Laboratorium programistyczne 3
równania różniczkowe i metoda Eulera

Projekt „Matematyka dla Ciekawych Świata”,
Piotr Morawiecki, Robert Paciorek

2022-04-30

1 Powtórzenie wykładu: metoda Eulera

Rozważmy równanie różniczkowe postaci:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = f(x(t), t) \tag{1}$$

$x(t)$ jest to pewna nieznana funkcja w czasie, a $f(x, t)$ jest znaną funkcją opisującą szybkość wzrostu x w czasie. Ponadto załóżmy, że znamy wartość początkową $x(0)$ (tzw. **warunek początkowy**. W tym skrypcie nauczymy się rozwiązywać to równanie z użyciem Pythona.

Jak wiemy z ostatnich zajęć pochodną można przybliżyć jako $\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{x(t+\Delta t)-x(t)}{\Delta t}$, gdzie Δt jest pewnym małym krokiem czasowym. Im mniejsze Δt wybierzemy tym oszacowanie pochodnej jest dokładniejsze. Równanie (1) możemy zatem przybliżyć jako:

$$\frac{x(t + \Delta t) - x(t)}{\Delta t} = f(x(t), t)$$

Po przekształceniu dostajemy:

$$\underbrace{x(t + \Delta t)}_{\text{nowa wartość x}} = \underbrace{x(t)}_{\text{stara wartość x}} + \underbrace{f(x(t), t)\Delta t}_{\text{przyrost wartości x}}$$

Równanie to pozwala znaleźć nam następną wartość zmiennej x w czasie $t + \Delta t$ znając jej wartość w czasie t. Powtarzając tą procedurę wielokrotnie można znaleźć przybliżone rozwiązanie naszego równania po dowolnym czasie. Jest to tak zwana **metoda Eulera**.

W tym skrypcie zaimplementujemy ją w Pythonie dla kilku prostych układów dynamicznych.

2 Modelowanie wzrostu populacji królików

Rozważmy najprostszy model populacji królików opisany równaniem:

$$\frac{\mathrm{d}n}{\mathrm{d}t} = kn$$

gdzie $n(t)$ to wielkość populacji zależna od czasu t, a k to stała opisująca szybkość wzrostu populacji. To równanie ma postać taką jak równanie 1 przy czym naszą nieznaną funkcją jest $n(t)$, natomiast $f(n, t) = kn$. Zakładamy warunek początkowy $n(0) = n_0$, gdzie n_0 to początkowa populacja królików.

Zatem możemy obliczyć wartość $n(t)$ w kolejnych krokach czasowych jako:

$$\underbrace{n(t + \Delta t)}_{\text{nowa wielkość populacji}} = \underbrace{n(t)}_{\text{stara wielkość populacji}} + \underbrace{kn(t)\Delta t}_{\text{przyrost populacji}} \tag{2}$$

Spróbujmy zatem zaimplementować to równanie krok po kroku w Pythonie.

Krok 1. Wprowadzamy parametry modelu, w tym przypadku zmienną k i n_0.

```
k = 1
n0 = 1
```

Krok 3. Inicjujemy dwa wektory, w których będziemy zapisywać kolejne wartości czasu oraz wielkości populacji. Początkowo powinny one zawierać jedną pozycję reprezentującą warunek początkowy, u nas $n(0) = n_0$.

```
t = [0]
n = [1]
```

Krok 2. Określamy do jakiego czasu t chcemy rozwiązać równanie (u nas do $t = 10$) oraz jakiej długości kroki czasowe chcemy użyć (u nas $\Delta t = 0.1$).

```
t_max = 10
dt = 0.1
```

Krok 4. W pętli `while` obliczamy wartości t i n w kolejnych krokach czasowych na podstawie ostatnich wartości. W przypadku t zwiększamy je w każdym kroku od Δt, a w przypadku $n(t)$ o $kn(t)\Delta t$, tak jak zapisaliśmy w równaniu (2). Pętlę jest wykonywana aż do momentu kiedy ostatni zapisany czas `t[-1]` dojdzie do wartości końcowej `t_max`:

```
while  t[-1] < t_max:
  t.append(t[-1] + dt)
  n.append(n[-1] + k*n[-1]*dt)
```

Krok 5. Na koniec rysujemy wykres $n(t)$ odpowiednio oznaczając osie.

```
plt.plot(t,n)
plt.xlabel('t')
plt.ylabel('n(t)')
plt.show()
```

Pamiętaj też, że żeby móc rysować wykresy na początku notebooka musisz załadować bibliotekę `matplotlib.pyplot`:

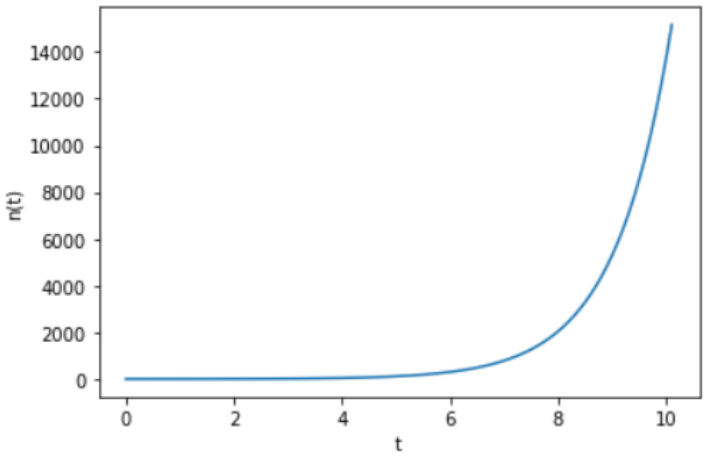
```
import matplotlib.pyplot as plt
```

Końcowy kod powinien wyglądać następująco:

```
import matplotlib.pyplot as plt
k = 1
n0 = 1
t = [0]
n = [n0]
t_max = 10
dt = 0.1
while t[-1] < t_max:
  t.append(t[-1] + dt)
  n.append(n[-1] + k*n[-1]*dt)
plt.plot(t,n)
plt.xlabel('t')
```

```
plt.ylabel('n(t)')
plt.show()
```

Powinniśmy otrzymać następujący wykres:



Zadanie 2.0.1

Wykonaj symulację dla trzech długości kroku czasowego: $\Delta t = 0.1$, $\Delta t = 0.01$ i $\Delta t = 0.001$, a następnie narysuj je na jednym wykresie. Spróbuj wykonać to zadanie używając pętlo `for` przechodzącej przez kolejne wartości Δt. Pamiętaj też o dodaniu legendy.
Które z otrzymanych rozwiązań jest najdokładniej? Odpowiedź uzasadnij.

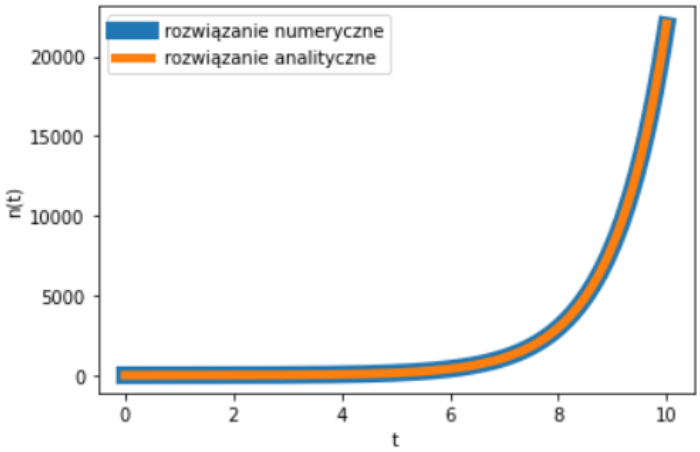
Ważna uwaga: Powinniśmy tak dobrać długość kroku czasowego Δt, żeby jego dalsze zmniejszanie nie wpływało w widoczny sposób na rozwiązanie. Dalsze zmniejszanie korku czasowego tylko będzie wydłużać czas obliczeń.

W naszym przypadku powinniśmy wybrać $\Delta t = 0.001$. Polecam sprawdzić doświadczalnie dodając wartość $\Delta t = 0.0001$ do rozwiązania poprzedniego zadania, że otrzymamy praktycznie to samo rozwiązanie co dla $\Delta t = 0.001$ (jednak Python będzie musiał wykonać 10 razy więcej iteracji). W przypadku prostych symulacji jak ta i tak otrzymalibyśmy szybko wynik, ale w przypadku zaawansowanych symulacji trwających kilka dni, zwiększenie 10-krotnie czasu obliczeń może być bardzo kosztowne.

Teraz spróbujemy sprawdzić czy otrzymane rozwiązanie numeryczne obliczone metodą Eulera jest zgodne z rozwiązaniem analitycznym wyprowadzonym na wykładzie $n(t) = n_0 e^{kt}$, gdzie n_0 to początkowa wartość n_0. W tym celu możemy zmodyfikować nasz kod następująco (wszystkie zmiany zostały odpowiednio skomentowane poprzez umieszczenie linijki rozpoczynającej się od `#` - te linijki są ignorowane przez interpreter):

```
import matplotlib.pyplot as plt
# załączamy bibliotekę math potrzebną do obliczenia funkcji wykładniczej
import math

k = 1
n0 = 1
t = [0]
n = [n0]
# inicjalizujemy listę, w której będziemy zapisywać rozwiązanie analityczne
n_analityczne = [1]

t_max = 10
dt = 0.001

while t[-1] < t_max:
  t.append(t[-1] + dt)
  n.append(n[-1] + k*n[-1]*dt)
  # obliczamy rozwiązanie analityczne w czasie t i dodajemy do listy
  n_analityczne.append(n0*math.exp(k*t[-1]))

# zwiększamy grubość wykresu rozwiązania numerycznego parametrem linewidth
plt.plot(t,n,linewidth=10)
# rysujemy wykres rozwiązania analitycznego
plt.plot(t,n_analityczne,linewidth=5)
plt.xlabel('t')
plt.ylabel('n(t)')
# dodajemy do wykresu legendę
plt.legend(['rozwiązanie numeryczne', 'rozwiązanie analityczne'])
plt.show()
```

Powinniśmy otrzymać następujący wykres:



Wyraźnie widać, że otrzymane rozwiązanie analityczne i numeryczne są ze sobą zgodne. Zwróć uwagę, że pogrubiliśmy wykresy w celu lepszej wizualizacji krzywych. Gdybyśmy zostawili domyślną wartość grubości to byłyby one nieodróżnialne.

Zadanie 2.0.2

Zmodyfikuj swoją symulację, tak aby wziąć pod uwagę ograniczone zasoby środowiska. Na wykładzie opisaliśmy ten model za pomocą tak zwanego równania logistycznego:

$$\frac{\mathrm{d}n}{\mathrm{d}t} = kn\left(1 - \frac{n}{N}\right) \tag{3}$$

gdzie N to maksymalna stabilna wielkość populacji.
Narysuj wykres $n(t)$ dla parametrów $k = 1$, $N = 10$, $n_0 = 1$ i $\Delta t = 0.001$.

Zadanie 2.0.3

Sprawdź, że rozwiązanie numeryczne otrzymane w poprzednim zadaniu jest zgodne z rozwiązaniem analitycznym wyprowadzonym na wykładzie:

$$n(t) = \frac{N}{n_0 + (N - n_0)e^{-kt}}$$

Zadanie 2.0.4

Narysuj rozwiązanie równania logistycznego dla różnych wartości warunku początkowego, np. $n(0) = 1$, $n(0) = 5$, $n(0) = 10$, $n(0) = 15$ i $n(0) = 20$. Wykorzystaj w tym celu komendę `for`. Co obserwujesz?

3 Zadanie dodatkowe

W tym zadaniu napiszemy symulację numeryczną opisującą jak koncentracja wirusów HIV w osoczu krwi zmienia się czasie. Oznaczmy koncentrację wirusów w czasie t jako $V(t)$.

Równanie opisujące zmianę $V(t)$ w czasie wygląda następująco:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = k - mV \tag{4}$$

Przyjęliśmy tutaj, że wirusy są produkowane w osoczu w stałym tempie, który oznaczymy jako k. Jednocześnie już istniejące wirusy są usuwane z krwi w tempie m.

Zadanie 3.0.1

Napisz symulację numeryczną, która pokaże jak zmienia się $V(t)$ w czasie. Przyjmij, że m wynosi 2.77 dni^{-1} oraz, że początkowo w osoczu nie ma wirusów HIV. Oszacuj (za pomocą symulacji lub analitycznie) ile wynosi k, jeśli eksperymentalnie zaobserwowano, że koncentracja wirusów krwi rośnie aż do ustabilizowania się na poziomie 300000 wirusów na ml osocza.

Teraz spróbujemy napisać model matematyczny opisujący działanie nowej terapii opartej na inhibitorach odwrotnej transkryptazy (https://pl.wikipedia.org/wiki/Inhibitory_odwrotnej_transkryptazy). W wyniku tej terapii komórki osocza są chronione przed zostanie zainfekowaniem. Wówczas zainfekowane do tej pory komórki stopniowo wymierają, a ich ilość maleje w tempie wykładniczym. W konsekwencji również tempo produkcji maleje wykładniczo.

Zadanie 3.0.2

Zmodyfikuj model opisany równaniem (4), żeby uwzględnić skutki terapii na szybkość produkcji nowych wirusów. Możesz wprowadzić nowy stały parametr (np. d), żeby opisać szybkość wykładniczego zaniku tempa produkcji wirusów HIV.

Zadanie 3.0.3

Napisz symulację numeryczną opisującą zmianę koncentracji wirusów $V(t)$ w trakcie trwania terapii. Wykorzystując ją wykonaj następujące zadania:

1. Zbadaj jak na wynik symulacji wpływa szybkość obumierania komórek zainfekowanych wirusem HIV.
2. Porównaj koncentrację wirusów $V(t)$ w przypadku stosowania i niestosowania terapii.
3. Narysuj wykres $V(t)$ zaznaczając na osi pionowej $\ln(V)$ zamiast $V(t)$. Jaką funkcją można opisać szybkość spadku koncentracji wirusów HIV w późnych etapach terapii (długo po osiągnięciu maksymalnej koncentracji wirusów)? Wskazówka: logarytm naturalny możesz obliczyć korzystając z funkcji `math.log()` należącej do biblioteki `math`.

Opisz swoje obserwacje popierając je odpowiednim wykresem.

4 Praca domowa nr 3

Rozwiązania zadań domowych należy przesłać do czasu następnych ćwiczeń. Prowadzący może również zmienić ostateczny termin przesłania pracy domowej, np. na inną godzinę lub dzień.

Za pracę domową można maksymalnie dostać 6 punktów. Można wybierać zarówno łatwiejsze zadania, jak i trudniejsze, za większą liczbę punktów. Można również rozwiązać zadania za większą liczbę punktów, np. za 10. Jeśli wtedy poprawnie będą rozwiązane zadania za 5 punktów, to zostanie przydzielone 5 punktów. Jeśli będą poprawnie rozwiązane zadania za więcej niż 6 punktów, np. 8 czy 10, to taka osoba otrzyma maksymalnie 6 punktów.

Pamiętaj, żeby notebook był czytelny – każde zadanie powinno być umieszczone w osobnym bloku tekstowym oraz poprzedzone numerem (i opcjonalnie opisem) zadania. Komentarze są mile widziane.

Zadanie 4.0.1 — 6 pkt (skok spadochronowy)

Na spadochroniarza działa siła grawitacji i siła oporów powietrza. Zmianę jego prędkości $v(t)$ w czasie opisuje równanie:

$$\frac{\mathrm{d}v}{\mathrm{d}t} = g - kv^2$$

gdzie g to przyspieszenie ziemskie, a k to współczynnik oporu powietrza zależny od powierzchni i kształtu spadochronu oraz gęstości powietrza. Ponadto prędkość początkową spadochroniarza w momencie otwarcia spadochronu oznaczmy jako v_0.

Uwaga: Każdy podpunkt tego zadania jest warty 1 punkt. Nie musisz rozwiązywać wszystkich podpunktów tego zadania. Jeśli nie masz pomysłu możesz je pominąć i zamiast tego zrobić wybrane podpunkty zadania drugiego.

1. Rozwiąż powyższe równanie wykorzystując symulację numeryczną opisującą prędkość spadochroniarza początkowo nie poruszającego się względem ziemi, tzn. $v_0 = 0$. Przyjmij $g = 9.81\mathrm{ms}^{-1}$ i $k = 1\mathrm{m}^{-1}$. Wyniki przedstaw na wykresie.
2. Pokaż na jedynym wykresie rozwiązania dla kilku różnych prędkości początkowych spadochroniarza v_0. Opisz jego wpływ na zmianę prędkości spadochroniarza (korzystając z pola tekstowego w Google Colab).
3. Pokaż na jedynym wykresie rozwiązania dla kilku różnych wartości k. Opisz jego wpływ na zmianę prędkości spadochroniarza. Spadochron o jakim współczynniku k powinniśmy wybrać, aby wylądować na ziemi z rekomendowaną prędkością około 5-6ms[1]?
4. Pokaż, że wyniki twojej symulacji dla $v_0 = 0$ są zgodne z następującym rozwiązaniem analitycznym:

$$v(t) = \sqrt{\frac{g}{k}}\frac{e^{2\sqrt{gk}t} - 1}{e^{2\sqrt{gk}t} + 1}$$

5. Napisz symulację numeryczną, która opisze jak prędkość spadochroniarza, który wyskoczył z samolotu w czasie $t = 0$, ale spadochron otworzył dopiero po 20 sekundach lotu. Przyjmij, że współczynnik oporu powietrza działającego na spadochroniarza bez otwartego spadochronu wynosi $k_1 = 0.004\mathrm{m}^{-1}$, a z otwartym spadochronie $k_2 = 1\mathrm{m}^{-1}$. Wyniki przedstaw na wykresie.
6. Napisz symulację, która pozwoli ci poza zmianą prędkości $v(t)$ opisze także zmianę wysokości nad ziemią w czasie $h(t)$. Przyjmij te same założenia, które zostały przedstawione w poprzednim zadaniu. Ponadto załóż, że wysokość z jakiej spadochroniarz wyskoczył z samolotu wynosiła $h_0 = 1000\mathrm{m}$. Przerwij symulację w momencie kiedy spadochroniarz wyląduje na Ziemi. Po jakim czasie to nastąpiło?

Zadanie 4.0.2 — 6 pkt (połów sandaczy)

Rozważmy, że w stawie hodujemy sandacze. Mają one dobre warunku do rozrodu, ale codziennie stała ich liczba jest wyławiana i sprzedawana na pobliskim rybnym targu. Zaproponujmy opisać zmianę populacji sandaczy $n(t)$ w czasie za pomocą równania:

$$\frac{\mathrm{d}n}{\mathrm{d}t} = n - k \tag{5}$$

gdzie k to tempo z jakim są one wyławiane. Oznaczmy początkową liczbę sandaczy jako n_0.
Uwaga: Każdy podpunkt tego zadania jest warty 1 punkt (z wyjątkiem ostatniego). Nie musisz rozwiązywać wszystkich podpunktów tego zadania. Jeśli nie masz pomysłu możesz je pominąć i zamiast tego zrobić wybrane podpunkty zadania pierwszego.

1. Rozwiąż równanie (5) za pomocą symulacji numerycznej. Przyjmij, że początkowa liczba sandaczy wynosi $n_0 = 10$, a tempo ich wyłowu wynosi $k = 5$ sandaczy/dzień. Narysuj rozwiązanie na wykresie.
2. W wyniku symulacji populacja sandaczy rośnie nieograniczenie. Jednak ograniczona przestrzeń w stawie nie pozwala na utrzymanie populacji sandaczy powyżej $N = 100$. Zmodyfikuj równanie (5) i swoją symulację, tak żeby uwzględnić ograniczenia zasobów w taki sam sposób jak w przypadku modelu logistycznego opisanego równaniem (3). Jednak cały czas musimy uwzględniać stałe tempo wyłowu sandaczy k.
3. Zwróć uwagę, że dla $k = 10$ populacja sandaczy w pewnym momencie spadnie poniżej zera. Zmodyfikuj swoją symulację w taki sposób, żeby to nigdy nie mogło nastąpić, tzn. jeśli wyłowimy wszystkie sandacze to ich liczba powinna zostać na poziomie 0.
4. Rysując stosowny wykres pokaż jak zależy rozwiązanie równania dla różnych wartości k z zakresu od 0 do 20. Opisz swoje obserwacje (w polu tekstowym w notatniku Google Colab).
5. (**podpunkt jest warty 2 punkty**) Sprawdź wartość populacji sandaczy po bardzo długim czasie, np. dla $t = 100$, tak żeby osiągnęła ona stan stacjonarny. Narysuj wykres pokazujący jak populacja końcowa sandaczy zależy od parametru k modelu dla k z przedziału od 0 do 50, przy założeniu, że ich początkowa populacja wynosi $n_0 = 100$. Opisz swoje obserwacje w polu tekstowym.