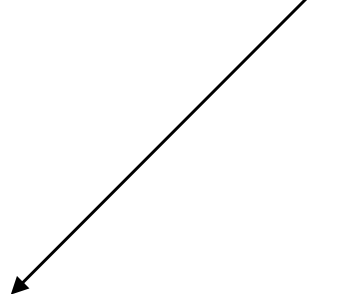


Animacja – Python na komputerze

```
import matplotlib.pyplot as plt
from matplotlib.patches import Circle
```

W środku pętli z ewolucją
(nie trzeba zapisywać trajektorii)

```
for t_i in range(len(time)):
    ...
    ewolucja
    ...
    if (t_i%100==0):      # co 100-na klatka
        plt.clf()        # wyczyść obrazek
        fig = plt.gcf()   # zdefiniuj nowy
        for p in particles: # pętla po cząstkach
            a = plt.gca()  # 'get current axes' (to add smth to them)
            cir = Circle((p.r[0],p.r[1]), radius=p.promien) # kółko tam gdzie jest cząstka
            a.add_patch(cir) # dodaj to kółko do rysunku
            plt.plot()      # narysuj
        plt.xlim((0,boxsize)) # obszar do narysowania
        plt.ylim((0,boxsize))
        fig.set_size_inches((6,6)) # rozmiar rysunku
        plt.title('Symulacja gazu Lennarda-Jonesa, krok {:06d}'.format(t_i))
        plt.savefig('img{:06d}.png'.format(t_i))
```



Animacja – Google Colab

„Montujemy” dysk:

```
# mount drive
from google.colab import drive
from google.colab import files # (opcjonalnie)
drive.mount('/content/gdrive')
images_dir = '/content/gdrive/My Drive/'
```

Zapisujemy klatki tak jak na poprzednim slajdzie:

```
plt.savefig(images_dir+'img{:06d}.png'.format(t_i))
# files.download(images_dir+'img{:06d}.png'.format(t_i))
```



Możemy je ewentualnie od razu ściągać

(ale to nie zawsze działa najlepiej przy bardzo wielu plikach, więc lepiej wejść na Dysk Google i ściągnąć klatki ręcznie, żeby zrobić animację)

Robienie filmu

convert -delay 0.1 *.png anim.gif (<https://imagemagick.org/script/index.php>)

opóźnienie
między klatkami

nazwa animacji

lub:

ffmpeg -f image2 -framerate 20 -i "img%06d.png" animation.avi
(<https://www.ffmpeg.org/>)

Używając *ffmpeg* musimy mieć klatki ponumerowane sekwencją co 1 bez przerw (0000, 0001, 0002, 0003, 0004, ..., a nie np. co 10: ~~0000, 0010, 0020, 0030, ...~~)

- Ściągnąć i zainstalować jeden z programów.
- Komendy wpisujemy w terminalu.

Animacja – Jupyter/Google Colab

(może nie działać)

(<https://colab.research.google.com/github/jckantor/CBE30338/blob/master/docs/A.03-Animation-in-Jupyter-Notebooks.ipynb>)

```
from matplotlib.patches import Ellipse
```

```
# create a figure and axes
```

```
fig = plt.figure(figsize=(12,5))
```

```
ax1 = plt.subplot(1,1,1)
```

```
# set up the subplots as needed
```

```
ax1.set_xlim(( 0, box_size))
```

```
ax1.set_ylim(( 0, box_size))
```

```
ax1.set_aspect('equal')
```

```
# create objects that will change in the animation. These are  
# initially empty, and will be given new values for each frame  
# in the animation.
```

```
txt_title = ax1.set_title('')
```

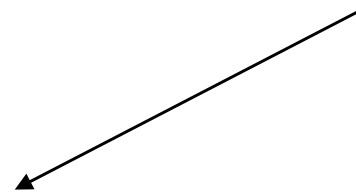
```
ellipses = []
```

```
for i in range(len(particles)):
```

```
    elp = Ellipse((0,0), width=2*particles[i].r,  
                  height=2*particles[i].r)
```

```
    ellipses.append(elp)
```

Zapisujemy trajektorię np. w
particle.history;
Animację robimy z
zapisanych trajektorii



Animacja – Jupyter/Google Colab

(<https://colab.research.google.com/github/jckantor/CBE30338/blob/master/docs/A.03-Animation-in-Jupyter-Notebooks.ipynb>)

```
# animation function. This is called sequentially
```

```
def drawframe(n):  
    for i in range(len(particles)):  
        ellipses[i].set_center((particles[i].history[n,0],  
                                particles[i].history[n,1]))  
        ax1.add_patch(ellipses[i])  
    txt_title.set_text(  
        'Lennard - Jones gas simulation, step: {0:4d}'.format(n))  
    return ellipses
```

Trajektoria cząstki zapisana w „history”

```
# animating
```

```
from matplotlib import animation
```

```
# blit=True re-draws only the parts that have changed.
```

```
anim = animation.FuncAnimation(fig, drawframe, frames=  
np.arange(0, len(time_range), 500), interval=75, blit=True)
```

Wywołuje „drawframe” w pętli z argumentami z listy „frames”

```
from IPython.display import HTML
```

```
HTML(anim.to_html5_video())
```

Odstęp pomiędzy klatkami (w ms)