by Falk Geurten

Explanation - ex06_cfd0

The first step in building the following code is declaring the formulas in a function, that we will later use to calculate the next coordinates of our graph after a certain amount of time passed:

```
def Lorenz(x, y, z):
    dx = (s * (y - x))
    dy = (x * (r - z) - y)
    dz = (x * y - b * z)
    return dx, dy, dz
```

Then we will declare the variables:

```
s = 10
r = 28
b = 8/3
```

We will simulate time, by having ticks we will iterate through. For each axis we will create an empty array with the length of ticks:

```
ticks = 5000
x_values = np.empty(ticks+1)
y_values = np.empty(ticks+1)
z_values = np.empty(ticks+1)
```

Now we will put the start coordinates of the Graph at the position 0 in each of the arrays:

```
x_values[0] = (-8)
y_values[0] = (-1)
z_values[0] = (33)
```

The next thing we define is the time change per calculation:

```
t_change = 0.01
```

This part is the most important one, this loop will go through the empty arrays, use the formulas and all of the other variables above to calculate the next coordinates of the Graph, save them in the empty array and repeat that process, till we went through all (in this case 5000) ticks.

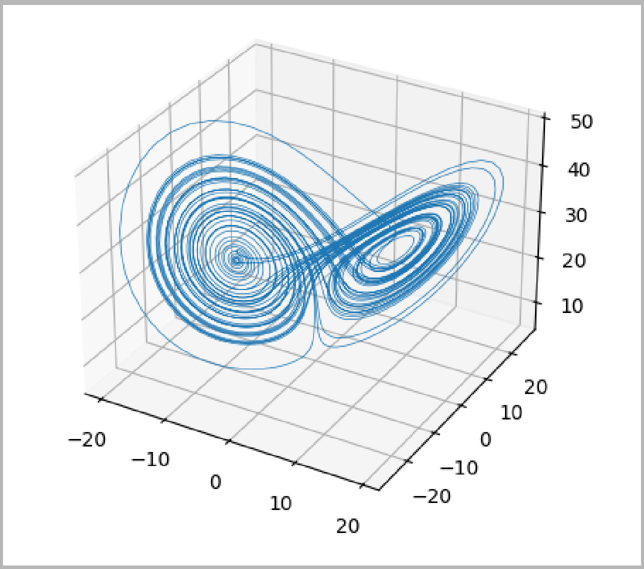
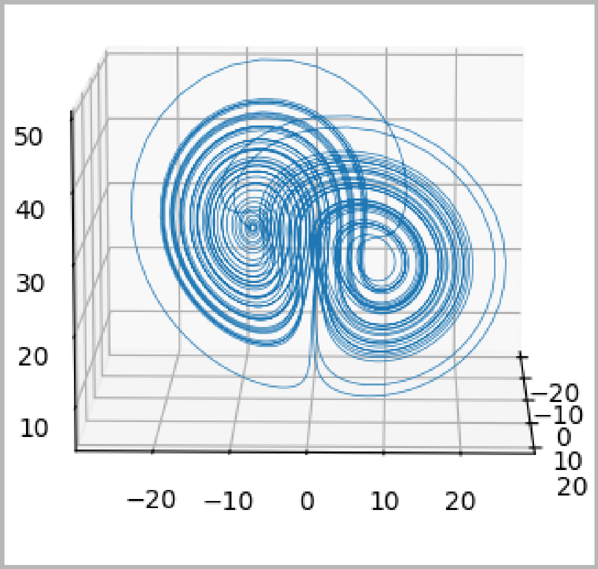
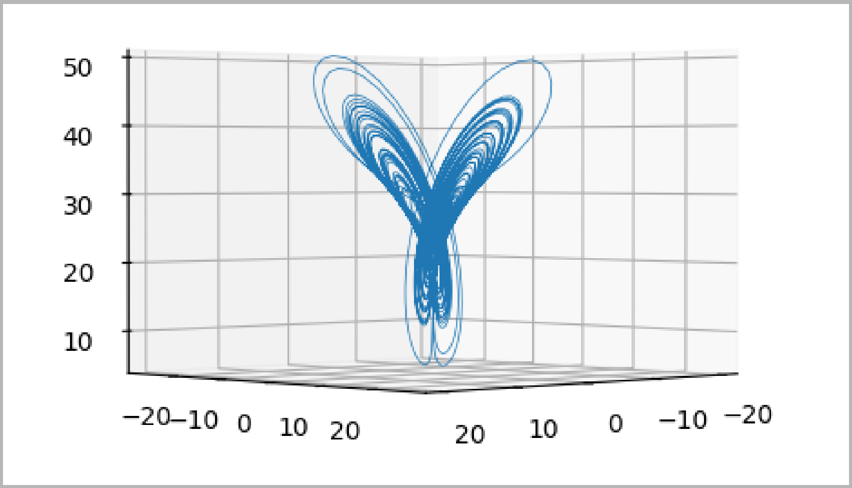
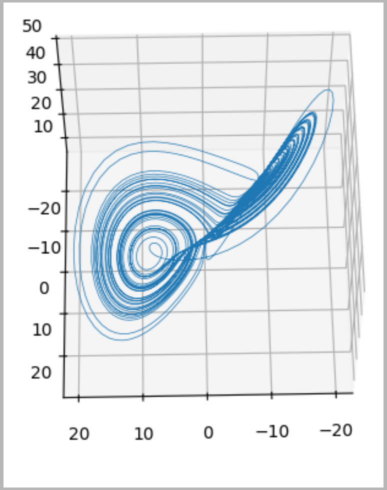
by Falk Geurten

```
for i in range(ticks):
        x_change, y_change, z_change = Lorenz(x_values[i], y_values[i], z_values[i])
        x_values[i + 1] = x_values[i] + (x_change * t_change)
        y_values[i + 1] = y_values[i] + (y_change * t_change)
        z_values[i + 1] = z_values[i] + (z_change * t_change)
```

The last thing that we need to do is 3D plotting all of the values we calculated. 3D plotting is in this case similar to 2D plotting, we just add another axis at the start. (fyi: the lw = 0.5 describes the line width)

```
ax = plt.figure().add_subplot(projection="3d")
ax.plot(x_values, y_values, z_values, lw = 0.5)
plt.show()
```

With the example values that are used above the Graph should look similar to this:


50
40
30
20
10
−20
−10
0
10
20
20
10
0
−10
−20

50
40
30
20
10
−20
−10
0
10
20
−20
−10
0
10
20

50
40
30
20
10
−20
−10
0
10
20
20
10
0
−10
−20

50
40
30
20
10
−20
−10
0
10
20
20
10
0
−10
−20