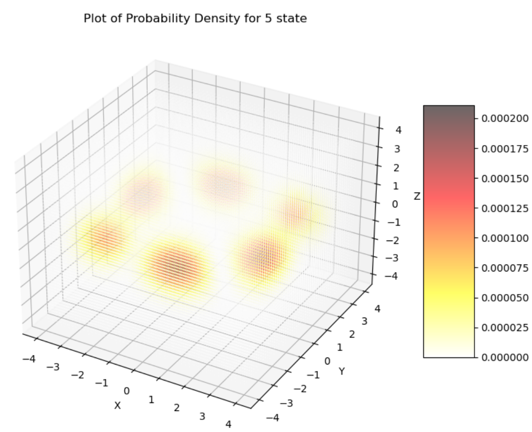

FIG. 82. Plot of the probability density for 5th state.

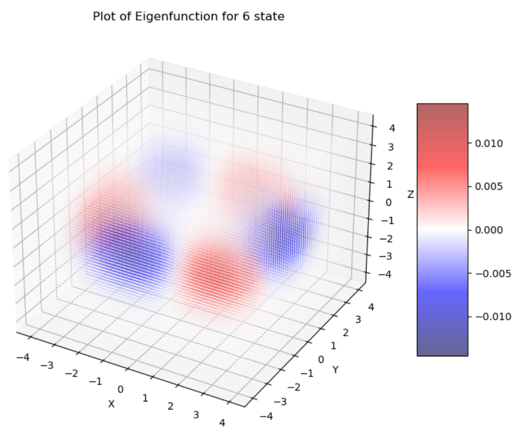

FIG. 83. Plot of the eigenfunctions for 6th state.

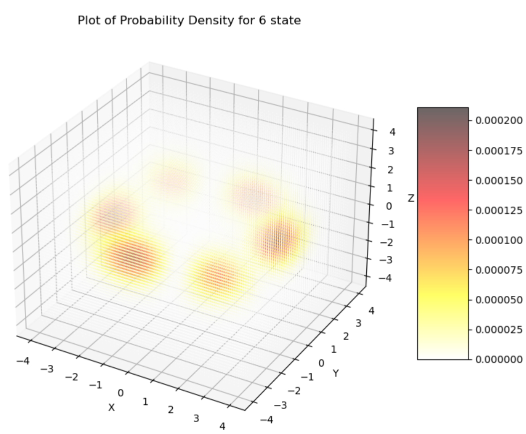

FIG. 84. Plot of the probability density for 6th state.

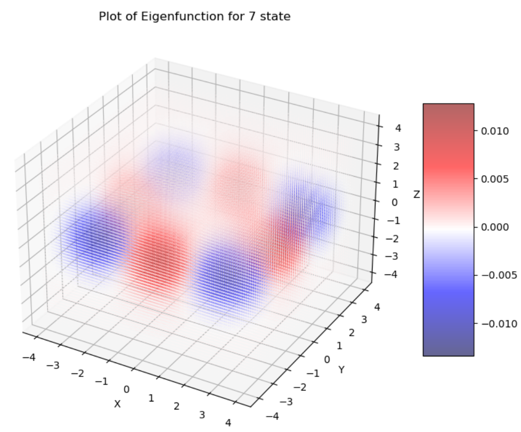

FIG. 85. Plot of the eigenfunctions for 7th state.

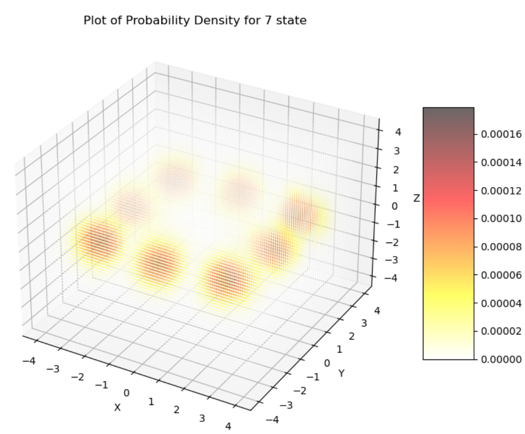

FIG. 86. Plot of the probability density for 7th state.

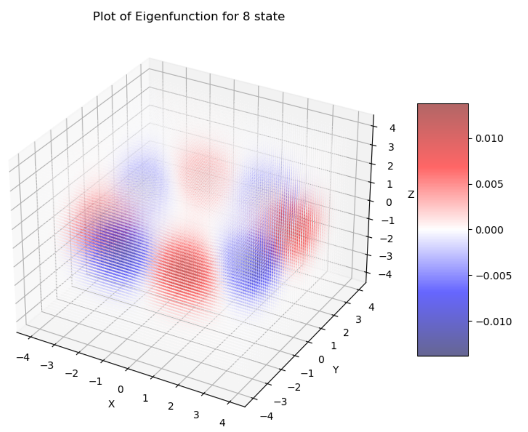

FIG. 87. Plot of the eigenfunctions for 8th state.

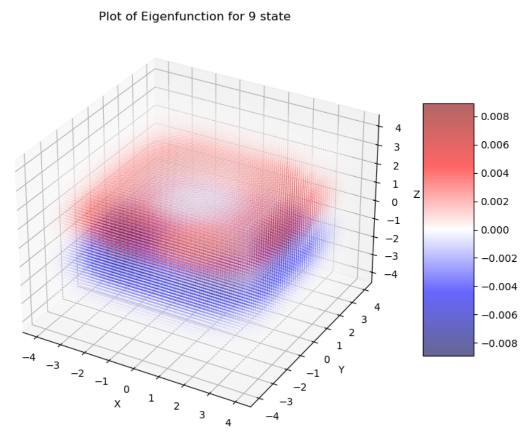

FIG. 89. Plot of the eigenfunctions for 9th state.

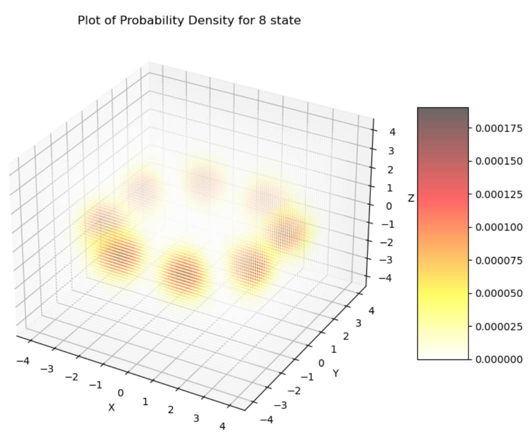

FIG. 88. Plot of the probability density for 8th state.

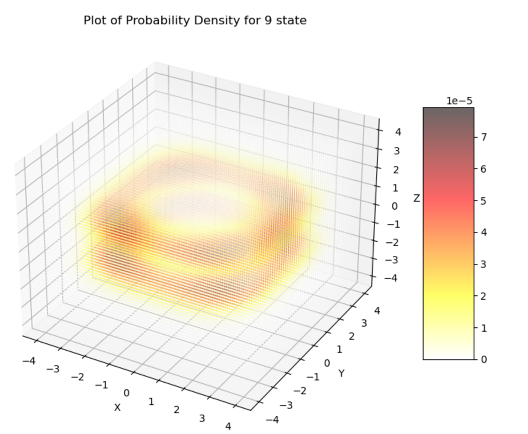

FIG. 90. Plot of the probability density for 9th state.

E. 3D *tanh* potential

The 5th case is the 3D *tanh* potential. The wavefunction is confined in

$$V = \tanh\left(\left(\frac{X}{L}\right)^2 + \left(\frac{Y}{L}\right)^2\right) \tag{32}$$

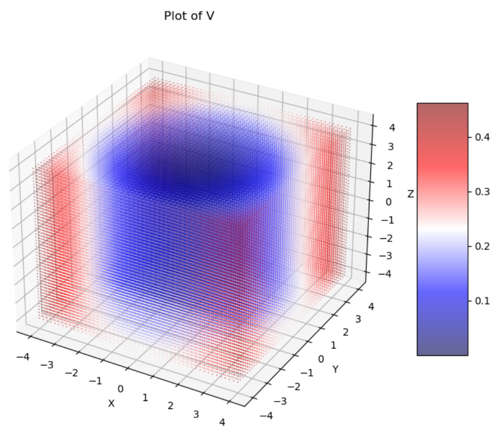

FIG. 91. Plot of the 3D circular well potential.

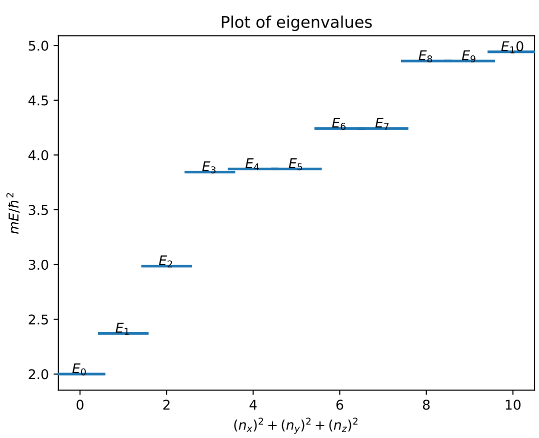

FIG. 92. Plot of the eigenvalues from ground state to 9th state.

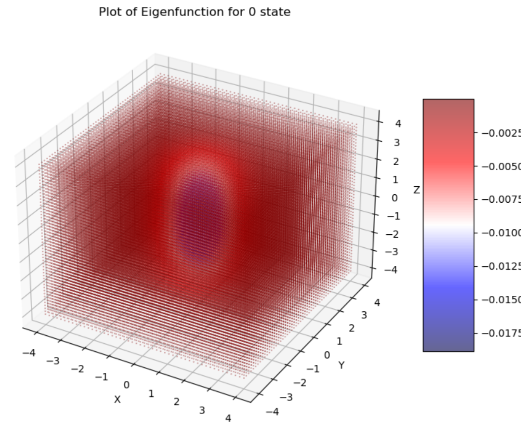

FIG. 93. Plot of the eigenfunctions for ground state.

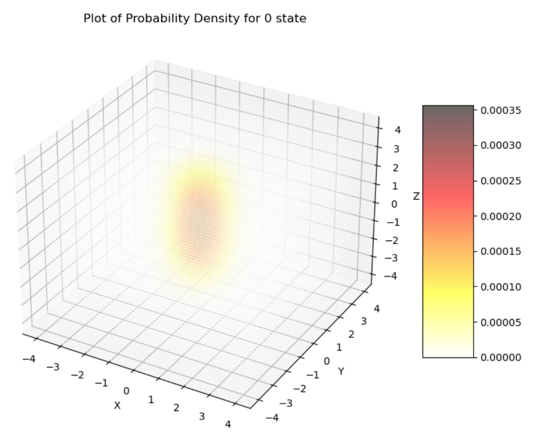

FIG. 94. Plot of the probability density for ground state.

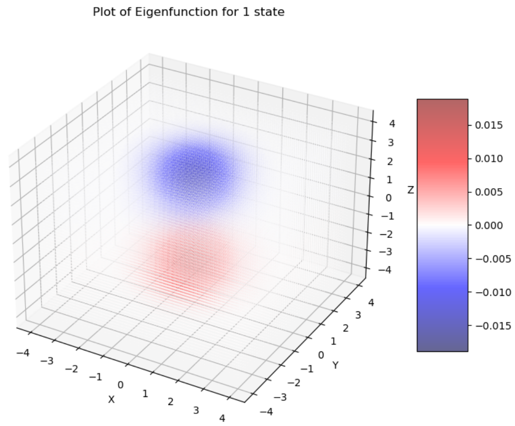

FIG. 95. Plot of the eigenfunctions for 1st state.

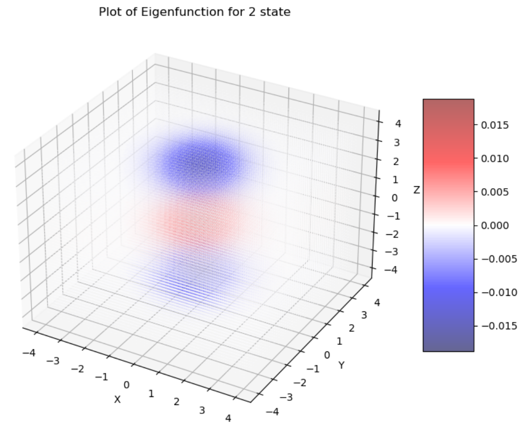

FIG. 97. Plot of the eigenfunctions for 2nd state.

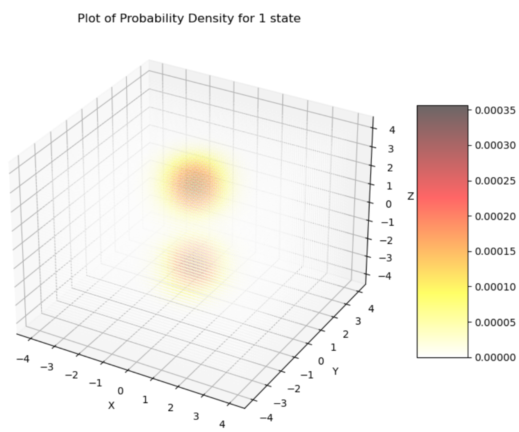

FIG. 96. Plot of the probability density for 1st state.

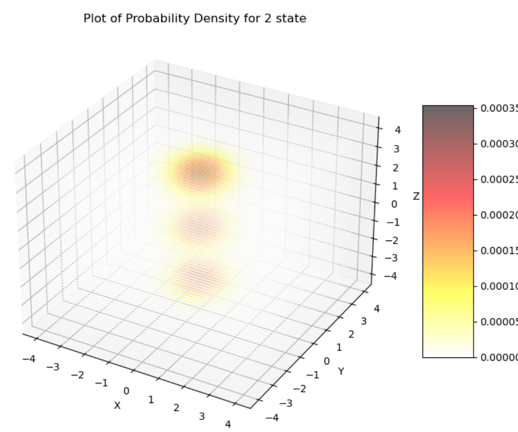

FIG. 98. Plot of the probability density for 2nd state.

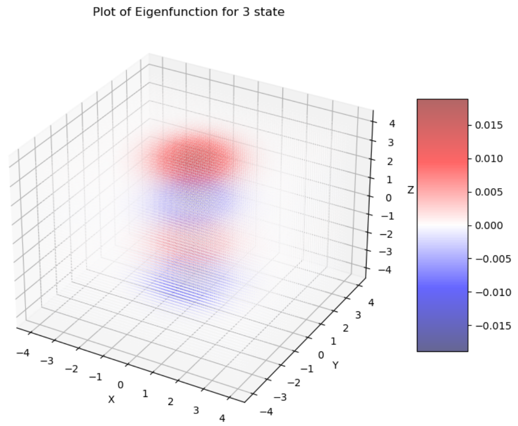

FIG. 99. Plot of the eigenfunctions for 3rd state.

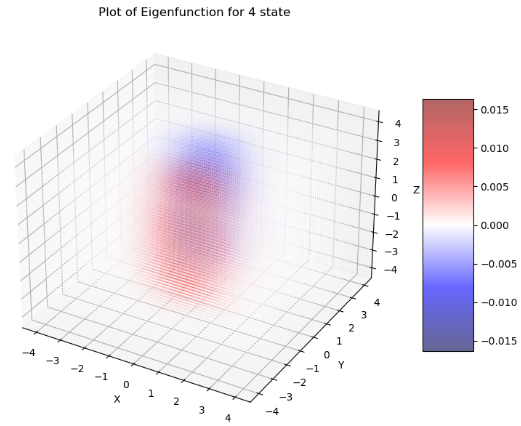

FIG. 101. Plot of the eigenfunctions for 4th state.

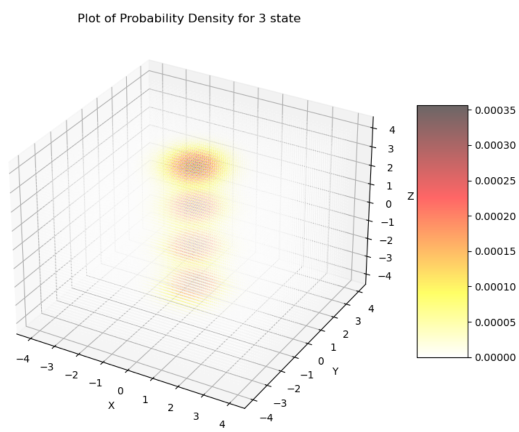

FIG. 100. Plot of the probability density for 3rd state.

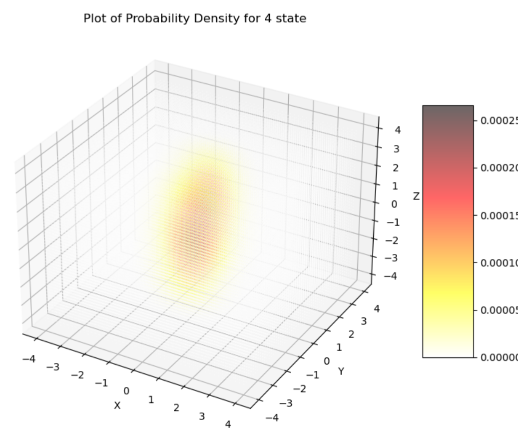

FIG. 102. Plot of the probability density for 4th state.

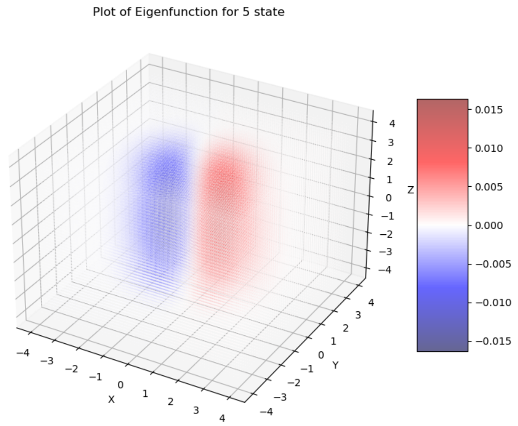

FIG. 103. Plot of the eigenfunctions for 5th state.

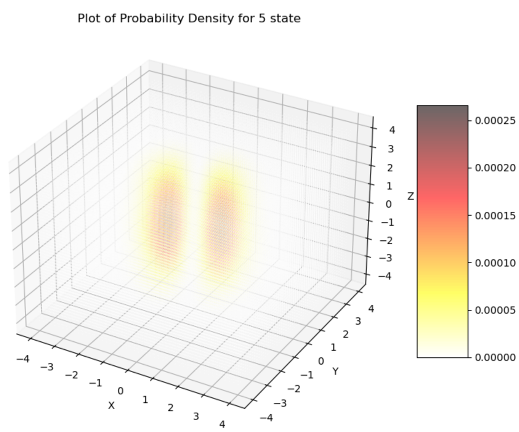

FIG. 104. Plot of the probability density for 5th state.

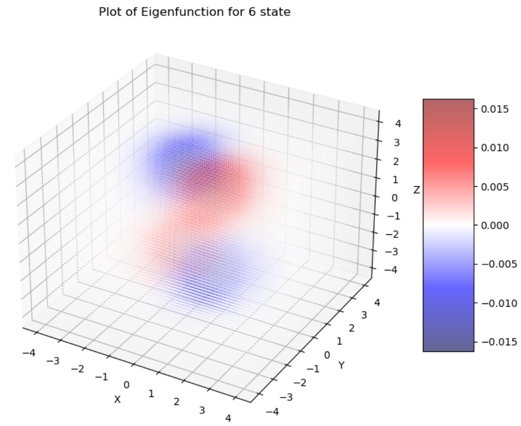

FIG. 105. Plot of the eigenfunctions for 6th state.

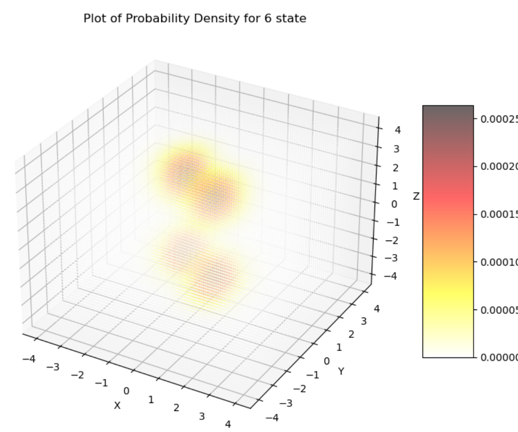

FIG. 106. Plot of the probability density for 6th state.

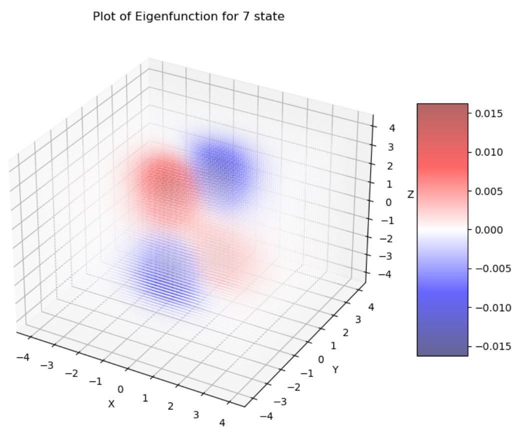

FIG. 107. Plot of the eigenfunctions for 7th state.

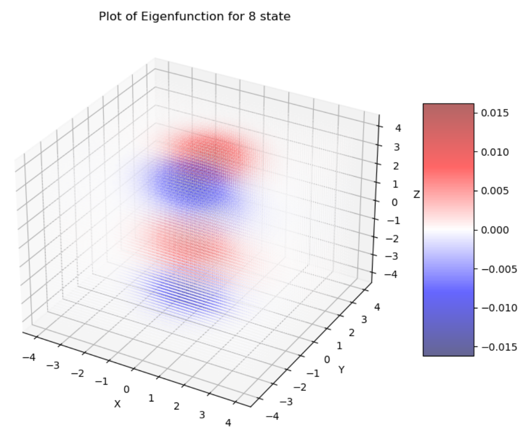

FIG. 109. Plot of the eigenfunctions for 8th state.

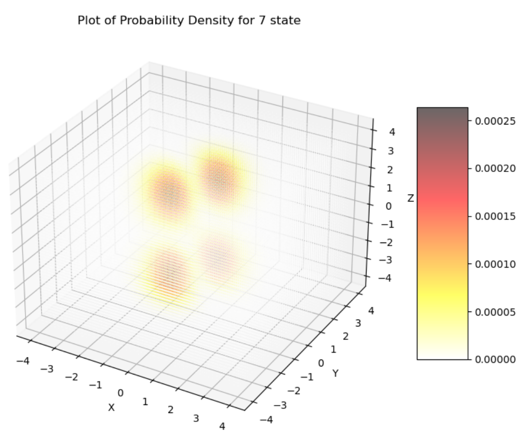

FIG. 108. Plot of the probability density for 7th state.

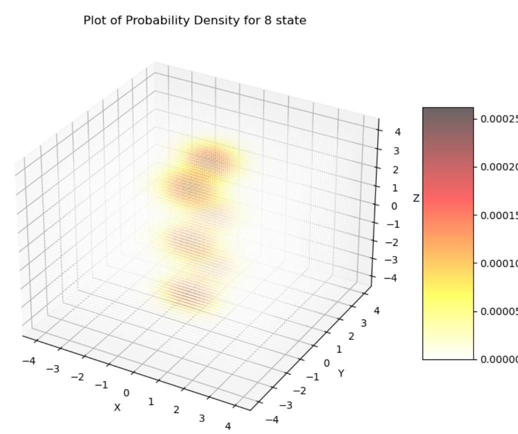

FIG. 110. Plot of the probability density for 8th state.

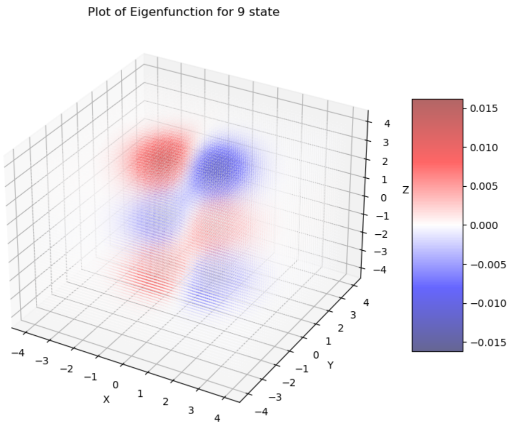

FIG. 111. Plot of the eigenfunctions for 9th state.

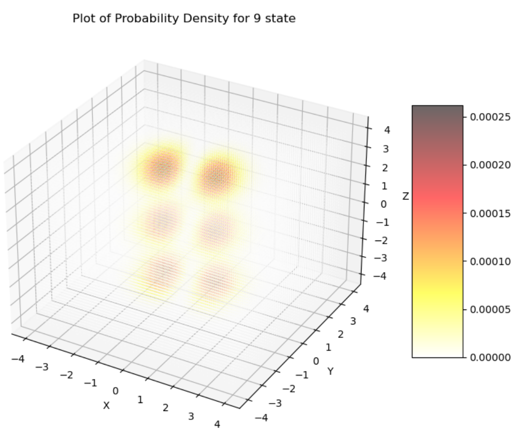

FIG. 112. Plot of the probability density for 9th state.

F. Hydrogen like atom potential potential

The 6th case is the Hydrogen like atom potential. The wavefunction is confined in

$$V = -\frac{1}{\sqrt{X^2 + Y^2 + Z^2}} \tag{33}$$

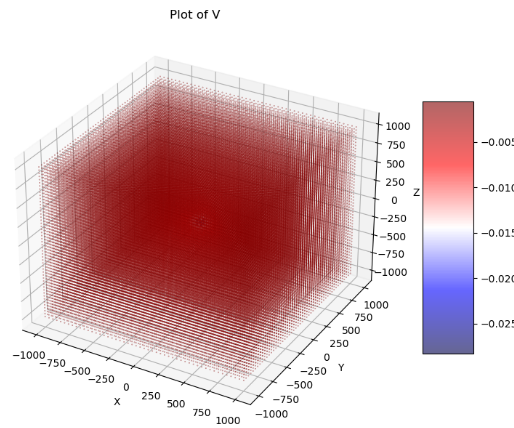

FIG. 113. Plot of the Hydrogen like atom potential.

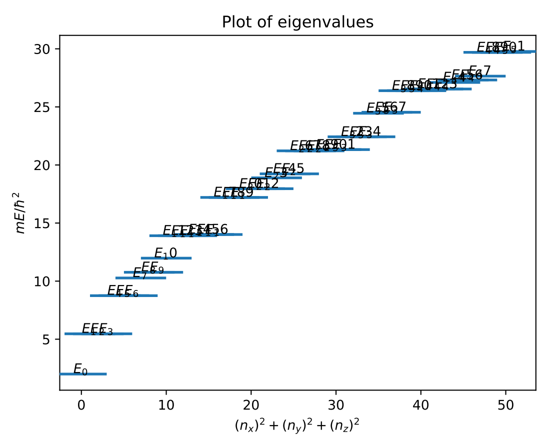

FIG. 114. Plot of the eigenvalues from ground state to 50th state.

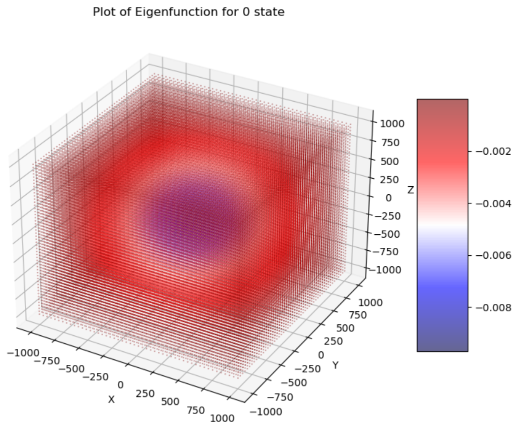

FIG. 115. Plot of the eigenfunctions for ground state.

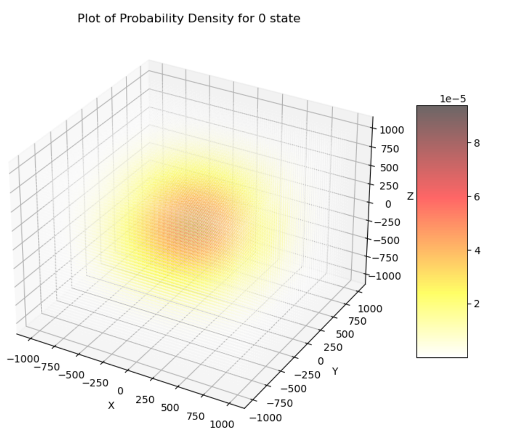

FIG. 116. Plot of the probability density for ground state.

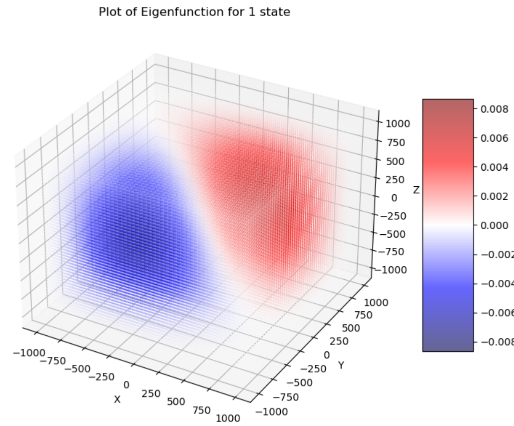

FIG. 117. Plot of the eigenfunctions for 1st state.

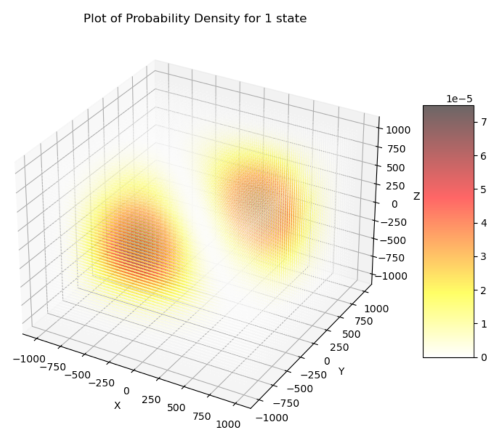

FIG. 118. Plot of the probability density for 1st state.

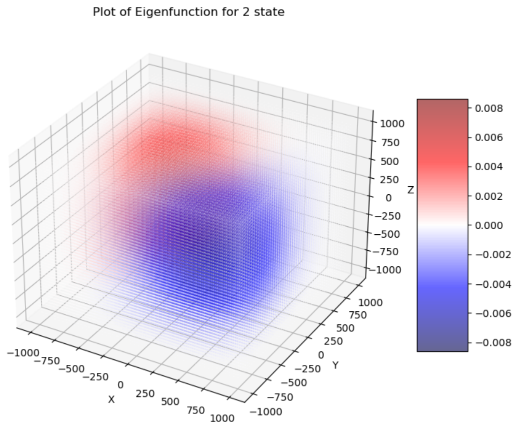

FIG. 119. Plot of the eigenfunctions for 2nd state.

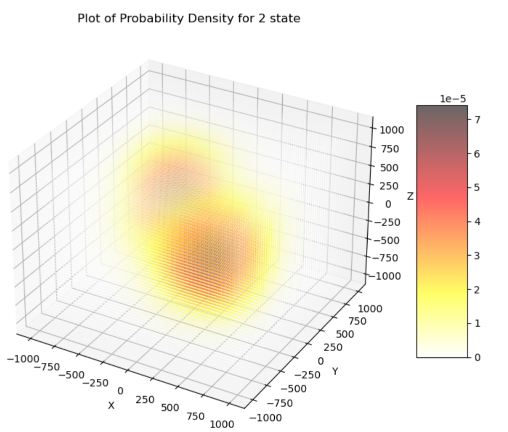

FIG. 120. Plot of the probability density for 2nd state.

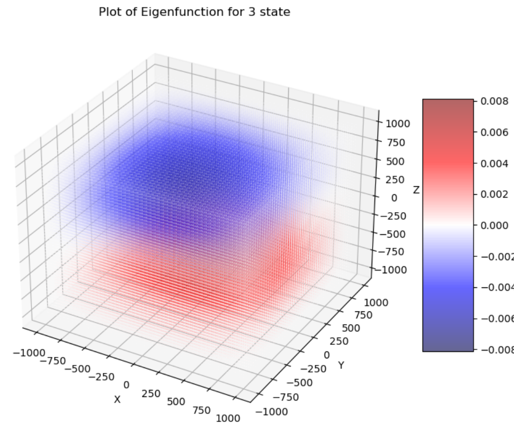

FIG. 121. Plot of the eigenfunctions for 3rd state.

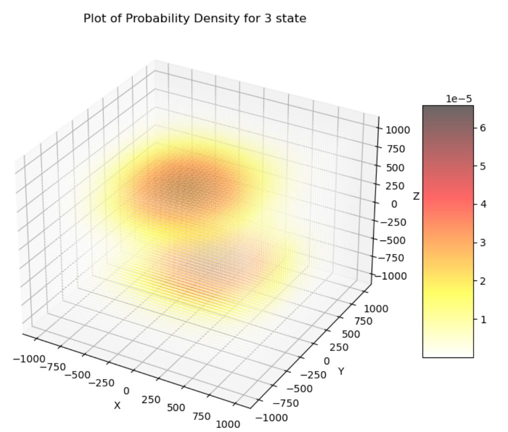

FIG. 122. Plot of the probability density for 3rd state.

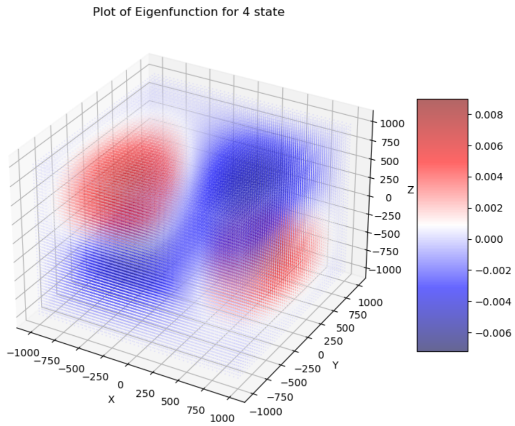

FIG. 123. Plot of the eigenfunctions for 4th state.

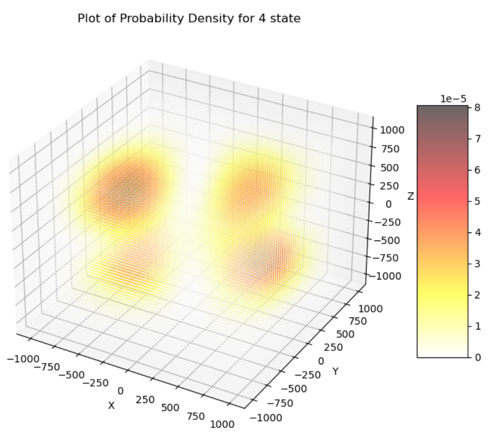

FIG. 124. Plot of the probability density for 4th state.

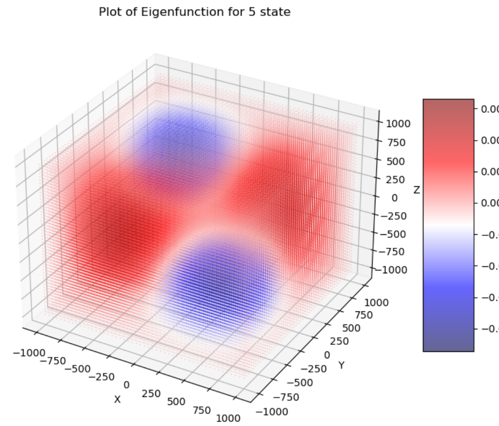

FIG. 125. Plot of the eigenfunctions for 5th state.

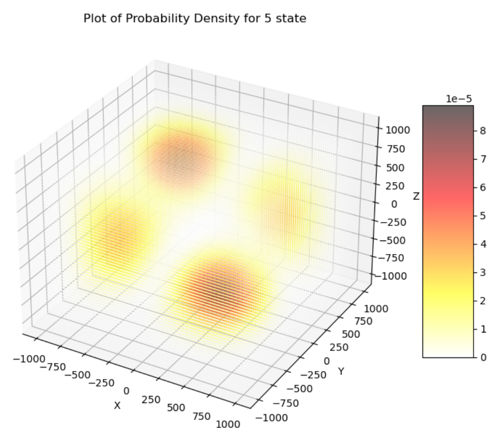

FIG. 126. Plot of the probability density for 5th state.

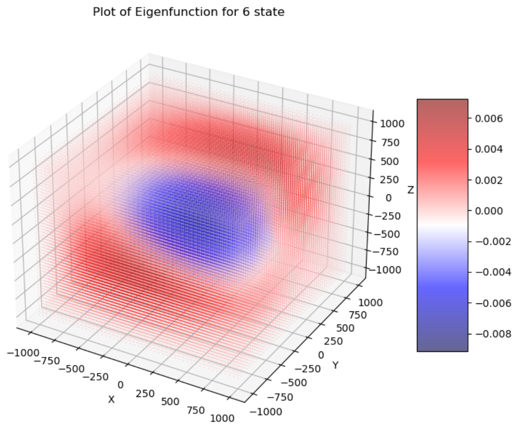

FIG. 127. Plot of the eigenfunctions for 6th state.

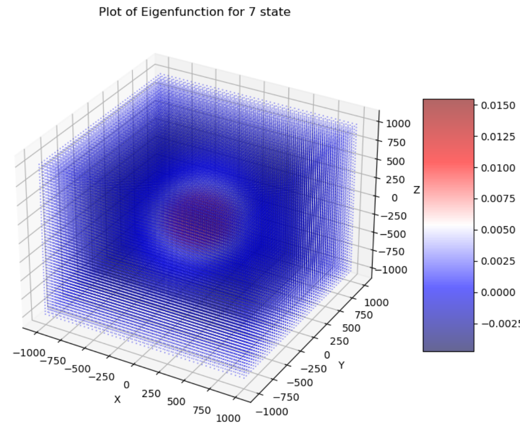

FIG. 129. Plot of the eigenfunctions for 7th state.

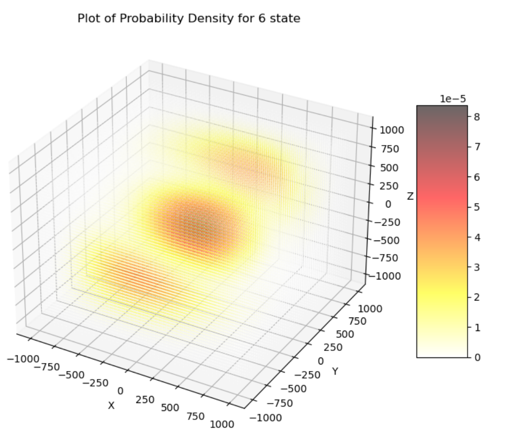

FIG. 128. Plot of the probability density for 6th state.

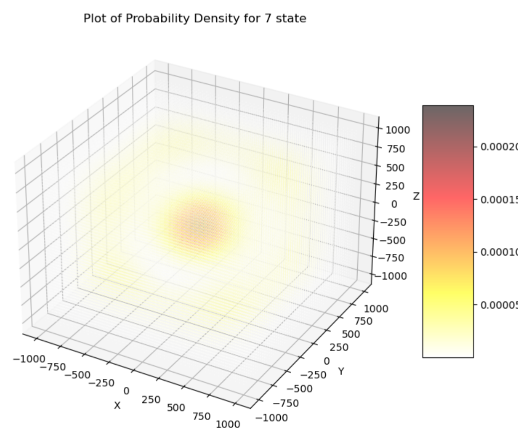

FIG. 130. Plot of the probability density for 7th state.

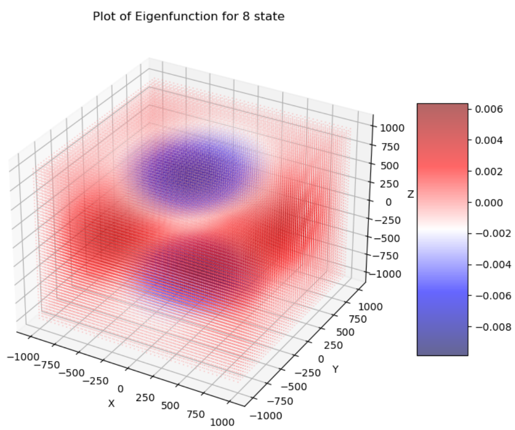

FIG. 131. Plot of the eigenfunctions for 8th state.

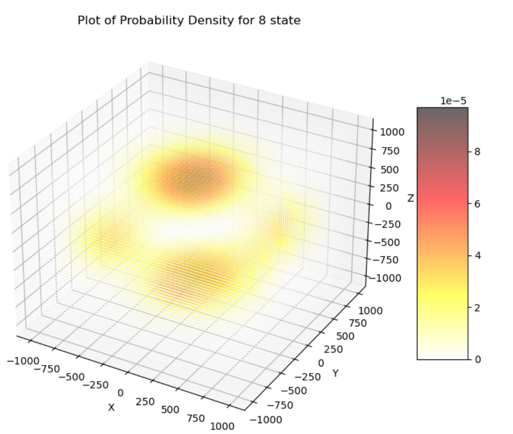

FIG. 132. Plot of the probability density for 8th state.

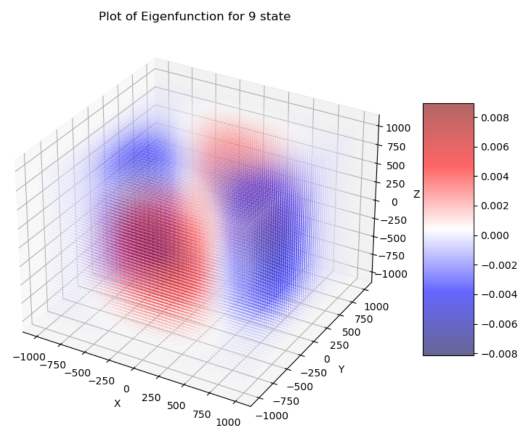

FIG. 133. Plot of the eigenfunctions for 9th state.

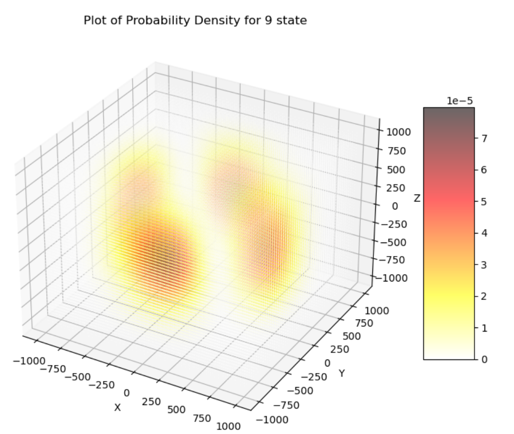

FIG. 134. Plot of the probability density for 9th state.

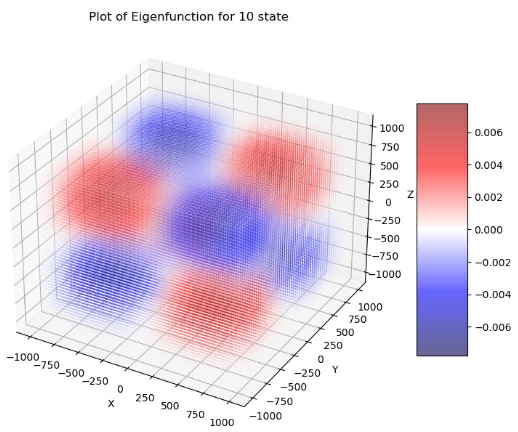

FIG. 135. Plot of the eigenfunctions for 10th state.

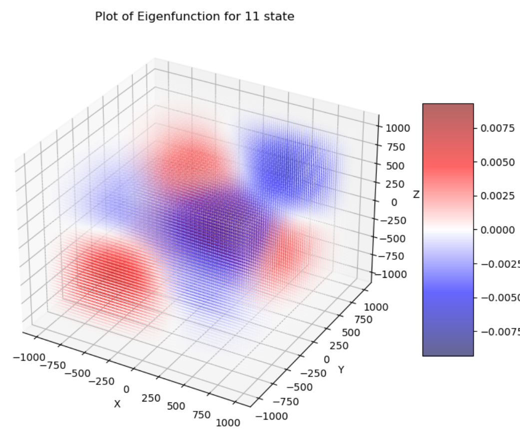

FIG. 137. Plot of the eigenfunctions for 11th state.

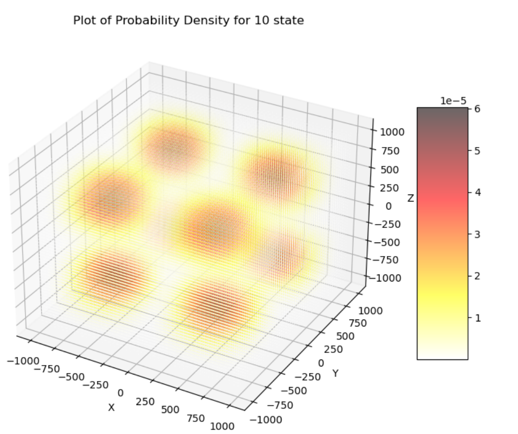

FIG. 136. Plot of the probability density for 10th state.

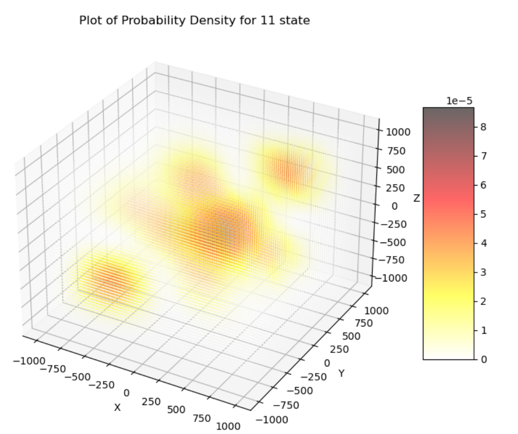

FIG. 138. Plot of the probability density for 11th state.

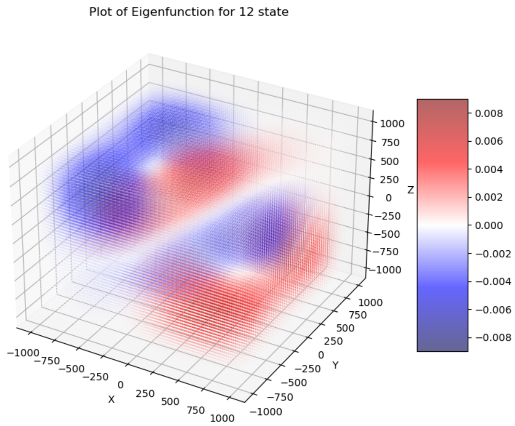

FIG. 139. Plot of the eigenfunctions for 12th state.

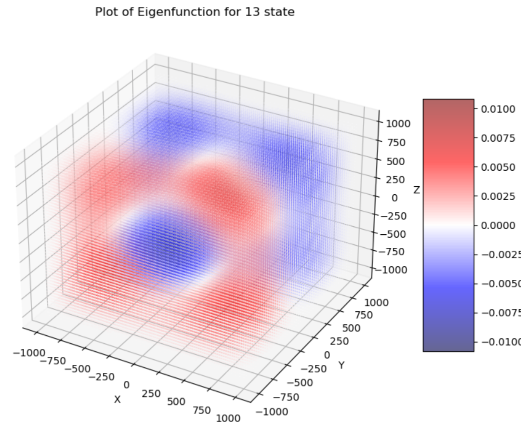

FIG. 141. Plot of the eigenfunctions for 13th state.

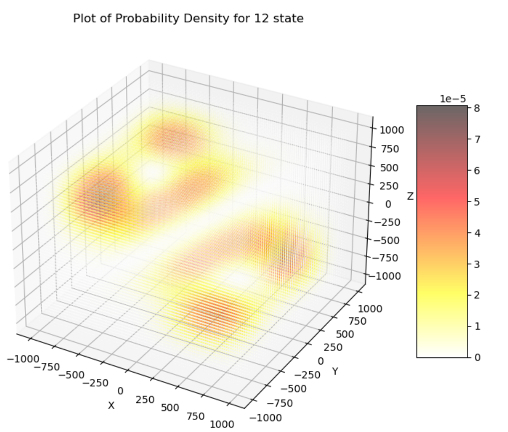

FIG. 140. Plot of the probability density for 12th state.

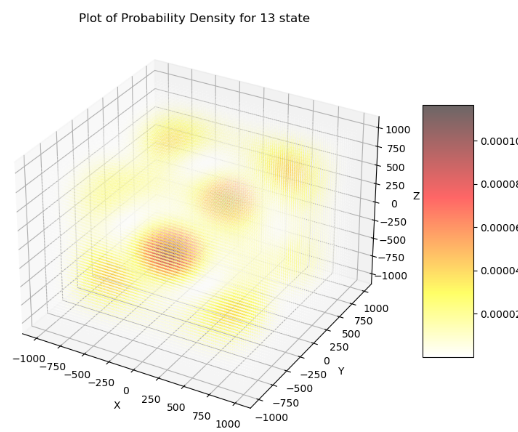

FIG. 142. Plot of the probability density for 13th state.

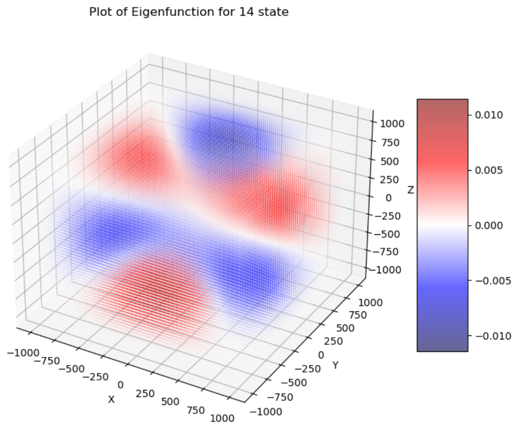

FIG. 143. Plot of the eigenfunctions for 14th state.

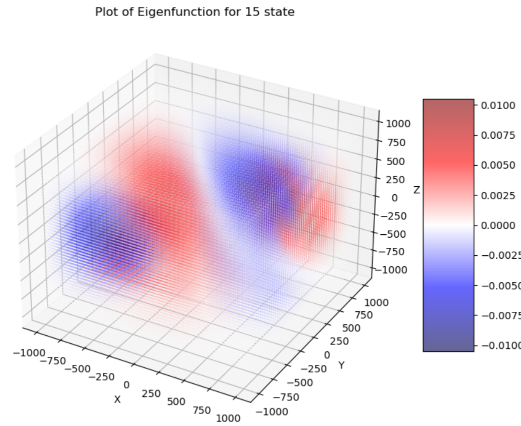

FIG. 145. Plot of the eigenfunctions for 15th state.

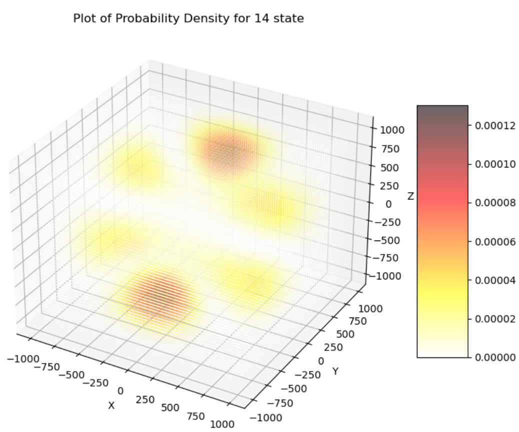

FIG. 144. Plot of the probability density for 14th state.

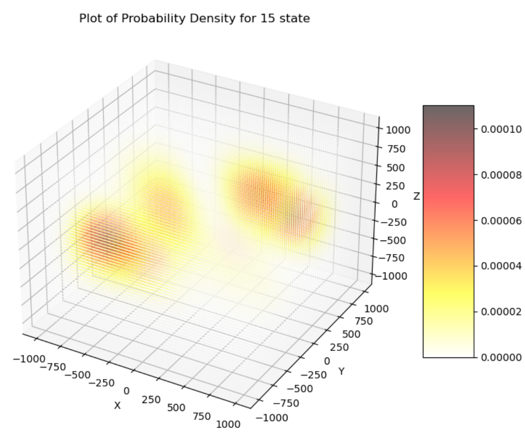

FIG. 146. Plot of the probability density for 15th state.

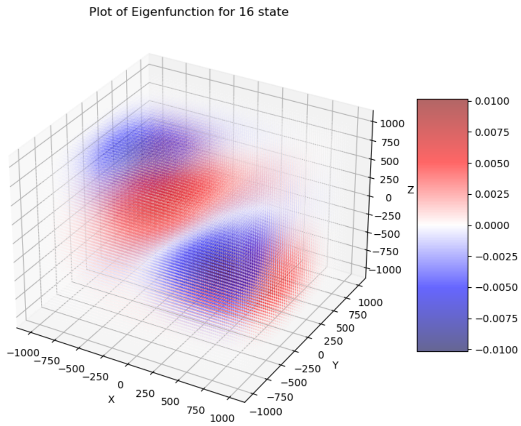

FIG. 147. Plot of the eigenfunctions for 16th state.

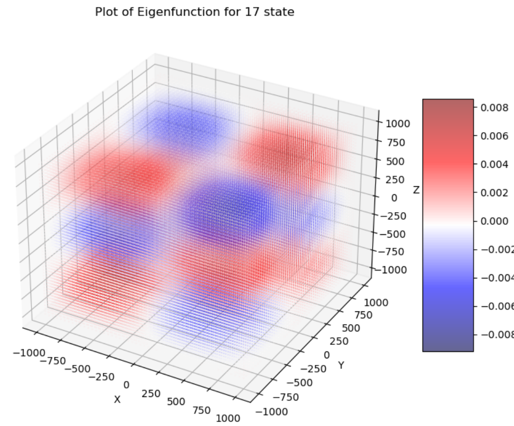

FIG. 149. Plot of the eigenfunctions for 17th state.

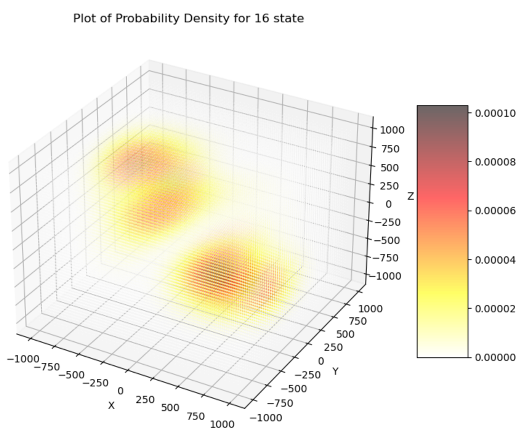

FIG. 148. Plot of the probability density for 16th state.

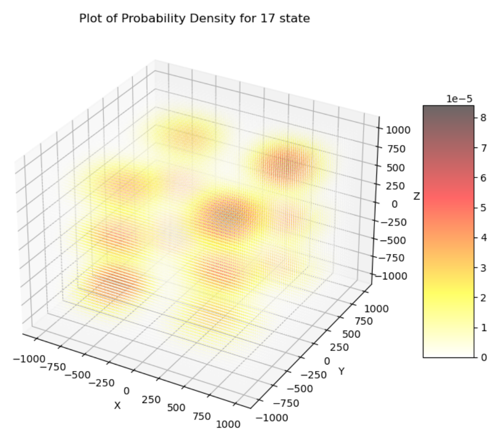

FIG. 150. Plot of the probability density for 17th state.

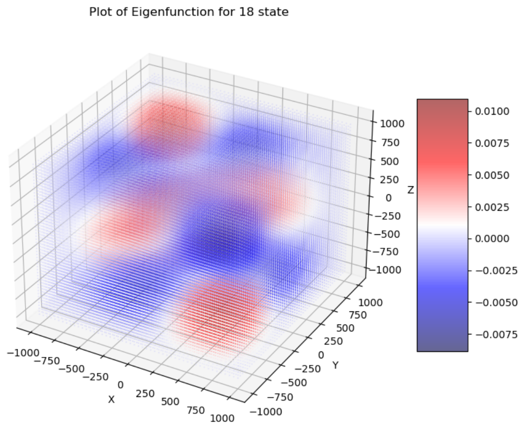

FIG. 151. Plot of the eigenfunctions for 18th state.

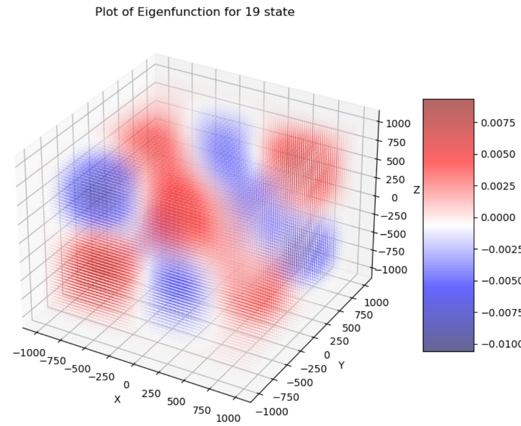

FIG. 153. Plot of the eigenfunctions for 19th state.

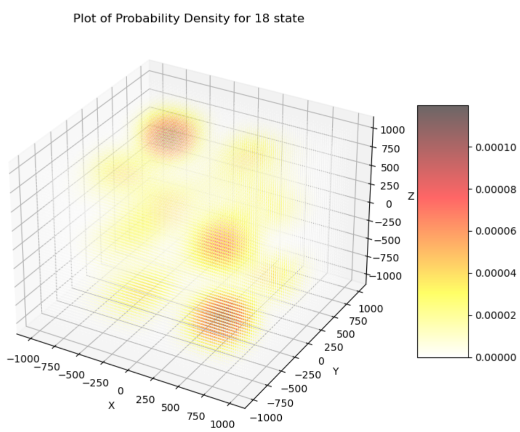

FIG. 152. Plot of the probability density for 18th state.

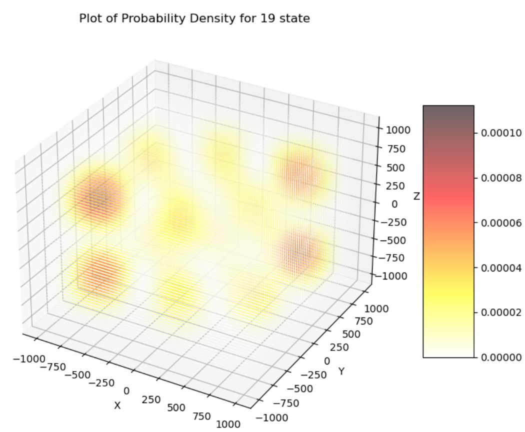

FIG. 154. Plot of the probability density for 19th state.

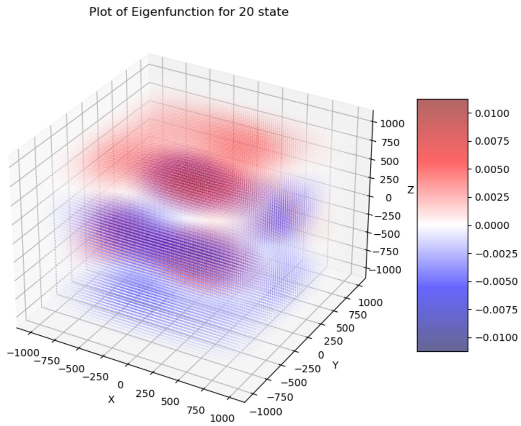

FIG. 155. Plot of the eigenfunctions for 20th state.

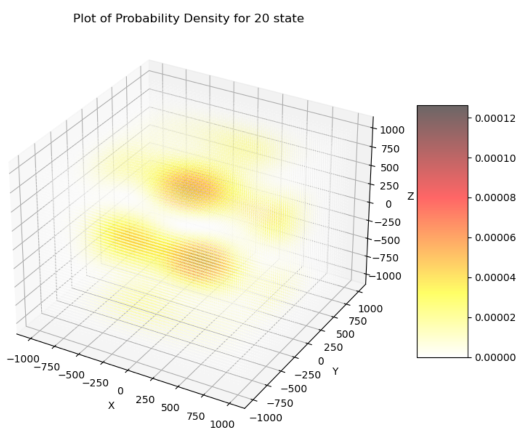

FIG. 156. Plot of the probability density for 20th state.

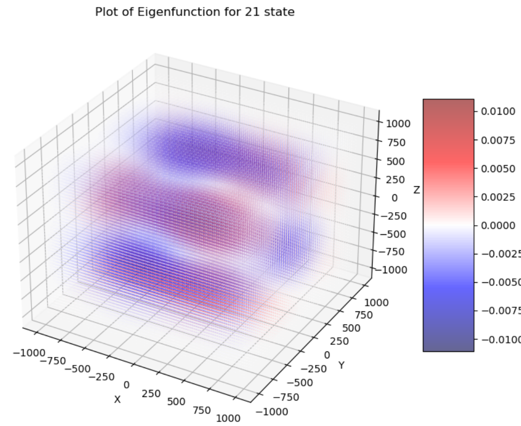

FIG. 157. Plot of the eigenfunctions for 21th state.

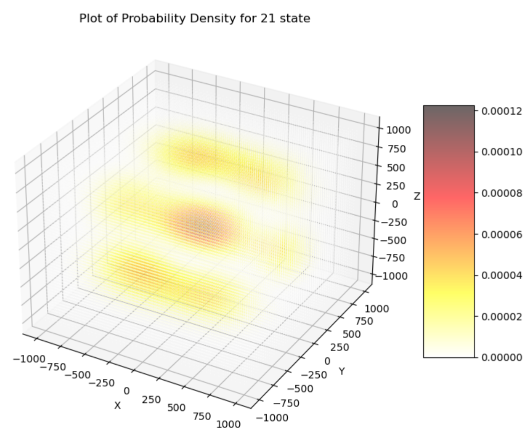

FIG. 158. Plot of the probability density for 21th state.

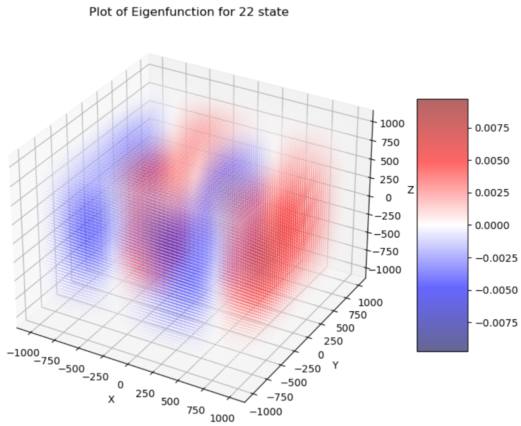

FIG. 159. Plot of the eigenfunctions for 22th state.

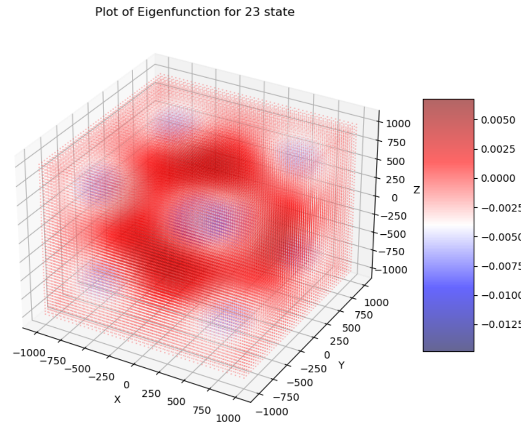

FIG. 161. Plot of the eigenfunctions for 23th state.

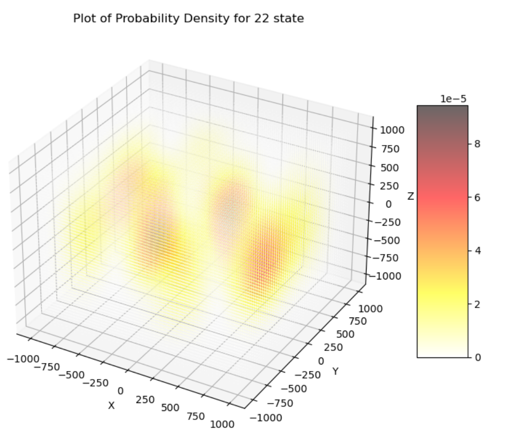

FIG. 160. Plot of the probability density for 22th state.

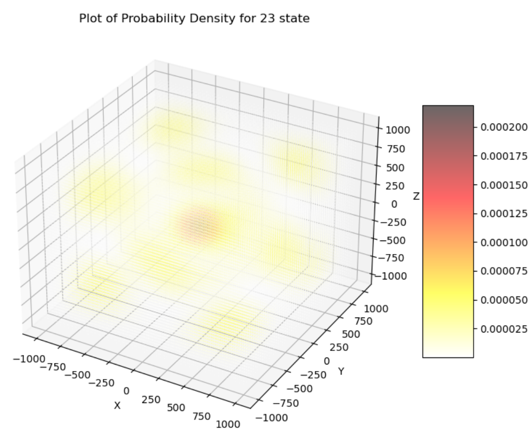

FIG. 162. Plot of the probability density for 23th state.

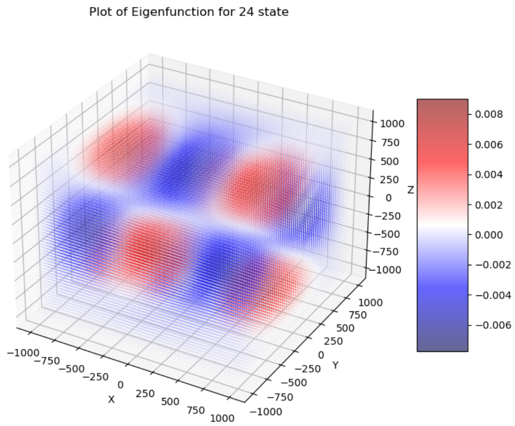

FIG. 163. Plot of the eigenfunctions for 24th state.

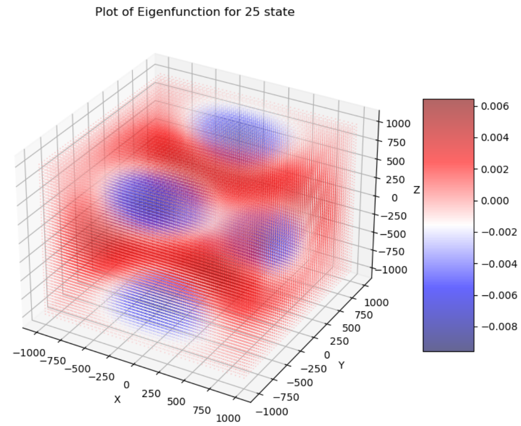

FIG. 165. Plot of the eigenfunctions for 25th state.

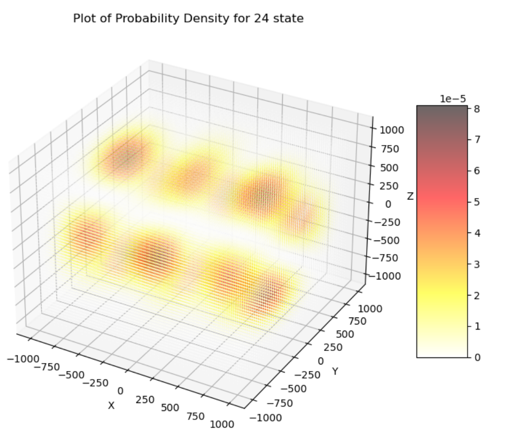

FIG. 164. Plot of the probability density for 24th state.

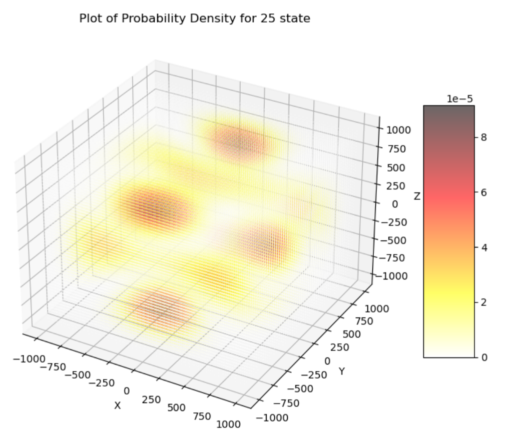

FIG. 166. Plot of the probability density for 25th state.

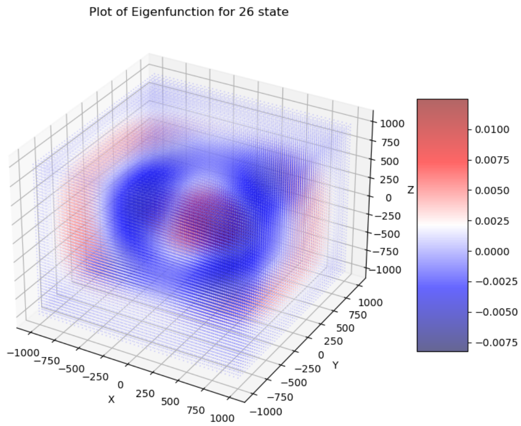

FIG. 167. Plot of the eigenfunctions for 26th state.

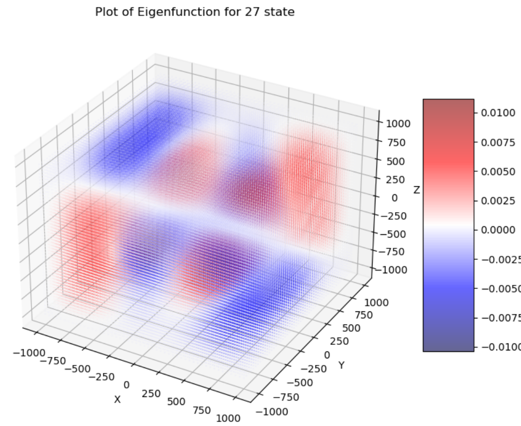

FIG. 169. Plot of the eigenfunctions for 27th state.

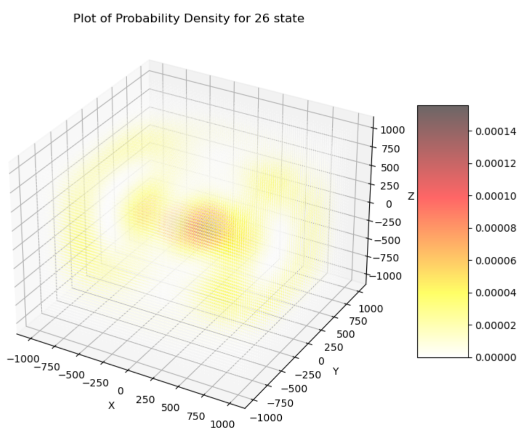

FIG. 168. Plot of the probability density for 26th state.

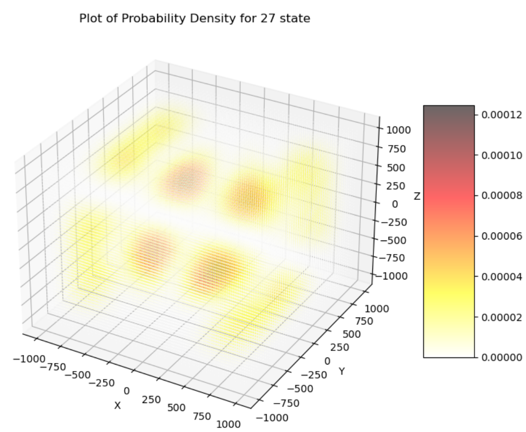

FIG. 170. Plot of the probability density for 27th state.

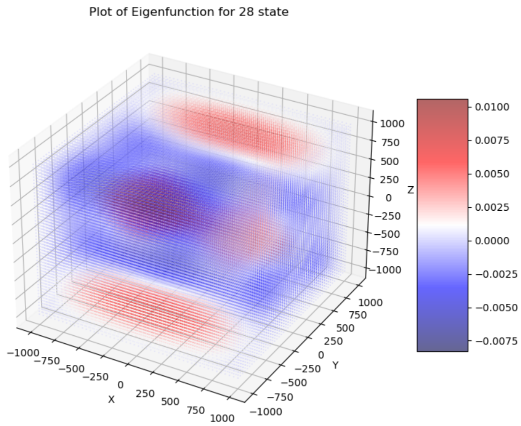

FIG. 171. Plot of the eigenfunctions for 28th state.

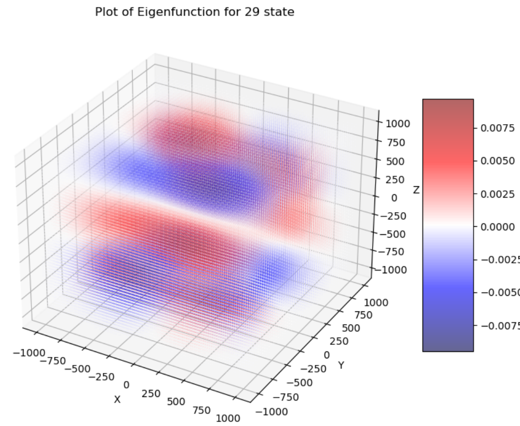

FIG. 173. Plot of the eigenfunctions for 29th state.

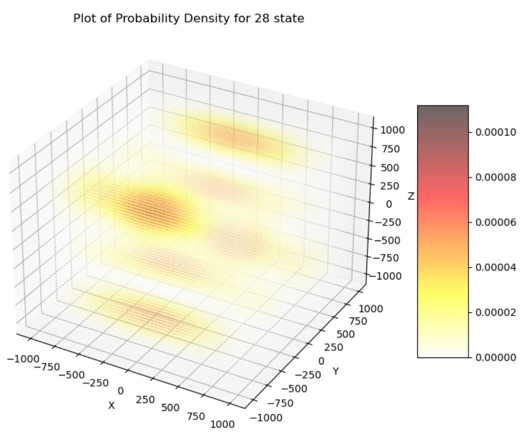

FIG. 172. Plot of the probability density for 28th state.

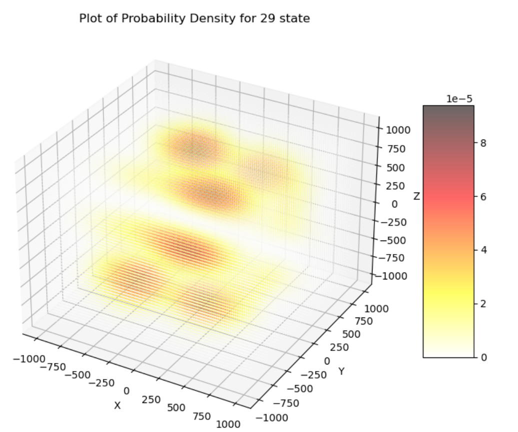

FIG. 174. Plot of the probability density for 29th state.

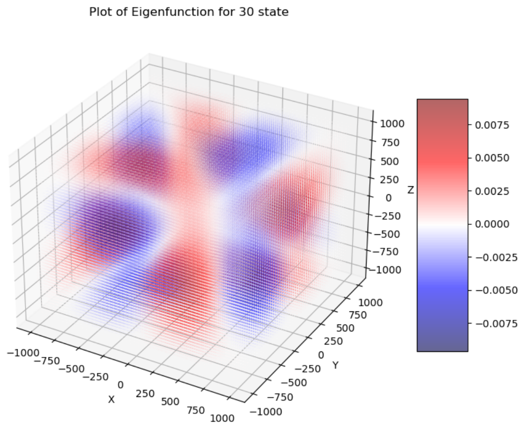

FIG. 175. Plot of the eigenfunctions for 30th state.

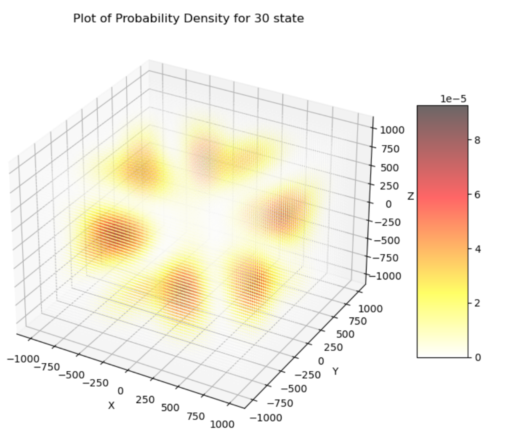

FIG. 176. Plot of the probability density for 30th state.

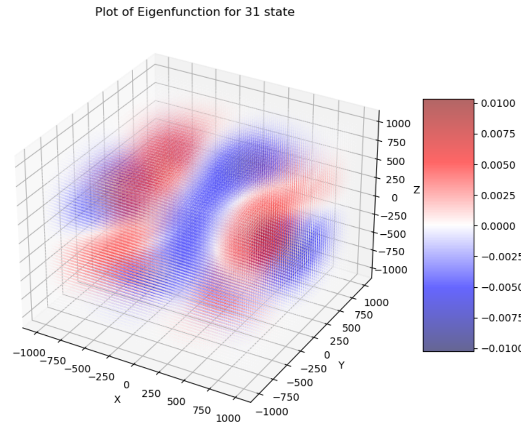

FIG. 177. Plot of the eigenfunctions for 31th state.

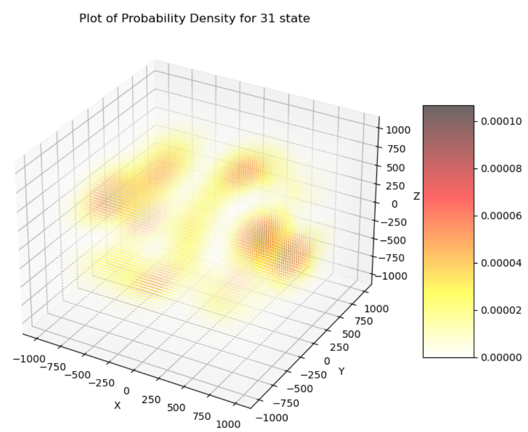

FIG. 178. Plot of the probability density for 31th state.

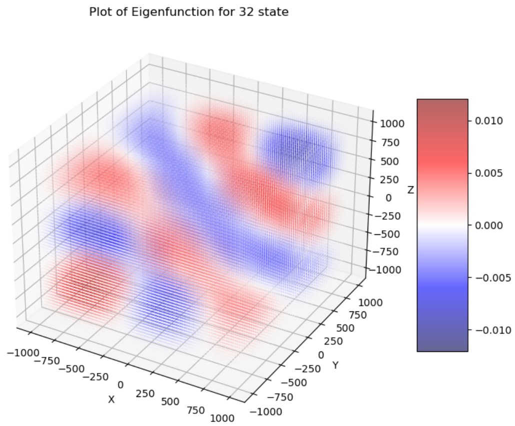

FIG. 179. Plot of the eigenfunctions for 32th state.

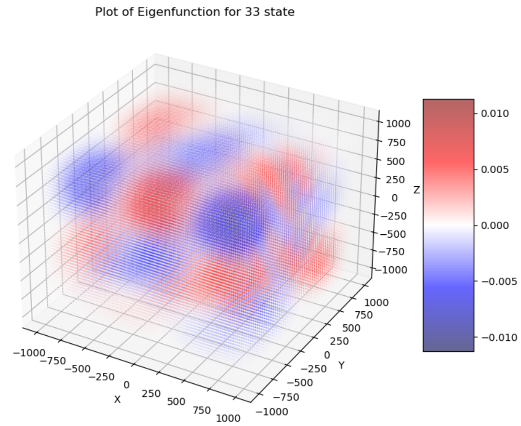

FIG. 181. Plot of the eigenfunctions for 33th state.

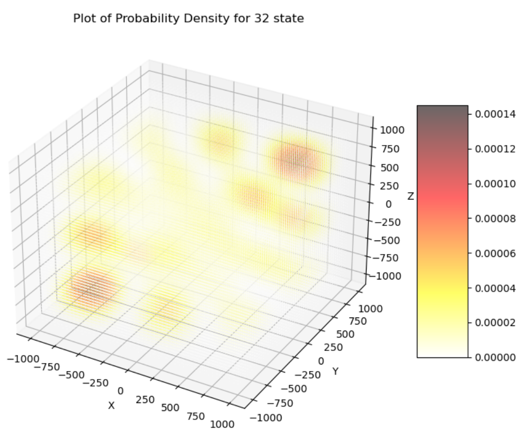

FIG. 180. Plot of the probability density for 32th state.

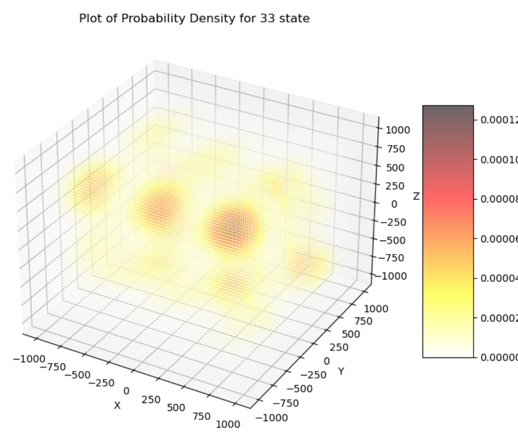

FIG. 182. Plot of the probability density for 33th state.

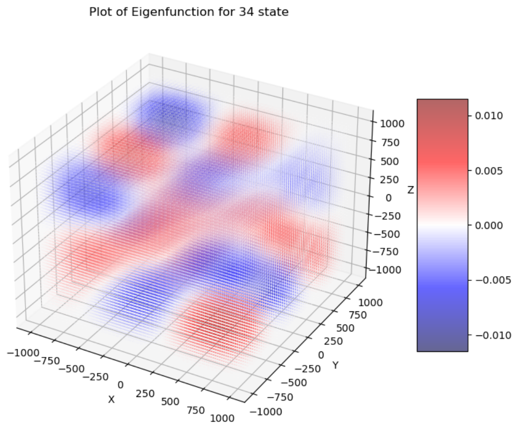

FIG. 183. Plot of the eigenfunctions for 34th state.

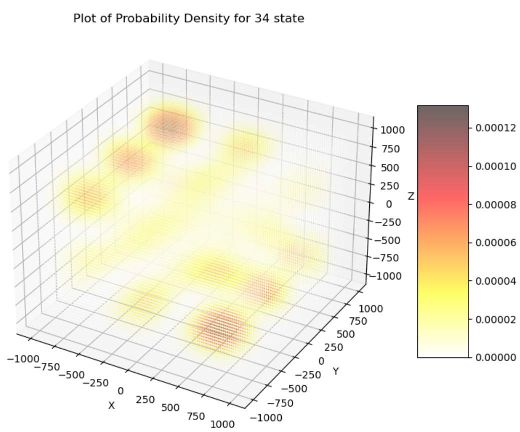

FIG. 184. Plot of the probability density for 34th state.

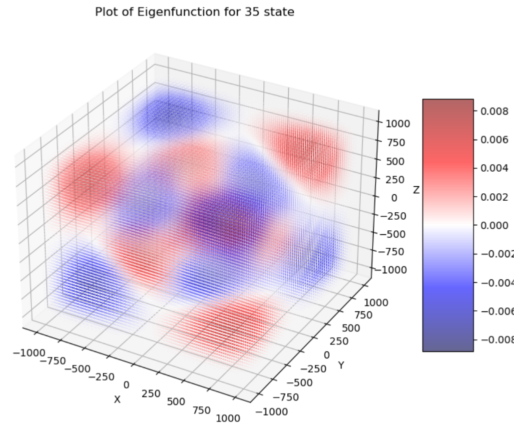

FIG. 185. Plot of the eigenfunctions for 35th state.

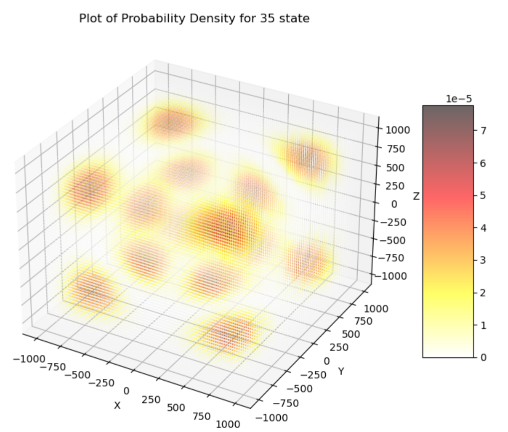

FIG. 186. Plot of the probability density for 35th state.

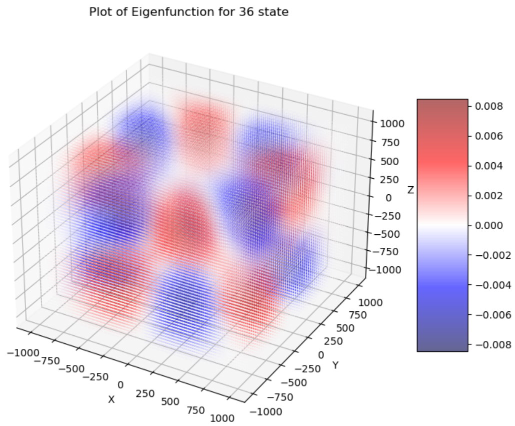

FIG. 187. Plot of the eigenfunctions for 36th state.

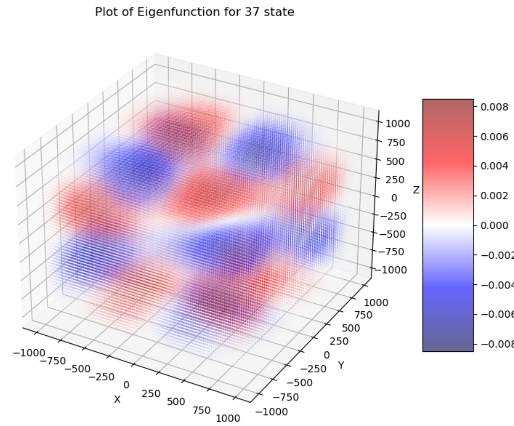

FIG. 189. Plot of the eigenfunctions for 37th state.

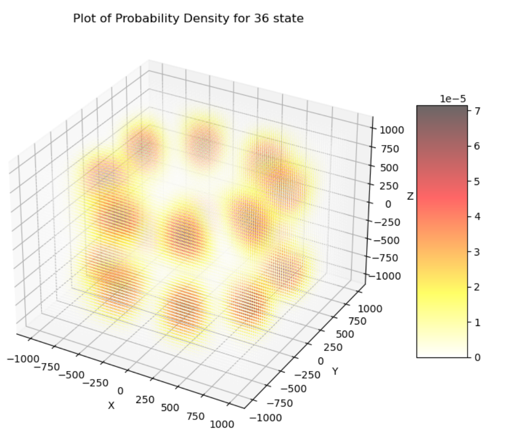

FIG. 188. Plot of the probability density for 36th state.

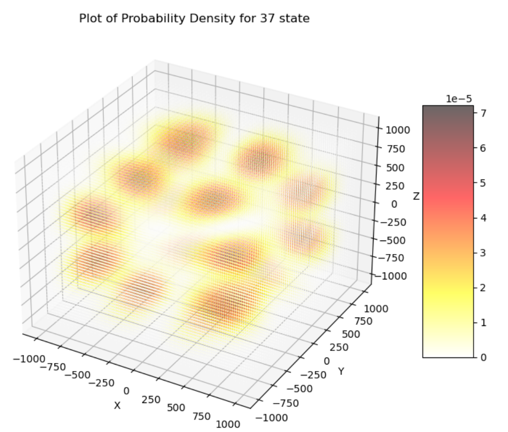

FIG. 190. Plot of the probability density for 37th state.

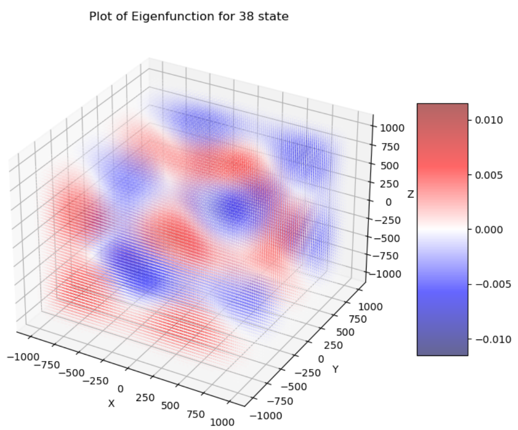

FIG. 191. Plot of the eigenfunctions for 38th state.

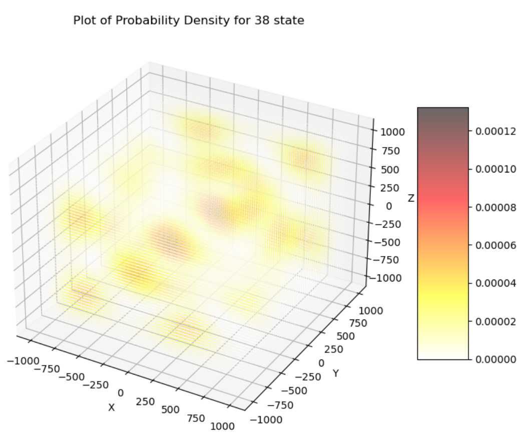

FIG. 192. Plot of the probability density for 38th state.

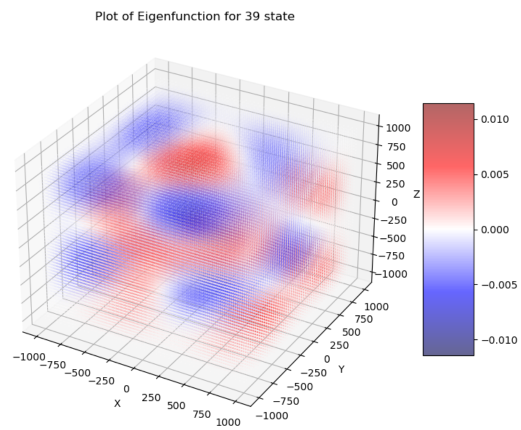

FIG. 193. Plot of the eigenfunctions for 39th state.

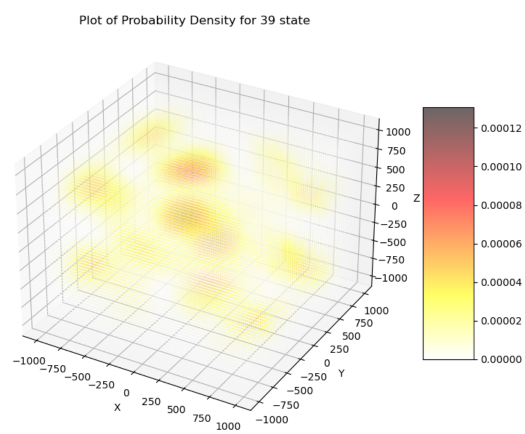

FIG. 194. Plot of the probability density for 39th state.

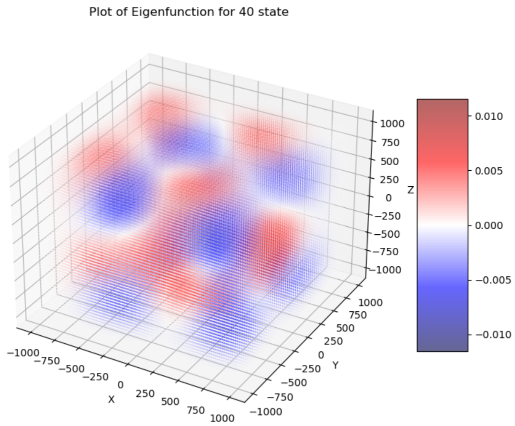

FIG. 195. Plot of the eigenfunctions for 40th state.

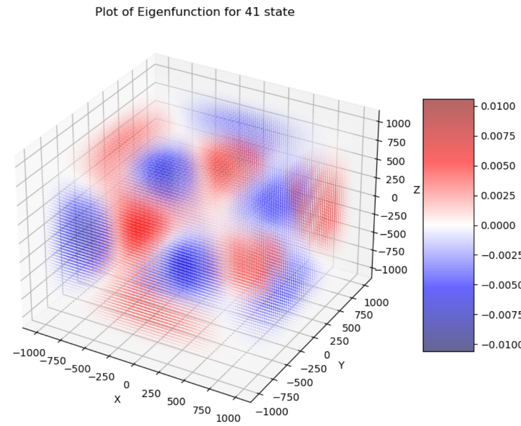

FIG. 197. Plot of the eigenfunctions for 41th state.

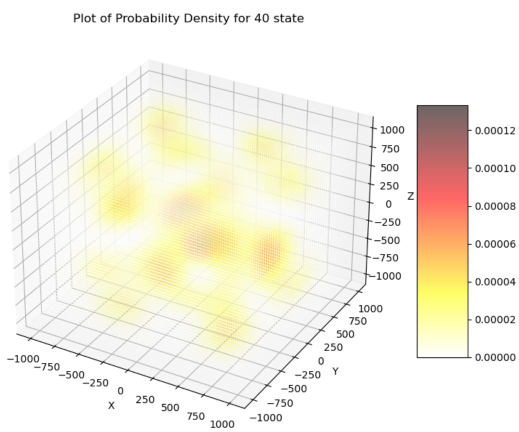

FIG. 196. Plot of the probability density for 40th state.

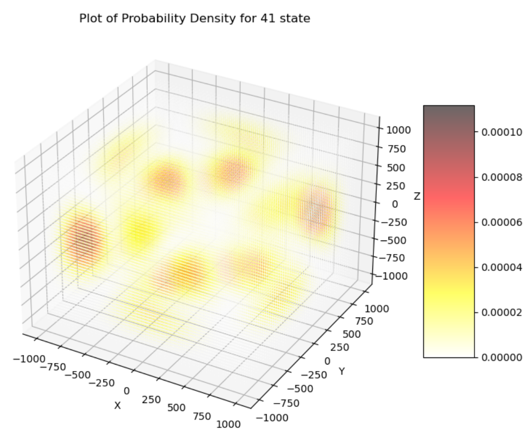

FIG. 198. Plot of the probability density for 41th state.

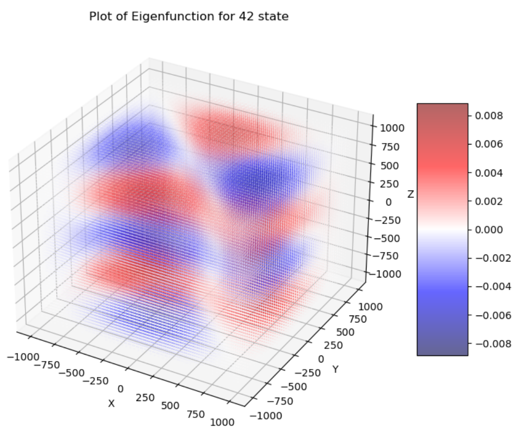

FIG. 199. Plot of the eigenfunctions for 42th state.

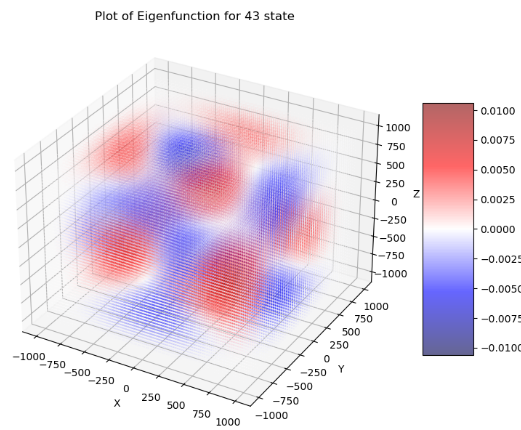

FIG. 201. Plot of the eigenfunctions for 43th state.

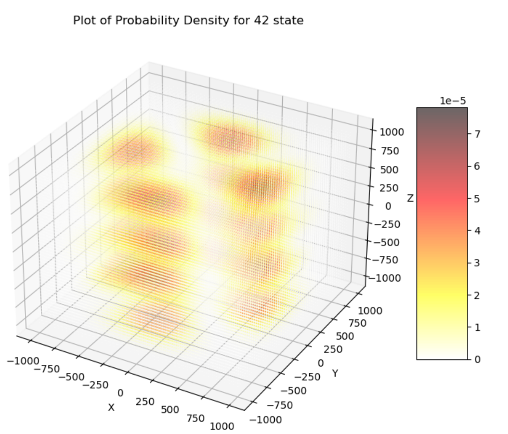

FIG. 200. Plot of the probability density for 42th state.

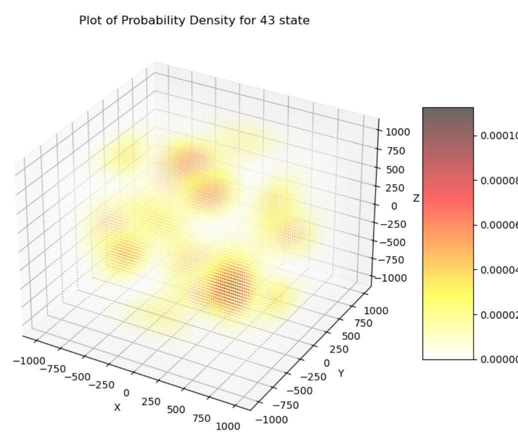

FIG. 202. Plot of the probability density for 43th state.

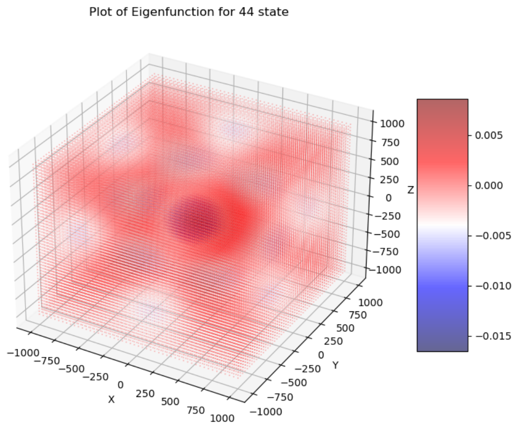

FIG. 203. Plot of the eigenfunctions for 44th state.

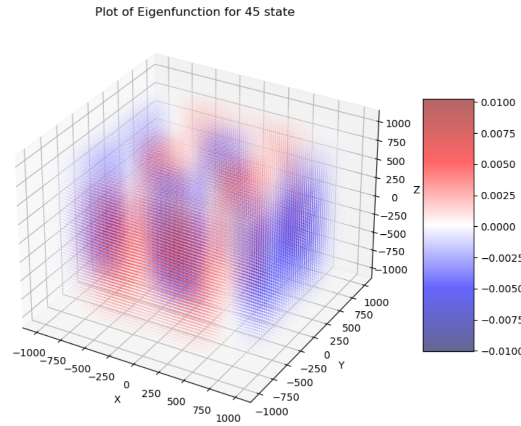

FIG. 205. Plot of the eigenfunctions for 45th state.

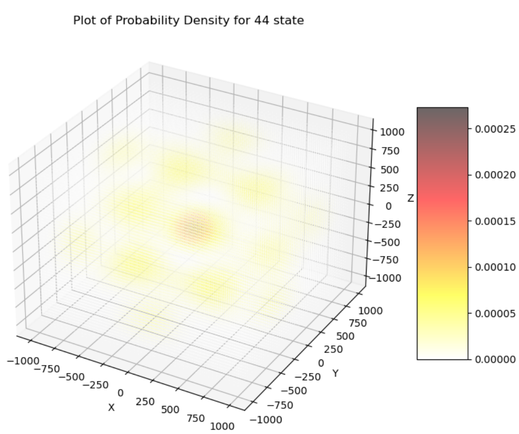

FIG. 204. Plot of the probability density for 44th state.

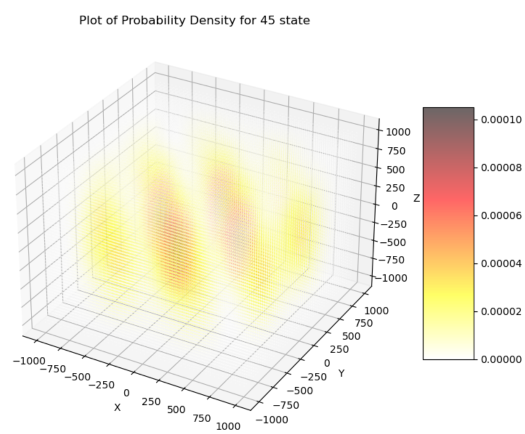

FIG. 206. Plot of the probability density for 45th state.

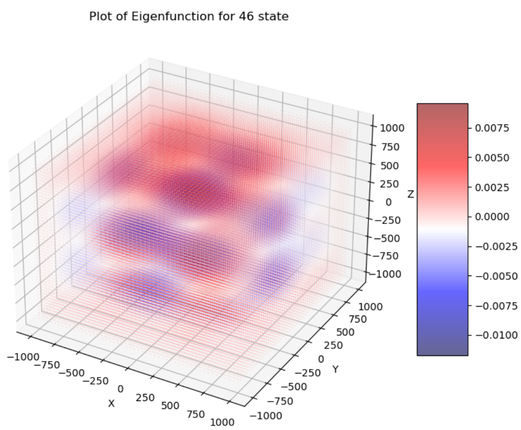

FIG. 207. Plot of the eigenfunctions for 46th state.

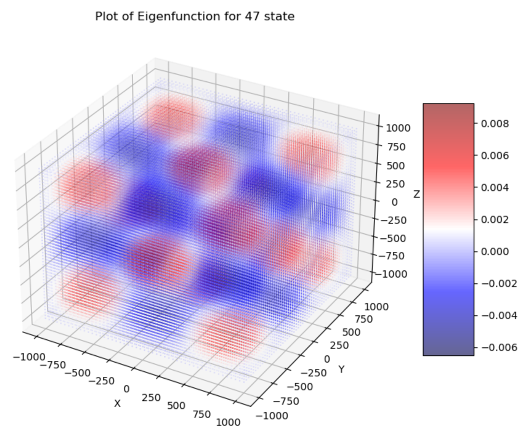

FIG. 209. Plot of the eigenfunctions for 47th state.

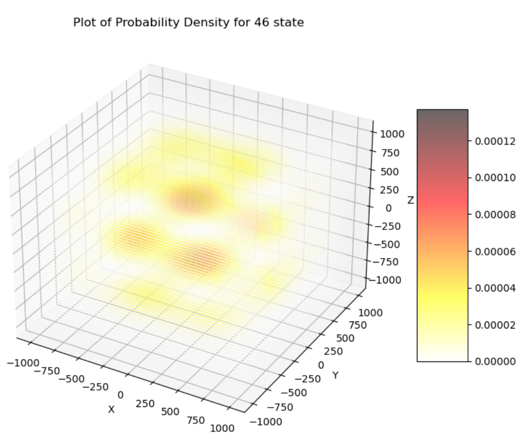

FIG. 208. Plot of the probability density for 46th state.

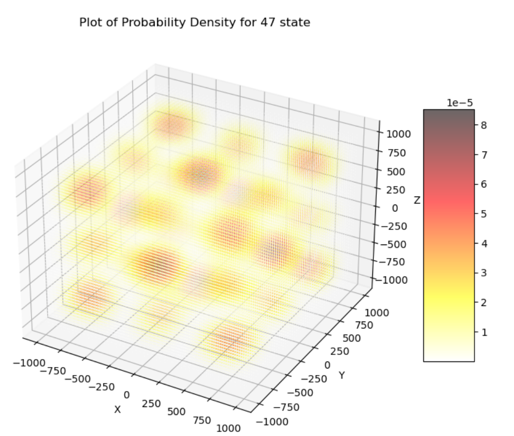

FIG. 210. Plot of the probability density for 47th state.

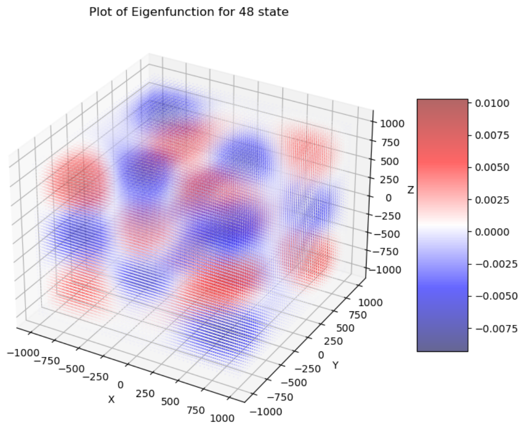

FIG. 211. Plot of the eigenfunctions for 48th state.

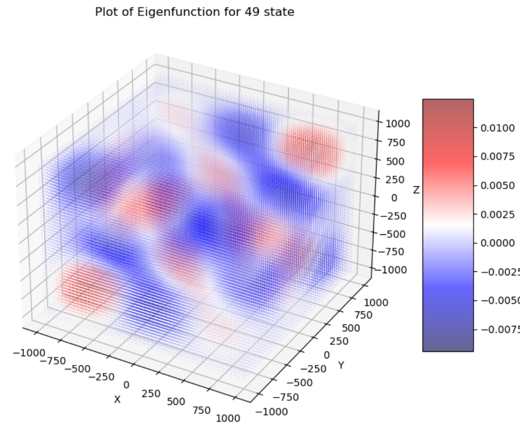

FIG. 213. Plot of the eigenfunctions for 49th state.

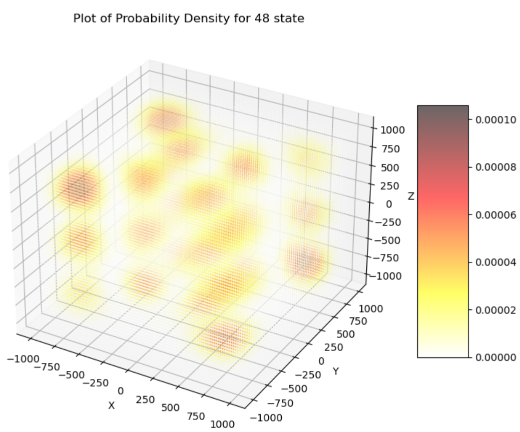

FIG. 212. Plot of the probability density for 48th state.

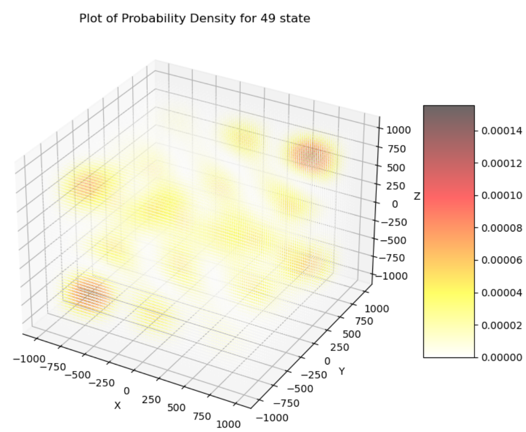

FIG. 214. Plot of the probability density for 49th state.

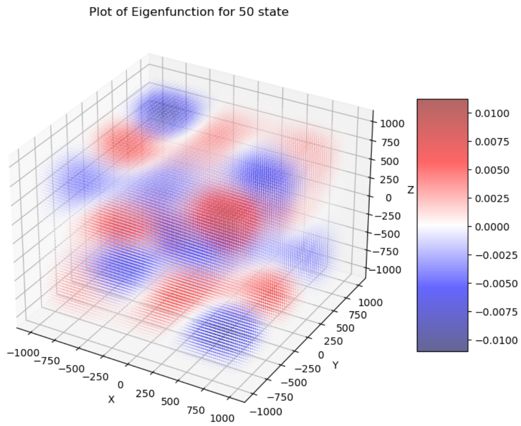

FIG. 215. Plot of the eigenfunctions for 50th state.

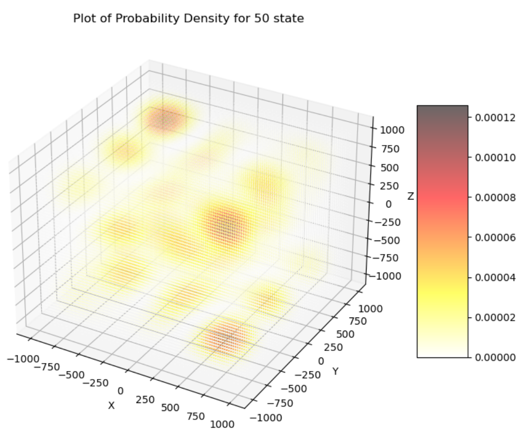

FIG. 216. Plot of the probability density for 50th state.

V. CONCLUSION

In conclusion, the 3D time-independent Schrodinger equation is solved by using the finite difference method. The eigenvalues and eigenvectors from the time-independent Schrodinger equation for an arbitrary potential energy function is found. We have solved the different 3D potential problems, for example, 3D infinite square wall potential, 3D simple harmonic oscillator potential, 3D circular well, 3D torus potential, 3D torus potential, hydrogen atom potential.

Appendix A: Code for calculation of 3D time independent Schrodinger equation

```
from datetime import datetime
start_time = datetime.now()

import numpy as np
from scipy.sparse.linalg import eigsh
from scipy.sparse.linalg import eigs
import matplotlib.pyplot as plt
from scipy import sparse
import os
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

my_path = os.path.abspath(r"#path")


##Create meshgrid for x y z
N = 100
L = 8
X,Y,Z= np.meshgrid(np.linspace(-L/2,L/2,N,
    dtype=float),
np.linspace(-L/2,L/2,N, dtype=float),
np.linspace(-L/2,L/2,N, dtype=float))

##potential m\deltax^2 unit
p = 'torus2'

# ISW
#V = np.zeros([N, N, N])

# SHO
#V =0.0001*((X)**2 + (Y)**2 +(Z)**2)/2.0

# SP
#from scipy.signal import square
#V = 200.0*(1.0 -
    square(2.1*np.pi*np.sqrt((X/L)**2 +
    (Y/L)**2) + (Z/L)**2))

# torus
r = 0.1*L
R = 0.3*L
V = ((-(r/L)**2) + (np.sqrt((X/L)**2 +
    (Y/L)**2)-R/L)**2)+(Z/L)**2)
```

```
# torus1
# r = 0.1*L
# R = 0.4*L
# V = ((-(r/L)**2) + (np.sqrt((X/L)**2 +
    (Y/L)**2)-R/L)**2)+(Z/L)**2)

#HA
#V = -1/np.sqrt((X)**2 + (Y)**2 + (Z)**2)

##create matrix
diag = np.ones([N])
diags = np.array([diag, -2*diag, diag])
D = sparse.spdiags(diags, np.array([-1, 0,1]),
    N, N)
I = np.identity(N)

##define energy
D1 = sparse.kronsum(D,D)
D2 = sparse.kronsum(D1,D)

#D1= np.kron(D,D)
#D2= np.kron(D1,D)

# T = -1/2 * (np.kron(np.kron(D,I),I)
#     + np.kron(np.kron(I,D),I)
#     + np.kron(np.kron(I,I),D))

# T = -1/2 * (sparse.kron(sparse.kron(D,I),I)
#     + sparse.kron(sparse.kron(I,D),I)
#     + sparse.kron(sparse.kron(I,I),D))

T = -1/2 * D2
U = sparse.diags(V.reshape(N**3),(0))
H = T+U

##Solve for eigenvector and eigenvalue
eigenvalues , eigenvectors = eigsh(H, k=11,
    which='SM')
def get_e(n):
return eigenvectors.T[n].reshape((N,N,N))

##plot V
fig = plt.figure(0,figsize=(9,9))
ax = fig.add_subplot(111, projection='3d')
plot0 = ax.scatter3D(X, Y, Z, c=V,
cmap=cm.seismic,
s=0.001,
alpha=0.6,
antialiased=True)

fig.colorbar(plot0, shrink=0.5, aspect=5)
ax.set_xlabel(r'X')
ax.set_ylabel(r'Y')
ax.set_zlabel(r'Z')
ax.set_title("Plot of V")
#plt.rcParams.update({"savefig.facecolor": (0,
    0, 0, 0)})
plt.savefig(os.path.join(my_path,
    'Figure_{}.0.png'.format(p)))
plt.close()
##number of state
for n in range (0,10):
##plot eigenvector
fig = plt.figure(1,figsize=(9,9))
ax = fig.add_subplot(111, projection='3d')
#ax.set_axis_off()
plot1 = ax.scatter3D(X, Y, Z, c=get_e(n),
cmap=cm.seismic,
s=0.001,
alpha=0.6,
antialiased=True)

fig.colorbar(plot1, shrink=0.5, aspect=5)
ax.set_xlabel(r'X')
ax.set_ylabel(r'Y')
ax.set_zlabel(r'Z')
ax.set_title("Plot of Eigenfunction for {}
    state".format(n))
plt.savefig(os.path.join(my_path,
    'Figure_{}_{}.1.png'.format(p,n)))
plt.close()
##plot probability density
fig = plt.figure(2,figsize=(9,9))
ax = fig.add_subplot(111, projection='3d')
#ax.set_axis_off()
plot2 = ax.scatter3D(X, Y, Z, c=get_e(n)**2,
cmap=cm.hot_r,
s=0.001,
alpha=0.6,
antialiased=True)

fig.colorbar(plot2, shrink=0.5, aspect=5)
ax.set_xlabel(r'X')
ax.set_ylabel(r'Y')
ax.set_zlabel(r'Z')
ax.set_title("Plot of Probability Density for {}
    state".format(n))
plt.savefig(os.path.join(my_path,
    'Figure_{}_{}.2.png'.format(p,n)))
plt.close()
##plot eigenvalues
plot3 = plt.figure(3)
alpha = eigenvalues[0]/2
E_a = eigenvalues/alpha
b = np.arange(0, len(eigenvalues),1)
plt.scatter(b, E_a, s=1444, marker="_",
    linewidth=2, zorder=3)
plt.title("Plot of eigenvalues")
plt.xlabel('$(n_{x})^2+(n_{y})^2+(n_{z})^2$')
plt.ylabel(r'$mE/\hbar^2$')

c = ['$E_{}$'.format(i) for i in
    range(0,len(eigenvalues))]
for i, txt in enumerate(c):
plt.annotate(txt, (np.arange(0,
    len(eigenvalues),1)[i], E_a[i]), ha="center")

plt.savefig(os.path.join(my_path,
    'Figure_{}.e.pdf'.format(p)))
plt.show()
plt.close('all')

end_time = datetime.now()
## time log
time = end_time - start_time
times = time.total_seconds()
print('Time: {}'.format(times))
```

```
test = '{}\n'.format(times)

name = r"Timedata"

with open(os.path.join(my_path, name+".txt" ),
    mode='a') as file:
file.write(test)
```


[1] D. Griffiths, *Introduction of Quantum Mechanics* (Prentice Hall, Inc., 1995).

[2] A. Doss, Solving toeplitz systems of equations and matrix conditioning.

[3] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis* (Cambridge University Press, 1991).

[4] A. Noreen and K. Olaussen, A python class for higher-dimensional schrödinger equations (2015), arXiv:1503.04607 [physics.comp-ph].

[5] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, Array programming with NumPy, Nature **585**, 357 (2020).

[6] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods **17**, 261 (2020).