3D Animations

```
print("Again, we can animate our plot:")
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.animation as animation
import matplotlib

N = 150 # size of our lattice of points in the xy-plane
fps = 10 # frames per second
frn = 50 # number of frames from the animation

x = np.linspace(-4,4,N) # N points between -4 and 4
x, y = np.meshgrid(x, x) # lattice of points in [-4,4]x[-4,4]
zarray = np.zeros((N, N, frn)) # NxNxfrn numbers

f = lambda x,y,sig : sig*np.exp(-(x**2+y**2)) # our function

for i in range(frn): # generating 'frn' plots (one for each frame)
    sig_i = np.sin(i*2*np.pi/frn) # sigma oscillating in time (with 'i')
    zarray[:,:,i] = f(x,y,sig_i) # global extra oscillation with time

def update_plot(frame_number, zarray, plot):
    plot[0].remove() # removing the old plot
    # the plot is now set equal to the 'frame_number' one
    plot[0] = ax.plot_surface(x, y, zarray[:,:,frame_number], cmap="turbo")


fig = plt.figure() # creating a figure
ax = fig.add_subplot(111, projection='3d') # creating an "axes" object (it's out plot)

plot = [ax.plot_surface(x, y, zarray[:,:,0], color='0.75', rstride=1, cstride=1)] # first plot (0-th
ax.set_zlim(0,1.1) # setting the limits on the 'z' axis
anim_3D = animation.FuncAnimation(fig, update_plot, frn, fargs=(zarray, plot), interval=1000/fps)

jsanim = HTML(anim_3D.to_jshtml()) # creating our animation
display(jsanim) # displaying our animation
```