

# Graficzna prezentacja rozprzestrzeniania się koronawirusa - projekt z przedmiotu Programowanie dla Fizyków

Gabriela Godek, Karol Maziarka i Katarzyna Olender

16.09.2020

## 1 Wstęp

W ramach zajęć z przedmiotu Programowanie dla Fizyków naszym zadaniem było pokazanie rozprzestrzeniania się wirusa Sars-Cov-2 w Polsce. Pokazaliśmy to na przykładzie metody Monte Carlo.

## 2 Wstęp teoretyczny

Wirusy (łac. virus – trucizna) to zbudowane z białek i kwasów nukleinowych skomplikowane cząsteczki organiczne nie posiadające struktury komórkowej, namnażają się przez infekowanie żywych komórek. Wirusy wykorzystują do namnażania aparat kopiujący zawarty w komórkach. Zawierają materiał genetyczny w postaci RNA (retrowirusy) lub DNA, wykazują jednak zarówno cechy komórkowych organizmów żywych, jak i materii nieożywionej. SARS-CoV-2 zawiera jednoniciowe RNA. Rozróżniamy również kilka kształtów: mieszany pałeczkowaty lub bryłowy. Wirusy przenoszą się na różne sposoby, przez kontakt seksualny (HIV, HPV), drogą pokarmową (wirusowe zapalenie wątroby typu A), czy kropelkową (SARS-CoV-2). Pierwsze wzmianki o ludzkich koronawirusach pochodzą z lat 60. XX wieku, kiedy to udało się wyizolować i opisać dwa patogeny – HCoV-229E oraz HCoV-OC43. Koronawirusy przez lata znajdowały się na uboczu głównego nurtu badań w wirusologii i medycynie, ponieważ te dwa gatunki wywołują łagodne przeziębienie, które bez żadnej interwencji ustępuje w ciągu kilku dni. Dopiero początek XXI wieku przyniósł światu epidemię choroby wywołanej przez nieznaną wcześniej, wysoce zakaźny gatunek koronawirusa SARS (ang. severe acute respiratory syndrome).

### 2.1 Czy wirus żyje?

Wirusów nie można zaliczyć do organizmów żywych ponieważ nie posiadają budowy komórkowej ani nie zachodzą w nich żadne procesy metaboliczne. Do

rozmnażania i przetrwania potrzebują innych organizmów by wnikać w ich komórki i namnażać się za ich pośrednictwem.

## 2.2 Jak wirusy się namnażają?

W każdej żywej komórce zachodzi szereg reakcji biochemicznych. Syntetyzowane i rozkładane są różnego typu związki organiczne, a także gromadzona i uwalniana energia. Wszystko dzięki organellom komórkowym i materiałowi genetycznemu, w które wyposażona jest każda taka komórka. Atakujący ją wirus wprowadza do niej własny materiał genetyczny, zyskując pełną kontrolę nad jej metabolizmem. W końcu komórka przestaje pracować na własny użytek. Zamiast tego tworzy nowe jednostki wirusów.

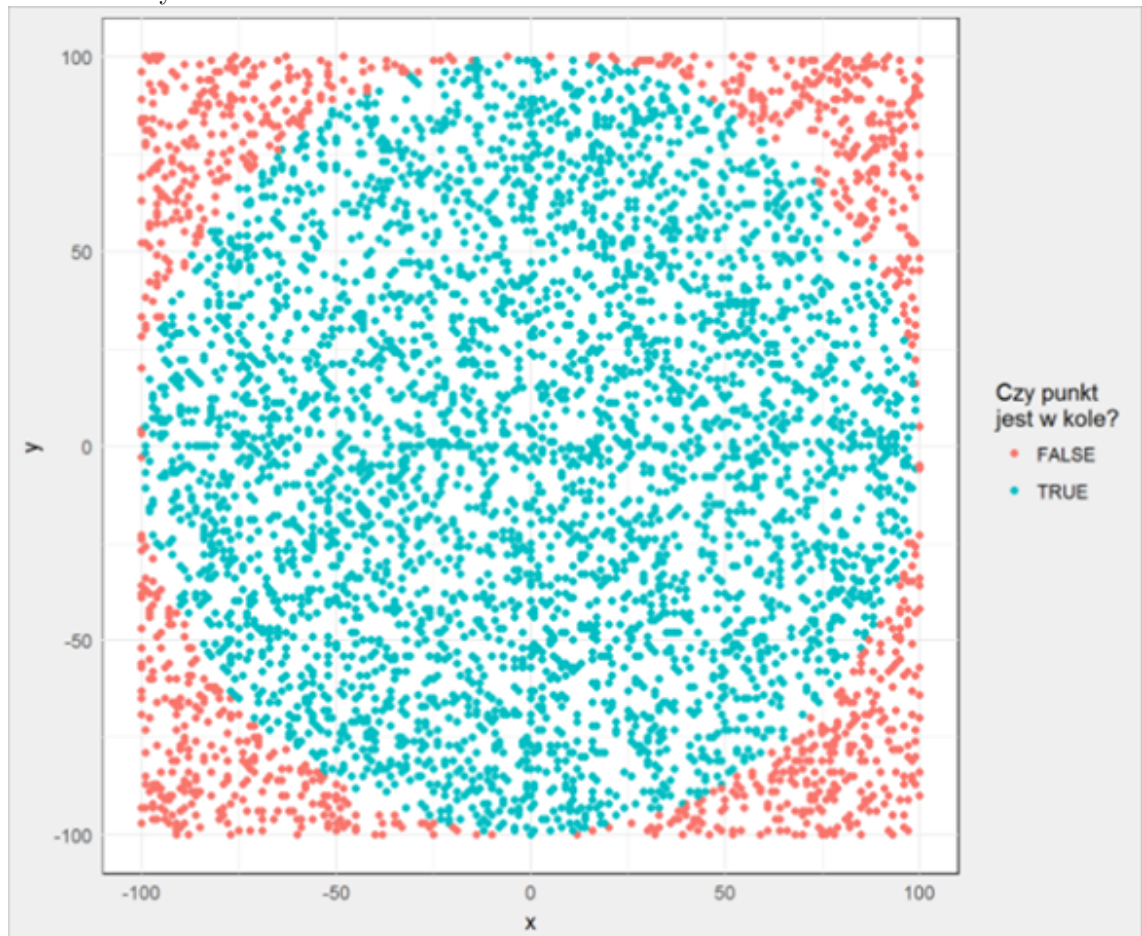
Cały ten proces można podzielić na kilka etapów. Najpierw wirus przyczepia się do powierzchni komórki, później przebija przez jej ścianę, a następnie wpuszcza do jej wnętrza własny materiał genetyczny. Na zewnątrz komórki pozostaje jedynie opróżniony kapsyd wirusa (w przypadku komórek zwierzęcych wirion wnika w całości). W wnętrzu komórki rozpoczyna się produkcja nowych białek kapsydowych. Obywa się ona zgodnie z instrukcją zapisaną w wirusowym DNA (lub RNA). Jednocześnie dochodzi do powielenia odcinków materiału genetycznego wirusa. Gdy zarówno białek, jak i nici DNA/RNA jest już odpowiednia ilość, łączą się one ze sobą, tworząc gotowe wiriony. Następnie uwalniają się z zaatakowanej komórki, a ta ulega zniszczeniu. Oczywiście nie każdy wirus może przyczepić się do każdej komórki i ją zainfekować. Na jego otocze zawsze znajduje się jakiś receptor (białko), który „pasuje” do białka znajdującego się na komórce i może się z nim połączyć. W przypadku koronawirusa tym receptorem jest białko ACE2.

## 3 Metoda Monte Carlo

Numeryczna metoda rozwiązywania problemów Monte Carlo, została wymyślona przez Stanisława Ulama, który należał do lwowskiej szkoły matematycznej. Metoda Monte Carlo to metoda stosowana do modelowania matematycznego procesów zbyt złożonych (obliczania całek, łańcuchów procesów statystycznych), aby można było przewidzieć ich wyniki za podejścia analitycznego. Istotną rolę w tej metodzie odgrywa losowanie (wybór przypadkowy) wielkości charakteryzujących proces, przy czym losowanie dokonywane jest zgodnie z rozkładem, który musi być znany. Dokładność wyniku uzyskanego tą metodą jest zależna od liczby sprawdzeń i jakości użytego generatora liczb pseudolosowych. Zwiększanie liczby prób nie zawsze zwiększa dokładność wyniku, ponieważ generator liczb pseudolosowych ma skończenie wiele liczb losowych w cyklu. Przykładowo całkowanie tą metodą jest używane w przypadkach, kiedy szybkość otrzymania wyniku jest ważniejsza od jego dokładności (np. obliczenia inżynierskie).

Metodę Monte Carlo i liczbę PI najlepiej obrazuje przykład z opuszczeniem ołówka. Mamy wyznaczony obszar (kwadrat) i upuszczamy na niego ołówek tak by wpadł w jego obszar. zynność ta jest powtórzona  $n$  razy. Liczba Pi to liczba

punktów w kole w stosunku do liczby wszystkich punktów, w przybliżeniu jest to 3.0896. Im liczba powtórzeń jest większa tym bliższą wartość liczby PI jesteśmy w stanie otrzymać.



Przykładowy kod Monte Carlo:

1. `import math`
2. `import random`
3. `random.seed()`
4. `n = 10000`
5. `hits = 0` `#initialize hits with 0`
6. `for i in range(n):`

```

7. x = random.random()
8. y = random.random()
9. z=(x,y)

10. if x**2+y**2 <= 1:
11.     hits += 1
12. else:
13.     HIT del z
14. hits = 0 # use hits and n compute PI

```

Metodą Monte Carlo można obliczyć pole figury zdefiniowanej nierównością:

$$x^2 + y^2 = R^2$$

czyli koła o promieniu  $R$  i środku w punkcie  $(0,0)$ .

1. Losuje się  $n$  punktów z opisanego na tym kole kwadratu – dla koła o  $R = 1$  współrzędne wierzchołków  $(-1,-1)$ ,  $(-1,1)$ ,  $(1,1)$ ,  $(1,-1)$ .
2. Po wylosowaniu każdego z tych punktów trzeba sprawdzić czy jego współrzędne spełniają powyższą nierówność (tj. czy punkt należy do koła).

Wynikiem losowania jest informacja, że z  $n$  wszystkich prób  $k$  było trafionych, zatem pole koła wynosi:

$$P_k = P \frac{k}{n},$$

gdzie  $P$  jest polem kwadratu opisanego na tym kole (dla  $R = 1 : P = 4$ ).

My obliczamy całkę metodą Monte Carlo przy pomocy następującego kodu:

```
#!/usr/bin/env python

from random import uniform
from math import pi, sin

import matplotlib.pyplot as plt

xpoints = []
ypoints = []

xfun = []
yfun = []

Nfun = 0
Nsquare = 0

Ntot = 100000

def fun(x):
    return sin(x)

a = 0
b = pi
h = 1

for i in range(Ntot):
    x = uniform(a, b)
    y = uniform(0, h)

    xpoints.append(x)
    ypoints.append(y)

    Nsquare += 1
    if y < fun(x):
        Nfun += 1
        xfun.append(x)
        yfun.append(y)

integral = h*(b-a) * float(Nfun) / float(Nsquare)

print("Value = ", integral)

print("Error = ", 2 - integral)

plt.plot(xpoints, ypoints, 'or')
plt.plot(xfun, yfun, 'ob')
plt.show()
```

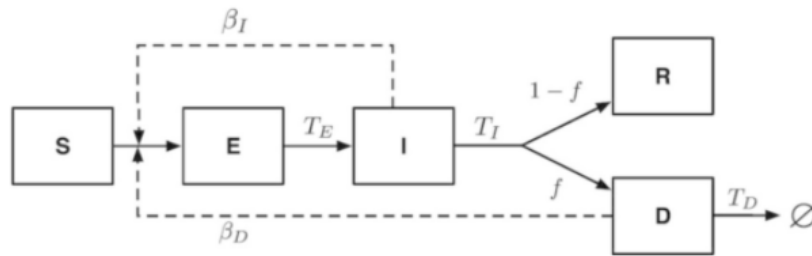
## 4 Model SEIRD

Model SEIRD (Susceptible-Exposed-Infected-Recovered-Dead) to kolejna metoda, która pozwala przeprowadzić symulacje rozprzestrzeniania się nowych, nieznanych chorób. Dzięki temu możemy w przybliżeniu przewidzieć jak będzie zachowywać się wirus, jak szybko będzie się rozprzestrzeniać i ile osób się nim zakazi.

Co dokładnie oznacza skrót SEIRD?

1. **susceptible**: grupa ryzyka
2. **exposed**: narażeni na zakażenie
3. **infected**: zarażeni z możliwością zarażania innych
4. **recovered**: ozdrowieńcy
5. **dead**: zmarli

Algorytm metody SEIRD jest następujący:



Strzałki zaznaczają kierunek etapów jakie powinien przejść potencjalny zarażony, linie przerywane zależą od interakcji między poszczególnymi grupami (np czy osoby zainfekowane będą mieć ponowny kontakt z osobami z grupy ryzyka i osobami podatnymi. Parametry  $\beta_I$  oraz  $\beta_D$  to wskaźnik szybkości z jaką dochodzi za zakażeń,  $T_E$ ,  $T_I$  oraz  $T_D$  to średnie okresy trwania choroby pomiędzy odpowiednimi etapami, a  $f$  to odsetek osób które umierają. Można przedstawić to za pomocą równań:

$$\begin{aligned}\frac{dS}{dt} &= -\beta SI \\ \frac{dE}{dt} &= -\beta_D SI - T_E E \\ \frac{dI}{dt} &= T_I E - \beta_I I \\ \frac{dR}{dt} &= \beta_D I(1 - f)\end{aligned}$$

$$\frac{dD}{dt} = \beta_D I f$$

Następnie należy to równanie zaimplementować do kodu. Nasz kod wygląda następująco:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
plt.ion()
plt.rcParams['figure.figsize'] = 10, 8

P = 0 # birth rate
d = 0.0001 # natural death percent (per day)
B = 0.0095 # transmission percent (per day)
G = 0.0001 # resurrect percent (per day)
A = 0.0001 # destroy percent (per day)

# solve the system dy/dt = f(y, t)
def f(y, t):
    Si = y[0]
    Zi = y[1]
    Ri = y[2]
    # the model equations (see Munz et al. 2009)
    f0 = P - B*Si*Zi - d*Si
    f1 = B*Si*Zi + G*Ri - A*Si*Zi
    f2 = d*Si + A*Si*Zi - G*Ri
    return [f0, f1, f2]

# initial conditions
S0 = 500. # initial population
Z0 = 0 # initial coronavirus population
R0 = 0 # initial death population
y0 = [S0, Z0, R0] # initial condition vector
t = np.linspace(0, 5., 1000) # time grid
# solve the DEs
soln = odeint(f, y0, t)
S = soln[:, 0]
Z = soln[:, 1]
R = soln[:, 2]
# plot results
plt.figure()
plt.plot(t, S, label='Living')
plt.plot(t, Z, label='coronavirus')
plt.xlabel('Days from outbreak')
plt.ylabel('Population')
plt.title('coronavirus - No Init. Dead Pop.; No New Births.')
plt.legend(loc=0)
plt.show()
```

```

# change the initial conditions
R0 = 0.01*S0 # 1% of initial pop is dead
y0 = [S0, Z0, R0]
# solve the DEs
soln = odeint(f, y0, t)
S = soln[:, 0]
Z = soln[:, 1]
R = soln[:, 2]
plt.figure()
plt.plot(t, S, label='Living')
plt.plot(t, Z, label='coronavirus')
plt.xlabel('Days from outbreak')
plt.ylabel('Population')
plt.title('coronavirus - 1% Init. Pop. is Dead; No New Births.')
plt.legend(loc=0)
plt.show()

# change the initial conditions
R0 = 0.01*S0 # 1% of initial pop is dead
P = 10 # 10 new births daily
y0 = [S0, Z0, R0]

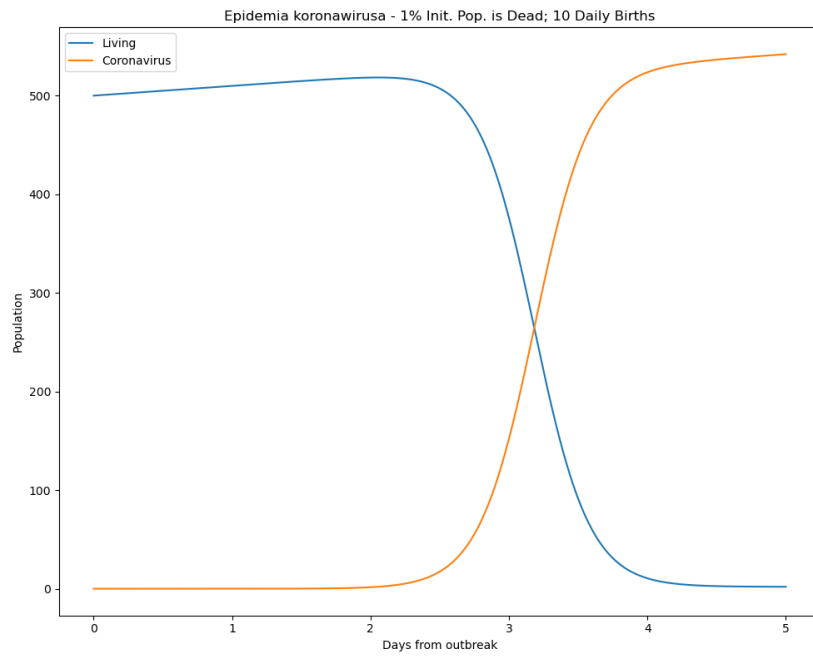
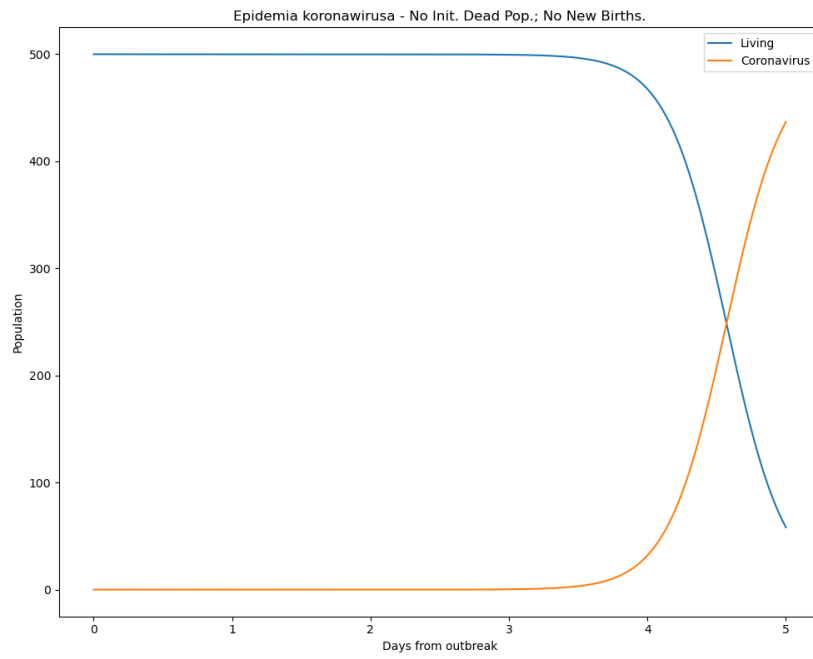
# solve the DEs
soln = odeint(f, y0, t)
S = soln[:, 0]
Z = soln[:, 1]
R = soln[:, 2]
plt.figure()
plt.plot(t, S, label='Living')
plt.plot(t, Z, label='coronavirus')
plt.xlabel('Days from outbreak')
plt.ylabel('Population')

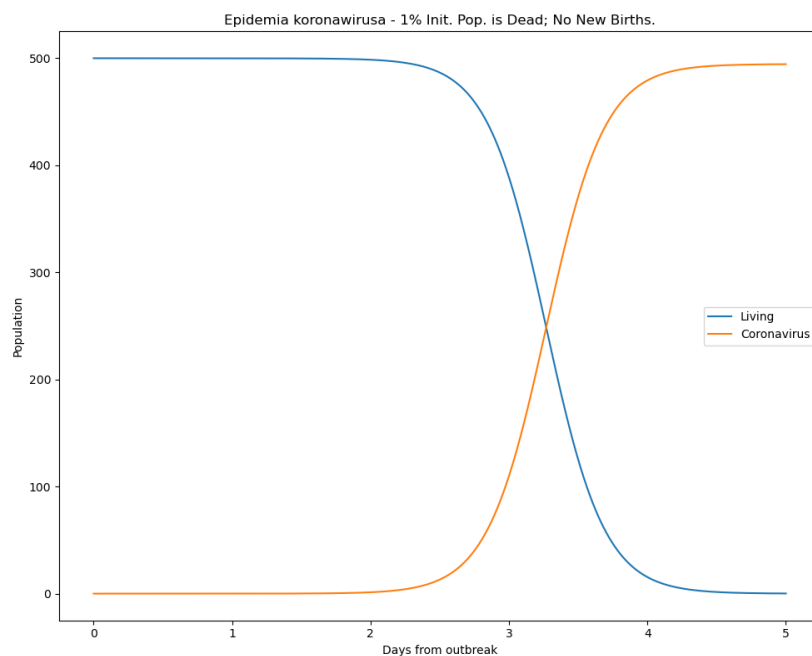
plt.title('coronavirus - 1% Init. Pop. is Dead; 10 Daily Births')
plt.legend(loc=0)
plt.show()

```

Dzięki tej metodzie możemy otrzymać następujące symulacje:

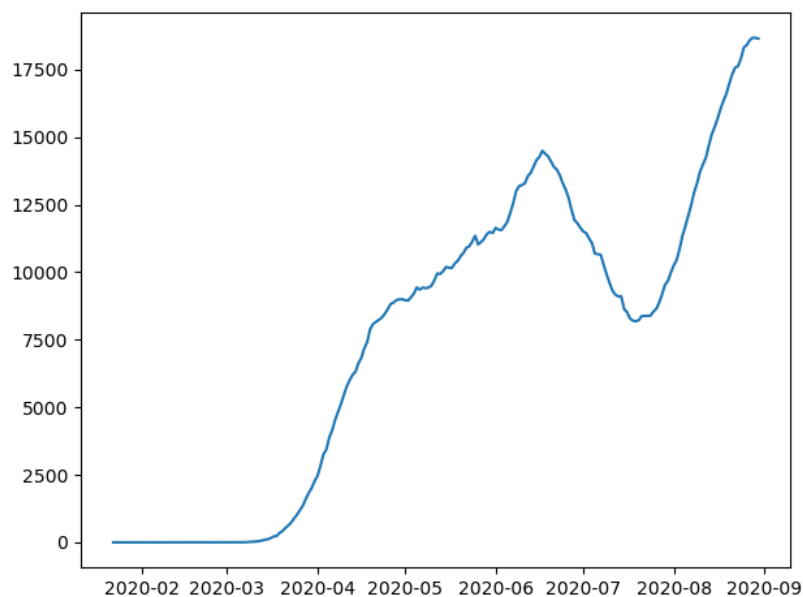






## 5 wnioski

Metoda Monte Carlo pozwala na zobrazowanie przebiegu pandemii w stopniu zadowalającym. By uzyskać dokładniejsze wyniki i bardziej precyzyjne wymagane byłoby powtórzenie doświadczenia znacznie więcej razy. Przy danych, które obecnie posiadamy wykres zachorowalności możemy przedstawić następująco:



Jest to liczba zakażonych w stosunku do kolejnych miesięcy. Z tego wykresu można wywnioskować, że pandemia rozwija się, ma tendencję wzrostową jednak na przełomie miesięcy gdzie są najwyższe temperatury zachorowalność spada. Przyrównując nasz przewidywany przebieg pandemii do realnej sytuacji w Polsce możemy dojść do wniosku, że przewidywania były słuszne z pewną nieścisłością co do miesięcy wakacyjnych. Jednak porównując wykres żółty z naszym wykresem możemy znaleźć dużo podobieństw.

## 6 Bibliografia

K. Pyrc - Ludzkie Koronawirusy, 2015  
<https://koronawirusunas.pl/>