Demo 7 cheat sheet: 2D harmonic oscillator

Miscellaneous

Function	Purpose	Inputs
np.arange(-1.0,1.0, 0.1)	Return an array evenly spaced values within a given interval according to a step size.	**x_init**, x_end, **step_size**
np.polynomial.hermite.hermval(x,c)	Evaluates a Hermite series at point x.	x: real c: array with coefficients for the series

Plotting related

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.	
np.meshgrid(x,y)	Return a coordinate matrix from coordinate vectors	x: array of x values; y: array of y values
plt.plot_surface(X,Y,Z)	Produces a 3D surface plot	X,Y: function arguments and Z: function values as 2D arrays
plt.contour(X,Y,Z)	Produces a 2D isocontour plot	X,Y: function arguments and Z: function values as 2D arrays

Demo specific common variables

x, y == array with x (y) values
xx, yy == arrays of x and y values obtained from meshgrid for the 3D plot.
Vxy == 2D harmonic potential
kx, ky == spring constants for the 2D harmonic potential.
omega, omega_A, omega_B == frequencies for the oscillator
mass, mass_A, mass_B == masses for oscillators
Psi_xy == wavefunction
hbar = value of the Plank's constant divided by 2*pi. Set to 1.

Demo specific common functions

Function	Purpose	Inputs
harmonic_potential_2D(xx, yy, kx, ky, x0=0, y0=0)	**Return a 2 dimensional harmonic potential**	**xx (yy) : x (y)-coordinate matrix of an x-y meshgrid;** **kx, ky: spring constants**
harmonic_oscillator_2D(xx, yy, l, m, mass = 1.0, omega = 1.0, hbar = 1.0)	**Returns the wavefunction for the 2D isotropic Harmonic Oscillator**	**xx (yy) : x (y)-coordinate matrix of an x-y meshgrid;** **l: l quantum number** **m: m quantum number** **mass: mass (defaults to atomic units)** **omega : oscillator frequency, defaults to atomic units.** **hbar: planck's constant divided by 2*pi**
harmonic_oscillator_2D_aniso (xx, yy, l, m, mass_1=1.0, mass_2=1, omega_1=1.0, omega_2=1, hbar = 1.0)	**Returns the wavefunction for the 2D anisotropic Harmonic Oscillator**	**xx (yy) : x (y)-coordinate matrix of an x-y meshgrid;** **l: l quantum number** **m: m quantum number** **mass_1(2): reduced mass of mode 1 (2)(defaults to atomic units)** **omega _1(2): oscillator frequency for mode 1(2), defaults to atomic units.** **hbar: planck's constant divided by 2*pi**