Rozwiązywanie równań różniczkowych metodą Eulera i RK

Przed Wami ostatnie zadanie, którego rozwiązanie pozwoli wam zakończyć tutorial, opuścić lokację startową i rozpocząć samodzielną walkę o przetrwanie w brutalnym świecie opanowanym przez zombie. W nagrodę za wasze dotychczasowe starania otwierają się przed wami trzy ścieżki, trzy różne zadania, które zdefiniują waszą przyszłość. Wybrać możecie tylko jedną. Wybierzcie mądrze, albowiem tego wyboru nie można zmienić. No chyba, że ktoś zdecyduje się wczytać savea. Nim się pożegnamy pozwólcie, że po raz ostatni przedstawię wam broń jaką będziecie musieli wykorzystać.

Problem początkowy Cauchego

PPC stanowi podstawę rozwiązywania równań różniczkowych. A mianowicie jest to minimalna ilość danych jakie musimy posiadać, aby rozwiązać dane zagadnienie. Na początku matematycznie: Jeżeli interesuje nas funkcja $f(x)$, ale znamy tylko wartość jej pochodnej np. $f'(x) = 2x$, możemy scałkować daną pochodną, tylko w takim przypadku otrzymamy wiele funkcji: $f(x) = x^2 + C$. Aby pozbyć się niewiadomej *C*, musimy znać jedną wartość funkcji *f*. Stąd ogólny wygląd PPC:

$$\begin{cases} f'(x, y) = coś \\ f(x_0) = y_0 \end{cases}$$

Jeżeli dostaniemy wyrażenie na druga pochodną sprawa wygląda tak samo, tylko do rozwiązania musimy znać wartość pierwszej pochodnej w jakimś punkcie:

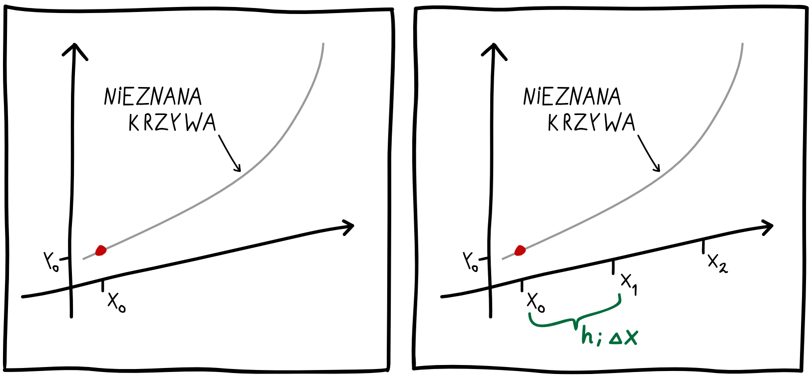
$$\begin{cases} f''(x, y, f') = coś \\ f'(x_0) = f'_0 \\ f(x_0) = y_0 \end{cases}$$

Metoda Eulera

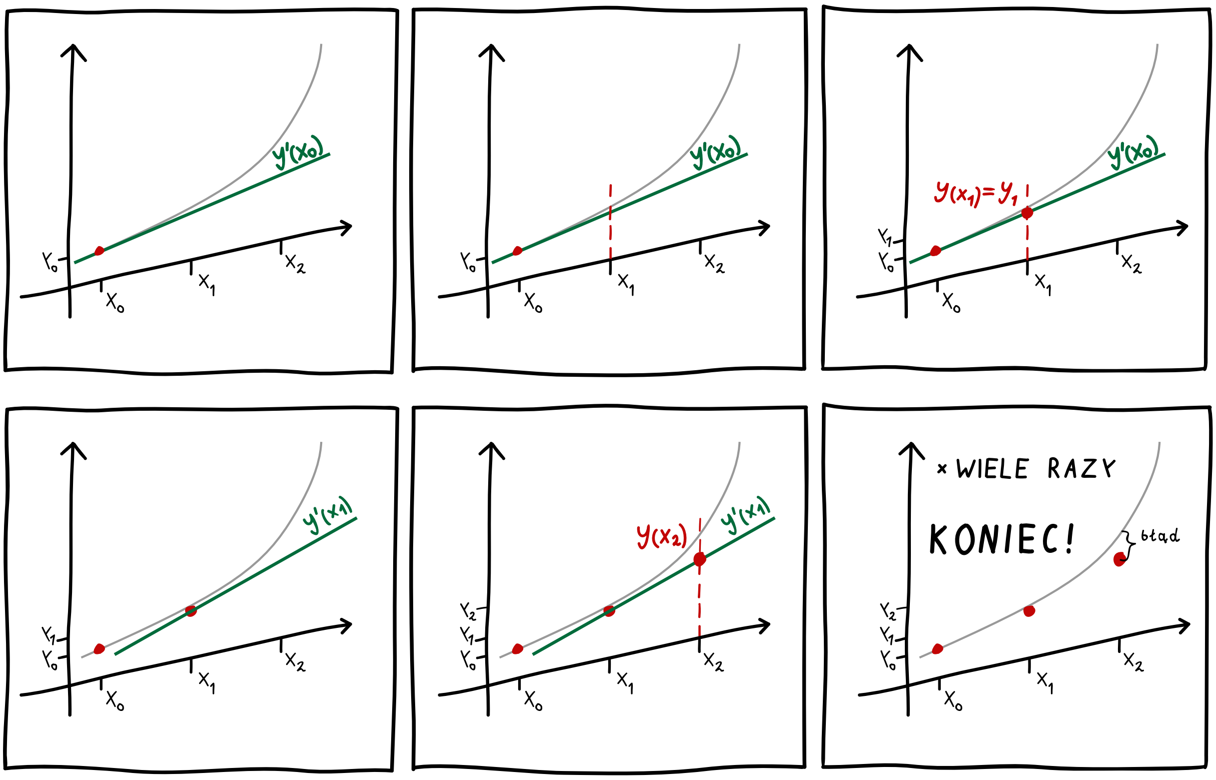
W metodach numerycznych nie poszukujemy funkcji. Nie odpowiedzą nam, że całką z *x* będzie x^2, ale mogą dość dobrze przybliżyć nam wartość funkcji w miejscach, których szukamy. I to się właśnie liczy. Metoda Eulera jest super sędziwą, bo liczącą już 255 lat, ale za to najprostszą z tych metod, której przybliżony sposób użycia wygląda następująco:

Startujemy od układu współrzędnych, na której mamy zaznaczony nasz punkt początkowy $y(x_0) = y_0$. Wybieramy sobie jakąś wartość kroku, *h* lub *dx*, która pozwoli nam zaznaczyć wiele punktów na osi x.

$$x_{i+1} = x_i + h, \quad i = 0, 1, 2, \ldots$$



Znamy wartość w punkcie *x₀*, teraz chcemy poznać tą wartość w punkcie *x₁*. Podobnie jak w metodzie Netwona, możemy wyznaczyć pochodną danej funkcji w punkcie. Wszak mamy tą pochodną wyrażoną jakimś wzorem. Po przeciągnięciu prostej wyrażającej pochodną do przecięcia z punktem *x₁* możemy zaznaczyć ten punkt jako wartość $y(x_1)$.



Otrzymaną wartość możemy wykorzystać by policzyć wartość funkcji w punkcie *x₂*, a następnie w punkcie *x₃*, *x₄* i tak dalej, i tak dalej… Ogólne wyrażenie na wartość funkcji w kolejnym kroku przyjmuje postać:

$$y(x_{i+1}) = y(x_i) + h \cdot f'(x_i, y_i)$$

Z obowiązku zaznaczam, że wartości szukanej funkcji możemy oczywiście przechowywać w postaci tablicy, dla każdego kolejnego kroku, ale bardzo często po prostu nadpisuje się pojedynczą zmienną, żeby oszczędzać miejsce i czas. A jeżeli jednak interesuje nas przebieg funkcji, to dokonuje się jedynie wypisania co którejś iteracji.

Wcale nie bardziej złożone przypadki:

Jeżeli problem, który chcemy rozwiązać składa się z wielu powiązanych z sobą równań różniczkowych sprawa wcale nie jest dużo bardziej skomplikowana. Do każdego z problemów stosujemy po prostu osobno metodę Eulera. Pamiętać trzeba tylko, że jeżeli liczymy przykładowo

$$y(x_{i+1}) = y(x_i) + h \cdot f'(x_i, y_i, z_i)$$

$$z(x_{i+1}) = z(x_i) + h \cdot g'(x_i, y_i, z_i)$$

gdy program próbuje policzyć $z(x_{i+1})$ to zamiast $y(x_i)$ dysponujemy już $y(x_{i+1})$. Czasami nie stanie się nic strasznego. Czasami jednak wszystko posypie się na skutek nawarstwiających się numerycznych. Dlatego warto albo zastosować wartość tymczasową y, albo policzyć pochodne jako osobne zmienne.

W przypadku rozwiazywania problemów wyższego rzędu stosujemy tą samą technikę jak powyżej:

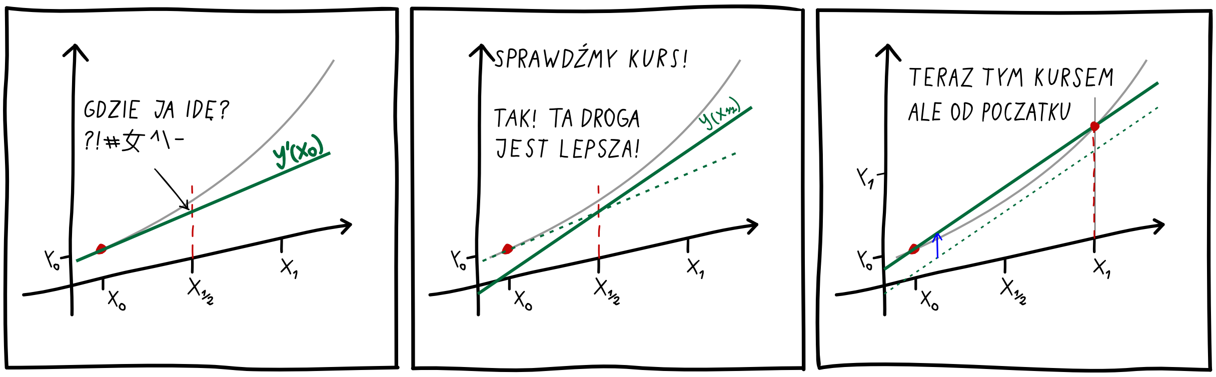
$$y(x_{i+1}) = y(x_i) + h \cdot f'(x_i, y_i, z_i, \dots)$$

$$y'(x_{i+1}) = y'(x_i) + h \cdot f''(x_i, y_i, z_i, \dots)$$

Znowu pamiętając tylko o kolejności wykonywanych obliczeń.

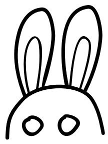
Metoda Runge-Kutty (bez s) drugiego rzędu

Jak widzicie, metoda Eulera obarczona jest błędem wynikającym z tego, że pochodna się zmienia. Błąd ten oczywiście możemy zmniejszyć skracając krok metody, ale czasami warto wykorzystać nieco bardziej wyrafinowane metody niż metoda Eulera. Kolejną w kolejce skomplikowanie jest metoda Runge-Kutty drugiego rzędu. W pierwszym kroku, liczymy tak samo jak w metodzie Eulera, ale zatrzymujemy się w środku kroku, liczymy nową, zazwyczaj bardziej poprawną pochodną. I to ją wykorzystujemy do obliczenia kolejnej wartości funkcji.



A ogólne równanie na następną wartość funkcji wygląda następująco:

$$y(x_{i+1}) = y(x_i) + h \cdot f'\left(x_i + \frac{h}{2}, y_i + \frac{1}{2} f'(x_i, y_i)\right)$$



Ścieżka 1:

Jedną z wielkich zagadek świata jest odpowiedź na pytanie jak to się stało, że pomimo tylu lat zombie nie wymarły? Wszystkie eksperymenty wykazały, że choć zombie nie człowiek, to żreć dalej musi. A skąd takie ilości jedzenia by utrzymać tak liczną populację? Drugą zagadką jest: dlaczego czasami populacja zombie się zmniejsza, by potem nagle znowu wzrosnąć?

Odpowiedzią na to pytanie mogą okazać się króliki! Tak! Te miłe, puchate stworzonka kicają sobie po łąkach, jedzą trawę, robią to co króliki robią najlepiej i przy okazji stanowią źródło wysokobiałkowej diety dla zombiaków. Taka hipoteza implikuje twierdzenie, że w swych gniazdach zombie nie tylko przeczekują dzień, ale równocześnie się rozmnażają. Nikt nie wie jak ten proces miałby się odbywać. Ba, nikt nawet nie próbuje wiedzieć, ale model to model. Z nim się nie dyskutuje. Tak postawiony problem może opisywać zagadnienie Voltera-Lotki, które w naszym przypadku wygląda następująco:

$$\frac{\partial K}{\partial t} = aK - bKZ$$

$$\frac{\partial Z}{\partial t} = cKZ - dZ$$

gdzie: K i Z to parametry opisujące populację króliczków i zombie Np. w 1000/km^2. Parametry *a, b, c* i *d* to parametry opisujące dynamikę kolejno: *a* – rozmnażania się królików (im więcej królików tym szybciej się rozmnażają), *b* – prawdopodobieństwo zastania upolowanym i zjedzonym, *c* – rozmnażanie się zombie (zależy zarówno od ilości zombie jak i dostępności futerkowanego jedzenia i *d* – śmiertelności zombie, gdy tego jedzenia brakuje.

Waszym zadaniem jest rozwiązanie ww. problemu startując od *t=0*, a kończąc na *t= 40* (lat) i przygotowanie wykresu prezentującego jak zmieniała się populacja królików i zombie przez te lata. $K_0=1$, $Z_0=0.5$, $a=b=c=d=1$. Zacznijcie od kroku czasowego $dt=0.1$ i sprawdźcie, jak zmienia się rozwiązanie wraz z jego zmniejszaniem.

Sprawdźcie również jak w tym przypadku spisze się metoda Runge-Kutty drugiego rzędu.

Ścieżka 2:

Ministerstwo szczęśliwości i dobrych nastrojów, dla poprawienia morale i przygotowania dzieci na możliwe przyszłe zagrożenia, postanowiło wydać grę komputerową „Space Zombies". W grze wcielasz się w dowódcę bazy kosmicznej, której zadaniem jest ochrona ziemi przed nadciągającymi z przestrzeni asteroidami pełnymi zielonych ludków. Niestety wszystkie silniki dające się odpalić na procesorze Intel 8080 uległy zniszczeniu wskutek burzy magnetycznej jaka powstała podczas nieudanej próby dekalaryfioryzacji Va Dinciego. Twoim zadaniem w tym projekcie byłoby stworzenie skryptu obliczającego trajektorię lotu pocisków w zmieniającym się polu grawitacyjnym.

Pozycję pocisku przechowujemy jako wektor $\boldsymbol{p} = [p_x, p_y]$, gdzie składowa x odzwierciedla położenie pocisku na dwuwymiarowej mapie gry. Standardowo x poziom, a składowa y pionie. Równania opisujące ruch pocisku to:

Zmiana pozycji pocisku $\boldsymbol{p} = [p_x, p_y]$:

$$\frac{\partial p_x}{\partial t} = v_x$$

$$\frac{\partial p_y}{\partial t} = v_y$$

Wpływ przyśpieszenia grawitacyjnego $\boldsymbol{a} = [a_x, a_y]$:

$$\frac{\partial v_x}{\partial t} = a_x$$

$$\frac{\partial v_y}{\partial t} = a_y$$

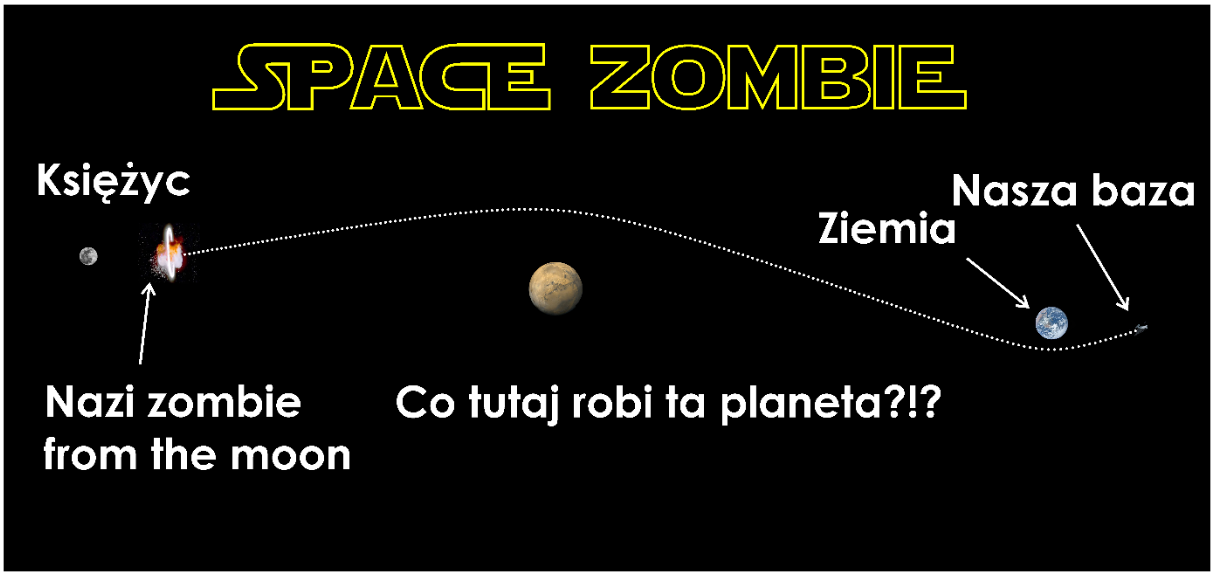
Jeżeli położenie pocisku względem ziemi wyraża się jako wektor: $\boldsymbol{o} = [o_x, o_y] = \boldsymbol{p}_z - \boldsymbol{p}$, to przyśpieszenie grawitacyjne działające na pierwsze ciało możemy obliczyć jako:

$$\boldsymbol{a} = \left[\frac{a_{sc} o_x}{|\boldsymbol{o}|}, \frac{a_{sc} o_y}{|\boldsymbol{o}|}\right]$$

$$a_{sc} = G\frac{m_2}{|\boldsymbol{o}|^2}$$

gdzie a_{sc} to skalarna wartość przyśpieszenia, G stała grawitacyjna, a m2 to masa drugiego ciała (ziemi).

Aby wykonać to zadanie trzeba policzyć trajektorię pocisku i stworzyć wykres prezentujący tor jego ruchu na mapie gry. Podobnie jak na screenshocie z bestseleru z roku 2023.



Dane do wersji demonstracyjnej, jaką macie stworzyć, to:
Maza ziemi: 5.972e24 kg
Promień ziemi: 6.371e6 m
Stała grawitacyjna: 6.6743e-11
Pozycja ziemi [0, 0] m
Pozycja bazy [2.5e7, -2.5e7] m
Zombie chciały być sprytne i próbowały podejść od przeciwnej strony, jednak chwilowo zatrzymały się na herbatę. Ich pozycja: [-3.5e7, 3.5e7] m.
Prędkość początkowa pocisku wystrzelonego z działa gausa: 4 000 m/s.
Kierunek: Wybierany przez gracza na podstawie kąta strzału.
Dodatkowe proponowane dane (te działają)
dt =1 s
t_end = 2 dni (2*24*3600) s

Ścieżka 3:

A może by tak rzucić to wszystko i polecieć w Bieszczady, nie! w Bieszczady jest za blisko, najlepiej tak polecieć z ziemi i nigdy nie wrócić. Rakieta już stoi na platformie. Waży 29 500 kg, zabiera 8 000 kg ładunku, a po zatankowaniu do pełna, czyli w jakieś 490 000 kg paliwa ponoć ma zabrać nas poza ziemię. Siłę zapewniają jej 9, chłepczących razem 875 kg paliwa na sekundę silników o łącznym ciągu 8227 kN. Nim zaryzykujecie lot, waszym zadaniem jest sprawdzenie czy ten pojazd jest w stanie spełnić oczekiwane przez was założenia. Policzcie wysokość na jaką uda się wlecieć i prędkość jaką uda się osiągnąć nim skończy się paliwo. Oczywiście uwzględniając zmianę masy i opór aerodynamiczny jaki stawia poruszająca się rakieta. (Funkcje na gęstość powietrza i opór aerodynamiczny w pliku na serwerze).

Przyjmijcie, że rakieta porusza się tylko pionowo w górę, a potrzebne równania to:

$$\frac{\partial h}{\partial t} = v$$

$$\frac{\partial v}{\partial t} = a$$

$$a = \frac{F_{tot}}{m} = \frac{F_{ciągu} - F_{graw} - F_{oporu}}{m_{rakiety} + m_{ładunku} + m_{paliwa}}$$