Anistortopic Oscillations and Electron orbitals in 3D

Aleksandr Petrosyan
ap886, Queens' College

Analysis

Oscillations

Isotropic Oscillations

Figure 1 shows the trajectory traced during the simulation. The initial conditions were set to be $k = \begin{pmatrix}10\\10\\10\end{pmatrix}, x_0 = \begin{pmatrix}10\\10\\10\end{pmatrix}, v_0 = \begin{pmatrix}20\\-0.4\\0\end{pmatrix}$ (units not specified)

The simulation is only a numerical approximate solution to the Physical problem, meaning that the curves don't re-trace themselves exactly. However a pattern is notable: the curve itself could be an ellipse, since the approximate curves converge very closely to that figure.

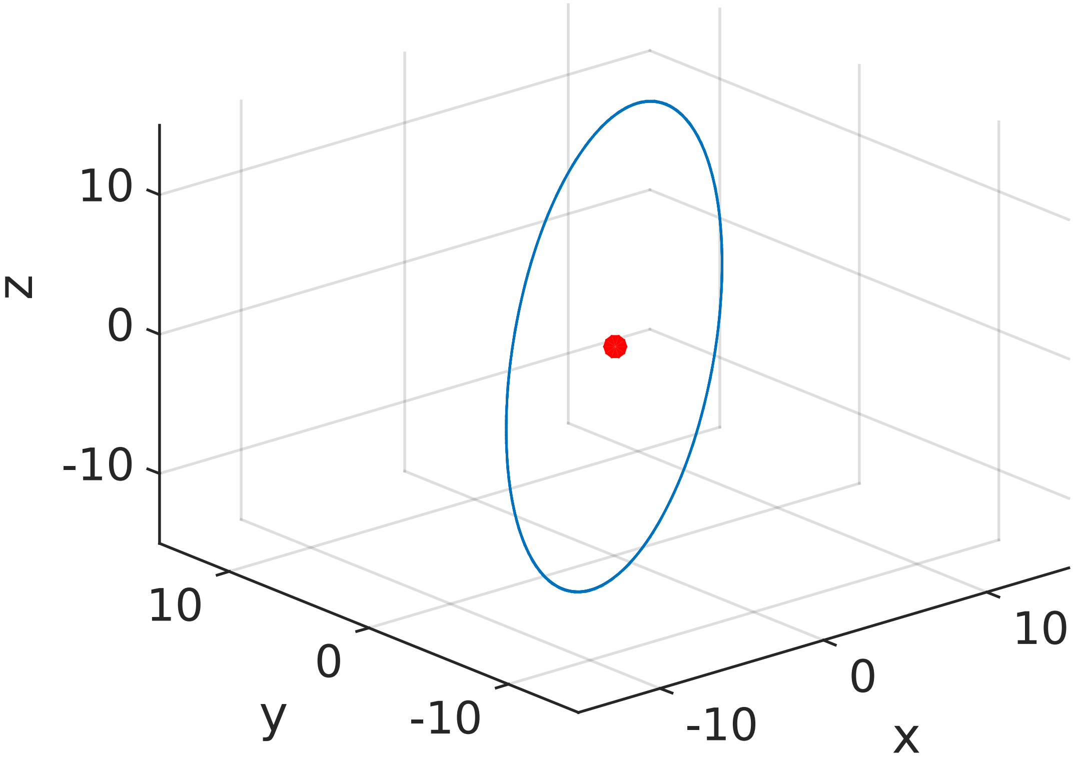

Figure 1: Isotropic Oscillations. The figure clearly Shows that the trajectory traced during one cycle of oscillation is an Ellipse.

Anisotropic Oscillations (Coplanar case)

Figure 2 shows the result of the simulation in the anisotropic case when the mass is initially not displaced and not moving in the vertical (e.g. z direction). Initial conditions used in this case:

$k = \begin{pmatrix} 10 \\ 17 \\ 10 \end{pmatrix}, x_0 = \begin{pmatrix} 10 \\ 10 \\ 0 \end{pmatrix}, v_0 = \begin{pmatrix} 20 \\ -0.4 \\ 0 \end{pmatrix}$ (units not specified)

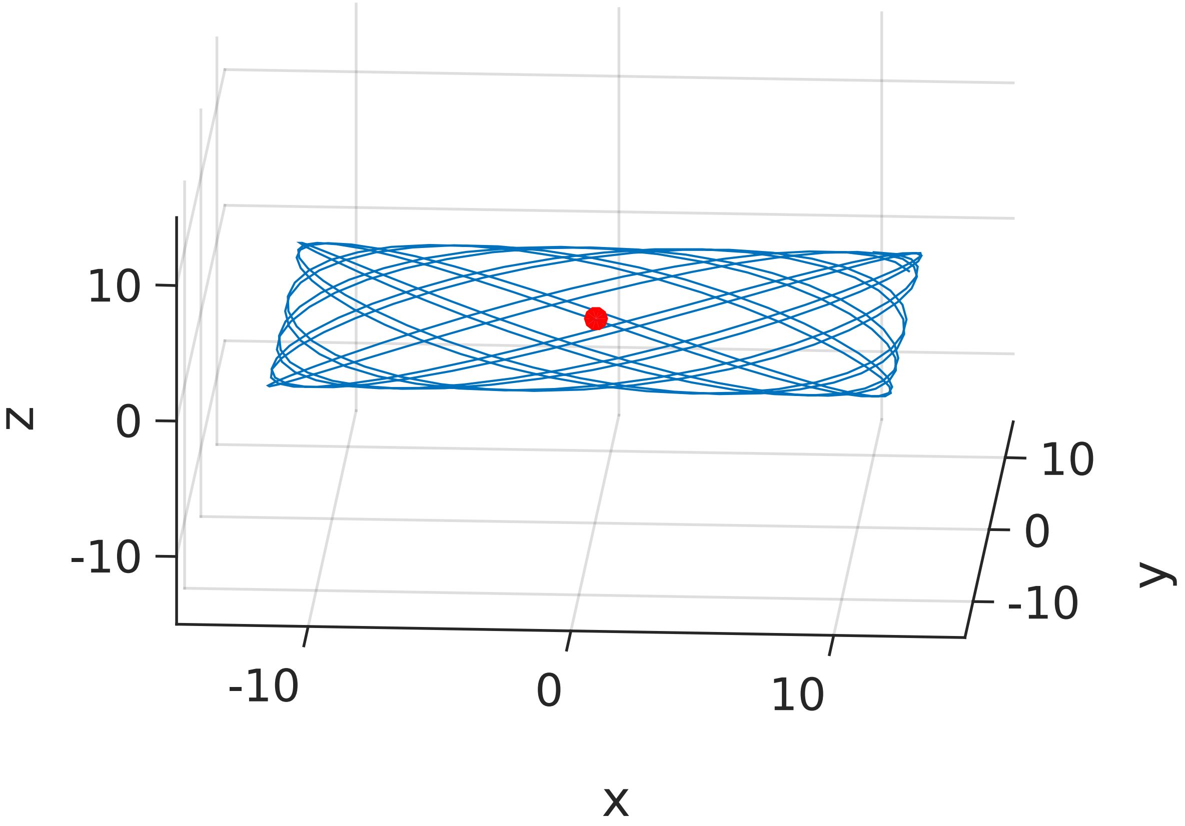

Figure 2: Anisotropic Oscillations. It can be observed that the trajectory lies inside the plane (x0y). The curve itself is known never to re-trace itself and unifromly fill the interior of a rectangle.

Anisotropic Oscillations (Non-coplanar case)

Figure 3 shows a more general type of oscillations that occur if the applied initial conditions (the same spring constants, different initial position and velocity) are:

$$k = \begin{pmatrix} 10 \\ 17 \\ 10 \end{pmatrix}, x_0 = \begin{pmatrix} 10 \\ 10 \\ 5 \end{pmatrix}, v_0 = \begin{pmatrix} 20 \\ -0.4 \\ -0.9 \end{pmatrix}, \text{(units not specified)}$$

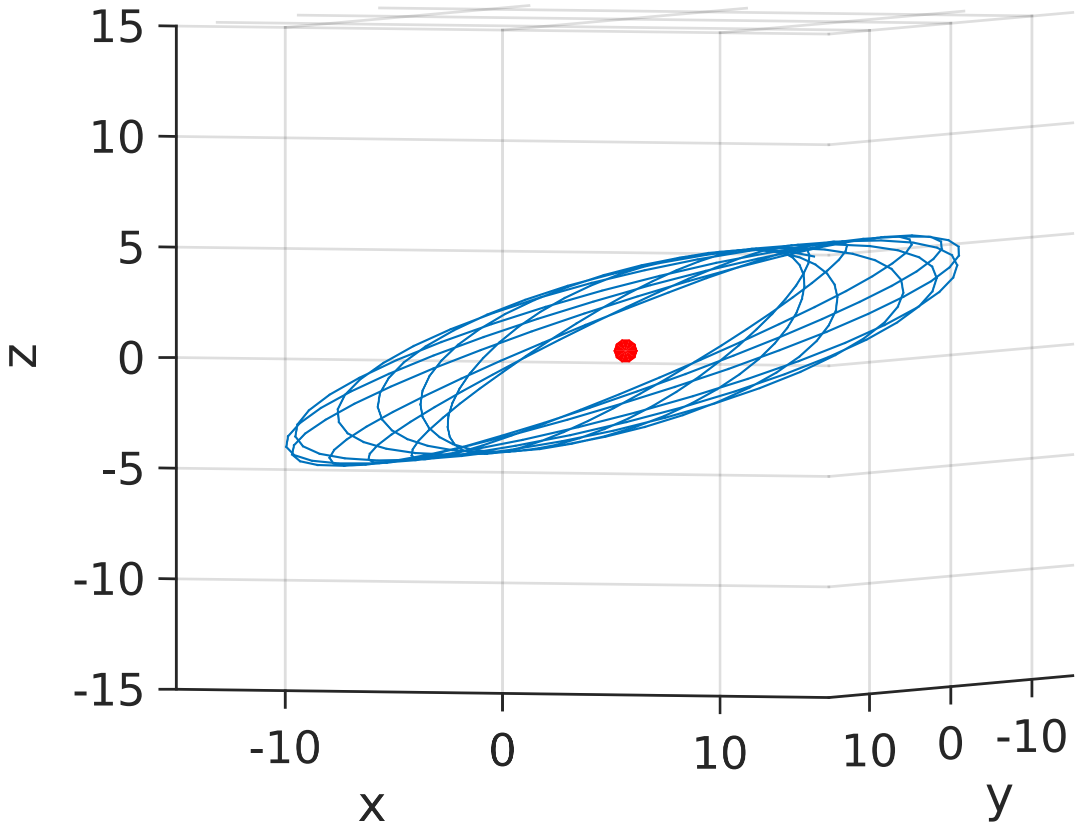

Figure 3: (Singly) Anisotropic Non-coplanar Oscillations. Note that the curve itself is reminiscent of the Langevin curves (see figure 2). All of the points cotained along that curve lie along an elliptic cylinder. This is not the most general case, hwoever since the spring constants along the vertical and one of the horizontal directions are the same.

Estimation of the Quality of the Numerical simulation

The Verlet's algorithm of simulation is merely an approximation. It assumes that time runs in discrete intervals during which the system doesn't change its state. During each time step the differential of the velocity is replaced by the increment which is only a rough approximation accurate depending on our choice of time intervals. For the Initial simulations, the time was quantised to the nearest $dt = 0.05$ unit, which in our case is not specified. The Verlets algorith gets more accurate as we decreease the time step. Thus the given time approxiamtion yields the acceptable accuracy.

However we can also exaggerate the time-quantisation to the nearest $dt = 0.2$; nearly four times the previous value.

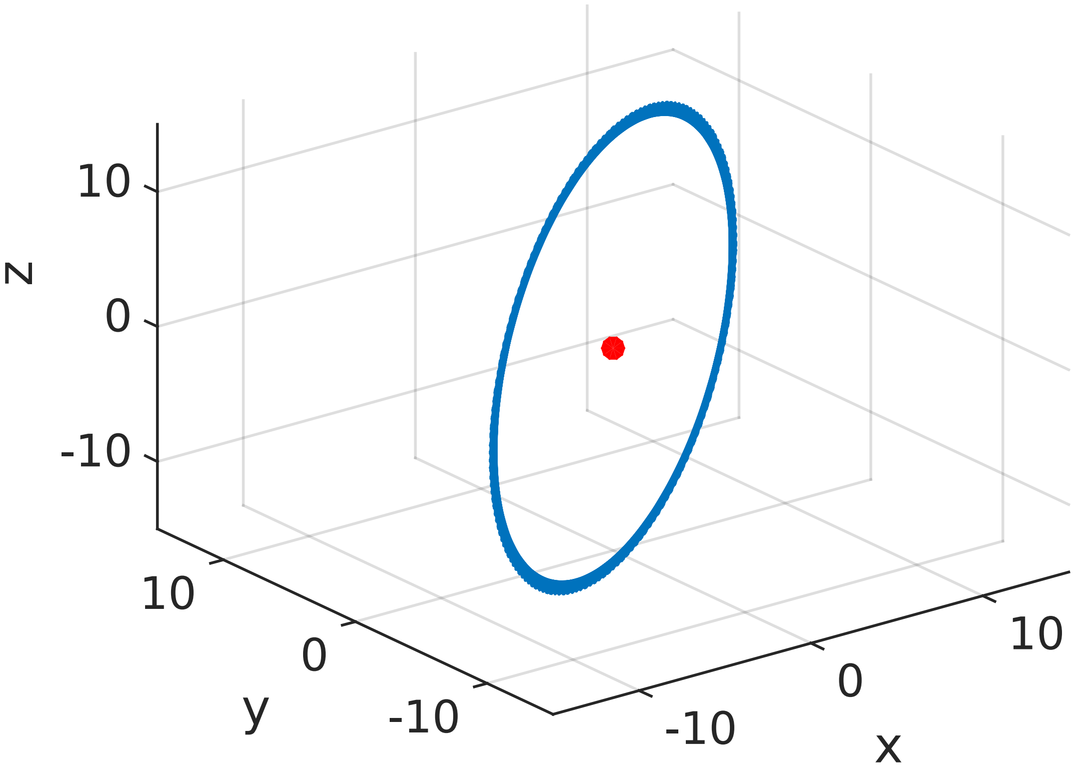

Figure 4: The simulation depicted in figure 1 with different time quantisation constant. The spacing between the curve(s) is noticeably larger, creating the effect of a thickened ellipse. For infinitesimally small time-quantisation constant the curve converges to an ellipse.

Electron Orbitals

The Following Figures show various plots of Electron Orbitals.

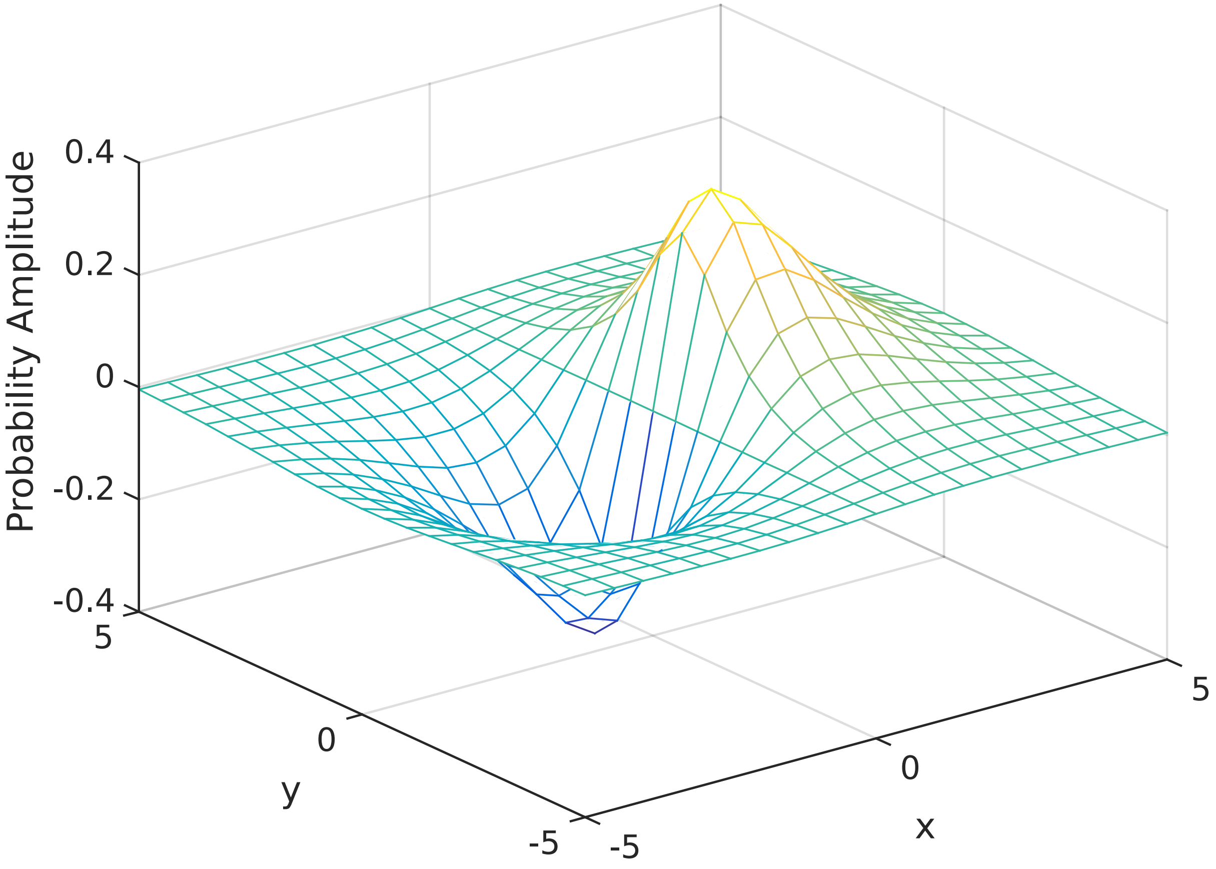

Figure 5: Plot of the Probability Amplitude (Wave function) of the P_x Orbital

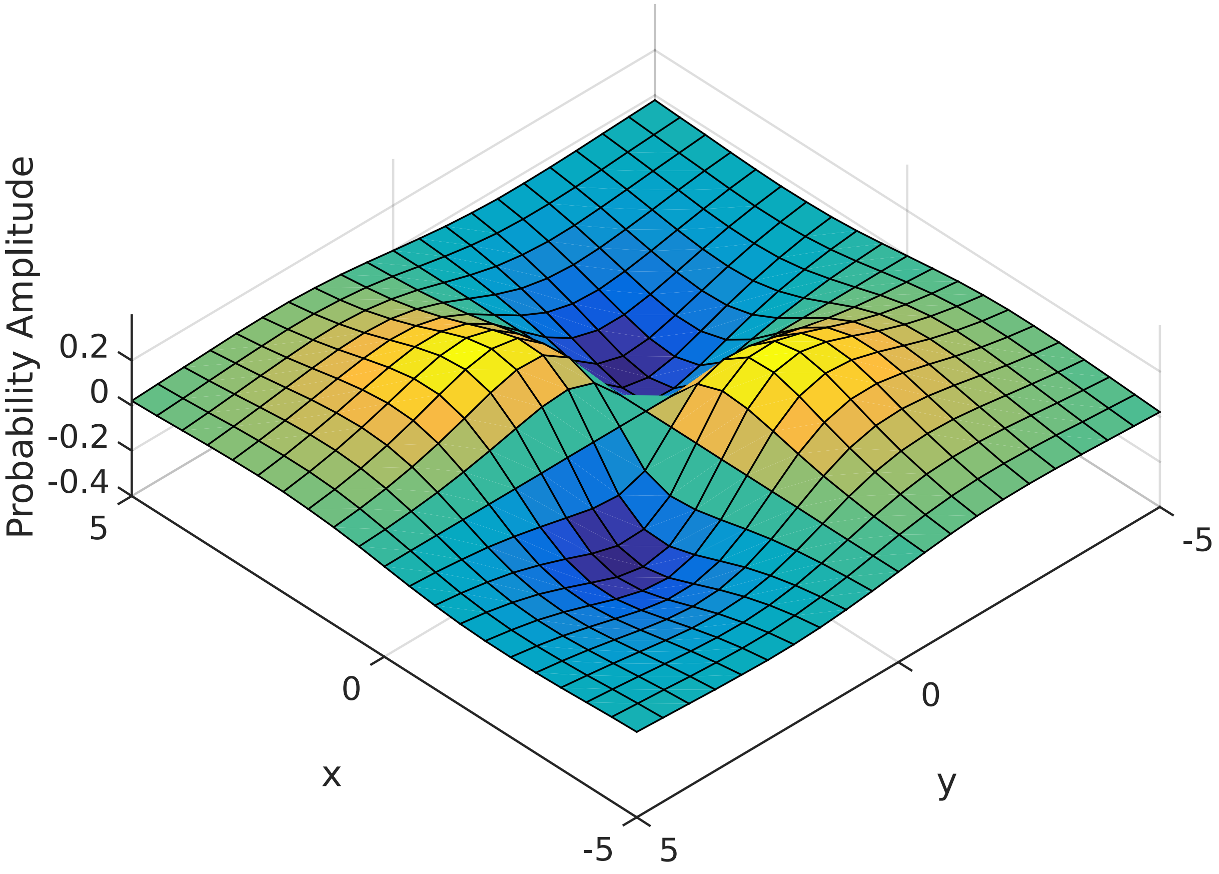

Figure 6: Plot of the Probability Amplitude (Wave function of the d_{xy} Orbital)

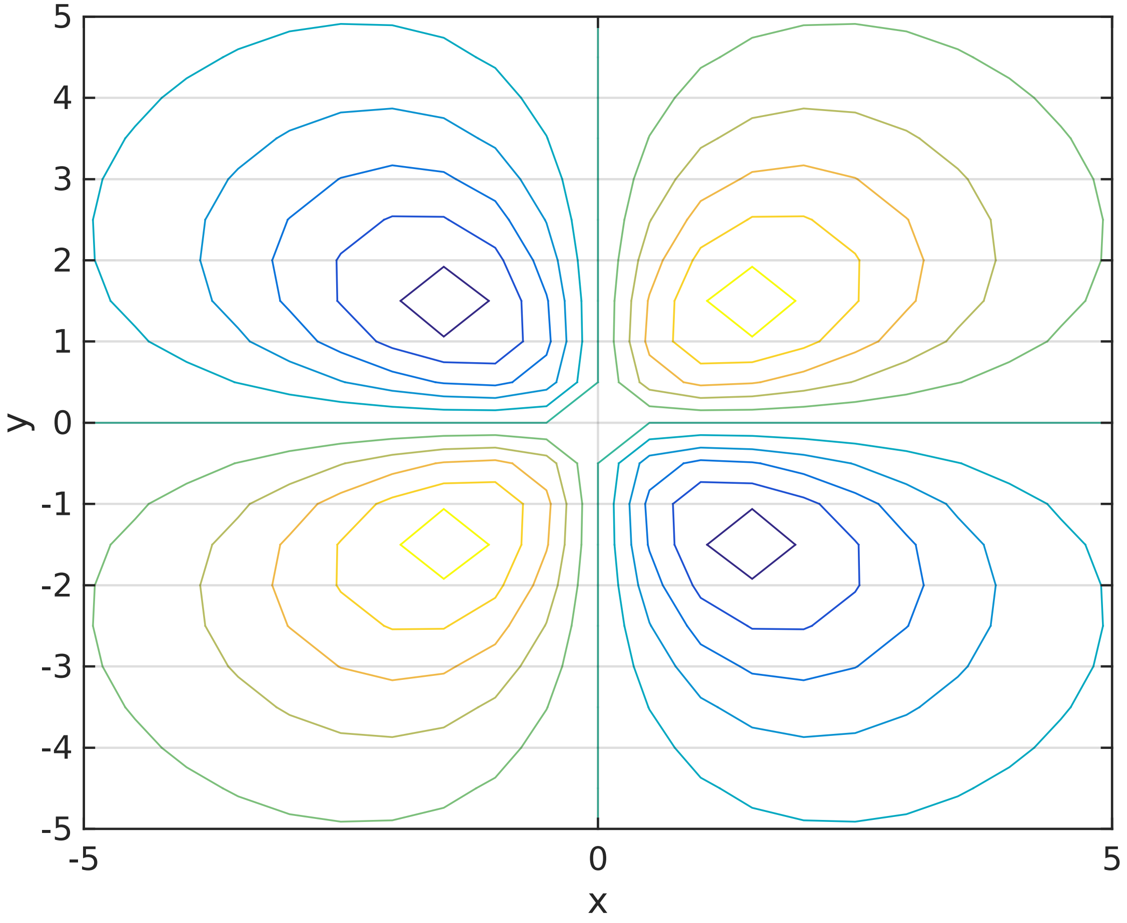

Figure 7: Contour Plot of the Probability Amplitude (Wave function of the d_{xy} Orbital)

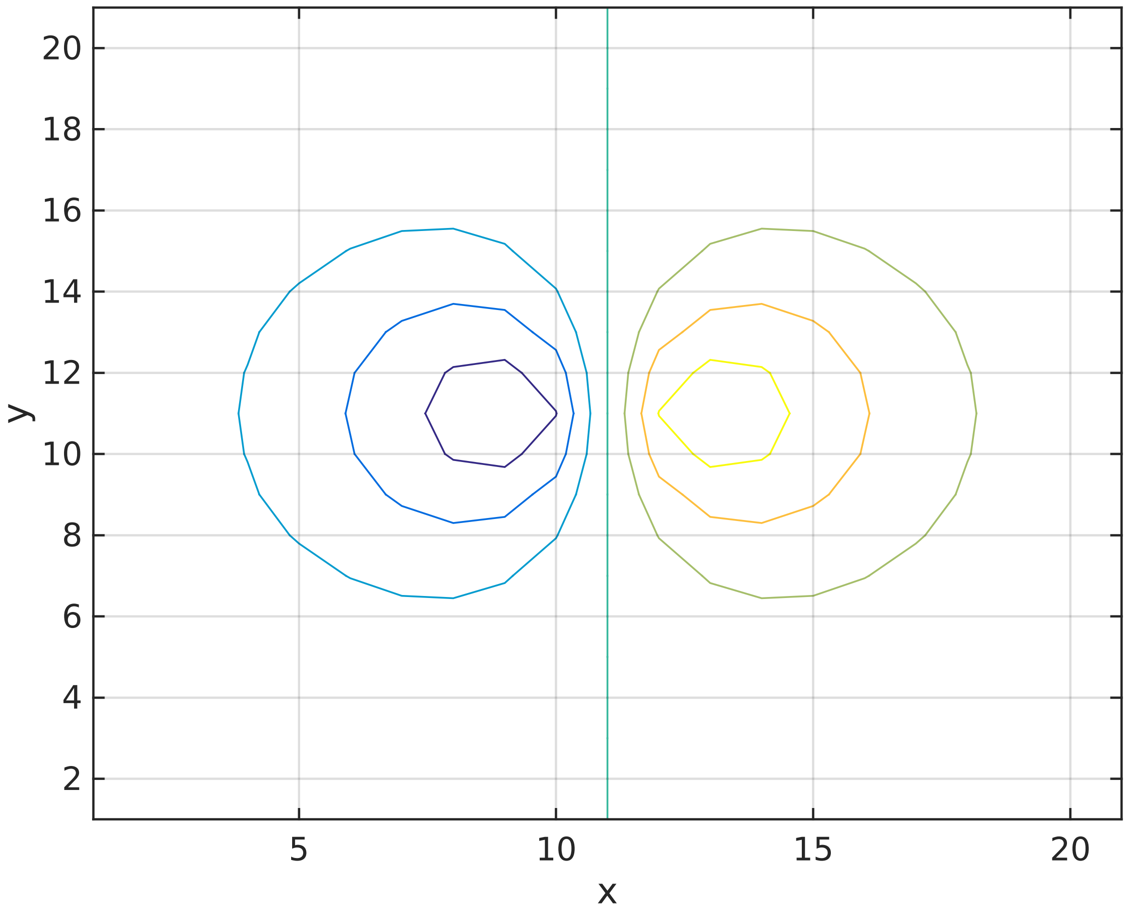

Figure 8: Contour Plot of the Probability Amplitude (Wave function of the P_x Orbital)

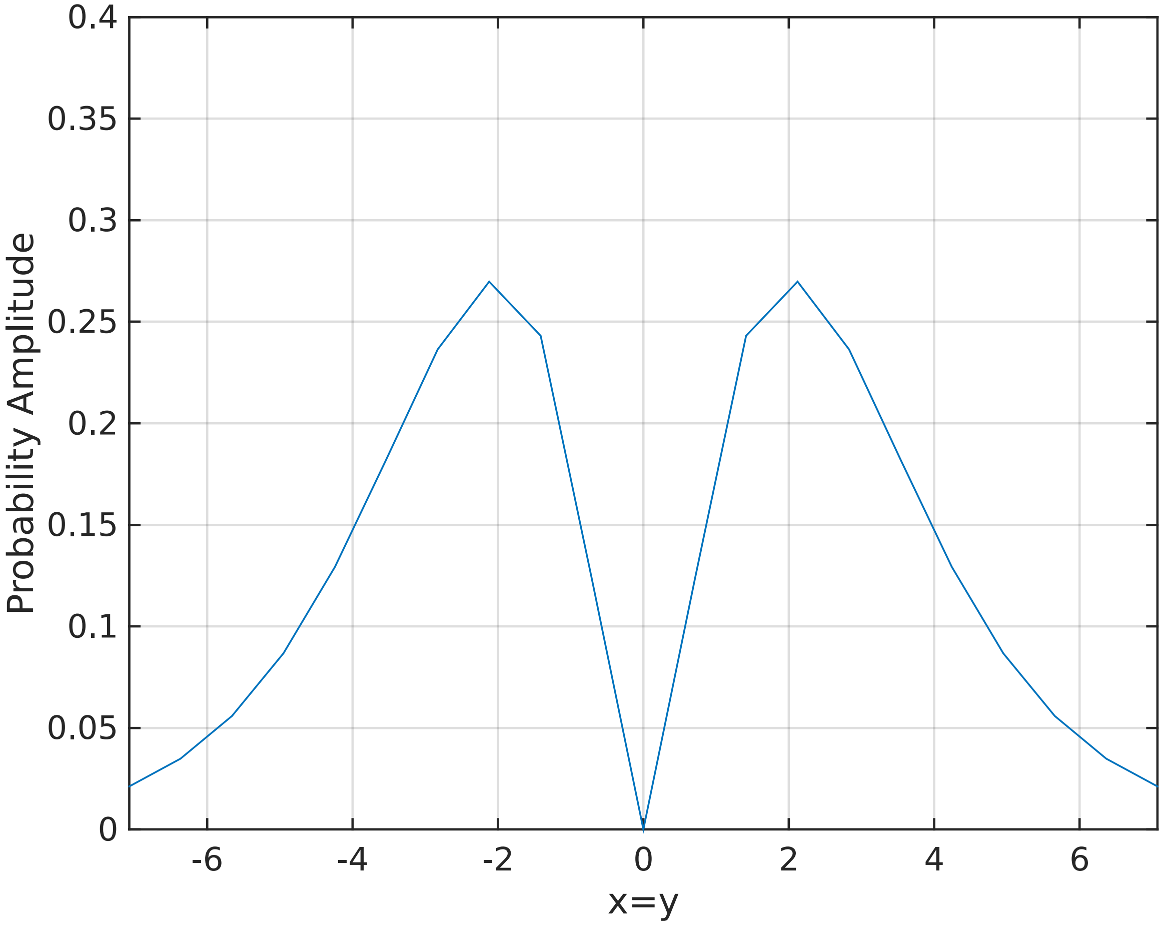

Figure 9: Slice of the d_{xy} Orbital Probability Amplitude through the line $x = y$.

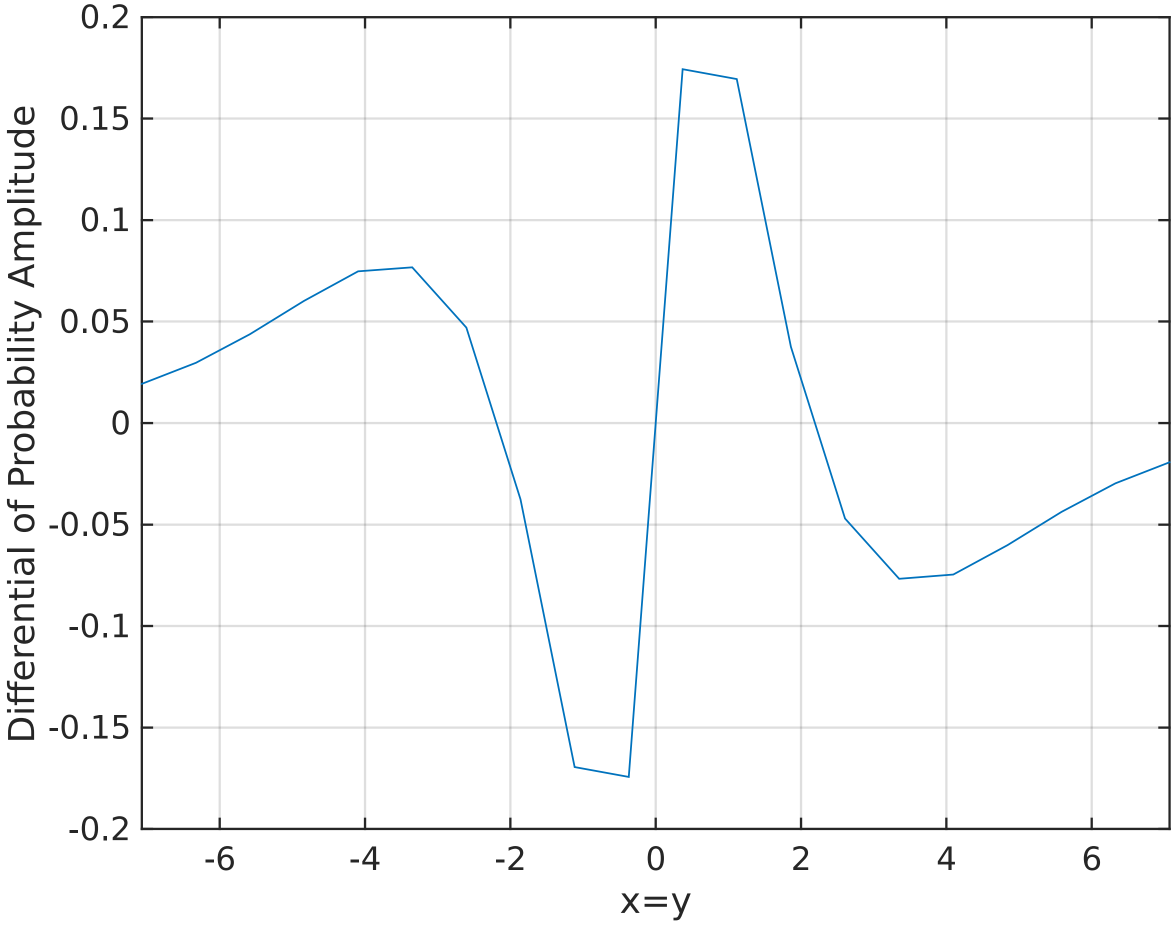

Figure 10: Slice of the d_{xy}'s differential along the $x = y$ direction, as a function of distance along that direction.

Appendix

Code for the Script for Electron Orbital Plotting.

```
%Plotting the Mesh Plot of Orbitals

%%
%Create Meshgrid for plotting
[X,Y] = meshgrid (-5:.5:5);

%%
%Helper Functions and Orbitals themselves

E = exp (-sqrt(X.^2+Y.^2+0));
Px = X.* E;
dxy = X.* Y.*E;

%%
%Plotting using mesh
%P_x
figure;

u=linspace (-5,5,21);
v=linspace (-5,5,21);
%We want the plot to agree to the coordinates
mesh (u,v,Px);
grid on;
set(gca,'fontsize',14,'linewidth',1);
xlabel('x');ylabel('y');zlabel('Probability Amplitude');

%%
%Printing to file.
print -dsvg 1P_x.svg;

%%
%Plotting using surf
%dxy
figure
surf(u,v,dxy);
grid on;
set(gca,'xlim',[-5 5]);
set(gca,'ylim',[-5  5]);
set(gca,'zlim',[-0.4 0.4])
view(-134,68);
set(gca,'fontsize',14,'linewidth',1);
xlabel('x');ylabel('y');zlabel('Probability Amplitude');
%%
%Printing to file
print -dsvg 1dxy.svg;
```

```
%%
%Plotting
%P_x
figure;
contour (Px);
grid on;
set(gca,'fontsize',14,'linewidth',1);
xlabel('x');ylabel('y');zlabel('Probability Amplitude');

%%
%Printing to file.
print -dsvg 2P_x_Contour.svg;

%%
%Plotting using surf
%dxy
figure
contour(u,v,dxy);
grid on;
set(gca,'xlim',[-5 5]);
set(gca,'ylim',[-5 5]);
set(gca,'zlim',[-0.4 0.4])
set(gca,'fontsize',14,'linewidth',1);
xlabel('x');ylabel('y');zlabel('Probability Amplitude');
%%
%Printing to file
print -dsvg 2dxy_Contour.svg;

%%
%Plotting along the Diagonal
figure

%Takes the Diagonal Matrix that contains
%All of the values of dxy along its diagonal.
f = diag(dxy);
plot(u.*sqrt(2),f);
%Need to re-scale the independent variable to
%agree with length along the direction

%If we needed to plot along an arbitrary
%direction we'd have to define a new linspace
%and a new function.
grid on;
set(gca,'xlim',[-5*sqrt(2) 5*sqrt(2)]);
set(gca,'ylim',[0 0.4]);
set(gca,'fontsize',14,'linewidth',1);
xlabel('x=y');ylabel('Probability Amplitude');
```

```
%%
%pirnting to file
print -dsvg 4dxy_along_diagonal.svg;

%%
%Obtaining differential
figure
df = diff (f);
U= linspace (-5,5,20);
fprime = df./ (sqrt(2)*0.5);
%The vector fprime is shorter
plot(U.*sqrt(2),fprime);grid on;
set(gca,'xlim',[-5*sqrt(2) 5*sqrt(2)]);
set(gca,'ylim',[-0.2 0.2]);
set(gca,'fontsize',14,'linewidth',1);
xlabel('x=y');ylabel('Differential of Probability Amplitude');

%%
%pirnting to file
print -dsvg 5Differential.svg;
```