Demo 6 cheat sheet: The poor's man molecule

qworld library

Function	Purpose	Inputs
probabilityDensity(psi_x)	Return the probability density associated to a wavefunction	**psi_x: array of wavefunction values**
my_plotting_function(x,functions_list,labels,title='Plot',xlab='x',ylab='f(x)',fts=12, lw=2,fs=(10,8))	Return a plot	- x: array with x values functions_list: list of arrays representing functions you want to plot labels: list of labels. It should have the same size as functions_list title: title of the plot (Default: 'Plot') other parameters defined by default
Normalize_wf(x, psi_x)	Return a normalized wavefunction	**psi_x: array of wavefunction values**

Miscellaneous (Libraries, numpy.linalg, arrays)

Function	Purpose	Inputs
dir(library)	Display the names of all the function in a module/library	**Name of the library (as import**
imp.load_source(name,path_of_library)	Import a library from path_of_library with the name specified	**name = string** **path_of_library = string**
help(name_of_function)	Return the documentation of the function.	**name of the function**
np.arange(-1.0,1.0, 0.1)	Return an array evenly spaced values within a given interval according to a step size.	**start, stop, step_size**
np.linspace(-1.0,1.0, 100)	Return an array of evenly spaced numbers over a specified interval.	**start, stop, number of points**
np.linalg.eigs(A) (imported as eigs)	Return the eigenvalues and eigenvector of a matrix A	**A: a squared matrix.**

Demo specific common variables and functions:

x == array with x values
V_x == potential
D == height of the barrier
R == width of the barrier,
r == width of the wells
r_pad == width of the boundary regions that "pad" the wells,
L == width of the entire box.
H == Hamiltonian in matrix form
psi0, psi1 == eigenvectors with the lowest eigenvalues
E1, E2 == lists with the two lowest energy eigenvalues for the list of R values.
pdf_0, pdf_1 == corresponding probability density functions
Rarray == array of values of R for step 6.

Function	Purpose	Inputs
box_with_barrier_V(x,R, r,r_pad,D)	**Return a double square potential**	**x -- array of positions; R -- Length of barrier; r -- length of the wells; r_pad -- length of regions that "pad" the wells at the boundaries; D -- height of barrier**
build_H_matrix(x, V_x, m = 1, h_bar = 1)	**Return the matrix representation of the Hamiltonian (H) for a given x and V_x**	**x—array of positions; V_x: potential defined over x.**

Matplotlib : Plotting (plt)

Usage	Purpose	Inputs
plt.figure(figsize=(10,8))	Setup parameter for a graphic, in this case we will use it change size.	figsize= (inches width, inches height)
plt.plot(x,y)	Plot lines	**x, y = vectors**
plt.xlabel("Axis x name")	Set the *x* axis label of the current plot.	**Name = string**
plt.xlim([xmin,xmax])	Set the *x* limits of the current axes.	xmin,xmax = scalars
plt.title("Plot name")	Set a title of the current plot.	**Name = string**
plt.show()	Display a figure.	