
Dynamika stochastyczna

Wydanie I

Jerzy Łuczka, Łukasz Machura

26/10/2013

Spis treści

1	Dynamika deterministyczna	3
1.1	Opis i modelowanie zjawisk oraz procesów przy pomocy równań różniczkowych	3
1.2	Modelowanie z czasem dyskretnym	8
1.3	Istnienie i jednoznaczność rozwiązań	8
1.4	Układy dynamiczne z czasem ciągłym	13
1.5	Dynamiczne układy zachowawcze i dysypatywne	25
1.6	Stany stacjonarne i ich stabilność	33
1.7	Atraktory	43
1.8	Dyskretnie układy dynamiczne	46
2	Chaos w układach dynamicznych	55
2.1	Determinizm a przewidywalność	55
2.2	Model chaosu. Układ bistabilny (oscylator Duffinga)	56
2.3	Scenariusz przejścia do chaosu	70
3	Dodatek numeryczny	83
3.1	Liczby losowe	83
3.2	Równania różniczkowe zwyczajne	94
3.3	Stochastyczne równania różniczkowe	101
3.4	Przykłady całkowania procesów stochastycznych	104
3.5	Lapunov	109
3.6	Bifurkacje	109
3.7	ARMA	109

Autorzy Jerzy Łuczka, Łukasz Machura

Wersja 0.2 10/2013

Pobierz podręcznik (v0.2, PDF)

Dynamika deterministyczna

1.1 Opis i modelowanie zjawisk oraz procesów przy pomocy równań różniczkowych

Jednym z podstawowych praw fizyki, jakie poznajemy w szkole średniej jest II zasada dynamiki Newtona. Opisuje ona klasyczne układy mechaniczne. Układy te są idealizacją realnych układów występujących w otaczającym nas świecie. W najprostszej wersji II zasada dynamiki Newtona w odniesieniu do jednej cząstki poruszającej się tylko wzdłuż jednej osi współrzędnych, np. wzdłuż osi OX, może być sformułowana w następującej postaci:

Ruch cząstki jest zdeterminowany przez siły jakie działają na cząstkę

Z punktu widzenia matematycznego, ruch cząstki opisany jest przez równanie Newtona:

$$ma = F \quad (1.1)$$

W równaniu tym występują trzy wielkości:

m to masa cząstki a jest przyspieszeniem cząstki F jest siłą działającą na cząstkę. Ponieważ ruch zachodzi tylko wzdłuż osi OX (tak zakładamy dla prostoty), siła F działa tylko w kierunku OX oraz przyspieszenie a jest wzdłuż osi OX.

Wiemy z kursu fizyki, że przyspieszenie cząstki jest pochodną (względem czasu) pierwszego rzędu prędkości v cząstki. Z kolei prędkość cząstki v jest pochodną pierwszego rzędu położenia cząstki x .

$$a = \frac{d}{dt}v = \frac{d}{dt} \frac{dx}{dt} = \frac{d^2x}{dt^2} \quad (1.2)$$

W ogólnej postaci siła

$$F = F(x, v, t) = F(x, dx/dt, t) \quad (1.3)$$

może zależeć od położenia x cząstki, jej prędkości $v = dx/dt$ oraz czasu t .

Jeżeli przyspieszenie a oraz siłę F w takiej postaci podstawimy do równania Newtona, to jego postać jest następująca:

$$m \frac{d^2x}{dt^2} = F \left(x, \frac{dx}{dt}, t \right) \quad (1.4)$$

W ten sposób otrzymujemy równanie różniczkowe, które opisuje jednowymiarowy ruch cząstki wzdłuż osi OX. Co możemy powiedzieć o tym równaniu:

Jest to równanie różniczkowe drugiego rzędu, ponieważ pojawia się pochodna drugiego rzędu d^2x/dt^2 . Jest to równanie różniczkowe zwyczajne, ponieważ nie występują pochodne cząstkowe a jedynie pochodne ze względu na jedną zmienną - w tym przypadku pochodne względem czasu t . Samo równanie Newtona nie wystarczy, aby opisać ruch cząstki. Musimy zadać warunki początkowe dla tego równania. Ponieważ jest to równanie drugiego rzędu, musimy zadać dwa warunki początkowe: początkowe położenie $x(t_0) = x_0$ oraz początkową prędkość $v(t_0) = v_0$. Warunki początkowe można zadać w dowolnej chwili czasu t_0 , ale zazwyczaj tą chwilą początkową jest umowna chwila $t_0 = 0$. Równanie (1.4) możemy zatem przedstawić w równoważnej postaci:

$$\frac{dx}{dt} = v \quad (1.5)$$

$$\frac{dv}{dt} = \frac{1}{m} F(x, v, t) \quad (1.6)$$

gdzie wprowadziliśmy nową zmienną v która ma interpretację prędkości cząstki. W ten sposób otrzymaliśmy układ 2 równań różniczkowych pierwszego rzędu. Jak później zobaczymy, taka manipulacja jest użyteczna przy wprowadzeniu pojęcia przestrzeni fazowej dla równań różniczkowych. Jeżeli siła F nie zależy w sposób jawny od czasu, to układ równań

$$\frac{dx}{dt} = v \quad (1.7)$$

$$m \frac{dv}{dt} = F(x, v) \quad (1.8)$$

nazywamy autonomicznym. Innymi słowy, jest to autonomiczny układ 2 równań różniczkowych zwyczajnych 1-rzędu. Mówimy wówczas, że jego przestrzeń fazowa jest 2-wymiarowa.

Jeżeli cząstka porusza się na płaszczyźnie (X, Y) , to równanie Newtona ma postać:

$$m \frac{d^2x}{dt^2} = F\left(x, y, \frac{dx}{dt}, \frac{dy}{dt}, t\right) \quad (1.9)$$

$$m \frac{d^2y}{dt^2} = G\left(x, y, \frac{dx}{dt}, \frac{dy}{dt}, t\right) \quad (1.10)$$

gdzie F i G są składowymi siły działającymi na cząstkę w kierunku x oraz y . W ogólnym przypadku siły te zależą od położenia cząstki (x, y) , jej składowych prędkości $(dx/dt, dy/dt)$ oraz czasu t .

Jeżeli składowe siły F i G nie zależą w sposób jawny od czasu, to postępując podobnie jak poprzednio otrzymamy układ:

$$\frac{dx}{dt} = v \quad (1.11)$$

$$\frac{dy}{dt} = u \quad (1.12)$$

$$m \frac{dv}{dt} = F(x, y, v, u) \quad (1.13)$$

$$m \frac{du}{dt} = G(x, y, v, u) \quad (1.14)$$

Jest to autonomiczny układ 4 równań różniczkowych zwyczajnych 1-rzędu. Mówimy wówczas, że jego przestrzeń fazowa jest 4-wymiarowa.

Dla cząstki poruszającej się w przestrzeni (X, Y, Z) , mamy 3 równania Newtona 2-rzędu. Jeżeli 3 składowe siły nie zależą w sposób jawny od czasu, to postępując podobnie jak poprzednio otrzymamy układ 6 równań różniczkowych 1-rzędu i przestrzeń fazowa jest 6-wymiarowa. W ogólności, dla N cząstek poruszających się w przestrzeni, przestrzeń fazowa ma wymiar $6N$. Analiza takich równań przekracza możliwości współczesnej matematyki w tym sensie, że mało wiemy o ogólnych własnościach konkretnych układów, które modelujemy. To powoduje, że musimy stosować numeryczne metody i komputer jest nieodzownym narzędziem analizy.

Powyżej podaliśmy jeden przykład modelowania. Bazuje on na formalizmie Newtona i równaniach ruchu Newtona. Może być stosowany do opisu dynamiki cząstek klasycznych. Czasami wygodnie jest stosować inny formalizm jak na przykład formalizm Lagrange’a lub formalizm Hamiltona. W wielu przypadkach wszystkie trzy formalizmy są równoważne. Dla tzw. układów z więzami, wygodnie jest stosować formalizm Lagrange’a lub formalizm Hamiltona.

Definiując układ równań różniczkowych jako autonomiczny, zakładaliśmy że siła nie zależy w sposób jawny od czasu. Może wydawać się, że jest to jakieś ograniczenie. Nie jest to prawdą. Układy nieautonomiczne można sprowadzić do układów autonomicznych wprowadzając dodatkową zmienną niezależną, dodatkowe “położenie”. Pokażemy to na prostym przykładzie. Rozpatrzmy cząstkę poruszającą się wzdłuż osi X . Na cząstkę działa siła tarcie proporcjonalna do prędkości cząstki, $F = -\gamma v$, działa siła potencjalna $F(x) = -V'(x)$ pochodząca od energii potencjalnej $V(x)$ (nazywanej skrótowo potencjałem). Siła ta jest ujemnym gradientem potencjału (czyli pochodną $V'(x)$). Dodatkowo na cząstkę działa periodyczna w czasie siła $F(t) = A \cos(\omega t)$. Równanie Newtona ma postać

$$m\ddot{x} = -\gamma\dot{x} - V'(x) + A \cos(\omega t) \quad (1.15)$$

gdzie kropki oznaczają pochodne względem czasu, a apostrof oznacza pochodną względem x . I tak

$$\dot{x} = \frac{dx}{dt}, \quad \ddot{x} = \frac{d^2x}{dt^2}, \quad V'(x) = \frac{dV(x)}{dx} \quad (1.16)$$

Równanie to możemy przedstawić w postaci układu 3 równań różniczkowych:

$$\dot{x} = v \quad (1.17)$$

$$m\dot{v} = -\gamma v - V'(x) + A \cos(z) \quad (1.18)$$

$$\dot{z} = \omega \quad (1.19)$$

Równoważność pokazujemy w następujący sposób:

w równaniu (1.18) należy zastąpić v z równania (1.17) wyrażeniem $v = \dot{x}$ pamiętając jednocześnie że $\dot{v} = \ddot{x}$ równanie (1.19) można scałkować i otrzymamy $z = \omega t$; wstawiamy to wyrażenie do równania (1.18). W ten sposób otrzymujemy znowu równanie (1.15). Tak więc jedno równanie różniczkowe nieautonomiczne 2-rzędu jest równoważne układowi 3 równań różniczkowych 1-rzędu. Odpowiadająca temu układowi przestrzeń fazowa jest 3-wymiarowa. Z przykładu tego płynie ważna wskazówka, jak otrzymywać autonomiczny układ równań różniczkowych 1-rzędu. Liczba tych równań definiuje przestrzeń fazową układu. Wymiar tej przestrzeni jest jedną z najważniejszych charakterystyk. Proszę to zapamiętać!

Fizyka stosuje też aparat równań różniczkowych cząstkowych. Studenci kierunku fizyka i pokrewnych kierunków znają przykłady takich równań. Równanie Schrodingera, równanie falowe, równanie dyfuzji, równania Maxwella to są równania różniczkowe cząstkowe. Ich analiza jest znacznie trudniejsza. Istnieją specjalne i specyficzne metody matematyczne pozwalające otrzymać informację o własnościach układów opisywanych takimi równaniami.

W wielu dziedzinach nauki (chemia, biologia, socjologia, nauki ekonomiczne) stosuje się fenomenologiczny sposób modelowania. Aby uzmysłowić, jak go stosować podamy jeden przykład.

1.1.1 Modelowanie procesu wzrostu

Procesy wzrostu pojawiają się na wielu obszarach. Nie trzeba być bystrym obserwatorem, aby zauważyć co wokół nas może wzrastać. My rozważamy jedną z możliwych klas procesów wzrostu: wzrost populacji zajęcy czy bakterii, wzrost depozytów pieniężnych na lokatach bankowych, wzrost stężenia substancji w reakcjach chemicznych czy wzrost liczby komórek nowotworowych. Często procesom wzrostu towarzyszą procesy malenia (zaniku, śmierci, ...). My je będziemy pomijać. Rozpatrzmy konkretny przykład: wzrost pieniędzy na lokacie bankowej. Załóżmy, że w chwili czasu t jest na lokacie $x(t)$ (np. złotych polskich). Pytamy, ile pieniędzy przyrośnie po pewnym czasie h , czyli ile pieniędzy będzie w chwili $t + h$. Zaczynamy modelować ten proces. Oznaczmy, że w chwili $t + h$ jest $x(t + h)$ pieniędzy. Na tę kwotę składają się pieniądze $x(t)$ oraz przyrost δ z odsetek, czyli

$$x(t + h) = x(t) + \delta \quad (1.20)$$

Przyrost δ zależy od wielkości oprocentowania k oraz od tego jak długo (h) trzymamy pieniądze na lokacie, czyli

$$\delta \propto x(t), \quad \delta \propto k, \quad \delta \propto h \quad (1.21)$$

Możemy to skomasować pisząc:

$$\delta = kx(t)h \quad (1.22)$$

Dlatego też

$$x(t + h) = x(t) + kx(t)h \quad (1.23)$$

Relacje tę możemy przepisać w postaci

$$\frac{x(t+h) - x(t)}{h} = kx(t) \quad (1.24)$$

W granicy małych przyrostów czasu $h \rightarrow 0$, lewa strona jest definicją pochodnej

$$\frac{dx(t)}{dt} = kx(t), \quad x(0) = x_0 \quad (1.25)$$

gdzie x_0 jest wartością początkową naszej lokaty. W ten oto sposób otrzymaliśmy równanie opisujące dynamikę wzrostu pieniędzy na naszej lokacie bankowej. Jest to równanie różniczkowe zwyczajne, 1-go rzędu, autonomiczne. Jego przestrzeń fazowa jest 1-wymiarowa.

Poniżej pokazujemy rozwiązania tego równania dla 3 różnych wartości k .

```
var('N1,N2,N3')
T = srange(0,3,0.01)
## rozwiązania dla różnych wartości k=0, 0.1, 0.2
sol=desolve_odeint( vector([0, 0.1*N2, 0.2*N3]), [5,5,5],T, [N1,N2,N3])
## wykresy rozwiązań dla różnych wartości k=-1, 0, 0.5
line(zip(T,sol[:,0]), figsize=(5, 3), legend_label="k=0") +\
  line(zip(T,sol[:,1]), color='red', legend_label="k=0.1")+ \
  line(zip(T,sol[:,2]), color='green', legend_label="k=0.2")
```

Inne procesy wzrostu także można modelować tym równaniem. Równanie to jest też punktem wyjściowym do modyfikacji, uogólnień, rozszerzeń, itp. Proste rozszerzenie polega na uzależnieniu współczynnika tempa wzrostu k od dodatkowych czynników. Na przykład przy modelowaniu wzrostu populacji zajęcy, możemy uzależnić tempo wzrostu k od liczby zajęcy w populacji: duża ilość zajęcy powoduje dużą konsumpcję pożywienia, a to z kolei skutkuje zmniejszeniem ilości pożywienia i utrudnieniami w zdobywaniu pożywienia. W efekcie zmniejsza się tempo wzrostu k . Innymi słowy, k powinno być malejącą funkcją $x(t)$ liczebników w populacji. Istnieje nieskończenie wiele takich funkcji. Na przykład

$$k \rightarrow k(x) = \exp(-bx), \quad b > 0 \quad (1.26)$$

jest malejącą funkcją x . Teraz równanie różniczkowe ma postać

$$\frac{dx}{dt} = xe^{-bx}, \quad x = x(t), \quad x(0) = x_0 \quad (1.27)$$

Jakie są skutki takiej zmiany? Pokazujemy to na poniższym rysunku. Zauważamy, że tempo wzrostu populacji zmniejsza się w porównaniu z poprzednim przypadkiem.

Model można rozszerzyć uwzględniając procesy śmierci: te naturalne i te wskutek istnienia drapieżników, które zjadają osobników populacji. Prosty model ofiara-drapieżca jest 2-wymiarowy: opisuje zmiany w populacji ofiar i zmiany w populacji drapieżników. Jest to autonomiczny układ 2 równań różniczkowych zwyczajnych.

```
var('x,y,z')
U = srange(0,300,0.01)
sol=desolve_odeint( vector([x*exp(-0.1*x), y*exp(-0.2*y), z*exp(-0.3*z)]), [5,5,5],U,
## pokazujemy rozwiązania dla różnych wartości k=-1, 0, 0.5
line(zip(U,sol[:,0]), figsize=(5, 3), legend_label="k=0")+ \
  line(zip(U,sol[:,1]), color='red', legend_label="k=0.1")+ \
  line(zip(U,sol[:,2]), color='green', legend_label="k=0.2")
```

1.2 Modelowanie z czasem dyskretnym

Powyżej otrzymaliśmy takie oto wyrażenie na przyrost:

$$x(t+h) = x(t) + khx(t) \quad (1.28)$$

Jeżeli zmiany następowałyby nie w sposób ciągły lecz dyskretny (np. co 1 dzień, co jedną godzinę) wówczas krok czasowy h jest dyskretny. Można wprowadzić oznaczenia

$$x_n = x(t), \quad x_{n+1} = x(t+h) \quad (1.29)$$

i wówczas równanie dla przyrostu ma postać

$$x_{n+1} = x_n + \alpha x_n, \quad \alpha = kh \quad (1.30)$$

W ten sposób otrzymujemy równanie z czasem dyskretnym. Ogólna postać tego typu równania to

$$x_{n+1} = f(x_n) \quad (1.31)$$

które mówi nam, jaką wartość przyjmuje dana wielkość w następnym kroku $n+1$ jeżeli znana jest wartość tej wielkości w kroku n . Równanie to nazywa się też równaniem rekurencyjnym. W zależności od postaci funkcji $f(x)$ otrzymujemy różne modele dynamiki układów.

Układ 2 równań z czasem dyskretnym ma postać

$$x_{n+1} = f(x_n, y_n) \quad (1.32)$$

$$y_{n+1} = g(x_n, y_n) \quad (1.33)$$

Analiza jakościowa takiego układu jest bardzo trudna. Czasami nieumiejętne stosowanie numerycznej analizy może skutkować tym, że umkną nam istotne cechy takiego układu, zwłaszcza gdy w układzie występują dodatkowo parametry których zmiana może powodować coś, co nazywa się bifurkacjami. Ale o tym w dalszej części książki.

1.3 Istnienie i jednoznaczność rozwiązań

Do opisu realnych zjawisk przy pomocy równań różniczkowych zwyczajnych z warunkami początkowymi zadanymi w chwili czasu $t = 0$, potrzebne nam są rozwiązania dla czasów $t > 0$ (ewolucja czasowa). Można też rozpatrywać przypadek $t < 0$ ale to zaliczyłbym do ćwiczeń matematycznych. Ważnym zagadnieniem jest istnienie rozwiązań równań różniczkowych. Możemy zapytać, czy zawsze rozwiązania równań różniczkowych istnieją i jeżeli istnieją, to czy to są jedyne rozwiązania z warunkiem początkowym. Oczywiście dla różnych warunków początkowych układ może różnie ewoluować, ale gdy startuje zawsze z tego samego stanu (warunku) początkowego to czy ewolucja jest taka sama? Na tym polega problem jednoznaczności rozwiązań. Jeżeli dla danego warunku początkowego istnieją np. 3 rozwiązania, to jak ewoluuje układ: istnieją 3 możliwości i którą możliwość wybiera układ? Gdyby tak

było dla realnych układów to nie moglibyśmy przewidywać ewolucji układu, nie moglibyśmy sterować układami, brak byłoby determinizmu. W rozwoju nauk ścisłych to właśnie determinizm stał się kołem napędowym rozwoju cywilizacyjnego ludzkości. To determinizm pozwala budować urządzenia, które działają tak jak my sobie tego życzymy: telewizor odbiera wybrany przeze mnie program, używam telefonu do komunikacji z moją rodziną, wystrzelona rakietą ma taką trajektorię jaką zaplanowałem, itd. Zbadamy 3 przykłady, które przybliżą nam powyższą problematykę. Źródło tych przykładów jest w książce: J. Hale, H. Kocak, “Dynamics and Bifurcations”. Książka jest znakomita.

1.3.1 Przykład 1

Równanie

$$\frac{dx}{dt} = -2x, \quad x(0) = x_0 \quad (1.34)$$

jest równaniem różniczkowym liniowym. Jest to jedno z najprostszych równań różniczkowych. Można łatwo sprawdzić, że funkcja

$$x(t) = x_0 e^{-2t} \quad (1.35)$$

jest rozwiązaniem i spełnia warunek początkowy $x(0) = x_0$. Funkcja ta jest dobrze określona dla wszystkich skończonych wartości czasu $t \in (-\infty, \infty)$. Nie ma tu większych ograniczeń. Jest to jedyne rozwiązanie. Poniższy rysunek daje wyobrażenie o rozwiązaniach $x(t)$ dla 3 różnych warunków początkowych. Przy okazji zauważmy, że wszystkie trzy rozwiązania dążą do tego samego stanu $x = 0$ dla długich czasów $t \rightarrow \infty$.

```
var('t')
g(t,a) = a*exp(-2*t)
p1 = plot(g(t,a=1), (t,0,2), legend_label=r"$x(0)=1$", color='blue')
p2 = plot(g(t,a=0), (t,0,2), legend_label=r"$x(0)=0$", color='red')
p3 = plot(g(t,a=-1), (t,0,2), legend_label=r"$x(0)=-1$", color='green')
show(p1+p2+p3, figsize=[6,3], axes_labels=[r'$t$', r'$x(t)$'], axes=False, frame=True)
```

1.3.2 Przykład 2

Równanie

$$\frac{dx}{dt} = 3x^2, \quad x(0) = x_0 \quad (1.36)$$

jest równaniem różniczkowym nieliniowym. Prawa strona tego równania jest określona dla wszystkich wartości x . Podobnie jak poprzednie równanie, można je rozwiązać metodą separacji zmiennych. Otrzymamy funkcję

$$x(t) = \frac{x_0}{1 - 3x_0 t} \quad (1.37)$$

która jest rozwiązaniem i spełnia warunek początkowy. Funkcja ta nie jest określona dla wszystkich skończonych wartości czasu $t \in (-\infty, \infty)$. Istnieją ograniczenia dla wartości czasu t . Ale jest to jedyne rozwiązanie.

```
var('t')
g = plot(-4.0/(1+12*t), (t,0,0.5), detect_poles='show', legend_label=r'$x(0)=-4$', color='blue')
g += plot(lambda t: 0.0, (t,0,0.5), legend_label=r'$x(0)=0$', color='red')
g += plot(1.0/(1-3*t), (t,0,1/3), detect_poles='show', legend_label=r'$x(0)=1$', color='blue')
g.show(axes_labels=[r'$t$', r'$x$'], ymin=-4, ymax=8, figsize=[6,3], axes=False, frame=True)
```

Wszystkie rozwiązania z ujemnym warunkiem początkowym $x(0) < 0$ są dobrze zdefiniowane dla wszystkich czasów $t > 0$ (krzywa niebieska). Podobnie jest z rozwiązaniem $x(t) = 0$ dla warunku początkowego $x(0) = 0$ (krzywa czerwona). Natomiast rozwiązanie z dodatnim warunkiem początkowym $x(0) > 0$ rozbiega się w skończonym czasie $t < 1/3x_0$. Gdyby to równanie miało opisywać ruch cząstki, to oznacza że w skończonym czasie cząstka przebywa nieskończoną odległość. To jest niefizyczne. Równanie to mogłoby opisywać proces wybuchu substancji: x mogłoby być objętością pęczniejącej substancji która wybucha po skończonym czasie.

1.3.3 Przykład 3

Równanie

$$\frac{dx}{dt} = 2\sqrt{x}, \quad x(0) = x_0 \geq 0 \quad (1.38)$$

jest równaniem różniczkowym nieliniowym. Prawa strona tego równania jest określona dla nieujemnych wartości $x \geq 0$. Podobnie jak 2 poprzednie równania, można je rozwiązać metodą separacji zmiennych. Otrzymamy rozwiązanie

$$x(t) = (t + \sqrt{x_0})^2 \quad (1.39)$$

Funkcja ta jest określona dla wszystkich wartości czasu $t > 0$. Jest to jedyne rozwiązanie z wyjątkiem jednego warunku początkowego: $x(0) = 0$. Dla tego warunku początkowego istnieje jeszcze jedno rozwiązanie, a mianowicie $x(t) = 0$. Tak więc dla $x(0) = 0$ mamy 2 różne rozwiązania

$$x(t) = t^2, \quad x(t) = 0 \quad (1.40)$$

Jak przebiega ewolucja, gdy układ startuje ze stanu początkowego $x(0) = 0$? W tym przypadku rozwiązania są niejednoznaczne.

```
var('t')
p1=plot(t**2, (t,0,1), legend_label=r"$x(0)=1$", color='blue')
p2=plot(0, (t,0,1), legend_label=r"$x(0)=0$", color='red')
show(p1+p2, figsize=[6,3], axes=False, frame=True)
```

Co jest takiego charakterystycznego w ostatnim przykładzie, że pojawia się niejednoznaczność rozwiązania równania różniczkowego? Na to pytanie daje odpowiedź twierdzenie o jednoznaczności rozwiązania równania różniczkowego. Potrzebna nam będzie własność funkcji:

Mówimy, że funkcja $f(x)$ spełnia warunek Lipschitza na zbiorze otwartym U jeżeli istnieje taka stała $L > 0$, że

$$|f(x_2) - f(x_1)| \leq L|x_2 - x_1| \quad (1.41)$$

dla wszystkich $x_1, x_2 \in U$.

Warunek Lipschitza można zapisać w postaci

$$|f(x+h) - f(x)| \leq Lh \quad \text{lub jako} \quad \left| \frac{f(x+h) - f(x)}{h} \right| \leq L \quad (1.42)$$

Z tego wynika że jeżeli $f(x)$ ma ograniczoną pochodną, to spełnia warunek Lipschitza. Są oczywiście nieróżniczkowalne funkcje, które spełniają warunek Lipschitza.

Twierdzenie Picarda Jeżeli funkcja $f(x)$ jest ciągła w U oraz spełnia warunek Lipschitza w U wówczas równanie różniczkowe

$$\frac{dx}{dt} = f(x), \quad x(0) = x_0 \quad (1.43)$$

ma dokładnie jedno rozwiązanie w U .

Istnieje kilka modyfikacji tego twierdzenia, ale na nasze potrzeby ta najprostsza wersja jest wystarczająca.

Teraz możemy odpowiedzieć, dlaczego w 3 przykładzie rozwiązanie jest niejednoznaczne: funkcja $f(x) = 2\sqrt{x}$ nie spełnia warunku Lipschitza ponieważ pochodna

$$\frac{df(x)}{dx} = \frac{1}{\sqrt{x}} \quad (1.44)$$

w punkcie $x = 0$ jest rozbieżna. W punktach $x > 0$ pochodna ma wartość skończoną i jest spełnione twierdzenie Picarda. Dlatego też rozwiązania są jednoznaczne.

1.3.4 Dodatek

Sage z powodzeniem jest w stanie rozwiązywać pewne równania różniczkowe zwyczajne. Zobaczmy jak poradzi sobie z powyższymi przykładami.

Przykład 1

$$\frac{dx}{dt} = -2x, \quad x(0) = x_0 \quad (1.45)$$

z rozwiązaniem

$$x(t) = x_0 e^{-2t}. \quad (1.46)$$

Na początek zadamy sobie zmienne. Druga linijka mówi o tym, że zmienna x będzie funkcją parametru t (czasu). Zamiast używać nazwy g użyjemy świeżo obliczonego rozwiązania `rozv`.

```
var('t x_0')
x = function('x', t)
rrz = diff(x,t) == -2*x
rozv = desolve(rrz, x)
rozv = rozv.subs(c=x_0)
print "rozwiązanie równania"
show(rozv)
p1 = plot(rozv(x_0=1), (t,0,2), legend_label=r"$x(0)=1$", color='blue')
p2 = plot(rozv(x_0=0), (t,0,2), legend_label=r"$x(0)=0$", color='red')
p3 = plot(rozv(x_0=-1), (t,0,2), legend_label=r"$x(0)=-1$", color='green')
show(p1+p2+p3, figsize=[6,3], axes_labels=[r'$t$', r'$x(t)$'], axes=False, frame=True)
```

Przykład 2

$$\frac{dx}{dt} = 3x^2, \quad x(0) = x_0 \quad (1.47)$$

z rozwiązaniem

$$x(t) = \frac{x_0}{1 - 3x_0 t}. \quad (1.48)$$

```
var('t x_0 c')
x = function('x', t)
print "Definiujemy równanie różniczkowe"
rrz = diff(x,t) == 3*x^2
rozv2 = desolve(rrz, x)
print "i je rozwiązujemy..."
show(rozv2)
print "krok 1\n obliczamy x(t) z poprzedniego kroku"
rozv2 = solve(rozv2,x)[0].rhs()
show(rozv2)
print "krok 2\n obliczamy x(0)"
buf = rozv2(t=0) == x_0
show(buf)
print "krok 3\n wyznaczamy stałą c"
buf = solve(buf,c)[0].rhs()
show(buf)
print "krok 4\n wstawiamy c do równania"
rozv2 = rozv2.subs(c=buf).full_simplify()
show(rozv2)
print "I na koniec prezentujemy wyniki"
x0 = -4
w = plot(rozv2(x_0=x0), (t,0,1), detect_poles='show', legend_label=r'$x(0)=%d$'%x0, color='blue')
x0 = 0
w += plot(rozv2(x_0=x0), (t,0,1), legend_label=r'$x(0)=%d$'%x0, color='red')
x0 = 1
w += plot(rozv2(x_0=x0), (t,0,1/3), legend_label=r'$x(0)=%d$'%x0, color='green')
w.show(axes_labels=[r'$t$',r'$x$'], tick_formatter='latex', xmin=0, xmax=0.5, ymin=-4.1,
```

Przykład 2

$$\frac{dx}{dt} = 2\sqrt{x}, \quad x(0) = x_0 \geq 0 \quad (1.49)$$

z rozwiązaniem

$$x(t) = (t + \sqrt{x_0})^2 \quad (1.50)$$

```
var('t x_0 c')
forget()
assume(x_0>=0)
assume(t+c>0)
print "równanie"
x = function('x', t)
rrz = diff(x,t) == 2*sqrt(x)
```



```

show(rrz)
print "i jego rozwiązanie"
rozv3 = solve(desolve(rrz, x), x)[0]
show(rozv3)
print "stała całkowania"
buf = solve(x_0 == rozv3.rhs()(t=0), c)
show(buf)
print "mamy dwa możliwe rozwiązania, wybieramy to z dodatnim c"
buf = buf[1]
show(buf)
print "i dostajemy ostatecznie"
rozv3 = rozv3.subs(c=buf.rhs())
show(rozv3)
print "I na koniec prezentujemy wyniki"
p1=plot(rozv3.rhs()(x_0=0), (t, 0, 1), legend_label=r"$x(0)=1$", color='blue')
show(p1, figsize=[6,3], axes=False, frame=True)

```

No tak, ale gdzie jest rozwiązanie $x(t) = 0$? Na chwilę obecną Sage nie rozróżni obu możliwych rozwiązań. Dlatego umiejętność analitycznego rozwiązania takich problemów wciąż jest niezbędna!

1.4 Układy dynamiczne z czasem ciągłym

We Wstępie podaliśmy kilka przykładów układów opisanych za pomocą równań różniczkowych zwyczajnych. Pierwsza klasa układów to układy opisywane przez mechanikę klasyczną i jej równania Newtona. Inna klasa układów to równania fenomenologiczne opisujące procesy wzrostu, procesy kinetyki chemicznej, dynamiki populacyjnej w układach biologicznych, itd. Te dwie klasy układów opisywane są układem równań różniczkowych zwyczajnych zapisanych w ogólnej postaci jako układ

$$\begin{aligned}
 \frac{dx_1}{dt} &= F_1(x_1, x_2, x_3, \dots, x_n) \\
 \frac{dx_2}{dt} &= F_2(x_1, x_2, x_3, \dots, x_n) \\
 &\vdots \\
 \frac{dx_n}{dt} &= F_n(x_1, x_2, x_3, \dots, x_n)
 \end{aligned} \tag{1.51}$$

Ten układ możemy zapisać w notacji wektorowej w postaci

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x}), \quad \vec{x}(0) = \vec{x}_0 \tag{1.52}$$

gdzie wektory

$$\vec{x} = [x_1, x_2, x_3, \dots, x_n], \quad \vec{F} = [F_1, F_2, F_3, \dots, F_n] \tag{1.53}$$

oraz dany jest zbiór warunków początkowych

$$\vec{x}(0) = [x_1(0), x_2(0), x_3(0), \dots, x_n(0)] = \vec{x}_0 = [x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, \dots, x_n^{(0)}] \tag{1.54}$$

Wskaźnik n mówi, ile równań różniczkowych jest “ukrytych” w powyższym zapisie wektorowym. Innymi słowy, rozważamy układ n równań różniczkowych scharaktaryzowanych przez n

funkcji skalarnych $F_i(x_1, x_2, x_3, \dots, x_n)$, ($i = 1, 2, 3, \dots, n$). Zauważmy, że rozważamy funkcje F_i które nie zależą w sposób jawny od czasu. W takim przypadku mówimy, że jest to układ autonomiczny n równań różniczkowych zwyczajnych. Ponadto wektor \vec{F} można traktować jako pole wektorowe stowarzyszone z układem równań różniczkowych lub pole wektorowe generowane przez układ równań różniczkowych. Do tej kwestii powrócimy jeszcze.

1.4.1 Istnienie i jednoznaczność rozwiązań

Do opisu realnych zjawisk przy pomocy równań różniczkowych zwyczajnych z warunkami początkowymi zadanymi w chwili czasu $t = 0$, potrzebne nam są rozwiązania dla czasów $t > 0$ (ewolucja czasowa). Można też rozpatrywać przypadek $t < 0$ ale to zaliczyłbym do ćwiczeń matematycznych. Ważnym zagadnieniem jest istnienie rozwiązań równań różniczkowych. Możemy zapytać, czy zawsze rozwiązania równań różniczkowych istnieją i jeżeli istnieją, to czy to są jedyne rozwiązania z warunkiem początkowym. Oczywiście dla różnych warunków początkowych układ może różnie ewoluować, ale gdy startuje zawsze z tego samego stanu (warunku) początkowego to czy ewolucja jest taka sama? Na tym polega problem jednoznaczności rozwiązań. Jeżeli dla danego warunku początkowego istnieją np. 3 rozwiązania, to jak ewoluuje układ: istnieją 3 możliwości i którą możliwość wybiera układ? Gdyby tak było dla realnych układów to nie moglibyśmy przewidywać ewolucji układu, nie moglibyśmy sterować układami, brak byłoby determinizmu. W rozwoju nauk ścisłych to właśnie determinizm stał się kołem napędowym rozwoju cywilizacyjnego ludzkości. To determinizm pozwala budować urządzenia, które działają tak jak my sobie tego życzymy: telewizor odbiera wybrany przeze mnie program, używam telefonu do komunikacji z moją rodziną, wystrzelona rakietą ma taką trajektorię jaką zaplanowałem, itd. Zbadamy 3 przykłady, które przybliżą nam powyższą problematykę. Źródło tych przykładów jest w książce: J. Hale, H. Kocak, "Dynamics and Bifurcations". Książka jest znakomita.

Przykład 1

Równanie

$$\frac{dx}{dt} = -2x, \quad x(0) = x_0 \quad (1.55)$$

jest równaniem różniczkowym liniowym. Jest to jedno z najprostszych równań różniczkowych. Można łatwo sprawdzić, że funkcja

$$x(t) = x_0 e^{-2t} \quad (1.56)$$

jest rozwiązaniem i spełnia warunek początkowy $x(0) = x_0$. Funkcja ta jest dobrze określona dla wszystkich skończonych wartości czasu $t \in (-\infty, \infty)$. Nie ma tu większych ograniczeń. Jest to jedyne rozwiązanie. Poniższy rysunek daje wyobrażenie o rozwiązaniach $x(t)$ dla 3 różnych warunków początkowych. Przy okazji zauważmy, że wszystkie trzy rozwiązania dążą do tego samego stanu $x = 0$ dla długich czasów $t \rightarrow \infty$.

```
var('t')
g(t,a) = a*exp(-2*t)
p1=plot(g(t,1),(t,0,2),figsize=(6, 3), legend_label="x(0)=1", color='blue' )
p2=plot(g(t,0),(t,0,2),figsize=(6, 3), legend_label="x(0)=0", color='red' )
p3=plot(g(t,-1),(t,0,2),figsize=(6, 3), legend_label="x(0)=-1", color='green' )
show(p1+p2+p3, axes_labels=[r'$t$', r'$x(t)$'], frame=True, axes=False)
```

Przykład 2

Równanie

$$\frac{dx}{dt} = 3x^2, \quad x(0) = x_0 \quad (1.57)$$

jest równaniem różniczkowym nieliniowym. Prawa strona tego równania jest określona dla wszystkich wartości x . Podobnie jak poprzednie równanie, można je rozwiązać metodą separacji zmiennych. Otrzymamy funkcję

$$x(t) = \frac{x_0}{1 - 3x_0 t} \quad (1.58)$$

która jest rozwiązaniem i spełnia warunek początkowy. Funkcja ta nie jest określona dla wszystkich skończonych wartości czasu $t \in (-\infty, \infty)$. Istnieją ograniczenia dla wartości czasu t . Ale jest to jedyne rozwiązanie.

```
var('t')
#detect_poles - wykrywanie i rysowanie biegunów
g=plot(-4.0/(1+12*t),t,0,5,detect_poles='show',legend_label=r'$x(0)=-4$',color='blue')
g+=plot(lambda t: 0.0,t,0,5,legend_label=r'$x(0)=0$',color='red')
g+=plot(1.0/(1-3*t),t,0,0.33,detect_poles='show',legend_label=r'$x(0)=1$',color='green')
g.show(axes_labels=[r'$t$',r'$x$'],tick_formatter='latex',xmin=0,xmax=0.5,ymin=-4.1,ymax=1.1)
```

Wszystkie rozwiązania z ujemnym warunkiem początkowym $x(0) < 0$ są dobrze zdefiniowane dla wszystkich czasów $t > 0$ (krzywa niebieska). Podobnie jest z rozwiązaniem $x(t) = 0$ dla warunku początkowego $x(0) = 0$ (krzywa czerwona). Natomiast rozwiązanie z dodatnim warunkiem początkowym $x(0) > 0$ rozbiega się w skończonym czasie $t < 1/3x_0$. Gdyby to równanie miało opisywać ruch cząstki, to oznacza że w skończonym czasie cząstka przebywa nieskończoną odległość. To jest niefizyczne. Równanie to mogłoby opisywać proces wybuchu substancji: x mogłoby być objętością pęczniejącej substancji która wybuchu po skończonym czasie.

Przykład 3

Równanie

$$\frac{dx}{dt} = 2\sqrt{x}, \quad x(0) = x_0 \geq 0 \quad (1.59)$$

jest równaniem różniczkowym nieliniowym. Prawa strona tego równania jest określona dla nieujemnych wartości $x \geq 0$. Podobnie jak 2 poprzednie równania, można je rozwiązać metodą separacji zmiennych. Otrzymamy rozwiązanie

$$x(t) = (t + \sqrt{x_0})^2 \quad (1.60)$$

Funkcja ta jest określona dla wszystkich wartości czasu $t > 0$. Jest to jedyne rozwiązanie z wyjątkiem jednego warunku początkowego: $x(0) = 0$. Dla tego warunku początkowego istnieje jeszcze jedno rozwiązanie, a mianowicie $x(t) = 0$. Tak więc dla $x(0) = 0$ mamy 2 różne rozwiązania

$$x(t) = t^2, \quad x(t) = 0 \quad (1.61)$$

Jak przebiega ewolucja, gdy układ startuje ze stanu początkowego $x(0) = 0$? W tym przypadku rozwiązania są niejednoznaczne.

```
var('t')
p1=plot(t*t, (t,0,1), figsize=(6, 3), legend_label="x(0)=1", color='blue' )
p2=plot(0, (t,0,1), figsize=(6, 3), legend_label="x(0)=0", color='red' )
show(p1+p2, frame=True, axes=False)
```

Co jest takiego charakterystycznego w ostatnim przykładzie, że pojawia się niejednoznaczność rozwiązania równania różniczkowego? Na to pytanie daje odpowiedź twierdzenie o jednoznaczności rozwiązania równania różniczkowego. Potrzebna nam będzie własność funkcji:

Mówimy, że funkcja $f(x)$ spełnia warunek Lipschitza na zbiorze otwartym U jeżeli istnieje taka stała $L > 0$, że

$$|f(x) - f(y)| \leq L|x - y| \quad (1.62)$$

dla wszystkich $x, y \in U$.

Warunek Lipschitza można zapisać w postaci

$$|f(x+h) - f(x)| \leq Lh \quad \text{lub jako} \quad \left| \frac{f(x+h) - f(x)}{h} \right| \leq L \quad (1.63)$$

Z tego wynika że jeżeli $f(x)$ ma ograniczoną pochodną, to spełnia warunek Lipschitza. Są oczywiście nieróżniczkowalne funkcje, które spełniają warunek Lipschitza.

Twierdzenie Picarda: Jeżeli funkcja $f(x)$ jest ciągła w U oraz spełnia warunek Lipschitza w U wówczas równanie różniczkowe

$$\frac{dx}{dt} = f(x), \quad x(0) = x_0 \quad (1.64)$$

ma dokładnie jedno rozwiązanie w U .

Istnieje kilka modyfikacji tego twierdzenia, ale na nasze potrzeby ta najprostsza wersja jest wystarczająca.

W przypadku układu równań różniczkowych, warunek Lipschitza ma postać

$$|F_i(x_1, x_2, x_3, \dots, x_n) - F_i(y_1, y_2, y_3, \dots, y_n)| \leq L \sum_{k=1}^n |x_k - y_k| \quad (1.65)$$

Nierówność ta musi być spełniona dla wszystkich funkcji F_i i twierdzenie Picarda brzmi podobnie. Warunek Lipschitza jest spełniony gdy pochodne cząstkowe są ograniczone,

$$\left| \frac{\partial F_i}{\partial x_k} \right| \leq K \quad (1.66)$$

dla dodatnich K .

Teraz możemy odpowiedzieć, dlaczego w 3 przykładzie rozwiązanie jest niejednoznaczne: funkcja $f(x) = 2\sqrt{x}$ nie spełnia warunku Lipschitza ponieważ pochodna

$$\frac{df(x)}{dx} = \frac{1}{\sqrt{x}} \quad (1.67)$$

w punkcie $x = 0$ jest rozbieżna. W punktach $x > 0$ pochodna ma wartość skończoną i jest spełnione twierdzenie Picarda. Dlatego też rozwiązania są jednoznaczne dla $x(0) > 0$.

1.4.2 Przestrzeń fazowa

Jeszcze raz przepiszemy równania różniczkowe (1.52) w notacji:

$$\begin{aligned}\frac{dx_1}{dt} &= F_1(x_1, x_2, x_3, \dots, x_n) \\ \frac{dx_2}{dt} &= F_2(x_1, x_2, x_3, \dots, x_n) \\ &\vdots \\ \frac{dx_n}{dt} &= F_n(x_1, x_2, x_3, \dots, x_n)\end{aligned}\tag{1.68}$$

Powyższy układ równań różniczkowych definiuje pewien układ dynamiczny (matematyczna definicja układu dynamicznego może być bardzo abstrakcyjna, ale na nasze potrzeby wystarczy to, co napisaliśmy). Zbiór wszystkich możliwych wartości $\{x_1, x_2, x_3, \dots, x_n\}$ tworzy zbiór który nazywamy przestrzenią fazową (?). Wymiar tej przestrzeni wynosi n , czyli tyle ile jest równań różniczkowych.

W zależności od kontekstu, będziemy stosowali różne zapisy tych samych równań.

Przykłady:

1. Jedno równanie różniczkowe. Zwykle będziemy stosowali zapis

$$\frac{dx}{dt} = \dot{x} = f(x)\tag{1.69}$$

Przestrzeń fazowa jest 1-wymiarowa.

2. Dwa równania różniczkowe. Zwykle będziemy stosowali zapis

$$\begin{aligned}\frac{dx}{dt} &= \dot{x} = f(x, y) \\ \frac{dy}{dt} &= \dot{y} = g(x, y)\end{aligned}\tag{1.70}$$

Przestrzeń fazowa jest 2-wymiarowa.

3. Trzy równania różniczkowe. Zwykle będziemy stosowali zapis

$$\begin{aligned}\frac{dx}{dt} &= \dot{x} = f(x, y, z) \\ \frac{dy}{dt} &= \dot{y} = g(x, y, z) \\ \frac{dz}{dt} &= \dot{z} = h(x, y, z)\end{aligned}\tag{1.71}$$

Przestrzeń fazowa jest 3-wymiarowa.

4. Równanie Newtona dla cząstki poruszającej się tylko wzdłuż osi OX na którą działa siła $F(x)$ zależna tylko od położenia ma postać

$$m\ddot{x} = F(x) \quad (1.72)$$

gdzie m jest masą cząstki. Jest to równanie różniczkowe 2-go rzędu i jest ono równoważne układowi 2 równań różniczkowych 1-go rzędu:

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= \frac{1}{m}F(x) \end{aligned} \quad (1.73)$$

Przestrzeń fazowa jest 2-wymiarowa: jest to zbiór możliwych położeń i prędkości cząstki, $\{x, v\}$. Mimo swej prostoty, ten model jest niesłychanie ważny. Stanowi on punkt wyjścia dla zrozumienia wielu ważnych aspektów układów dynamicznych.

Geometryczne własności przestrzeni fazowej

Krzywa fazowe

Aby uniknąć na tym etapie abstrakcyjnych definicji, będziemy rozważać dla przykładu 2-wymiarowy układ dynamiczny

$$\begin{aligned} \dot{x} &= f(x, y), & x(0) &= x_0 \\ \dot{y} &= g(x, y), & y(0) &= y_0 \end{aligned} \quad (1.74)$$

Przestrzeń fazowa jest 2-wymiarowa. Może to być płaszczyzna lub jej część. Ale może to być bardziej skomplikowany zbiór 2-wymiarowy. Na przykład może to być sfera (podobna do powierzchni piłki), może to być torus (podobny do dętki rowerowej). Mogą to być jeszcze bardziej skomplikowane obiekty 2-wymiarowe. Ale dla naszych celów wystarczy rozważać płaszczyznę. Na płaszczyźnie można estetycznie przedstawiać coś w formie rysunków. Wprowadzamy na płaszczyźnie kartezjański układ współrzędnych o osiach OX i OY. Warunek początkowy $\{x_0 = x(0), y_0 = y(0)\}$ jest punktem o odpowiednich współrzędnych. Rozwiążemy powyższy układ równań różniczkowych numerycznie przy pomocy najprostszego schematu:

$$\begin{aligned} \frac{x(t+h) - x(t)}{h} &= f(x(t), y(t)) \\ \frac{y(t+h) - y(t)}{h} &= g(x(t), y(t)) \end{aligned} \quad (1.75)$$

Przepiszemy to w postaci

$$\begin{aligned} x(t+h) &= x(t) + f(x(t), y(t))h \\ y(t+h) &= y(t) + g(x(t), y(t))h \end{aligned} \quad (1.76)$$

1. Obliczenia numeryczne musimy zacząć od warunku początkowego w chwili $t = 0$, czyli w pierwszym kroku obliczamy

$$\begin{aligned}x_1 &= x(h) = x(0) + f(x(0), y(0))h \\ y_1 &= y(h) = y(0) + g(x(0), y(0))h\end{aligned}\tag{1.77}$$

Na płaszczyźnie otrzymujemy punkt o współrzędnych $\{x_1, y_1\}$. Zaznaczmy go na płaszczyźnie. Teraz mamy już 2 punkty:

$$\{x_0, y_0\}, \quad \{x_1, y_1\}\tag{1.78}$$

2. W następnym kroku wybieramy czas $t = h$:

$$\begin{aligned}x_2 &= x(h + h) = x(2h) = x(h) + f(x(h), y(h))h \\ y_2 &= y(h + h) = y(2h) = y(h) + g(x(h), y(h))h\end{aligned}\tag{1.79}$$

Wykorzystamy oznaczenie jak wyżej: $x_1 = x(h), y_1 = y(h)$ i przepisujemy te równania w postaci

$$\begin{aligned}x_2 &= x_1 + f(x_1, y_1)h \\ y_2 &= y_1 + g(x_1, y_1)h\end{aligned}\tag{1.80}$$

Na płaszczyźnie otrzymujemy punkt o współrzędnych $\{x_2, y_2\}$. Zaznaczmy go na płaszczyźnie. Teraz mamy już 3 punkty:

$$\{x_0, y_0\}, \quad \{x_1, y_1\}, \quad \{x_2, y_2\}\tag{1.81}$$

3. Widzimy od razu, że w 3 kroku otrzymujemy równania

$$\begin{aligned}x_3 &= x_2 + f(x_2, y_2)h \\ y_3 &= y_2 + g(x_2, y_2)h\end{aligned}\tag{1.82}$$

i otrzymujemy punkt o współrzędnych $\{x_3, y_3\}$.

4. Zauważamy, że w n -tym kroku otrzymujemy równania

$$\begin{aligned}x_n &= x_{n-1} + f(x_{n-1}, y_{n-1})h \\ y_n &= y_{n-1} + g(x_{n-1}, y_{n-1})h\end{aligned}\tag{1.83}$$

22. Częściej pisze się, co się otrzymuje w następnym kroku, czyli $n+1$:

$$\begin{aligned}x_{n+1} &= x_n + f(x_n, y_n)h \\ y_{n+1} &= y_n + g(x_n, y_n)h\end{aligned}\tag{1.84}$$

Otrzymujemy równania rekurencyjne, które pozwalają wyznaczyć ewolucję układu, czyli rozwiązanie numeryczne układu równań różniczkowych. Na płaszczyźnie XY otrzymujemy ciąg punktów o współrzędnych

$$\{x_n, y_n\} \quad (1.85)$$

Jeżeli przyrost czasu h jest odpowiednio mały, to ten ciąg punktów łączymy linią ciągłą i otrzymujemy krzywą na płaszczyźnie. Ta krzywa nazywa się krzywą fazową układu dynamicznego. Mając narysowaną taką krzywą fazową, możemy wnioskować o ewolucji układu i cechach charakterystycznych zachowania się układu w czasie t . Poniżej przedstawiamy dwa przykłady: krzywe fazowe dla oscylatora harmonicznego i oscylatora harmonicznego tłumionego.

Oscylator harmoniczny Przykładem oscylatora harmonicznego jest ciało o masie m przyłączone do sprężyny i poruszające się wzdłuż osi OX . Siła działająca na to ciało jest proporcjonalna do wychylenia x od położenia równowagi i przeciwnie skierowana do wychylenia; gdy rozciągamy sprężynę w kierunku większych dodatnich wartości x to siła działa w kierunku ujemnych wartości x :

$$F = -kx \quad (1.86)$$

gdzie k charakteryzuje “sprężystość” sprężyny. Równanie Newtona ma postać:

$$m\ddot{x} = -kx, \quad \text{lub w postaci} \quad \ddot{x} = -(k/m)x = -\omega^2 x \quad (1.87)$$

gdzie $\omega^2 = k/m$. To równanie drugiego rzędu jest równoważne 2 równaniom pierwszego rzędu:

$$\begin{aligned} \dot{x} &= y, & x(0) &= x_0 \\ \dot{y} &= -\omega^2 x, & y(0) &= y_0 \end{aligned} \quad (1.88)$$

Tłumiony oscylator harmoniczny Jeżeli w poprzednim przykładzie założymy bardziej realistyczną sytuację, w której układ nie jest w próżni, ale znajduje się w środowisku (np. w powietrzu, w wodzie lub innej cieczy), to na ciało działa dodatkowa siła, a mianowicie siła tarcia (tłumienia) F_d . Siła tarcia jest proporcjonalna do prędkości cząstki i przeciwnie skierowana do kierunku ruchu

$$F_d = -\gamma_0 v = -\gamma_0 \dot{x} \quad (1.89)$$

gdzie γ_0 nazywa się współczynnikiem tarcia (tłumienia).

Siła tarcia jest związana z oddziaływaniem ciała z cząsteczkami otoczenia. Otoczenie stawia opór gdy ciało porusza się w nim i im większa jest prędkość ciała tym większy jest opór otoczenia. Doświadczamy to, gdy biegniemy cali zanurzeni w wodzie.

Uwzględniając siłę tarcia, równanie Newtona przyjmuje postać

$$m\ddot{x} = -kx - \gamma_0 \dot{x}, \quad \text{lub w postaci} \quad \ddot{x} = -\frac{k}{m}x - \frac{\gamma_0}{m}\dot{x} = -\omega^2 x - \gamma \dot{x} \quad (1.90)$$

gdzie $\omega^2 = k/m$ oraz $\gamma = \gamma_0/m$. Równanie powyższe jest równoważne 2 równaniom pierwszego rzędu:

$$\begin{aligned} \dot{x} &= y, & x(0) &= x_0 \\ \dot{y} &= -\gamma y - \omega^2 x, & y(0) &= y_0 \end{aligned} \quad (1.91)$$

Oczywiście gdy $\gamma = 0$, wówczas otrzymujemy równanie oscylatora harmonicznego bez tarcia (nie tłumionego).

Poniżej przedstawiamy krzywe fazowe dla tych 2 przykładów.

Oscylator harmoniczny bez tarcia

```
var('x y')
def schemat_eulera2D(vec, ics, Tlist):
    i = 0
    dx = [ics[0]]
    dy = [ics[1]]
    h = Tlist[i+1] - Tlist[i]
    iks(x,y) = vec[0R]*h
    igrek(x,y) = vec[1R]*h
    for time in Tlist:
        dx.append(dx[i] + iks(dx[i],dy[i]))
        dy.append(dy[i] + igrek(dx[i],dy[i]))
        i += 1
    return zip(dx,dy)
#
@interact(layout={'top':[['omega','x0','y0']], 'bottom':[['T','h']]])
def _(title=['a','b'], h=selector(['0.005','0.01','0.05','0.1','0.5','1'], default='0.1'),
    global oscylator_nietlumiony, background
    if T == 0:
        T = 2*pi/omega
    listT = srange(0,T,float(h), include_endpoint=True)
    background = desolve_odeint(vector([y,-omega^2*x]), [x0, y0], srange(0,T+0.1,0.1,includ
    oscylator_nietlumiony = schemat_eulera2D([y,-omega^2*x], [x0, y0], listT)
    print r'Parametry modelu'
    html(r'$\omega$=%s, x_0=%s, y_0=%s$'%(omega,x0,y0))
    print r'Parametry symulacji'
    html(r'$h$=%s, T=%s$'%(h,T))
    print '\nDla T=0 lista generowana jest automatycznie dla jednego okresu własnego oscyla
#
@interact
def _(krok=slider(1, len(oscylator_nietlumiony), 1, default=1, label=r'krok')):
    ...
    buf = zip(*oscylator_nietlumiony)
    minx, maxx, miny, maxy = min(buf[0]), max(buf[0]), min(buf[1]), max(buf[1])
    kroki = oscylator_nietlumiony[:krok]
    kroki_plot = list_plot(kroki, figsize=(4,4), axes_labels=[r'$x$',r'$y$'], size=30, xmin
    ...
    txt_plot = text(r'$[x_0,y_0]$', kroki[0], vertical_alignment='bottom', horizontal_alignmen
    for i in range(1,len(kroki)):
        txt_plot += text(r'$[x_{%d},y_{%d}]$'%(i,i), kroki[i], vertical_alignment='bottom', ho
    ...
    full_plot = list_plot(oscylator_nietlumiony, plotjoined=1, figsize=(4,4), axes_labels=
    full_plot += list_plot(background.tolist(), plotjoined=1, color='grey', alpha=0.5)
    html.table(["krzywe fazowe dla oscylatora bez tarcia",""],[full_plot+kroki_plot,kroki_
```

W przypadku oscylatora nietłumionego, krzywe fazowe są zamknięte. Częstka z biegiem czasu porusza się tak, że położenie x i prędkość $v = y$ leżą na krzywej fazowej. Ponieważ jest to krzywa zamknięta, to po pewnym czasie częstka znowu “przebiega” punkty, które się powtarzają. Powtarzanie się jest cechą charakterystyczną ruchu okresowego. Tak więc krzywa fazowa zamknięta przedstawia ruch okresowy (periodyczny). Okres takiego ruchu periodycznego to czas potrzebny na to, aby częstka startując od punktu np. $\{x_0, y_0\}$ i poruszając się po krzywej fazowej dotarła znowu do tego samego punktu $\{x_0, y_0\}$.

W przypadku oscylatora tłumionego, krzywą fazową jest spirala zwijająca się do punktu $\{0, 0\}$. Ruch po spirali oznacza, że zarówno x jak i $v = y$ maleją w czasie i dla długich czasów położenie x oraz prędkość v dążą do zera, czyli częstka zwalnia i na końcu zatrzymuje się. To jest ruch tłumiony: amplituda drgań maleje w czasie. To jest to, co obserwujemy w ruchu kulki zawieszanej na nitce: kulka wykonuje coraz to mniejsze drgania i po długim czasie wisi pionowo (to jest coś co nazywa się stanem równowagi lub położeniem stacjonarnym).

Gdy mamy bardziej skomplikowane krzywe fazowe, ich “rozszyfrowanie” może być trudniejsze. Ale ogólna zasada jest taka: gdy x rośnie to oznacza wzrost położenia częstki. Gdy y maleje to oznacza, że maleje prędkość częstki. Gdy x maleje to maleje współrzędna położenia częstki. Gdy y rośnie to rośnie prędkość częstki.

Tłumiony oscylator harmoniczny

```
var('x y')
def schemat_eulera2D(vec, ics, Tlist):
    i = 0
    dx = [ics[0]]
    dy = [ics[1]]
    h = Tlist[i+1] - Tlist[i]
    iks(x,y) = vec[0R]*h
    igrek(x,y) = vec[1R]*h
    for time in Tlist:
        dx.append(dx[i] + iks(dx[i],dy[i]))
        dy.append(dy[i] + igrek(dx[i],dy[i]))
        i += 1
    return zip(dx,dy)
#
@interact(layout={'top':[['omega','ggamma','x0','y0']], 'bottom':[['T','h']]])
def _(title=['a','b'], h=selector(['0.05','0.01','0.1','0.5','1'], default='0.1', button=
global oscylator_tlumiony, globggamma, globomega, background2
globggamma = ggamma
globomega = omega
if T == 0:
    T = 2*pi/omega
listT = srange(0,T,float(h),include_endpoint=True)
background2 = desolve_odeint(vector([y,-omega^2*x-ggamma*y]), [x0, y0], srange(0,2*pi/c
oscylator_tlumiony = schemat_eulera2D([y,-omega^2*x-ggamma*y], [x0, y0], listT)
print r'Parametry modelu'
html(r'\gamma=%s, \omega=%s, x_0=%s, y_0=%s'% (ggamma,omega,x0,y0))
print r'Parametry symulacji'
html(r'h=%s, T=%s'% (h,T))
print '\nDla T=0 lista generowana jest automatycznie dla jednego okresu własnego oscyla
#
@interact
def _(krok=slider(1, len(oscylator_tlumiony), 1, default=1, label=r'krok')):
    buf = zip(*oscylator_tlumiony)
```

```

minx, maxx, miny, maxy = min(buf[0]), max(buf[0]), min(buf[1]), max(buf[1])
kroki = oscylator_tlumiony[:krok]
kroki_plot = list_plot(kroki, figsize=(4,4), axes_labels=[r'$x$', r'$y$'], size=30, xmin=
txt_plot = text(r'$[x_0, y_0]$', kroki[0], vertical_alignment='bottom', horizontal_alignmen
for i in range(1, len(kroki)):
    txt_plot += text(r'$[x_{%d}, y_{%d}]$'%(i,i), kroki[i], vertical_alignment='bottom', h
full_plot = list_plot(oscylator_tlumiony, plotjoined=1, figsize=(4,4), axes_labels=[r'$
full_plot += list_plot(background2.tolist(), plotjoined=1, color='grey', alpha=0.5)
html.table([[ "krzywe fazowe dla oscylatora tłumionego", "" ], [full_plot+kroki_plot, kroki_

```

1.4.3 Pole wektorowe

Prawe strony układu równań różniczkowych

$$\begin{aligned}\dot{x} &= f(x, y), & x(0) &= x_0 \\ \dot{y} &= g(x, y), & y(0) &= y_0\end{aligned}\quad (1.92)$$

można traktować jak składowe pewnego pola wektorowego:

$$\vec{F} = [F_x, F_y] = [f(x, y), g(x, y)] \quad (1.93)$$

W każdym punkcie płaszczyzny o współrzędnych $\{x, y\}$ rysujemy wektor o składowych $[f(x, y), g(x, y)]$. W ten sposób otrzymujemy pole wektorowe. No dobrze, ale jaką informację o układzie można “wyciągnąć” z tego pola wektorowego. Wykonajmy takie oto ćwiczenie numeryczne: Startujemy z warunku początkowego $\{x_0, y_0\}$ i rysujemy w tym punkcie wektor o składowych $[f(x_0, y_0), g(x_0, y_0)]$ czyli

$$\text{w punkcie } \{x_0, y_0\} \text{ rysujemy wektor o składowych } [f(x_0, y_0), g(x_0, y_0)] \quad (1.94)$$

Jak poprzednio, rozwiązujemy numerycznie układ równań różniczkowych i obliczamy $\{x_1, y_1\}$:

$$\text{w punkcie } \{x_1, y_1\} \text{ rysujemy wektor o składowych } [f(x_1, y_1), g(x_1, y_1)] \quad (1.95)$$

Następnie obliczamy $\{x_2, y_2\}$:

$$\text{w punkcie } \{x_2, y_2\} \text{ rysujemy wektor o składowych } [f(x_2, y_2), g(x_2, y_2)] \quad (1.96)$$

W n-tym kroku iteracji obliczamy $\{x_n, y_n\}$:

$$\text{w punkcie } \{x_n, y_n\} \text{ rysujemy wektor o składowych } [f(x_n, y_n), g(x_n, y_n)] \quad (1.97)$$

Ponieważ wszystkie powyższe punkty $\{x_i, y_i\}$ leżą na krzywej fazowej, to wektory $[f(x_i, y_i), g(x_i, y_i)]$ są przyłączone do tych krzywych fazowych. Zauważamy, że wektory te są styczne do krzywej fazowej. Jeżeli $\{x_i, y_i\}$ miałyby interpretację położenia cząstki na płaszczyźnie, to wektory $[f(x_i, y_i), g(x_i, y_i)]$ miałyby interpretację prędkości ponieważ $\dot{x} = f(x, y)$ oraz $\dot{y} = g(x, y)$. Wiemy, że $\dot{x} = v_x$ jest x-ową składową prędkości, z kolei $\dot{y} = v_y$ jest y-ową składową prędkości. Innymi słowy, otrzymane pole wektorowe to pole prędkości fikcyjnej cząstki.

```

var('x y')
@interact(layout={'top':[['omega','ggamma','x0','y0']], 'bottom':[['T','h']]])
def _(title=['a','b'], h=selector(['0.05','0.01','0.1','0.5','1'], default='0.1', button
    global oscylator_tlumiony, globggamma, globomega
    globggamma = ggamma
    globomega = omega
    listT = srange(0,T,float(h))
    oscylator_tlumiony = desolve_odeint(vector([y,-omega^2*x-ggamma*y]), [x0, y0], listT,
    print r'Parametry modelu'
    html(r'\gamma=%s, \omega=%s, x_0=%s, y_0=%s'%(ggamma,omega,x0,y0))
    print r'Parametry symulacji'
    html(r'h=%s, T=%s'%(h,T))
vf = lambda u,a,b: (u[0]+u[1],u[1]-a*u[0]-b*u[1])
#
@interact
def _(krok=slider(1, len(oscylator_tlumiony), 1, default=1, label=r'krok')):
    kroki = oscylator_tlumiony[:krok]
    kroki_plot = list_plot(kroki.tolist(), figsize=(4,4), axes_labels=[r'$x$',r'$y$'], size
    pole_wektorowe = arrow(kroki[0],vf(kroki[0],globomega^2,globggamma),color='red',xmax=vf
    for krok in kroki[1:]:
        pole_wektorowe += arrow(krok,vf(krok,globomega^2,globggamma),color='red', width=.4)
    txt_plot = text(r'$[x_0,y_0]$',kroki[0],vertical_alignment='bottom',horizontal_alignmen
    for i in range(1,len(kroki)):
        txt_plot += text(r'$[x_{%d},y_{%d}]$'%(i,i),kroki[i],vertical_alignment='bottom',ho
    shadowplot = list_plot(oscylator_tlumiony.tolist(), plotjoined=1, figsize=(4,4), axes_la
    full_plot = list_plot(oscylator_tlumiony.tolist(), plotjoined=1, figsize=(4,4), axes_la
    html.table(["krzywe fazowe dla oscylatora tłumionego",""],[full_plot+kroki_plot,shadow

```

Poniżej znajdziecie komórkę, w której zachęcamy wszystkich do poeksperymentowania z różnymi modelami. Miłej zabawy...

Wskazówka: Aby całość zadziałała poprawnie musicie zadeklarować model podając dx i dy , podać wartości wszystkich parametrów (teraz jest tylko α) oraz warunki początkowe x_0 i y_0 . Na koniec zdecydujcie jaki okres dynamiki punktu chcecie symulować przypisując do zmiennej T odpowiednią wartość.

```

#####
# Model #
#####
# zmienne
var('x y')
#
# parametry
# UWAGA: jeżeli Twój model będzie zależny od innych parametrów
#         tu właśnie musisz je wszystkie wyspecyfikować
alpha = 1
#
# warunki początkowe
x0 = 1
y0 = 1
#
# model
dx = y
dy = -alpha*x - y
#
# czas (T) symulacji

```

```

T = 12
#
#####
# Symulacje + wizualizacja
#####
listT = srange(0,T,0.1,include_endpoint=True)
numeryka = desolve_odeint(vector([dx, dy]), [x0, y0], listT, [x,y])
przestrzen_fazowa = list_plot(numeryka.tolist(), plotjoined=1, figsize=(4,4), axes_labels=['x', 'y'])
pole_wektorowe = plot_vector_field([dx,dy], (x, numeryka[:,0].min(), numeryka[:,0].max()))
show(przestrzen_fazowa+pole_wektorowe)

```

1.5 Dynamiczne układy zachowawcze i dysypatywne

1.5.1 Układy zachowawcze

Niektóre układy równań różniczkowych mają specyficzną strukturę i ukryte własności. Przykłady z fizyki mają taką specyficzną strukturę. Rozpatrzmy równanie Newtona dla cząstki o jednym stopniu swobody w postaci:

$$m\ddot{x} = F(x) = -\frac{dV(x)}{dx} = -V'(x) \quad (1.98)$$

gdzie siła $F(x)$ jest potencjalna, tzn. istnieje taka funkcja $V(x)$, że $F(x) = -V'(x)$. Oznaczenie $V'(x)$ jest pochodną funkcji $V(x)$ względem zmiennej x . Funkcja $V(x)$ nazywa się energią potencjalną, ale my będziemy pisali krótko - potencjał. Jeżeli znamy siłę $F(x)$ to potencjał można znaleźć obliczając całkę:

$$V(x) = -\int_a^x F(y)dy \quad (1.99)$$

gdzie a jest dowolną liczbą wybieraną tak jak nam wygodnie. Np. możemy wybrać tak, aby w pewnym punkcie potencjał był zerowy lub nieskończony.

Równanie Newtona jest równaniem 2-go rzędu, autonomicznym. Zapiszemy je w postaci

$$\begin{aligned} \dot{x} &= v, & x(0) &= x_0, \\ m\dot{v} &= F(x) = -V'(x), & v(0) &= v_0. \end{aligned} \quad (1.100)$$

Z tego wynika, że przestrzeń fazowa układu jest 2-wymiarowa $\{x, v\}$. W tej przestrzeni fazowej może analizować krzywe fazowe. Można zauważyć, że położenie $x(t)$ oraz prędkość $v(t)$ cząstki zmieniają się w czasie zgodnie z równaniem Newtona, to istnieje pewna funkcja (kombinacja) tych 2 funkcji $x(t)$ oraz $v(t)$, która nie zmienia się w czasie:

$$E[x(t), v(t)] = \frac{1}{2}mv^2(t) + V(x(t)) = \frac{1}{2}mv^2(0) + V(x(0)) = E[x(0), v(0)] \quad (1.101)$$

Wielkość E nazywa się w fizyce całkowitą energią układu i składa się z 2 części: energii kinetycznej cząstki $E_k = mv^2/2$ oraz energii potencjalnej cząstki $E_p = V(x)$. Jeżeli E nie zmienia się w czasie, to znaczy że jest to funkcja stała ze względu na czas i pochodna względem czasu powinna być zero. Sprawdźmy to:

$$\frac{dE}{dt} = \frac{d}{dt}E[x(t), v(t)] = \frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial v} \frac{dv}{dt} = V'(x)\dot{x} + mv\dot{v} = -F(x)v + vF(x) \quad (1.102)$$

gdzie skorzystaliśmy ze związku pomiędzy siłą i energią potencjalną oraz z równania Newtona.

Ponieważ E nie zmienia się w czasie, to mówimy że jest to stała ruchu lub całka ruchu, lub całka pierwsza układu (ostatnie nazwy wydają się być dziwaczne, bo w wyrażeniu dla E nie widać żadnej całki). Istnienie stałych czy też całek ruchu ułatwia analizę układów. Pokażemy to na przykładzie oscylatora harmonicznego dla którego postać siły jest dobrze znana:

$$F(x) = -kx = -m\omega^2 x, \quad V(x) = \frac{1}{2}kx^2 = \frac{1}{2}m\omega^2 x^2, \quad \omega^2 = \frac{k}{m} \quad (1.103)$$

Prawo zachowania energii mówi, że

$$E = \frac{1}{2}mv^2(t) + \frac{1}{2}kx^2(t) = \text{const.} = \frac{1}{2}mv^2(0) + \frac{1}{2}kx^2(0) \quad (1.104)$$

Ponieważ ta wielkość jest niezmienna w czasie, to określa równanie krzywej fazowej na płaszczyźnie XY . Łatwo zauważyć, że powyższe równanie w zmiennych $\{x, y = v\}$ ma postać

$$my^2 + kx^2 = 2E \quad (1.105)$$

Jest to równanie elipsy:

$$\frac{x^2}{(2E/k)} + \frac{y^2}{(2E/m)} = 1 \quad (1.106)$$

o osiach $a = 2E/k$ oraz $b = 2E/m$. Narysujmy sobie taką elipsę dla, powiedzmy, $E = 2, k = 0.2$ oraz $m = 1$. Wiadomo, że każdy wie jak taka elipsa będzie wyglądać, ale zrobimy to bardziej po to, żeby wyrobić sobie naturalną umiejętność używania programów typu Sage do wizualizacji i interpretacji wyników.

```
@interact(layout={'top':[['E','k','m']],})
def _(title=['elipsa'], E=input_box(2,label=r'$E$', width=10), k=input_box(0.2,label=r'$k$', width=10), m=input_box(1,label=r'$m$', width=10)):
    a = 2*E/k
    b = 2*E/m
    ellipse((0,0),a,b,0,fill=True,alpha=0.3).show()
```

Elipsa jest krzywą zamkniętą, więc ruch jest periodyczny. Można sobie wyobrazić, że ruch cząstki w potencjale $V(x)$ jest podobny do ruchu cząstki we wnętrzu połówki sfery (w czasie). Nie jest to prawdą, ale takie wyobrażenie wyrabia w nas intuicję o własnościach ruchu. Poniżej przedstawiamy krok po kroku co zrobić, aby narysować krzywe fazowe układu.

Rysujemy wykres przedstawiający potencjał $V(x)$. Poniżej tego wykresu, z osią pionową ustawioną jak w wykresie dla potencjału, rysujemy 2 symetryczne krzywe zadane przez prawo zachowania energii $\frac{1}{2}mv^2 + V(x) = E$ czyli stąd wynika że $v = \pm\sqrt{\frac{2}{m}[(E - V(x))]}$. Te dwie krzywe $v = v(x, E)$ są krzywymi fazowymi. Cząstka porusza się w prawo gdy prędkość jest dodatnia $v > 0$ (zielona krzywa) i w lewo gdy prędkość jest ujemna $v < 0$ (czerwona krzywa). Prędkość jest zero wówczas, gdy $V(x) = E$. Wynika to z prawa zachowania energii (podstaw tam $v = 0$). Równanie $V(x) = E$ wyznacza punkty zwrotu x_i : cząstka w tych punktach ma zerową prędkość i zmienia kierunek ruchu (zawraca).

Spróbujemy, krok po kroku zanalizować równanie Newtona aby uzyskać krzywe fazowe.

$$m\ddot{x} = F \quad (1.107)$$

Jeżeli siła będzie liniowa $F = -kx$ to dostaniemy wyżej opisane zagadnienie oscylatora harmonicznego. Na początku musimy zadeklarować nazwy zmiennych oraz parametrów użytych w modelu. Pamiętaj - każdorazowo, jeżeli chcesz obliczać coś symbolicznie, trzeba taką linijkę napisać i ją wykonać. W kolejnych liniijkach ustalimy parametry układu, zdefiniujemy siły z jakimi mamy do czynienia i obliczymy potencjał (całka z siły brana ze znakiem minus). W następnym kroku, z prawa zachowania energii, obliczymy teraz jak prędkość zależy od położenia (owe krzywe fazowe).

```
#0 (kilka zmiennych)
var('x v')
#parametry dla wizualizacji
x0 = 1.3
v0 = 0.3
k = 0.2
m = 1
F = -k*x
#1
V = -integral(F,x)
p1 = plot(V, xmin=-x0, xmax=x0)
p1.show(figsize=4, axes_labels=[r'$x$', r'$V(x)=%s$'%V])
#
#prawo zachowania energii
E = m*v0^2 + V(x=x0)
PZE = m*v^2 + V == E
#i jego rozwiązanie
rozv = solve(PZE, v); show(rozv)
v1=rozv[0].rhs()
v2=rozv[1].rhs()
#
#ekstremalne wychylenie
#prawo zachowania energii dla v=0
rozv = solve(PZE(v=0), x); show(rozv)
xmin = rozv[0].rhs()
xmax = rozv[1].rhs()
#punkt początkowy (tak jak powyżej)
ball = (x0,V(x=x0))
p0 = point(ball,size=30)
p0 += text(r" punkt startowy",ball,vertical_alignment='bottom',horizontal_alignment='left')
#
#ekstrema
ball = (xmax,V(x=xmax))
p0 += point(ball,size=30,color='red')
p0 += text("ekstremum_",ball,vertical_alignment='bottom',horizontal_alignment='right',color='red')
p12a = line((ball,(xmax,0)),linestyle='dotted',color='grey')
ball = (xmin,V(x=xmin))
p0 += point(ball,size=30,color='red')
p0 += text("_ekstremum",ball,vertical_alignment='bottom',horizontal_alignment='left',color='red')
p12a += line((ball,(xmin,0)),linestyle='dotted',color='grey')
#
#potencjał
p1 = plot(V, xmin=xmin, xmax=xmax)
#
#krzywe fazowe
p12b = line(((xmin,0),(xmin,v2(x=0))),linestyle='dotted',color='grey')
p12b += line(((xmax,0),(xmax,v2(x=0))),linestyle='dotted',color='grey')
p2 = plot(v1, (x,xmin,xmax), color='red')
p2 += plot(v2, (x,xmin,xmax), color='green')
```

```
#
(p0+p1+p12a).show(figsize=4, axes_labels=['$x$', '$V(x)$'])
(p12b+p2).show(figsize=4, xmax=xmax)
sage:
var('x y z t')
xy_wsp = [( 'x' , 'x' ), ( 'y' , 'y' )]+[( 'z' , 'z' )]
N = len(xy_wsp)
J = matrix(SR,N)
```

1.5.2 Układy potencjalne

Układ o 1 stopniu swobody jest potencjalny (tzn. istnieje potencjał $V(x)$ pod warunkiem, że siła zależy tylko od położenia cząstki, tzn. $F = F(x)$). Jeżeli siła zależy także od prędkości cząstki, tzn. gdy $F = F(x, v)$, nie istnieje potencjał V taki aby $F = -V' = -dV/dx$. Dla układów o wielu stopniach swobody, opisywanych układem równań Newtona

$$m_i \frac{d^2 \vec{r}_i}{dt^2} = \vec{F}_i(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots, \vec{r}_N) \quad (1.108)$$

dla N cząstek, układ jest potencjalny, gdy istnieje taka funkcja skalarna $V(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots, \vec{r}_N)$, że siła działająca na i -tą cząstkę jest gradientem potencjału ze znakiem minus. Prościej jest to wyjaśnić na przykładzie 1 cząstki poruszającej się w przestrzeni 3-wymiarowej:

$$\begin{aligned} m \frac{d^2 x}{dt^2} &= F_1(x, y, z) = -\frac{\partial}{\partial x} V(x, y, z), \\ m \frac{d^2 y}{dt^2} &= F_2(x, y, z) = -\frac{\partial}{\partial y} V(x, y, z), \\ m \frac{d^2 z}{dt^2} &= F_3(x, y, z) = -\frac{\partial}{\partial z} V(x, y, z). \end{aligned} \quad (1.109)$$

W ogólnym przypadku, gdy mamy zadane 3 składowe siły F_1, F_2 oraz F_3 , nie musi istnieć tylko jedna funkcja V taka aby powyższe równania były spełnione. Nasuwa się pytanie, czy istnieje proste kryterium mówiące, że układ jest potencjalny. Jeżeli

$$\vec{F} = -\text{grad } V \quad \text{to} \quad \text{rot } \vec{F} = -\text{rot grad } V = -\vec{\nabla} \times \vec{\nabla} V \equiv 0 \quad (1.110)$$

gdzie operator $\vec{\nabla}$ jest operatorem różniczkowania

$$\vec{\nabla} = \hat{e}_x \frac{\partial}{\partial x} + \hat{e}_y \frac{\partial}{\partial y} + \hat{e}_z \frac{\partial}{\partial z} \quad (1.111)$$

Wystarczy zatem sprawdzić, czy rotacja pola sił \vec{F} jest 0.

Zadanie

Sprawdzić, czy siły $\vec{F}(x, y, z)$ o składowych

$$1. \quad F_1(x, y, z) = \frac{y}{x^2 + y^2 + z^2}, \quad F_2(x, y, z) = -\frac{x}{x^2 + y^2 + z^2}, \quad F_3(x, y, z) = \frac{z}{x^2 + y^2 + z^2} \quad (1.112)$$

$$2. \quad F_1(x, y, z) = \frac{x - z}{x^2 + y^2}, \quad F_2(x, y, z) = xe^{-y^2}, \quad F_3(x, y, z) = z + 5 \quad (1.113)$$

$$3. \quad F_1(x, y, z) = 25x^4y - 3y^2, \quad F_2(x, y, z) = 5x^5 - 6xy - 5, \quad F_3(x, y, z) = 0 \quad (1.114)$$

są potencjalne

Jeżeli układ jest potencjalny to łatwo sprawdzić, podobnie jak wyżej w przypadku układu o 1-stopniu swobody, że istnieje stała ruchu - całkowita energia układu:

$$E = \sum_i \frac{m\vec{v}^2}{2} + V(\vec{r}_1, \vec{r}_2, \vec{r}_3, \dots, \vec{r}_N) = \text{constant}, \quad \frac{dE}{dt} = 0 \quad (1.115)$$

Dlatego takie pole sił nazywa się zachowawczym polem sił. Wszystkie siły związane z potencjalnym polem sił są siłami zachowawczymi. Istnieją jednak siły, które nie są siłami potencjalnymi, mimo to pozostają siłami zachowawczymi. Przykładem może być siła Lorentza działająca na naładowaną cząstkę poruszającą się w polu magnetycznym. Nie należy tego mylić z zachowawczymi układami dynamicznymi. Tę kwestię postaramy się teraz wyjaśnić.

1.5.3 Dynamiczne układy zachowawcze i dysypatywne

W teorii układów dynamicznych ważną rolę pełnią dwa pojęcia: zachowawcze układy dynamiczne i dysypatywne układy dynamiczne. Znowu dla jasności wywodu rozpatrzmy przykład układu o 3-wymiarowej przestrzeni fazowej:

$$\begin{aligned} \dot{x} &= F_1(x, y, z), & x(0) &= x_0, \\ \dot{y} &= F_2(x, y, z), & y(0) &= y_0, \\ \dot{z} &= F_3(x, y, z), & z(0) &= z_0. \end{aligned} \quad (1.116)$$

Wybieramy w przestrzeni fazowej obszar $D(0)$ o objętości $M(0)$. Zawiera on wszystkie możliwe warunki początkowe

$$\{x_0, y_0, z_0\} \in D(0) \quad (1.117)$$

Pod wpływem ewolucji każdy punkt (x_0, y_0, z_0) z tego obszaru przejdzie po czasie t do punktu $(x(t), y(t), z(t))$. Zbiór tych punktów w chwili t tworzy obszar $D(t)$ o objętości $M(t)$. Zachodzi pytanie:

$$\text{w jakich przypadkach} \quad M(t) = M(0) \quad (1.118)$$

Innymi słowy, kiedy układ dynamiczny zachowuje objętość fazową. Zbadamy ten problem. Wprowadzimy nowe oznaczenia, aby ułatwić notację:

$$x_t = x(t), \quad y_t = y(t), \quad z_t = z(t) \quad (1.119)$$

Objętość fazowa warunków początkowych w chwili $t = 0$ wynosi

$$M(0) = \int \int \int_{D(0)} dx_0 dy_0 dz_0 \quad (1.120)$$

Objętość fazowa w chwili t wynosi

$$M(t) = \int \int \int_{D(t)} dx_t dy_t dz_t \quad (1.121)$$

Ewolucja układu to nic innego jak zamiana zmiennych $(x_0, y_0, z_0) \rightarrow (x_t, y_t, z_t)$. Dokonajmy tej zamiany zmiennych w drugiej całce:

$$M(t) = \int \int \int_{D(t)} dx_t dy_t dz_t = \int \int \int_{D(0)} \frac{\partial(x_t, y_t, z_t)}{\partial(x_0, y_0, z_0)} dx_0 dy_0 dz_0 = \int \int \int_{D(0)} J(t) dx_0 dy_0 dz_0 \quad (1.122)$$

gdzie J jest jacobianem transformacji $(x_t, y_t, z_t) \rightarrow (x_0, y_0, z_0)$. Jeżeli objętość fazowa nie zmienia się w czasie (jest funkcją stałą), to jej pochodna

$$\frac{dM(t)}{dt} = \int \int \int_{D(0)} \frac{dJ(t)}{dt} dx_0 dy_0 dz_0 \quad (1.123)$$

wynosi zero. Jeżeli

$$\frac{dJ(t)}{dt} = 0 \quad \text{to} \quad \frac{dM(t)}{dt} = 0 \quad \text{czyli} \quad M(t) = M(0) \quad (1.124)$$

Więc rozpoczynamy obliczenia

$$\frac{dJ(t)}{dt} = \frac{d}{dt} \frac{\partial(x_t, y_t, z_t)}{\partial(x_0, y_0, z_0)} = \frac{d}{dt} \begin{bmatrix} \frac{\partial x_t}{\partial x_0} & \frac{\partial x_t}{\partial y_0} & \frac{\partial x_t}{\partial z_0} \\ \frac{\partial y_t}{\partial x_0} & \frac{\partial y_t}{\partial y_0} & \frac{\partial y_t}{\partial z_0} \\ \frac{\partial z_t}{\partial x_0} & \frac{\partial z_t}{\partial y_0} & \frac{\partial z_t}{\partial z_0} \end{bmatrix} \quad (1.125)$$

Należy powyższy wyznacznik rozwinąć i pamiętać, że rozwiązania równań różniczkowych

$$x_t = x_t(x_0, y_0, z_0), \quad y_t = y_t(x_0, y_0, z_0), \quad z_t = z_t(x_0, y_0, z_0) \quad (1.126)$$

zależą od warunków początkowych $\{x_0, y_0, z_0\}$. Po rozwinięciu wyznacznika pojawiają się wyrażenia typu

$$\frac{d}{dt} \frac{\partial x_t}{\partial z_0} = \frac{\partial}{\partial z_0} \frac{dx_t}{dt} = \frac{\partial}{\partial z_0} \dot{x}_t = \frac{\partial}{\partial z_0} F_1(x_t, y_t, z_t) = \frac{\partial F_1}{\partial x_t} \frac{\partial x_t}{\partial z_0} + \frac{\partial F_1}{\partial y_t} \frac{\partial y_t}{\partial z_0} + \frac{\partial F_1}{\partial z_t} \frac{\partial z_t}{\partial z_0} \quad (1.127)$$

Jak widać, w tym prostym przypadku musimy przeprowadzić uciążliwe rachunki. Znacznie lepiej jest posłużyć się rachunkiem symbolicznym z wykorzystaniem SAGE.

Aby przeprowadzić dowód, najlepiej jest obejść ograniczenia operacji na wyrażeniach z pochodnymi w Sage. Pochodna wyznacznika jest zrobiona automatycznie, potem jest ręcznie wykonane podstawienie:

$$\frac{\partial}{\partial z_0} \dot{x}_t = \frac{\partial F_1}{\partial x_t} \frac{\partial x_t}{\partial z_0} + \frac{\partial F_1}{\partial y_t} \frac{\partial y_t}{\partial z_0} + \frac{\partial F_1}{\partial z_t} \frac{\partial z_t}{\partial z_0} \quad (1.128)$$

```

for i, (v,lv) in enumerate(xy_wsp):
for j, (u,lu) in enumerate(xy_wsp):
    J[i,j] = var("d%sd%s"%(v,u), latex_name=r'\displaystyle\frac{\partial %s_t}{\partial %s_t}')
    var("dF%sd%s"%(v,u), latex_name=r'\displaystyle\frac{\partial F_s}{\partial %s_t}')
#
to_fun = dict()
for v in J.list():
    vars()[str(v).capitalize()] = function(str(v).capitalize(),t)
    var("%sd"%str(v))
    to_fun[v]=vars()[str(v).capitalize()]
    to_fun[vars()[str(v)+"d"]]=vars()[str(v).capitalize()].diff()
to_var = dict((v,k) for k,v in to_fun.items())
#
to_rhs = dict()
for i, (v,lv) in enumerate(xy_wsp):
    for j, (u,lu) in enumerate(xy_wsp):
        to_rhs[vars()[str(v)+"d%sd%s"%(v,u)]] = sum([vars()[str(v)+"dF%sd%s"%(v,w)]*vars()[str(v)+"d%sd%s"%(w,u)] for w in J.list()])
print "Zaczynamy od macierzy Jacobiego:"
show(J)
print "Wszystkie pochodne cząstkowe są reprezentowane przez niezależne zmienne, aby polubić"
show(J.subs(to_fun))
print "Liczymy wyznaczniki pochodną, oraz wracamy do zmiennych symbolicznych:"
J.subs(to_fun).det().diff(t).subs(to_var).show()
#
print "Używając słownika to_rhs, podstawiamy prawe strony ODE:"
J.subs(to_fun).det().diff(t).subs(to_var).subs(to_rhs).show()
#
print "Ostatecznie dzielimy otrzymany wzór przez Jacobian:"
#
final = J.subs(to_fun).det().diff(t).subs(to_var).subs(to_rhs)/J.det()
final.simplify_full().show()

```

Ostatecznie otrzymamy wyrażenie

$$\frac{dJ(t)}{dt} = J(t) \left[\frac{\partial F_1}{\partial x_t} + \frac{\partial F_2}{\partial y_t} + \frac{\partial F_3}{\partial z_t} \right] = J(t) \operatorname{div} \vec{F} \quad (1.129)$$

To, co jest w nawiasie kwadratowym nazywa się dywergencją pola wektorowego \vec{F} . Wstawiamy to wyrażenie do równania (1.123) i otrzymamy

$$\frac{dM(t)}{dt} = \int \int \int_{D(0)} \frac{dJ(t)}{dt} dx_0 dy_0 dz_0 = \int \int \int_{D(0)} J(t) \operatorname{div} \vec{F} dx_0 dy_0 dz_0 = \int \int \int_{D(t)} \operatorname{div} \vec{F} dx_t dy_t dz_t \quad (1.130)$$

gdzie dokonaliśmy odwrotnego przejścia (z prawej strony na lewą stronę) jak w równaniu (1.122).

Można teraz uogólnić ten wynik na dowolną ilość wymiarów przestrzeni fazowej dla układu równań

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x}), \quad \vec{x} = [x_1, x_2, x_3, \dots, x_n], \quad \vec{F} = [F_1, F_2, F_3, \dots, F_n] \quad (1.131)$$

i otrzymamy

Twierdzenie Jeżeli dywergencja pola wektorowego \vec{F} danego równania różniczkowego jest zero,

$$\operatorname{div} \vec{F} = \sum_i \frac{\partial F_i}{\partial x_i} = 0 \quad (1.132)$$

wówczas objętość fazowa jest zachowana, $M(t) = M(0)$. Takie układy dynamiczne nazywamy zachowawczymi. Jeżeli objętość fazowa maleje w czasie, to układ nazywamy dysypatywnym. Innymi słowy, układ jest dysypatywny gdy objętość $M(t) < M(0)$ dla $t > 0$. Oznacza to, że dla układów dysypatywnych

$$\frac{dM(t)}{dt} < 0 \quad (1.133)$$

Gdyby

$$\operatorname{div} \vec{F} = C_0 = \text{const.} \quad (1.134)$$

wówczas z równania (1.130) otrzymujemy prostą relację

$$\frac{dM(t)}{dt} = C_0 M(t) \quad (1.135)$$

która pozwala rozstrzygnąć czy układ jest dysypatywny.

Przykład 1: Oscylator harmoniczny tłumiony

$$\begin{aligned} \dot{x} &= y = F_1(x, y), & x(0) &= x_0, \\ \dot{y} &= -\gamma y - \omega^2 x = F_2(x, y), & y(0) &= y_0. \end{aligned} \quad (1.136)$$

Łatwo obliczyć dywergencję pola

$$\operatorname{div} \vec{F} = \frac{\partial F_1}{\partial x} + \frac{\partial F_2}{\partial y} = -\gamma < 0 \quad (1.137)$$

Równanie (1.134) przyjmuje postać

$$\frac{dM(t)}{dt} = -\gamma M(t), \quad \text{jego rozwiązaniem jest funkcja malejąca} \quad M(t) = M(0)e^{-\gamma t} \quad (1.138)$$

czyli objętość fazowa (w tym przypadku powierzchnia fazowa) maleje w czasie i dlatego jest to dysypatywny układ dynamiczny.

Przykład 2: Model Lorenza

$$\begin{aligned} \dot{x} &= \sigma(y - x) = F_1(x, y, z), & x(0) &= x_0, \\ \dot{y} &= x(\rho - z) - y = F_2(x, y, z), & y(0) &= y_0, \\ \dot{z} &= xy - \beta z = F_3(x, y, z), & z(0) &= z_0. \end{aligned} \quad (1.139)$$

gdzie wszystkie parametry są dodatnie: $\sigma, \rho, \beta > 0$.

Obliczymy dywergencję 3-wymiarowego pola $\vec{F} = [F_1, F_2, F_3]$. Proste rachunki pokazują, że

$$\operatorname{div} \vec{F} = \frac{\partial F_1}{\partial x} + \frac{\partial F_2}{\partial y} + \frac{\partial F_3}{\partial z} = -\sigma - 1 - \beta < 0 \quad (1.140)$$

Objętość fazowa (w tym przypadku faktycznie objętość w 3 wymiarowej przestrzeni) maleje eksponencjalnie w czasie, podobnie jak w poprzednim przykładzie. Dlatego też jest to dysypatywny układ dynamiczny.

1.6 Stany stacjonarne i ich stabilność

Czasami rozwiązanie równań różniczkowych dąży do stałej wartości dla długich czasów, $t \rightarrow \infty$. Mówimy wówczas, że istnieje rozwiązanie stacjonarne (stan stacjonarny, punkt stały, punkt równowagi):

$$\lim_{t \rightarrow \infty} \vec{x}(t) = \vec{x}_s \quad (1.141)$$

Może istnieć kilka stanów stacjonarnych, a nawet nieskończenie wiele stanów stacjonarnych. Który z tych stanów się realizuje, zależy to od warunku początkowego $\vec{x}(0) = \vec{x}_0$. Rozwiązanie stacjonarne \vec{x}_s nie zależy od czasu. Nazwa “rozwiązanie stacjonarne” nie jest bezpodstawne. Faktycznie jest to rozwiązanie układu równań różniczkowych

$$\frac{d\vec{x}}{dt} = \vec{F}(\vec{x}), \quad \vec{x}(0) = \vec{x}_0 \quad (1.142)$$

z warunkiem początkowym $\vec{x}(0) = \vec{x}_s$. Jeżeli $\vec{x}(t) = \vec{x}_s$ jest rozwiązaniem stacjonarnym to musi spełniać układ (1.142), czyli

$$\text{jeżeli } \vec{x}(t) = \vec{x}_s \quad \text{to} \quad \frac{d\vec{x}(t)}{dt} = \frac{d\vec{x}_s}{dt} = 0 = \vec{F}(\vec{x}(t)) = \vec{F}(\vec{x}_s) = 0 \quad (1.143)$$

Innymi słowy, stan stacjonarny wyznaczony jest z warunku:

$$\vec{F}(\vec{x}_s) = 0 \quad (1.144)$$

Powyższy warunek to układ n -równań algebraicznych. Zwykle udaje nam się go rozwiązać analitycznie w niewielu przypadkach, w szczególności gdy wymiar przestrzeni fazowej $n > 1$. Natomiast możemy taki układ rozwiązywać numerycznie. Jeżeli już wyznaczymy stan stacjonarny, to nasuwa się pytanie na ile jest on stabilny, tzn. gdy nieco wytrącimy układ z tego stanu to czy powróci on do poprzedniego stanu stacjonarnego czy też oddali się od niego. Na przykład stanem stacjonarnym kulki poruszającej się na nitce w polu ziemskim jest położenie pionowe. Gdy kulkę wychylimy z tego położenia, po dostatecznie długim czasie powróci ona do pozycji pionowej i tak tam pozostanie nieruchoma. Jest to stabilny stan równowagi. Rozważmy teraz kulkę mogącą poruszać się tylko po sferze. Gdy umieścimy kulkę na biegunie północnym sfery w polu ziemskim to nieskończenie małe zaburzenie spowoduje, że kulka spadnie z tego położenia i nigdy do niego nie powróci. W obu tych przypadkach zakładamy rzeczywiste warunki ruchu z tarcie. Pominięcie tarcia spowoduje radykalnie różne zachowanie. Te dwa przykłady pozwalają nabyć intuicję, co to znaczy że stan stacjonarny jest stabilny lub jest niestabilny. Z grubsza można powiedzieć, że stan stacjonarny \vec{x}_s jest stabilny jeśli każda trajektoria startująca z punktu bliskiego \vec{x}_s pozostaje blisko \vec{x}_s wraz z upływem czasu. Natomiast \vec{x}_s jest niestabilny gdy każda trajektoria startująca z punktu bliskiego \vec{x}_s oddala się od tego punktu gdy $t \rightarrow \infty$. Można podać bardziej precyzyjne definicje.

Definicja Mówimy, że stan stacjonarny \vec{x}_s jest stabilny jeżeli dla dowolnego $\epsilon > 0$ istnieje takie $\delta(\epsilon) > 0$, że dla każdego \vec{x}_0 takiego że $|\vec{x}_0 - \vec{x}_s| < \delta$ rozwiązanie $\vec{x}(t)$ spełnia nierówność: $|\vec{x}(t) - \vec{x}_s| < \epsilon$ dla dowolnych czasów $t > 0$. Jeżeli dodatkowo $\lim_{t \rightarrow \infty} \vec{x}(t) = \vec{x}_s$ to stan stacjonarny \vec{x}_s jest asymptotycznie stabilny.

Innymi słowy dla stabilnych stanów rozwiązanie $\vec{x}(t)$ jest cały czas blisko rozwiązania stacjonarnego \vec{x}_s , a dla asymptotycznie stabilnych stanów rozwiązanie $\vec{x}(t)$ dąży do \vec{x}_s gdy czas $t \rightarrow \infty$.

1.6.1 Przypadek A: Jedno równanie różniczkowe

Przykład: Równanie różniczkowe liniowe

$$\dot{x} = \lambda x = f(x), \quad \lambda \in R \quad (1.145)$$

Stan stacjonarny znajdujemy jako rozwiązanie równania

$$f(x_s) = \lambda x_s = 0 \quad \text{stąd otrzymujemy stan stacjonarny} \quad x_s = 0 \quad (1.146)$$

Pytamy, czy ten stan jest stabilny. Musimy zaburzyć rozwiązanie stacjonarne $x(t) = x_s = 0$ i wystartować z dostatecznie bliskiego w stosunku do $x_s = 0$ warunku początkowego X_0 . Rozwiązaniem równania różniczkowego liniowego jest funkcja

$$x(t) = X_0 e^{\lambda t} \quad (1.147)$$

Jeżeli

$$\lambda < 0 \quad \text{to} \quad x(t) \rightarrow 0 \quad \text{dla wszystkich } X_0 \text{ bliskich } 0 \quad (1.148)$$

Wówczas stan stacjonarny $x_s = 0$ jest stabilny i dodatkowo jest asymptotycznie stabilny ponieważ rozwiązanie to dąży do $x_s = 0$ gdy $t \rightarrow \infty$.

Jeżeli

$$\lambda > 0 \quad \text{to} \quad x(t) \rightarrow \infty \quad \text{dla wszystkich } X_0 \text{ bliskich } 0 \quad (1.149)$$

Wówczas stan stacjonarny $x_s = 0$ jest niestabilny.

```
var('x')
pa1=plot(0.005*exp(-x), (x,0,1), figsize=(6,3), color="red")
pa2=plot(0.005*exp(x), (x,0,1), figsize=(6,3))
pa3=plot(-0.005*exp(-x), (x,0,1), figsize=(6,3), color="red")
pa4=plot(-0.005*exp(x), (x,0,1), figsize=(6,3))
show(pa1+pa2+pa3+pa4)
```

Na rysunku przedstawiono zagadnienie stabilności: czerwone krzywe dążą do 0 gdy $t \rightarrow \infty$. Niebieskie krzywe uciekają od 0 gdy $t \rightarrow \infty$.

Warunki początkowe $x(0) = \pm 0.05$ są blisko stanu stacjonarnego $x_s = 0$. Przykład ten sugeruje nam metodę badania stabilności stanu stacjonarnego. Teraz podamy tę metodę.

1.6.2 Metoda linearyzacji badania stabilności

Rozpatrujemy układ 1-wymiarowy:

$$\frac{dx}{dt} = \dot{x} = f(x), \quad f(x_s) = 0 \quad (1.150)$$

Aby zbadać stabilność stanu x_s , analizujemy zaburzenie

$$h(t) = x(t) - x_s, \quad |h(0)| = |x(0) - x_s| < \delta \quad (1.151)$$

Funkcja $h(t)$ powinna być mała, gdy stan x_s jest stabilny. Jak daleko jest rozwiązanie $x(t)$ od rozwiązania x_s . Zobaczmy, jakie równanie różniczkowe spełnia $h(t)$:

$$\frac{dh}{dt} = \frac{d}{dt}[x(t) - x_s] = \frac{dx}{dt} = f(x) = f(x_s + h) = f(x_s) + f'(x_s)h + \frac{1}{2}f''(x_s)h^2 + \frac{1}{3!}f'''(x_s)h^3 + \dots \quad (1.152)$$

Ponieważ $f(x_s) = 0$, otrzymujemy równanie różniczkowe dla odchylenia $h(t)$ od stanu stacjonarnego

$$\frac{dh}{dt} = f'(x_s)h + \frac{1}{2}f''(x_s)h^2 + \frac{1}{3!}f'''(x_s)h^3 + \dots \quad (1.153)$$

Jeżeli $f'(x_s) \neq 0$, to pierwszy istotny wyraz w tym równaniu jest liniowy względem h . Wyrazy h^2, h^3 i wyższych potęg są pomijalnie małe. Jeżeli np. $h = 10^{-2}$ to $h^2 = 10^{-4}, h^3 = 10^{-6}$. Wówczas h^2, h^3 i wyższe potęgi h można pominąć jako bardzo małe. Otrzymujemy równanie różniczkowe liniowe

$$\frac{dh}{dt} = \lambda h, \quad \lambda = f'(x_s) \quad (1.154)$$

z rozwiązaniem

$$h(t) = h(0)e^{\lambda t} \quad (1.155)$$

Wiemy już z powyższego przykładu, że gdy $\lambda < 0$ to $h(t) \rightarrow 0$ gdy $t \rightarrow \infty$. To oznacza, że zaburzenie $x(t) \rightarrow x_s$ gdy $t \rightarrow \infty$ i wówczas stan stacjonarny x_s jest stabilny. Jeżeli $\lambda > 0$ to $|h(t)| \rightarrow \infty$ gdy $t \rightarrow \infty$. To oznacza, że zaburzenie $x(t)$ ucieka od x_s gdy $t \rightarrow \infty$ i wówczas stan stacjonarny x_s jest niestabilny. Otrzymujemy następujące kryterium na stabilność stanu stacjonarnego:

Jeżeli $\lambda = f'(x_s) < 0$ to stan stacjonarny jest stabilny; jeżeli $\lambda = f'(x_s) > 0$ to stan stacjonarny jest niestabilny.

Jeżeli $\lambda = f'(x_s) = 0$ to nie wiem nic na temat stabilności. Musimy badać następne niezerowe wyrazy. Jeżeli $f''(x_s) \neq 0$ zatrzymujemy pierwszy nieznikający wyraz czyli badamy równanie

$$\frac{dh}{dt} = \gamma h^2, \quad \gamma = \frac{1}{2}f''(x_s) \quad (1.156)$$

Jeżeli $f'(x_s) = 0$ oraz $f''(x_s) = 0$ to bierzemy następny nieznikający wyraz i badamy równanie

$$\frac{dh}{dt} = \nu h^3, \quad \nu = \frac{1}{3!}f'''(x_s) \quad (1.157)$$

Jeżeli w tych przypadkach $h(t) \rightarrow 0$ gdy $t \rightarrow \infty$, to stan stacjonarny x_s jest stabilny. W przeciwnym przypadku - nie jest stabilny.

1.6.3 Metoda potencjału badania stabilności

W jednym wymiarze, równanie różniczkowe zawsze można przedstawić w postaci

$$\frac{dx}{dt} = \dot{x} = f(x) = -\frac{dV(x)}{dx} = -V'(x), \quad f(x_s) = 0 \quad (1.158)$$

gdzie funkcja

$$V(x) = - \int f(x) dx \quad (1.159)$$

nazywana jest “potencjałem”. W ogólności to nie jest potencjał fizyczny który pojawia się w równaniu Newtona dla cząstki z tłumieniem:

$$m\ddot{x} + \gamma\dot{x} = -V'(x) \quad (1.160)$$

Jeżeli ruch cząstki odbywa się w środowisku o dużym tarcu, w tzw. reżimie przetłumionym, w którym przyspieszenie cząstki jest znikomo małe (formalnie gdy $m = 0$), wówczas równanie Newtona ma postać

$$\gamma\dot{x} = -V'(x) \quad (1.161)$$

które po przeskalowaniu ma postać:

$$\dot{x} = -\frac{1}{\gamma}V'(x) = -\tilde{V}'(x) \quad (1.162)$$

Stąd też “historycznie” wywodzi się nazwa potencjał dla abstrakcyjnego układu dynamicznego:

$$\dot{x} = f(x) = -V'(x) \quad (1.163)$$

Stan stacjonarny x_s określony równaniem

$$f(x_s) = -V'(x_s) = 0 \quad (1.164)$$

to punkt ekstremalny potencjału (o ile pochodna parzystego rzędu $V^{(2k)}(x_s) \neq 0$). Punkt x_s jest stabilny gdy

$$\lambda = f'(x_s) = -V''(x_s) < 0 \quad (1.165)$$

czyli $V''(x_s) > 0$. To odpowiada minimum potencjału. Natomiast punkt x_s niestabilny odpowiada maksimum potencjału. Mamy więc klarowy obraz: Rysujemy potencjał $V(x)$. Minima potencjału to stabilne stany stacjonarne; maksima potencjału to niestabilne stany stacjonarne.

Zadania

Wyznacz stany stacjonarne i zbadaj ich (asymptotyczną) stabilność. Korzystaj z metody linearyzacji i metody potencjału.

1. $\dot{x} = 4x - x^3$
2. $\dot{x} = 1 + x^4$
3. $\dot{x} = 3 \sin(x)$
4. $\dot{x} = 2x \sin(x)$
5. $\dot{x} = -x^2 \sin(x)$

Poniżej pokazujemy wyniki dla zadania 1. Są 3 stany stacjonarne: $x_{s1} = 2, x_{s2} = 0, x_{s3} = -2$. Stany $x_{s1} = 2$ oraz $x_{s3} = -2$ są asymptotycznie stabilne (rozwiązania dążą do tych stanów). Stan $x_{s2} = 0$ jest niestabilny (rozwiązania uciekają od tego stanu).


```

var('x,y,z,u,Z,Y,t')
T0 = srange(0,1.5,0.01)
f11=4*x-x^3
f12=4*y-y^3
f13=4*z-z^3
f15=4*u-u^3
f16=0
sol5=desolve_odeint( vector([f11, f12, f13, 0, 0,f15]), [4,0.2,-0.2,2,-2,-4],T0,[x,y,z,u,Z,Y,t])
line( zip ( T0,sol5[:,0]) ,figsize=(7, 4)) +\
... line( zip ( T0,sol5[:,1]) ,color='red')+\
... line( zip ( T0,sol5[:,2]) ,color='green') +\
... line( zip ( T0,sol5[:,4]) ,color='gray') +\
... line( zip ( T0,sol5[:,5]) ,color='black')+\
... line( zip ( T0,sol5[:,3]) ,color='violet')

```

Z powyższego przykładu zauważmy następujące cechy:

1. Zbiór warunków początkowych dzieli się na dwa podzbiory $A_1 = (-\infty, 0)$ oraz $A_2 = (0, \infty)$. Dla warunków początkowych ze zbioru A_1 rozwiązania $x(t) \rightarrow -2$ dla $t \rightarrow \infty$, a dla warunków początkowych ze zbioru A_2 rozwiązania $x(t) \rightarrow 2$ dla $t \rightarrow \infty$.
2. Krzywe $x(t)$ są funkcjami monotonicznymi: albo cały czas rosną w czasie, albo cały czas maleją gdy czas rośnie. Dlaczego? Rozważmy 2 przykłady warunków początkowych widocznych na rysunku:
 1. gdy $x(0) = 4$ to $(dx/dt)(x = 4) = 4 * 4 - 4^3 = -48 < 0$. Pochodna jest ujemna, a to oznacza że funkcja maleje. Podobnie jest dla wszystkich warunków początkowych $x(0) > 2$
 2. gdy $x(0) = 0.2$ to $(dx/dt)(x = 0.2) = 4 * 0.2 - (0.2)^3 > 0$. Pochodna jest dodatnia, a to oznacza że funkcja rośnie. Podobnie jest dla wszystkich warunków początkowych $x(0) \in (0, 2)$
3. Różne krzywe $x(t)$ nigdy się nie przecinają. Wynika to z tego, że rozwiązania są jedyne i jednoznaczne. Jeżeli by się przecinały, to z punktu przecięcia wychodziłoby kilka rozwiązań. A to jest niemożliwe na mocy twierdzenia o jednoznaczności rozwiązań.

1.6.4 Przypadek B: Układ 2 równań różniczkowych

Dla jasności prezentacji, rozpatrujemy układ 2 równań różniczkowych

$$\begin{aligned}\dot{x} &= f(x, y), \\ \dot{y} &= g(x, y).\end{aligned}\tag{1.166}$$

Stany stacjonarne (x_s, y_s) określone są jako rozwiązania układu równań algebraicznych

$$\begin{aligned}f(x_s, y_s) &= 0, \\ g(x_s, y_s) &= 0.\end{aligned}\tag{1.167}$$

Gdyby istniał potencjał $V(x, y)$, powyżej przedstawione wnioski otrzymane z metody potencjału są także prawdziwe dla układów wielowymiarowych. Ponieważ w ogólnym przypadku

nie musi istnieć “potencjał”, zbadamy stabilność stanów (x_s, y_s) stosując metodę linearyzacji. Definiujemy nowe funkcje

$$\begin{aligned} X(t) &= x(t) - x_s, \\ Y(t) &= y(t) - y_s. \end{aligned} \quad (1.168)$$

Charakteryzują one odchylenie od stanu stacjonarnego, które są małe gdy stan stacjonarny jest stabilny. Zobaczymy, jakie równania różniczkowe spełniają te funkcje:

$$\dot{X}(t) = \dot{x}(t) - \dot{x}_s = \dot{x}(t) = f(x(t), y(t)) = f(x_s + X(t), y_s + Y(t)) = f(x_s, y_s) + \frac{\partial f}{\partial x} X + \frac{\partial f}{\partial y} Y + \dots \quad (1.169)$$

$$\dot{Y}(t) = \dot{y}(t) - \dot{y}_s = \dot{y}(t) = g(x(t), y(t)) = g(x_s + X(t), y_s + Y(t)) = g(x_s, y_s) + \frac{\partial g}{\partial x} X + \frac{\partial g}{\partial y} Y + \dots \quad (1.170)$$

Wszystkie pochodne cząstkowe są obliczane w punkcie (x_s, y_s) . Ponieważ w punkcie stacjonarnym $f(x_s, y_s) = g(x_s, y_s) = 0$ otrzymujemy zlinearyzowane równania różniczkowe w postaci

$$\dot{X} = a_{11}X + a_{12}Y \quad (1.171)$$

$$\dot{Y} = a_{21}X + a_{22}Y \quad (1.172)$$

gdzie macierz współczynników

$$J = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \text{obliczona w punkcie } (x_s, y_s) \quad (1.173)$$

jest macierzą Jacobiego. Ponieważ otrzymaliśmy liniowy układ równań różniczkowych, jego rozwiązanie poszukujemy w postaci funkcji

$$X(t) = Ae^{\lambda t}, \quad Y(t) = Be^{\lambda t} \quad (1.174)$$

gdzie stałe A oraz B są określone przez warunki początkowe $(X(0), Y(0))$.

Zauważamy, że

$$\dot{X} = \lambda X, \quad \dot{Y} = \lambda Y \quad (1.175)$$

Wstawiamy te relacje do układu równań zlinearyzowanych:

$$\lambda A = a_{11}A + a_{12}B \quad (1.176)$$

$$\lambda Y = a_{21}A + a_{22}B \quad (1.177)$$

Jest to zagadnienie własne dla macierzy Jacobiego, gdzie λ to są wartości własne macierzy Jacobiego. To jest także równoważne liniowemu układowi równań algebraicznych

$$(a_{11} - \lambda)A + a_{12}B = 0 \quad (1.178)$$

$$a_{21}A + (a_{22} - \lambda)B = 0 \quad (1.179)$$

Taki układ ma niezerowe rozwiązanie dla A oraz B gdy wyznacznik

$$\text{Det}|J - \lambda I| = \text{Det} \begin{bmatrix} a_{11} - \lambda & a_{12} \\ a_{21} & a_{22} - \lambda \end{bmatrix} = (a_{11} - \lambda)(a_{22} - \lambda) - a_{12}a_{21} = \lambda^2 - (a_{11} + a_{22})\lambda + a_{11}a_{22} - a_{12}a_{21} \quad (1.180)$$

Macierz I jest macierzą jednostkową, tzn. diagonalną 2×2 i wszystkie elementy diagonalne to 1. Z powyższej relacji otrzymujemy równanie kwadratowe dla nieznanych wartości własnych $\lambda = \lambda_1$ oraz $\lambda = \lambda_2$.

Rozwiązanie zlinearyzowanego układu jest w postaci kombinacji liniowej :

$$X(t) = A_1 e^{\lambda_1 t} + A_2 e^{\lambda_2 t}, \quad Y(t) = B_1 e^{\lambda_1 t} + B_2 e^{\lambda_2 t} \quad (1.181)$$

Pytanie o stabilność stanu stacjonarnego (x_s, y_s) jest równoważne pytaniu: kiedy

$$\lim_{t \rightarrow \infty} X(t) = 0 \quad \lim_{t \rightarrow \infty} Y(t) = 0 \quad (1.182)$$

Funkcje eksponencjalne dążą do zera jeżeli część rzeczywista wartości własnych macierzy Jacobiego λ_i są ujemne:

$$\text{Re}(\lambda_1) < 0, \quad \text{Re}(\lambda_2) < 0 \quad (1.183)$$

Wówczas stan stacjonarny jest asymptotycznie stabilny. Jeżeli

$$\text{Re}(\lambda_1) > 0, \quad \text{i/lub} \quad \text{Re}(\lambda_2) > 0 \quad (1.184)$$

to stan stacjonarny jest niestabilny. Jeżeli

$$\text{Re}(\lambda_1) = \text{Re}(\lambda_2) = 0 \quad (1.185)$$

to stan stacjonarny jest stabilny, ale nie jest asymptotycznie stabilny. Jeżeli wartości własne macierzy Jacobiego wynoszą zero, metoda linearyzacji nie rozstrzyga o stabilności.

Zamiast wyznaczyć wartości własne (λ_1, λ_2) tej macierzy, wystarczy sprawdzić, kiedy część rzeczywista wartości własnych jest ujemna (lub dodatnia). Ponieważ macierz Jacobiego jest macierzą 2×2 , więc otrzymujemy równanie kwadratowe dla λ . Aby wartości własne miały część rzeczywistą ujemną muszą zachodzić dwie relacje:

$$\lambda_1 + \lambda_2 = a_{11} + a_{22} < 0 \quad \text{oraz} \quad \lambda_1 \lambda_2 = a_{11}a_{22} - a_{12}a_{21} > 0, \quad (1.186)$$

to oznacza że

$$\text{Tr } J < 0, \quad \det J > 0 \quad (1.187)$$

gdzie Tr oznacza ślad macierzy, czyli sumę elementów diagonalnych macierzy oraz Det jest wyznacznikiem macierzy.

Przykład 1

$$\begin{aligned}\dot{x} &= 1 - xy = f(x, y), \\ \dot{y} &= x - y^2 = g(x, y).\end{aligned}\tag{1.188}$$

Łatwo znaleźć stany stacjonarne

$$\begin{aligned}1 - xy &= 0, \\ x - y^2 &= 0.\end{aligned}\tag{1.189}$$

Z drugiego równania $x = y^2$ wstawiamy do pierwszego równania: $1 - y^3 = 0$ czyli $y^3 = 1$. Stąd $y = 1$ oraz $x = 1$. Otrzymujemy stan stacjonarny

$$(x_s, y_s) = (1, 1)\tag{1.190}$$

Obliczmy elementy macierzy Jacobiego

$$J = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \end{bmatrix} = \begin{bmatrix} -y & -x \\ 1 & -2y \end{bmatrix} = \begin{bmatrix} -1 & -1 \\ 1 & -2 \end{bmatrix} \quad \text{w punkcie } (x = 1, y = 1)\tag{1.191}$$

Ślad macierz J , czyli suma elementów diagonalnych $\text{Tr}: \mathbf{J} = -3 < 0$ oraz wyznacznik $\text{Det}: \mathbf{J} = 3 > 0$. Spełnione są relacje dla stanu stacjonarnego który jest asymptotycznie stabilny. Wniosek: stan stacjonarny $(x_s, y_s) = (1, 1)$ jest asymptotycznie stabilny.

Można sprawdzić to, wyliczając explicite wartości własne macierzy Jacobiego:

$$\lambda_1 = \frac{1}{2}(-3 + i\sqrt{3}), \quad \lambda_2 = \frac{1}{2}(-3 - i\sqrt{3})\tag{1.192}$$

Ich części rzeczywiste są ujemne: $\Re(\lambda_1) = -3/2$, $\Re(\lambda_2) = -3/2$.

1.6.5 Przypadek C: Stabilność stanów stacjonarnych układów wielowymiarowych

Dla układu równań różniczkowych o wymiarze większym niż 2 stosujemy te same metody co dla układu 2 równań różniczkowych. Oczywiście istnieje więcej różnych przypadków i większe bogactwo własności stanów stacjonarnych. Nie będziemy przedstawiali tu przypadku o dowolnym wymiarze. Rozważymy przypadek 3 równań, aby pokazać podobieństwo do przypadku 2 równań. Analizujemy układ 3 równań różniczkowych:

$$\begin{aligned}\dot{x} &= F_1(x, y, z), & x(0) &= x_0, \\ \dot{y} &= F_2(x, y, z), & y(0) &= y_0, \\ \dot{z} &= F_3(x, y, z), & z(0) &= z_0.\end{aligned}\tag{1.193}$$

Stany stacjonarne są określone przez rozwiązania układu równań algebraicznych:

$$F_1(x, y, z) = 0, \quad F_2(x, y, z) = 0, \quad F_3(x, y, z) = 0\tag{1.194}$$

Z równań tych otrzymujemy stan stacjonarny $(x, y, z) = (x_s, y_s, z_s)$. Następnie obliczymy macierz Jacobiego

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial x} & \frac{\partial F_1}{\partial y} & \frac{\partial F_1}{\partial z} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y} & \frac{\partial F_2}{\partial z} \\ \frac{\partial F_3}{\partial x} & \frac{\partial F_3}{\partial y} & \frac{\partial F_3}{\partial z} \end{bmatrix} \quad (1.195)$$

Obliczymy macierz Jacobiego w punkcie stacjonarnym $J = J(x_s, y_s, z_s)$. W ten sposób otrzymujemy macierz liczbową. Kolejnym krokiem jest znalezienie wartości własnych $\lambda = \lambda_i (i = 1, 2, 3)$ tej macierzy, czyli rozwiązanie wielomianu 3-go stopnia dla λ :

$$\text{Det}(J - \lambda I) = \text{Det} \begin{bmatrix} \frac{\partial F_1}{\partial x} - \lambda & \frac{\partial F_1}{\partial y} & \frac{\partial F_1}{\partial z} \\ \frac{\partial F_2}{\partial x} & \frac{\partial F_2}{\partial y} - \lambda & \frac{\partial F_2}{\partial z} \\ \frac{\partial F_3}{\partial x} & \frac{\partial F_3}{\partial y} & \frac{\partial F_3}{\partial z} - \lambda \end{bmatrix} = 0 \quad (1.196)$$

Macierz I jest macierzą jednostkową, tzn. diagonalną 3×3 i wszystkie elementy diagonalne to 1.

Jeżeli części rzeczywiste wszystkich wartości własnych macierzy Jacobiego λ_i są ujemne:

$$\text{Re}(\lambda_1) < 0, \quad \text{Re}(\lambda_2) < 0, \quad \text{Re}(\lambda_3) < 0 \quad (1.197)$$

to stan stacjonarny jest asymptotycznie stabilny. Jeżeli chociaż jedna z wartości własnych λ_i ma część rzeczywistą dodatnią

$$\text{Re}(\lambda_1) > 0, \quad \text{i/lub} \quad \text{Re}(\lambda_2) > 0, \quad \text{i/lub} \quad \text{Re}(\lambda_3) > 0 \quad (1.198)$$

to stan stacjonarny jest niestabilny.

W przypadkach wielowymiarowych wygodnie jest stosować metody numeryczne. Obliczanie wartości własnych macierzy jest numerycznie zadaniem łatwym.

Przykład 2: Model Lorenza

$$\begin{aligned} \dot{x} &= \sigma(y - x) = F_1(x, y, z), & x(0) &= x_0, \\ \dot{y} &= x(\rho - z) - y = F_2(x, y, z), & y(0) &= y_0, \\ \dot{z} &= xy - \beta z = F_3(x, y, z), & z(0) &= z_0. \end{aligned} \quad (1.199)$$

gdzie wszystkie parametry są dodatnie: $\sigma, \rho, \beta > 0$.

KROK 1: Znajdujemy stany stacjonarne czyli rozwiązujemy układ równań algebraicznych

$$\begin{aligned} \sigma(y - x) &= 0, \\ x(\rho - z) - y &= 0, \\ xy - \beta z &= 0. \end{aligned} \quad (1.200)$$

Możemy posłużyć się programem w Sage, ale układ ten jest na tyle prosty, że możemy go rozwiązać sami. Z pierwszego równania wynika, że

$$y = x \quad (1.201)$$

Z trzeciego równania otrzymujemy

$$z = \frac{1}{\beta}x^2 \quad (1.202)$$

Wstawiamy x oraz z do drugiego równania otrzymując relację:

$$x\rho - x\frac{1}{\beta}x^2 - x = 0 \quad (1.203)$$

czyli

$$x[\rho - 1 - \frac{1}{\beta}x^2] = 0 \quad (1.204)$$

Ta relacja jest prosta i wynika z niej że

$$x = x_1 = 0 \quad \text{lub} \quad x = x_2 = \sqrt{\beta(\rho - 1)} \quad \text{lub} \quad x = x_3 = -\sqrt{\beta(\rho - 1)} \quad (1.205)$$

Otrzymujemy 3 stany stacjonarne:

$$\begin{aligned} P_1 &= (x_1, y_1, z_1) = (0, 0, 0), \\ P_2 &= (x_2, y_2, z_2) = (\sqrt{\beta(\rho - 1)}, \sqrt{\beta(\rho - 1)}, \rho - 1), \\ P_3 &= (x_3, y_3, z_3) = (-\sqrt{\beta(\rho - 1)}, -\sqrt{\beta(\rho - 1)}, \rho - 1). \end{aligned} \quad (1.206)$$

Dla każdego stanu stacjonarnego musimy zbadać jego stabilność analizując zagadnienie własne dla macierzy Jacobiego. No to do dzieła...

```
#kilka zmiennych
reset()
var('x y z sigma rho beta alpha')
#i kilka zalozen
assume(sigma>0)
assume(rho>0)
assume(beta>0)
#definiujemy rownania rozniczkowe
F1 = sigma*(y-x)
F2 = x*(rho-z) - y
F3 = x*y - beta*z
html.table([[ '$F_1$', '$F_2$', '$F_3$', ], [F1,F1,F3]])
print "stany stacjonarne"
rozv = solve([F1==0,F2==0,F3==0],[x,y,z])
P1 = rozv[2]
P2 = rozv[0]
P3 = rozv[1]
html.table([[ '$P_1$', '$P_2$', '$P_3$', ], [P1,P2,P3]])
print 'Macierz Jakobiego'
J = matrix([
[diff(F1,x),diff(F1,y),diff(F1,z)],
[diff(F2,x),diff(F2,y),diff(F2,z)],
[diff(F3,x),diff(F3,y),diff(F3,z)]
])
show(J)
#
@interact
def _(punkt=['P1','P2','P3']):
```

```

P = dict(zip(['P1', 'P2', 'P3'], [P1, P2, P3])) [punkt]
J = matrix([
    [
        diff(F1, x) (x=P[0].rhs(), y=P[1].rhs(), z=P[2].rhs()),
        diff(F1, y) (x=P[0].rhs(), y=P[1].rhs(), z=P[2].rhs()),
        diff(F1, z) (x=P[0].rhs(), y=P[1].rhs(), z=P[2].rhs())
    ],
    [
        diff(F2, x) (x=P[0].rhs(), y=P[1].rhs(), z=P[2].rhs()),
        diff(F2, y) (x=P[0].rhs(), y=P[1].rhs(), z=P[2].rhs()),
        diff(F2, z) (x=P[0].rhs(), y=P[1].rhs(), z=P[2].rhs())
    ],
    [
        diff(F3, x) (x=P[0].rhs(), y=P[1].rhs(), z=P[2].rhs()),
        diff(F3, y) (x=P[0].rhs(), y=P[1].rhs(), z=P[2].rhs()),
        diff(F3, z) (x=P[0].rhs(), y=P[1].rhs(), z=P[2].rhs())
    ]
])
print "macierz Jakobiego w punkcie %s" % P
show(J)
#zagadnienie własne macierzy
dJ = det(J - alpha*matrix(3,3,1)) == 0
rozwdJ1 = solve(dJ, alpha)
show(rozwdJ1)
b = 1
s = 2
r = 3
i = 0
for a in rozwdJ1:
    print "sprawdzamy"
    print "wartość własna alpha_%d"%(i+1)
    buf = real(a.rhs() (rho=r, beta=b, sigma=s))
    show(n(buf))
    if buf < 0:
        print "jest < 0"
    else:
        print "jest > 0"
        i += 1
if i == 0:
    print "\n"
    print "Wynika z tego, że dla"
    print "beta=%s, sigma=%s, rho=%s"%(b, s, r)
    print "%s jest punktem stabilnym"%P
else:
    print "\n"
    print "Wynika z tego, że dla"
    print "beta=%s, sigma=%s, rho=%s"%(b, s, r)
    print "%s nie jest punktem stabilnym"%P

```

1.7 Atraktory

Atraktor A to taki zbiór w przestrzeni fazowej układu dynamicznego, że wiele trajektorii startujących nawet bardzo daleko od tego zbioru dąży w miarę upływu czasu do tego zbioru A . Najprościej jest to zrozumieć na przykładzie oscylatora tłumionego. Realizacją takiego oscy-

latora może być wahadło matematyczne czyli kulka na nieważkim pręcie zawieszona w jakimś środowisku (byleby nie w próżni). Na kulkę działa siła przyciągania ziemskiego i siła tarcia. Kulkę wychylamy o dowolny kąt od pozycji pionowej i obserwujemy trajektorię kulki. Kulka wykonuje coraz to mniejsze wahania i po dostatecznie długim czasie zatrzymuje się w pozycji pionowej. Kulkę możemy wychylać o dowolny kąt, nadawać jej dowolną prędkość, a i tak po pewnym czasie kulka zatrzyma się w pozycji pionowej, która odpowiada zerowemu wychyleniu kulki. Ten zerowy kąt wychylenia jest atraktorem. W tym przypadku atraktorem jest punkt w przestrzeni fazowej. Ponieważ przestrzeń fazowa oscylatora harmonicznego tłumionego jest 2-wymiarowa położenie-prędkość, atraktorem jest punkt $A = (p_{ooenie} = 0, prdko = 0)$.

Podamy teraz bardziej formalną definicję.

Atraktor Atraktorem nazywamy ograniczony zbiór w przestrzeni fazowej, do którego dążą asymptotycznie w czasie (tzn. gdy $t \rightarrow \infty$) obszary warunków początkowych o niezerowej objętości w przestrzeni fazowej. Atraktor to inaczej zbiór przyciągający: przyciąga on trajektorie o różnych warunkach początkowych. Ale nie musi on przyciągać wszystkich trajektorii. Dla danego układu dynamicznego może istnieć wiele atraktorów, nawet nieskończenie wiele. Atraktory mogą mieć nieskomplikowaną strukturę: to może być punkt, kilka punktów, krzywa taka jak okrąg czy zdeformowana elipsa, część płaszczyzny, torus (podobny do dętki), część przestrzeni. Atraktory mogą też mieć skomplikowaną strukturę: może to być zbiór fraktalny, tzn. zbiór o niecałkowitym wymiarze, np. 0.63, 2.06. Taki atraktor nazywa się *dziwnym atraktorem*.

Z atraktorami związane są obszary przyciągania lub baseny przyciągania $B(A)$. Basenem przyciągania atraktora A nazywamy zbiór warunków początkowych x_0 , dla których trajektorie są przyciągane do A , czyli

$$B(A) = \{x_0 : \lim_{t \rightarrow \infty} x(t; x_0) \in A\} \quad (1.207)$$

gdzie $x(t; x_0)$ jest trajektorią startującą z warunku początkowego x_0 , np. rozwiązaniem układu równań różniczkowych z odpowiednimi warunkami początkowymi \vec{x}_0 .

1.7.1 Przykład 1: oscylator harmoniczny tłumiony

Jest tylko jeden atraktor: to punkt $(0, 0)$. Basenem przyciągania jest cała płaszczyzna fazowa.

```
var('x,y')
T = srange(0,50,0.01)
sol1=desolve_odeint(vector([y,-x-0.2*y]), [0,1], T, [x,y])##warunek początkowy x=2, y=4
sol2=desolve_odeint(vector([y,-x-0.2*y]), [0,0.85], T, [x,y])##warunek początkowy x=-1, y=0.85
sol3=desolve_odeint(vector([y,-x-0.2*y]), [0,0.7], T, [x,y])##warunek początkowy x=0, y=0.7
p1=plot(x^2, -2, 2,figsize=(6,3), )
g1=list_plot(sol1.tolist(), plotjoined=1, figsize=(6,3),axes_labels=[r'$x$',r'$y$'])
g1 +=list_plot(sol2.tolist(), plotjoined=1, figsize=(6,3),color="red", axes_labels=[r'$x$',r'$y$'])
g1 +=list_plot(sol3.tolist(), plotjoined=1, figsize=(6,3),color="green", axes_labels=[r'$x$',r'$y$'])
html.table([["potencjał kwadratowy","oscylator tłumiony"],[p1,g1]])
html("<p align='center'>wszystkie rozwiązania dążą do punktu (0,0) </p>")
```


1.7.2 Przykład 2: oscylator nieliniowy (bistabilny) tłumiony

Są dwa atraktory: punkt $(-x_s, 0)$ oraz symetryczny punkt $(x_s, 0)$, gdzie x_s jest stanem stacjonarnym. Płaszczyzna fazowa dzieli się na 2 baseny przyciągania, które są “pasiastymi” naprzemiennymi zbiorami. Przejrzysta wizualizacja jest opracowana na naszej stronie internetowej:

<http://visual.icse.us.edu.pl/wizualizacje/mechanika-teoretyczna/zobacz/BasenyPrzyciagania/>

```
var('x,y')
T1 = srange(0,30,0.01)
sol1=desolve_odeint(vector([y,2*x-1.2*x^3-0.2*y]), [0,1], T1, [x,y])##warunek początkowy
sol2=desolve_odeint(vector([y,2*x-1.2*x^3-0.2*y]), [0,2], T1, [x,y])##warunek początkowy
sol3=desolve_odeint(vector([y,2*x-1.2*x^3-0.2*y]), [0,3], T1, [x,y])##warunek początkowy
sol4=desolve_odeint(vector([y,2*x-1.2*x^3-0.2*y]), [0,4], T1, [x,y])##warunek początkowy
p11=plot(0.3*x^4 - x^2, -2, 2,figsize=(6,3), )
g11=list_plot(sol1.tolist(), plotjoined=1, figsize=(6,3),axes_labels=[r'$x$',r'$y$'])
g11 +=list_plot(sol2.tolist(), plotjoined=1, figsize=(6,3),color="red", axes_labels=[r'$x$',r'$y$'])
g11 +=list_plot(sol3.tolist(), plotjoined=1, figsize=(6,3),color="green", axes_labels=[r'$x$',r'$y$'])
g11 +=list_plot(sol4.tolist(), plotjoined=1, figsize=(6,3),color="black", axes_labels=[r'$x$',r'$y$'])
html.table([[["potencjał bistabilny","oscylator nieliniowy tłumiony"],[p11,g11]])
html("<p align='center'> rozwiązania dążą albo do punktu  $(-x_s,0)$  albo to punktu  $(x_s,0)$ ")
```

1.7.3 Przykład 3: Cykl graniczny

Atraktorem jest krzywa zamknięta (okrąg, elipsa, inne dowolne krzywe zamknięte). Poniżej przedstawiamy dwa przykłady zaczerpnięte z modeli biologicznych.

```
var('x,y')
T3 = srange(0,50,0.01)
de1=y+x*(0.2-(x^2+y^2))
de2=-x+y*(0.2-(x^2+y^2))
s1=desolve_odeint(vector([de1, de2]), [0.5,0.5], T3, [x,y])##warunek początkowy x=2, y=4
s2=desolve_odeint(vector([de1, de2]), [0.01, 0.01], T3, [x,y])##warunek początkowy x=2, y=4
h1=list_plot(s1.tolist(), plotjoined=1, figsize=(6,3),color="red",axes_labels=[r'$x$',r'$y$'])
h2=list_plot(s2.tolist(), plotjoined=1, figsize=(6,3),axes_labels=[r'$x$',r'$y$'])
show(h1+h2)

var('x,y')
a, b, d = 1.3, 0.33, 0.1
F(x,y)=x*(1-x) - a*x*y/(x+d)
G(x,y)= b*y*(1-y/x)
T = srange(0,80,0.01)
sl1=desolve_odeint(vector([F,G]), [0.2,0.3], T, [x,y])
sl2=desolve_odeint(vector([F,G]), [0.2,0.2], T, [x,y])
j1=list_plot(sl1.tolist(), plotjoined=1, color="red", figsize=(6, 3))
j2=list_plot(sl2.tolist(), plotjoined=1, figsize=(6, 3))
show(j1+j2)

var('x,y')
a, b, d = 1.3, 0.33, 0.1
F(x,y)=x*(1-x) - a*x*y/(x+d)
G(x,y)= b*y*(1-y/x)
T = srange(0,200,0.01)
sl1=desolve_odeint(vector([F,G]), [0.2,0.3], T, [x,y])
sl2=desolve_odeint(vector([F,G]), [0.2,0.2], T, [x,y])
```

```
j1=list_plot(sl1.tolist(), plotjoined=1, color="red", figsize=(6, 3))
j2=list_plot(sl2.tolist(), plotjoined=1,  figsize=(6, 3))
show(j1+j2)
```

1.7.4 Przykład 4: Atraktor Lorenza

Jest to przykład tak zwanego dziwnego atraktora. Najprostsza jego definicja to taka, że posiada on strukturę fraktala. O układzie Lorenza generującym ten fraktal można poczytać w poprzednim rozdziale tego skryptu, traktującym o stanach stacjonarnych.

```
var('x y z')
rho=28
sigma=10
beta=8/3
F1 = sigma*(y-x)
F2 = x*(rho-z) - y
F3 = x*y - beta*z
T = srange(0,100,0.01)
atraktor_lorenza = desolve_odeint(vector([F1,F2,F3]), [0,0.5,1], T, [x,y,z])
p2d = list_plot(zip(atraktor_lorenza[:,0],atraktor_lorenza[:,1]), plotjoined=1, figsize=
p3d = list_plot(atraktor_lorenza.tolist(), plotjoined=1, viewer='tachyon', figsize=4)
print "2D rysunek atraktora Lorenza"
p2d.show()
print "3D rysunek atraktora Lorenza"
p3d.show()
```

1.8 Dyskretne układy dynamiczne

We Wprowadzeniu pokazaliśmy w jaki sposób mogą pojawiać się zagadnienia modelowane równaniami rekurencyjnymi

$$x_{n+1} = f(x_n), \quad x_0 = c \quad (1.208)$$

Z jednej strony mogą to być przybliżenia równań różniczkowych, gdy pochodną aproksymujemy skończonym ilorazem różniczkowym. Z drugiej strony, mogą to być układy dynamiczne w których czas jest dyskretny i jest mierzony co pewien ustalony interwał czasowy (co sekundę, co godzinę, co dzień, co miesiąc, itd.). Tego typu modele spotykamy w biologii. Kolejnym przykładem jest algorytm Newtona pozwalający znajdować rekurencyjnie pierwiastki równania

$$F(x) = 0 \quad (1.209)$$

W metodzie Newtona kolejne przybliżenie pierwiastka dane jest przez równanie

$$x_{n+1} = x_n + \frac{F(x_n)}{F'(x_n)} \equiv f(x_n) \quad (1.210)$$

gdzie zakładamy, że pochodna $F'(x)$ istnieje (w przeciwnym przypadku metoda ta nie może być stosowana). Oznacza to, że nie możemy wpaść w ekstremum funkcji $F(x)$. W szczególności dla $F(x) = x^2 - 2$ ciąg (1.210) w zależności od warunku początkowego zmierza do liczby $x_+ = \sqrt{2}$ lub $x_- = -\sqrt{2}$ (co jest oczywiste skądinąd).

Wizualizacja iteracji Używając komputera, kolejne wyrazy x_n jest łatwo uzyskać. Dla przykładu, niech

$$x_{n+1} = x_n - 0.5x_n^2 + 1.7, \quad x_0 = 1 \quad (1.211)$$

Poniżej prezentujemy program w SAGE:

```
# warunek początkowy
x0 = 1
# tu można wstawić dowolną funkcję f(x)
f(x) = x - 0.5*x*x+1.7
for i in range(20):
    print x0
    x0 = f(x0)
```

Poniżej jest prezentacja graficzna dla ogólniejszego przypadku

$$x_{n+1} = x_n - bx_n^2 + 1.7, \quad x_0 = 1 \quad (1.212)$$

z dowolnym parametrem b . Zachęcamy do zabawy: zmieniajcie wartość parametru b oraz warunek początkowy. Czytelnik łatwo może podać swoje własne równanie, zmieniając odpowiednie wyrażenia w programie.

```
def ne(b,X):
    return X - b*X*X +1.7
#
def pophis(startp,b,length):
    his = [startp]
    for i in range(length):
        his.append( ne(b,his[i]) )
    return his
#
#warunek pocz. przed b,
#po b ilość iteracji,
#skala wykresu podana w ymin, ymax
@interact
def _(b=slider(0.05,3,0.05,default=0.5,label='Factor b')):
    show(list_plot(pophis(1,b,35),plotjoined=True,marker='o',ymin=0.5,ymax=3))
```

1.8.1 Metoda pajęczyny

Metoda pajęczyny pozwala na wykresie śledzić własności równań rekurencyjnych. Za pomocą tego wykresu można oberwować kolejne kroki iteracji.

Sposób rysowania:

Dla danego równania $x_{n+1} = f(x_n)$, rysujemy wykres funkcji $y = f(x)$ oraz linię prostą $y = x$. Prosta ta pozwala przenosić wartość x_{n+1} z osi OY na oś OX.

Na osi OX zaznaczamy warunek początkowy x_0 . Znajdujemy graficznie punkt $x_1 = f(x_0)$, który jest na osi OY.

Przy pomocy prostej $y = x$ przenosimy teraz punkt x_1 na oś OX.

Mając na osi OX punkt x_1 , traktujemy go jako następny warunek początkowy i znajdujemy graficznie na osi OY punkt $x_2 = f(x_1)$.

Powtarzamy kroki od 2,3,4 z ostatnim otrzymanym punktem jako początkowym x_0 .

```
var('r,x0,n')
@interact
def cobweb(a=slider(0.4,1.4,0.1,default=1),x0=slider(0,1,0.1,default=1)): ## zmiana para
    f(x)=x - a*x*x+1.7      ## postać funkcji f(x), która można zmieniać
    p = plot(f(x)==0,(x,-1,3),ymin=-2,ymax=3,xmin=-1,xmax=3,color='black')+plot(x,(x,-1,4),
    for n in range(20):
        th = 1
        if n>45:
            th = 1.5
            color='red'
        elif n < 5:
            color='blue'
            th=1.5
        else:
            color='grey'
            th =0.5
        l1 = line([(x0,x0),(x0,f(x0))],color=color,thickness=th)
        l2 = line([(x0,f(x0))),(f(x0),f(x0))],color=color,thickness=th)
        p = p+l1+l2
        x0 = f(x0)
    p.axes_labels(["$x_n$","$x_{n+1}$"])
    p.show()
```

1.8.2 Stany stacjonarne (punkty stałe) o okresie 1

Układ dynamiczny z czasem dyskretnym ma postać równania rekurencyjnego

$$x_{n+1} = f(x_n), \quad x_0 \text{ znany} \quad (1.213)$$

Wartość funkcji w następnym kroku x_{n+1} jest obliczana z wartości funkcji x_n w kroku poprzednim.

Z punktu widzenia modelowania, chcielibyśmy wiedzieć, czy układ w trakcie ewolucji dąży do jakiegoś stanu stacjonarnego i czy ten stan stacjonarny jest stabilny. Pojawia się też pytanie, jakiego typu stany stacjonarne mogą pojawiać się dla układów modelowanych równaniami rekurencyjnymi.

A. Stany stacjonarne (punkty stałe o okresie 1)

Przypomnijmy sobie, co oznacza istnienie stanu stacjonarnego dla układów modelowanych równaniem różniczkowym:

$$\frac{dx(t)}{dt} = F(x(t)) \quad (1.214)$$

Stan stacjonarny to taki stan, który nie zmienia się w trakcie ewolucji, nie zmienia się wraz ze zmianą czasu, czyli $x(t) = x_s$ jest wielkością stałą, niezmienną. Skoro tak, to

$$\frac{dx(t)}{dt} = \frac{dx_s}{dt} = 0 \quad (1.215)$$

Aby lewa strona równania różniczkowego była równa prawej stronie, musi zachodzić równość:

$$F(x_s) = 0 \quad (1.216)$$

To jest równanie (warunek), z którego wyznaczamy stan stacjonarny.

Podobnie rozumiemy w przypadku równań rekurencyjnych: Stan stacjonarny to taki stan, który nie zmienia się w trakcie ewolucji (dyskretnej), czyli z punktu x_s otrzymujemy znowu x_s . Dla równania dyskretnego ta niezmiennosc oznacza, że

$$\text{jeżeli } x_n = x_s \text{ to } x_{n+1} = x_s \quad (1.217)$$

i równanie rekurencyjne

$$x_{n+1} = f(x_n) \text{ ma postać } x_s = f(x_s) \quad (1.218)$$

Stąd otrzymujemy równanie dla stanu stacjonarnego

$$x_s = f(x_s) \text{ lub w uproszczonym zapisie } x = f(x) \quad (1.219)$$

Zapamiętajmy to równanie! Matematycy zamiast nazwy “stan stacjonarny” stosują nazwę “punkt stały odwzorowania” lub “punkt stały o okresie 1”.

Poniżej przedstawiamy to graficznie. Po wielu krokach iteracji obserwujemy powtarzanie się tej samej wartości: kolejna iteracji już nic nie zmienia, jest ta sama, stała.

```
def newpop(a,prevpop):
    return a*prevpop
#
def populationhistory(startpop,a,length):
    history = [startpop]
    for i in range(length):
        history.append( newpop(a,history[i]) )
    return history
#
@interact
def _( a=slider(0.5,1.1,0.05,default=0.5,label='Malthus Factor a') ):
    myplot=list_plot( populationhistory(1,a,30) ,plotjoined=True,marker='o',ymin=0,ymax=2) #
    myplot.show()
```

Zobaczmy jak to “działa” w przypadku tzw. równania logistycznego gdy $f(x) = ax(1 - x)$:

$$x = f(x) \text{ oznacza } x = ax = ax - ax^2 \quad (1.220)$$

Jest to równanie kwadratowe:

$$ax^2 - ax + x = 0 \text{ czyli } ax^2 + x = x[ax + 1] = 0 \quad (1.221)$$

Otrzymujemy dwa rozwiązania

$$x_1 = 0 \text{ oraz } x_2 = \frac{a-1}{a} = 1 - \frac{1}{a} \quad (1.222)$$

Są to dwa stany stacjonarne układu. Występuje tu podobieństwo do ciągłego modelu Verhulsta, które też posiada dwa stany stacjonarne $N_1 = 0$ (niestabilne) oraz $N_2 = K$ (stabilne).

Który ze stanów stacjonarnych x_1 i x_2 jest stabilny, a który niestabilny. Zbadamy teraz ten problem.

B. Stabilność stanów stacjonarnych

Stabilność stanu stacjonarnego x_s jest podobnie definiowana jak w przypadku równań różniczkowych zwyczajnych: dowolnie małe odchylenie od stanu stacjonarnego jest w trakcie ewolucji redukowane i odchylenie dąży do zera. Innymi słowy: jeżeli stan początkowy x_0 mało różni się od stanu stacjonarnego x_s , to ciąg liczb:

$$x_0, \quad x_1 = f(x_0), \quad x_2 = f(x_1), \quad x_3 = f(x_2), \quad x_4 = f(x_3), \dots \quad (1.223)$$

dąży do stanu stacjonarnego x_s .

Innymi słowy, jeżeli $|x_0 - x_s| < \delta$ dla odpowiednio małej liczby δ to $|x_n - x_s| \rightarrow 0$ gdy $n \rightarrow \infty$. Wyprowadzimy kryterium które pozwala badać stabilność x_s .

Wprowadzamy oznaczenie dla małego odchylenia od stanu stacjonarnego:

$$y_n = x_n - x_s \ll 1 \quad (1.224)$$

stąd

$$x_n = x_s + y_n. \quad (1.225)$$

Wówczas

$$y_{n+1} = x_{n+1} - x_s = f(x_n) - x_s = f(x_s + y_n) - x_s \quad (1.226)$$

Ponieważ y_n jest małą wielkością, to funkcję $f(x_s + y_n)$ rozwijamy w szereg Taylora:

$$y_{n+1} = f(x_s + y_n) - x_s \approx [f(x_s) + f'(x_s)y_n + \dots] - x_s = f(x_s) - x_s + f'(x_s)y_n + \dots = \lambda y_n + \dots \quad \text{gdzi} \quad (1.227)$$

Wykorzystaliśmy tu równość dla stanu stacjonarnego: $x_s = f(x_s)$. W ten sposób otrzymaliśmy równanie rekurencyjne dla odchylenia od stanu stacjonarnego:

$$y_{n+1} = \lambda y_n \quad (1.228)$$

Jeżeli $y_n \rightarrow 0$ to stan stacjonarny x_s jest stabilny. Musimy teraz zbadać, dla jakich wartości liczby $\lambda = f'(x_s)$ równanie dla y_n ma rozwiązania dążące do zera dla warunku początkowego $|y_0| \ll 1$.

Rozpatrzmy 4 przypadki:

1. Przypadek $\lambda > 1$

Aby łatwiej zrozumieć, przyjmijmy $\lambda = 2$. Wówczas równanie na odchylenie od stanu stacjonarnego ma postać

$$y_{n+1} = 2y_n \quad (1.229)$$

Otrzymamy ciąg liczbowy

$$y_0, \quad y_1 = 2y_0, \quad y_2 = 2y_1 = 2 \times 2y_0 = 2^2y_0, \quad y_3 = 2y_2 = 2^3y_0, \quad y_4 = 2y_3 = 2^4y_0, \dots \quad (1.230)$$

Widzimy, że kolejne liczby rosną, ponieważ są mnożone przez czynnik 2 i ciąg liczbowy jest rozbieżny. Otrzymujemy stąd wniosek, że dla $\lambda > 1$, stan stacjonarny nie jest stabilny: nieskończenie małe odchylenie od wartości stacjonarnej rośnie wraz z kolejnym krokiem iteracji.

2. Przypadek $\lambda < -1$

Aby łatwiej zrozumieć, przyjmijmy $\lambda = -2$. Wówczas równanie na odchylenie od stanu stacjonarnego ma postać

$$y_{n+1} = -2y_n \quad (1.231)$$

Otrzymamy ciąg liczbowy

$$y_0, \quad y_1 = -2y_0, \quad y_2 = -2y_1 = (-2) \times (-2)y_0 = 2^2y_0, \quad y_3 = (-2)y_2 = -2^3y_0, \quad y_4 = (-2)y_3 = 2^4y_0, \dots \quad (1.232)$$

Widzimy, że wartości bezwzględne kolejnych liczby rosną, ponieważ są mnożone przez czynnik -2 i ciąg liczbowy jest rozbieżny. Otrzymujemy stąd wniosek, że dla $\lambda < -1$, stan stacjonarny nie jest stabilny: nieskończenie małe odchylenie od wartości stacjonarnej rośnie wraz z kolejnym krokiem iteracji.

3. Przypadek $\lambda \in (-1, 1)$

Aby łatwiej zrozumieć, przyjmijmy $\lambda = (1/2)$. Wówczas równanie na odchylenie od stanu stacjonarnego ma postać

$$y_{n+1} = \frac{1}{2}y_n \quad (1.233)$$

Otrzymamy ciąg liczbowy

$$y_0, \quad y_1 = \frac{1}{2}y_0, \quad y_2 = \frac{1}{2}y_1 = \frac{1}{2} \times \frac{1}{2}y_0 = \frac{1}{2^2}y_0, \quad y_3 = \frac{1}{2}y_2 = \frac{1}{2^3}y_0, \quad y_4 = \frac{1}{2}y_3 = \frac{1}{2^4}y_0, \dots \quad (1.234)$$

Widzimy, że kolejne liczby maleją, ponieważ są mnożone przez czynnik $1/2$ i ciąg liczbowy dąży do zera. Otrzymujemy stąd wniosek, że dla $\lambda \in (-1, 1)$, stan stacjonarny jest stabilny: nieskończenie małe odchylenie od wartości stacjonarnej maleje do zera wraz z kolejnym krokiem iteracji.

Wniosek z tego jest następujący:

Uwaga: Stan stacjonarny x_s jest stabilny jeżeli $\lambda = f'(x_s) \in (-1, 1)$.

1.8.3 Stany stacjonarne o okresie 2

Jeżeli dla długich iteracji (formalnie dla $n \rightarrow \infty$) otrzymujemy ciągle tą samą wartość, to mówimy o stanie stacjonarnym o okresie 1. Pokazano to na powyższym rysunku. Jednakże mogą pojawić się inne stany stacjonarne. Dla przykładu, zobaczmy jak zachowuje się układ dla $n \rightarrow \infty$, który jest pokazany poniżej.

```
def ne(b,pre):
    return 1+b*pre*(1-(1/16)*pre)
#
def pophis(startp,b,length):
    his = [startp]
    for i in range(length):
        his.append( ne(b,his[i]) )
    return his
#
@interact
def _( b=slider(0.05,3.8,0.05,default=3.05,label='Factor b') ):
    aplot=list_plot( pophis(4,b,35) ,plotjoined=True,marker='o',ymin=0,ymax=18)##warunek po
    aplot.show()
```

Obserwujemy, że dla dużych wartości iteracji $n \gg 1$, dwie wartości iteracji powtarzają się i układ “skacze” pomiędzy dwoma stanami. Mówimy wówczas o stanie stacjonarnym o okresie 2. Możemy także powiedzieć, że jest to stan periodyczny. Jak wyznaczyć takie stany? Skorzystamy ze wzoru na kolejne iteracje:

$$x_{n+2} = f(x_{n+1}), \quad x_{n+1} = f(x_n) \quad (1.235)$$

Należy zauważyć, że wartość $x_{n+2} = x_n$ ponieważ co drugi krok jest ten sam stan. Dlatego też

$$x_{n+2} = f(x_{n+1}) = f[f(x_n)] = x_n \quad (1.236)$$

Jeżeli oznaczymy stan stacjonarny x_s o okresie 2 jako x^* to powyższe równanie przepiszemy jako

$$f[f(x^*)] = x^* \quad (1.237)$$

Ale to samo zachodzi dla stanu $x_{n+3} = x_{n+1}$. Dlatego równanie powyższe ma 2 rozwiązania $x^* = x_1^*$ oraz $x^* = x_2^*$.

W praktyce rozwiążemy równanie w postaci bardziej przyjaznej:

$$x = f[f(x)] \equiv g(x) \quad (1.238)$$

Pamiętajmy, że jest to złożenie 2 funkcji (co prawda takich samych funkcji, ale to jest drugorzędne).

To jest bardzo ważne równanie!

Napisanie tego równania w jawnej postaci nastręcza duże kłopoty przeciętnemu studentowi. Dlatego podamy 1 przykład. Niech układ dynamiczny będzie opisany równaniem

$$x_{n+1} = 1 - 2x_n^2, \quad f(x_n) = 1 - 2x_n^2 \quad \text{czyli} \quad f(x) = 1 - 2x^2 \quad (1.239)$$

Ile wynosi $f[f(x)]$? Obliczamy

$$f[f(x)] = 1 - 2[f(x)]^2 = 1 - 2[1 - 2x^2]^2 = 1 - 2[1 - 4x^2 + 4x^4] = -8x^4 + 8x^2 - 1 \quad (1.240)$$

Dlatego równanie które określa stan stacjonarny o okresie 2 ma postać

$$x = -8x^4 + 8x^2 - 1 = g(x) \quad (1.241)$$

Jest to wielomian 4-go stopnia.

Stabilność stanów stacjonarnych o okresie 2

Badanie stabilności stanów o okresie 2 jest analogiczne do badania stabilności stanów stacjonarnych o okresie 1. Jeżeli stan stacjonarny jest określony przez równanie

$$x^* = g(x^*) \quad (1.242)$$

to stan ten jest stabilny gdy

$$\lambda_2 = g'(x^*) \in (-1, 1) \quad (1.243)$$

Ponieważ funkcja $g(x)$ jest funkcją złożoną, więc należy stosować reguły różniczkowania funkcji złożonej:

$$\lambda_2 = g'(x^*) = [\{f[f(x)]\}']_{x=x^*} = \left[\frac{df(u)}{du} \right]_{u=f(x^*)} \left(\frac{df(x)}{dx} \right)_{x=x^*} \in (-1, 1) \quad (1.244)$$

Powyższy związek można przepisać w postaci:

$$\lambda_2 = f'(u^*)f'(x^*) \in (-1, 1) \quad (1.245)$$

gdzie $u^* = f(x^*)$

Warto coś tu zauważyć i uprościć. Ponieważ stan x^* jest o okresie 2 to jak nadmieniliśmy powyżej, faktycznie są 2 stany: $x^* = x_1^*$ (np. górny) oraz $x^* = x_2^*$ (np. dolny), co widać doskonale z powyższego rysunku. Dlatego też dolny stan przechodzi w górny i następnie górny stan przechodzi w dolny. Można to zapisać jako:

$$x_2^* = f(x_1^*), \quad x_1^* = f(x_2^*) \quad (1.246)$$

Stąd otrzymujemy relację:

$$\lambda_2(x_1^*) = f'(x_2^*)f'(x_1^*) \in (-1, 1) \quad (1.247)$$

Podobna relacja zachodzi dla drugiego stanu

$$\lambda_2(x_2^*) = f'(x_1^*)f'(x_2^*) \in (-1, 1) \quad (1.248)$$

Jest to ta sama relacja co dla x_1^* . Więc oznacza to, że wystarczy zbadać wielkość $f'(x_1)f'(x_2)$, gdzie x_1 oraz x_2 to dwa rozwiązania równania $x = f(f(x))$.

Podobnie można badać punkty stałe o dowolnym okresie n :

$$x^* = f\{f[\dots f(x^*)\dots]\} = G(x^*) \quad (1.249)$$

gdzie mamy n -krotne złożenie odwzorowania f .

Stan x^* jest stabilny gdy

$$\lambda_n = G'(x^*) \in (-1, 1) \quad (1.250)$$

Ponieważ funkcja $G(x)$ jest funkcją złożoną, więc należy stosować reguły różniczkowania funkcji złożonej, podobnie jak to jest pokazane powyżej w przypadku $n = 2$.

Chaos w układach dynamicznych

2.1 Determinizm a przewidywalność

Determinizm jest terminem wieloznacznym. W odniesieniu do nauk przyrodniczych możemy mówić o tym że :

Każde zjawisko jest wyznaczone przez swoje przyczyny i całokształt warunków. Każde zjawisko podlega prawidłowościom przyrody, które wyrażane są w postaci odpowiednich praw nauki. Jeżeli dysponujemy odpowiednią wiedzą (o przyczynach, prawidłowościach), to można (przynajmniej w zasadzie) przewidywać przyszły bieg zdarzeń. Poznawcze znaczenie zasady przyczynowości sprowadza się do możliwości przewidywania. Determinizm i indeterminizm występują już w starożytnej filozofii przyrody. Demokryt twierdził, że nie ma w świecie zdarzeń przypadkowych, lecz „wszystko dzieje się wskutek konieczności”. Determinizm łączono z poglądem, że prawa przyrody mają charakter jednoznaczny. Okazało się jednak że wiele zjawisk podlega jedynie prawom statystycznym. Pogląd, że pewne zjawiska przyrody nie podlegają prawom jednoznacznym, ale jedynie statystycznym to indeterminizm. Doniosła rola praw statystycznych w fizyce współczesnej wymaga rozszerzenia formuły determinizmu: każde zdarzenie podlega prawom przyrody jednoznacznym (determinizm jednoznaczny, mocny) bądź statystycznym (determinizm probabilistyczny, umiarkowany). Wówczas indeterminizm (umiarkowany) mieści się w ramach determinizmu w szerszym ujęciu. Wydawało się, że w mechanice, stworzonej przez Galileusza i Newtona, znając położenie i prędkość (lub położenie i pęd) ciała materialnego można przewidzieć cały przyszły ruch tego ciała (a więc podać położenie i pęd w każdej chwili późniejszej), a także ustalić, jaki był ruch tego ciała w przeszłości. Wszystko to jest słuszne przy założeniu, że znane są wszystkie siły działające na ciało w przeszłości i w przyszłości. Wynikało to z twierdzeń matematycznych o jednoznaczności rozwiązań równań różniczkowych, a takimi są równania Newtona.

Jednakże historia pokazała, że w fizyce klasycznej są znane teorie statystyczne, niedeterministyczne. Rozważmy dla przykładu gaz zamknięty w jakimś naczyniu. Wiemy, że w warunkach równowagi termodynamicznej dwie równe co do objętości części tego naczynia będą zawierać jednakową liczbę cząsteczek gazu. Nie wiemy jednak, które cząsteczki znajdą się w której z dwu połówek naczynia. Sytuacja pozornie przypomina prawo rozpadu: można podać taki czas, w którym rozpadnie się połowa atomów, i w ten sposób podzielić wszystkie atomy na dwie równe części - te, które się w tym czasie rozpadną, i te, które się nie rozpadną, podobnie

jak podzieliliśmy cząsteczki gazu według kryterium, w której połowie naczynia się znajdują. W klasycznej fizyce statystycznej znamy prawa rządzące zachowaniem pojedynczych cząstek (są nimi z założenia prawa mechaniki newtonowskiej), a nasza niewiedza co do tego zachowania jest spowodowana po pierwsze niemożliwością śledzenia ruchu wielu miliardów obiektów, a po drugie, brakiem potrzeby, aby to czynić: wystarczy nam znać właśnie tylko pewne wielkości średnie, które ujawniają się fenomenologicznie, na przykład jako temperatura gazu, czy też jego ciśnienie. Tak więc rzeczywisty kompletny opis stanu gazu musiałby zawierać informację dotyczącą N wektorów położenia i N wektorów pędu (N - liczba cząsteczek gazu), co jest liczbą ogromną, podczas gdy opis statystyczny ogranicza się do kilku potrzebnych liczb. Opis statystyczny odnosi się do ogromnej liczby cząstek i jest to opis oparty o teorię prawdopodobieństwa i teorię procesów stochastycznych. Z definicji jest to opis niedeterministyczny. Ale jak powiedzieliśmy, opis jednej cząstki jest w pełni deterministyczny. Twierdzenia o jednoznaczności rozwiązań równań różniczkowych dawały nadzieję na totalny determinizm i przewidywalność ruchu pojedynczych cząstek. Nadzieja ta z praktycznego punktu widzenia okazała się mrzonką. W latach 50-tych XX wieku pokazano, że z praktycznego punktu widzenia determinizm mechaniki Newtona jest złudny i ugruntowana wiara w przewidywalność zachowania się prostych układów mechanicznych załamała się. Pojawiły się liczne przykłady, a później teoria matematyczna, pokazujące niemożliwość przewidywania czasowej ewolucji prostych układów mechanicznych. Podkreślamy, że chodzi tu o praktyczne aspekty przewidywalności. Z matematycznego punktu widzenia, przewidywalność jest ciągle słuszna. Dobrym przykładem nieprzewidywalności w praktyce jest prognoza pogody, co udowadnia codzienne życie. Poniżej przedstawimy zagadnienia, które ukażą nam, co oznacza nieprzewidywalność w teorii deterministycznej. Pokażemy, dlaczego ewolucja określona przez determinizm równań Newtona jest nieprzewidywalna. Ta deterministyczna nieprzewidywalność ma swoją nazwę: deterministyczny chaos.

2.2 Model chaosu. Układ bistabilny (oscylator Duffinga)

Aby dobrze zrozumieć istotę chaotycznego zachowania się układów dynamicznych, posłużymy się prostym przykładem z mechaniki klasycznej Newtona. Rozpatrujemy jednowymiarowy ruch cząstki wzdłuż osi OX opisany równaniem Newtona:

$$m\ddot{x} = F(x, \dot{x}, t) = ax - bx^3 - \gamma\dot{x} + A \cos(\Omega t), \quad x(0) = x_0, \quad \dot{x}(0) = v(0) = v_0 \quad (2.1)$$

Model ten wydaje się być banalnie prosty.

jest to ruch cząstki w polu siły $F(x) = ax - bx^3$ ($a, b > 0$) jest to ruch tłumiony tarciem $F(v) = -\gamma v = -\gamma\dot{x}$ oraz na cząstkę działa siła periodyczna w czasie $F(t) = A \cos(\Omega t)$. Zauważmy, że średnia wartość siły $F(t)$ na jednym okresie wynosi zero, czyli średnio działa zerowa siła! Siła $F(x) = x - x^3$ to jest siła potencjalna. Dlatego warto wykreślić potencjał $V(x)$ odpowiadający tej sile:

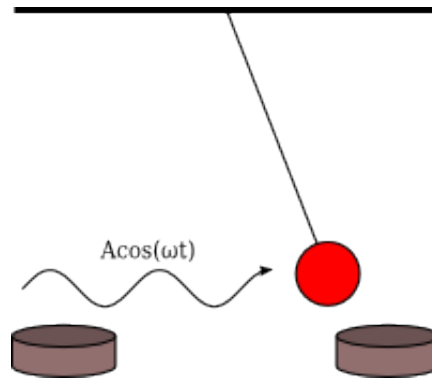
$$V(x) = - \int F(x) dx = \frac{1}{4}bx^4 - \frac{1}{2}ax^2 \quad (2.2)$$

Potencjał ten nazywa się potencjałem bistabilnym i jest jednym z najważniejszych modelowych potencjałów w fizyce, począwszy od teorii Newtona, poprzez teorię przejść fazowych,

teorię aktywacji reakcji chemicznych, a na teorii cząstek elementarnych (mechanizm Higgsa) skończywszy. Poniżej pokazujemy wykres tego potencjału (mówiąc precyzyjnie: energii potencjalnej).

Powyższe równanie Newtona ma kilka realizacji.

1 Pierwszy przykład realizacji: Metalowa kulka zawieszona na nieważkim pręcie w polu dwóch magnesów (które modelują bistabilny potencjał). Na kulkę działa w kierunku poziomym periodyczne pole elektryczne $A \cos(\omega t)$. Układ ten jest przedstawiony na rysunku.



Rysunek 2.1: Przykład realizacji układu bistabilnego.

2 Drugi przykład realizacji: Obwód elektroniczny, który jest szczegółowo opisany w pracy:

B. K. Jones and G. Trefan, Am. J. Phys. 69 (2001) str. 464. “The Duffing oscillator: A precise electronic analog chaos demonstrator for the undergraduate laboratory”

```
# Przeskalowany potencjał bistabilny: a=b=1
p = plot(0.25*x^4 - 0.5*x^2, (x,-1.6,1.6), figsize=(6,4), axes_labels=[r'$x$', r'$V(x)$'])
p += text("$-x_s$", (-1,0.025), fontsize=16, color='black')
p += text("$x_s$", (1,0.025), fontsize=16, color='black')
p.show()
```

2.2.1 Skalowanie

Układ opisany powyżej zawiera 6 parametrów. Część parametrów można wyeliminować poprzez przeskalowanie równania do postaci bezwymiarowej. Istnieje kilka możliwości. Zwykle zaczynamy od skalowania czasu i położenia. Nowy bezwymiarowy czas τ ma postać:

$$s = \frac{t}{\tau_0}, \quad \tau_0^2 = \frac{m}{a} \quad (2.3)$$

Nowe bezwymiarowe położenie definiujemy jako

$$X = \frac{x}{L}, \quad L^2 = \frac{a}{b} \quad (2.4)$$

Wówczas bezwymiarowa postać równania ruchu jest następująca:

$$\ddot{X} = X - X^3 - \gamma_0 \dot{X} + A_0 \cos(\omega_0 s), \quad X(0) = X_0, \quad \dot{X}(0) = \dot{X}_0 \quad (2.5)$$

Obecnie występują 3 przeskalowane parametry:

$$\gamma_0 = \frac{\tau_0^2}{mL} \gamma, \quad A_0 = \frac{\tau_0^2}{mL} A, \quad \omega_0 = \tau_0 \Omega \quad (2.6)$$

W dalszej części będziemy posługiwali się tylko i wyłącznie przeskalowanym równaniem. Dlatego wygodnie będzie używać “starych” oznaczeń: Bedziemy analizowali równanie w postaci

$$\ddot{x} = x - x^3 - \gamma \dot{x} + A \cos(\omega_0 t), \quad x(0) = x_0, \quad \dot{x}(0) = \dot{y}_0 = v_0 \quad (2.7)$$

gdzie przeskalowany potencjał

$$V(x) = - \int F(x) dx = \frac{1}{4} x^4 - \frac{1}{2} x^2 \quad (2.8)$$

Przeskalowane równanie jest w takiej postaci, że przyjmujemy wartości parametrów $m = 1, a = 1, b = 1$.

2.2.2 Krok 1. Układ zachowawczy

W pierwszym kroku rozpatrujemy najprostszypadek (pamiętajmy o przeskalowanej postaci, w której masa cząstki $m = 1$):

$$\ddot{x} = x - x^3 = -V'(x), \quad x(0) = x_0, \quad \dot{x}(0) = v(0) = v_0 \quad (2.9)$$

Jest on równoważny układowi 2 równań różniczkowych, autonomicznych, pierwszego rzędu:

$$\begin{aligned} \dot{x} &= v, & x(0) &= x_0, \\ \dot{v} &= x - x^3, & v(0) &= v_0. \end{aligned} \quad (2.10)$$

Oznacza to, że przestrzeń fazowa jest 2-wymiarowa.

Taki przypadek był już rozpatrywany: jest to układ zachowawczy o jednym stopniu swobody. Istnieje jedna stała ruchu (jedna całka ruchu), a mianowicie całkowita energia układu:

$$\frac{1}{2} \dot{x}^2(t) + V(x(t)) = \text{const.} = E = E_k + E_p = \frac{1}{2} \dot{x}^2(0) + V(x(0)) = \frac{1}{2} v_0^2 + V(x_0) \quad (2.11)$$

na którą składa się energia kinetyczna E_k oraz energia potencjalna E_p . Stała E jest określona przez warunki początkowe $x(0) = x_0$ oraz $v(0) = v_0$. Ponieważ jest zachowana całkowita energia układu, ruch jest periodyczny. Nie istnieją atraktory i nie istnieją asymptotycznie stabilne stany stacjonarne. Krzywe fazowe są zamknięte co oznacza że cząstka porusza się periodycznie w czasie. W zależności od warunków początkowych, amplituda drgań jest większa lub mniejsza, ponieważ warunki początkowe wyznaczają wartość stałej ruchu E . Jeżeli dwa warunki początkowe (x_{01}, v_{01}) oraz (x_{02}, v_{02}) nieznacznie się różnią, np. w sensie odległości:

$$|[x_{01}^2 + v_{01}^2] - [x_{02}^2 + v_{02}^2]| \ll 1 \quad (2.12)$$

to krzywe fazowe nieznacznie się różnią i ruch cząstki dla tych dwóch warunków początkowych nieznacznie się różni. Mówimy wówczas, że układ jest nieczuły na zmianę warunków początkowych. Jak widać z powyższego wzoru, dwa różne warunki początkowe oznaczają,

że układ ma dwie różne energie E . To z kolei oznacza, że częstotści ruchu periodycznego także będą różne. Różnica częstotści powoduje, że cząstki będą się powoli oddalać od siebie, ale tempo oddalania będzie liniowe w czasie. Gdyby tempo oddalania było znacznie szybsze, a mianowicie rośnie eksponencjalnie w czasie, zachowanie takie nazwalibyśmy chaotycznym. Do tego problemu powrócimy poniżej, ponieważ jest on kluczowym dla zrozumienia chaotycznego zachowania się układu.

Poniżej przedstawiamy potencjał i krzywe fazowe dla tego przypadku.

```
#parametry dla wizualizacji
var('x v')
x0 = 1.5
v0 = 0.2
E = 0.25*x0^4 - 0.5*x0^2 + v0^2
#
#prawo zachowania energii
V=0.25*x^4 - 0.5*x^2
PZE = v^2 + V == E
#
#wychylenia ekstremalne
print "ekstremalne wychylenia dla (x0,v0) = (%.2f,%.2f)"%(x0,v0)
rozv = solve(PZE(v=0), x); show(rozv)
xmin = min([i.rhs() for i in rozv if imag(i.rhs()) == 0])
xmax = max([i.rhs() for i in rozv if imag(i.rhs()) == 0])
#
#i jego rozwiązanie
print "ekstremalne prędkości dla (x0,v0) = (%.2f,%.2f)"%(x0,v0)
rozv = solve(PZE, v); show(rozv)
v1=rozv[0].rhs()
v2=rozv[1].rhs()
vmax = abs(v1(x=0))
#
#krzywe fazowe
start_point = (x0,V(x=x0))
p0 = point(start_point,size=30) + text(r"$ x_0$",start_point,vertical_alignment='bottom')
p1 = plot(V, (x,xmin,xmax))
p21 = plot(v1, (x,xmin,xmax), color='red')
p22 = plot(v2, (x,xmin,xmax), color='green')
(p0+p1).show(figsize=4)
(p21+p22).show(figsize=4)
```

2.2.3 Krok 2. Układ dysypatywny czyli wpływ tarcia.

W drugim kroku dodajemy tarcie i rozpatrujemy równanie ruchu w postaci:

$$\ddot{x} = x - x^3 - \gamma \dot{x}, \quad x(0) = x_0, \quad \dot{x}(0) = v(0) = v_0 \quad (2.13)$$

Jest on równoważny układowi 2 równań różniczkowych, autonomicznych, pierwszego rzędu:

$$\begin{aligned} \dot{x} &= v, & x(0) &= x_0, \\ \dot{v} &= x - x^3 - \gamma v, & v(0) &= v_0. \end{aligned} \quad (2.14)$$

Oznacza to, że przestrzeń fazowa jest 2-wymiarowa.

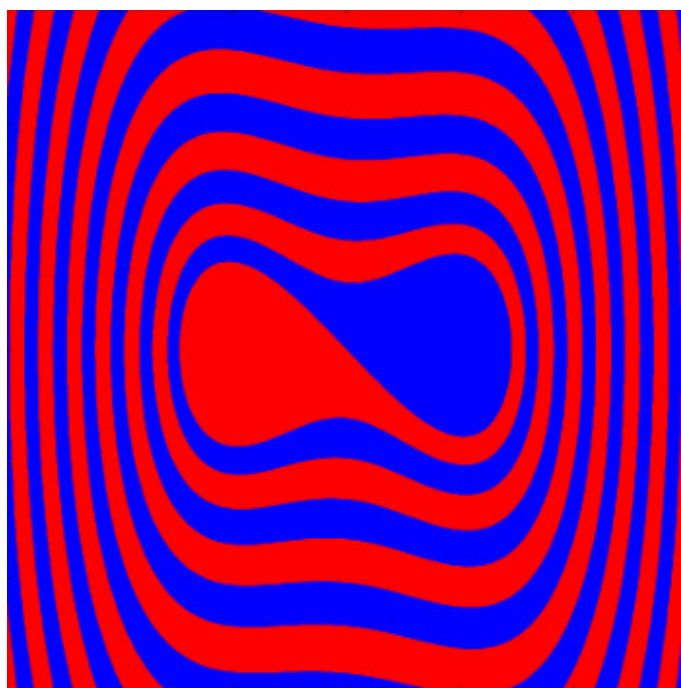
Taki przypadek był także rozpatrywany: jest to układ dysypatywny o jednym stopniu swobody. Nie istnieje już stała ruchu E . Całkowita energia układu maleje w czasie. W tym układzie istnieją 3 stany stacjonarne. Stany te określone są przez równanie:

$$x - x^3 = 0, \quad \text{stad} \quad x_{s0} = 0, \quad x_{s1} = 1, \quad x_{s2} = -1 \quad (2.15)$$

Stany stacjonarne $x_{s1} = 1$ oraz $x_{s2} = -1$ są stabilne. Stan $x_{s0} = 0$ jest niestabilny. Istnieją 2 atraktory $A_1 = x_{s1} = 1$ oraz $A_2 = x_{s2} = -1$ i 2 obszary przyciągania $B(A_1)$ oraz $B(A_2)$, których suma mnogościowa $B(A_1) \cup B(A_2) = R^2$ jest całą płaszczyzną. Krzywe fazowe zawsze dążą do jednego z atraktorów lub do niestabilnego stanu stacjonarnego. Jeżeli dwa warunki początkowe (x_{01}, v_{01}) oraz (x_{02}, v_{02}) nieznacznie się różnią np. w sensie odległości:

$$|[x_{01}^2 + v_{01}^2] - [x_{02}^2 + v_{02}^2]| \ll 1 \quad (2.16)$$

i są w tym samym obszarze przyciągania, to krzywe fazowe nieznacznie się różnią i ruch cząstki dla tych dwóch warunków początkowych nieznacznie się różni. Mówimy wówczas, że układ jest nieczuły na zmianę warunków początkowych. Natomiast jeżeli dwa warunki początkowe $(x_{01}, v_{01}) \in B(A_1)$ oraz $(x_{02}, v_{02}) \in B(A_2)$ nieznacznie się różnią, ale są w dwóch obszarach przyciągania $B(A_1)$ oraz $B(A_2)$, to trajektorie zaczną po pewnym czasie różnić się znacznie, będą przyciągane do dwóch różnych atraktorów i będą dążyć do dwóch różnych stanów stacjonarnych: $x_{s1} = 1$ oraz $x_{s2} = -1$. Tym niemniej, w takiej sytuacji mówimy, że układ jest nieczuły na zmianę warunków początkowych w sensie o którym mowa powyżej.



Rysunek 2.2: Diagram basenów przyciągania dla potencjału bistabilnego

Kolor niebieski to obszar warunków początkowych które są “przyciągane” do atraktora $(1, 0)$, do prawego minimum potencjału. Kolor czerwony to obszar warunków początkowych które są “przyciągane” do atraktora $(-1, 0)$, do lewego minimum potencjału. W zależności od wartości stałej tłumienia γ , diagram ten przybiera nieco inne kształty, ale struktura dwu-kolorowych

pasów pozostaje. Brzeg obszarów przyciągania jest gładką krzywą, której wymiar wynosi 1. Jeżeli warunki początkowe są położone dokładnie na tym brzegu, to cząstka porusza się do niestabilnego stanu stacjonarnego ($x = 0, v = 0$) (maksimum potencjału).

```
# wykresy dla przypadku z tłumieniem
var('x v')
x01, v01 = 1.50, 0
x02, v02 = 1.52, 0
#
# siła
F = x-x^3
V = -integrate(F,x)
#
# tarcie: parametr gamma
g = 0.1
#
# numeryczne rozwiązanie równań ruchu
T = srange(0,20*pi,0.01)
num1 = desolve_odeint(vector([v,F-g*v]), [x01,v01], T, [x,v])
num2 = desolve_odeint(vector([v,F-g*v]), [x02,v02], T, [x,v])
#
#krzywe fazowe
lt = plot(V, (x, -max([abs(x01),abs(x02)]),max([abs(x01),abs(x02)])), color='black', fi
lt += point((x01,V(x=x01)), color='green', size=50, axes_labels=['$x$','$V(x)$'])
lt += point((x02,V(x=x02)), color='red', size=50)
lb = list_plot(num1.tolist(), plotjoined=1, color='green', axes_labels=['$x(t)$','$v(t)$'])
lb += list_plot(num2.tolist(), plotjoined=1, color='red', figsize=4)
rt = list_plot(zip(T,num1[:,0].tolist()), plotjoined=1, color='green', axes_labels=['$t$','$x(t)$'])
rt += list_plot(zip(T,num2[:,0].tolist()), plotjoined=1, color='red', figsize=4)
rb = list_plot(zip(T,num1[:,1].tolist()), plotjoined=1, color='green', axes_labels=['$t$','$v(t)$'])
rb += list_plot(zip(T,num2[:,1].tolist()), plotjoined=1, color='red', figsize=4)
#
html("""
<p align='center'>rozwiązania z warunkami początkowymi
<span style="color:green">($x_{01},v_{01}$)=(%.2f,%.2f)</span>
<span style="color:red">($x_{02},v_{02}$)=(%.2f,%.2f)</span>
dążą do tego samego atraktora:
(x,v)=(-1,0)
</p>
""")(x01,v01,x02,v02)
html.table([[lt,rt],[lb,rb]])
```

Na powyższym zestawie rysunków, 2 warunki początkowe leżą w tym samym obszarze przyciągania atraktora $(-1, 0)$. Oznacza to, że 2 warunki początkowe są umiejscowione w czerwonym obszarze na diagramie basenów przyciągania pokazanym powyżej. Układ nie jest czuły na zmianę warunków początkowych, gdy leżą one w tym samym basenie przyciągania.

```
# wykresy dla przypadku z tłumieniem
var('x v')
x01, v01 = 1.58, 0
x02, v02 = 1.57, 0
#
# siła
F = x-x^3
V = -integrate(F,x)
#
# tarcie: parametr gamma
g = 0.1
```

```
#
# numeryczne rozwiązanie równań ruchu
T = srange(0,20*pi,0.01)
num1 = desolve_odeint(vector([v,F-g*v]), [x01,v01], T, [x,v])
num2 = desolve_odeint(vector([v,F-g*v]), [x02,v02], T, [x,v])
#
# wykresy funkcji
lt = plot(V, (x, -max([abs(x01),abs(x02)]),max([abs(x01),abs(x02)])),color='black', fi
lt += point((x01,V(x=x01)), color='blue', size=50, axes_labels=['$x$','$V(x)$'])
lt += point((x02,V(x=x02)), color='red', size=50)
lb = list_plot(num1.tolist(), plotjoined=1, color='blue', axes_labels=['$x(t)$','$v(t)$'])
lb += list_plot(num2.tolist(), plotjoined=1, color='red', figsize=4)
rt = list_plot(zip(T,num1[:,0]).tolist(), plotjoined=1, color='blue', axes_labels=['$t$','$x(t)$'])
rt += list_plot(zip(T,num2[:,0]).tolist(), plotjoined=1, color='red', figsize=4)
rb = list_plot(zip(T,num1[:,1]).tolist(), plotjoined=1, color='blue', axes_labels=['$t$','$v(t)$'])
rb += list_plot(zip(T,num2[:,1]).tolist(), plotjoined=1, color='red', figsize=4)
#
html("""
<p align='center'>rozwiązania z warunkami początkowymi
<span style="color:blue">($x_{01},v_{01}$)=(%.2f,%.2f)</span>
<span style="color:red">($x_{02},v_{02}$)=(%.2f,%.2f)</span>
dążą do różnych atraktorów:
<span style="color:blue">(x,v)=(1,0)</span>
<span style="color:red">(x,v)=(-1,0)</span>
</p>
""%(x01,v01,x02,v02))
html.table([[lt,rt],[lb,rb]])
```

Na powyższym zestawie rysunków, 2 warunki początkowe leżą w dwóch różnych obszarach przyciągania. Oznacza to, że 1 warunek początkowy leży w niebieskim obszarze na diagramie basenów przyciągania, natomiast 2 warunek początkowy leży w czerwonym obszarze na diagramie basenów przyciągania. Te dwa warunki początkowe leżą blisko brzegu 2 basenów przyciągania. Dlatego układ jest czuły na zmianę warunków początkowych, pod warunkiem że leżą one w dwóch różnych basenach przyciągania. Ale to nie jest jeszcze kryterium własności chaotycznych układu.

2.2.4 Krok 3. Układ z tarciem i periodyczną siłą.

W trzecim kroku dodajemy siłę periodyczną w czasie i rozpatrujemy równanie ruchu w wyjściowej pełnej postaci:

$$\ddot{x} = x - x^3 - \gamma \dot{x} + A \cos(\omega_0 t), \quad x(0) = x_0, \quad \dot{x}(0) = v(0) = v_0 \quad (2.17)$$

Jest on równoważny układowi 3 równań różniczkowych, autonomicznych, pierwszego rzędu:

$$\begin{aligned} \dot{x} &= v, & x(0) &= x_0, \\ \dot{v} &= x - x^3 - \gamma v + A \cos z, & v(0) &= v_0, \\ \dot{z} &= \omega_0, & z(0) &= 0. \end{aligned} \quad (2.18)$$

Oznacza to, że przestrzeń fazowa jest 3-wymiarowa.

Matematycy wolą przepisać powyższy układ równań dla “tradycyjnych” 3 zmiennych (x, y, z)

w postaci:

$$\begin{aligned}\dot{x} &= y, & x(0) &= x_0, \\ \dot{y} &= x - x^3 - \gamma y + A \cos z, & y(0) &= y_0, \\ \dot{z} &= \omega_0, & z(0) &= 0.\end{aligned}\tag{2.19}$$

czyli prędkość cząstki v jest teraz oznaczona jako $v = y$.

Okazuje się, że pełny układ wykazuje radykalnie inne własności od poprzednich 2 przypadków. Z punktu widzenia fizyki mamy taki oto proces: Cząstka porusza się w bistabilnym potencjale. Ponieważ potencjał dąży do nieskończoności gdy położenie dąży do nieskończoności, ruch cząstki jest ograniczony; cząstka jest uwięziona w potencjale i nie może uciec do nieskończoności. Siła tarcia pcha cząstkę do jednego ze (“starych”) stanów stacjonarnych x_{s1} lub x_{s2} . Z kolei zewnętrzna siła periodyczna w czasie pompuje energię do układu i przeciwdziała sile tarcia. Cząstka już nie dąży do stanu stacjonarnego, nie zatrzyma się dla długich czasów ale będzie ciągle poruszać się i nigdy już nie spocznie. Istotne stają się efekty inercyjne związane z masą cząstki, które są odzwierciedlone w wyrazie \dot{y} , czyli przyspieszeniu cząstki. Istotne jest to, że nie jest to ruch przetłumiony. W konsekwencji układ nie posiada stanu stacjonarnego w postaci punktu w przestrzeni fazowej jak to było w przypadku 2. Wszystkie te powyższe czynniki stają się istotne dla zrozumienia skomplikowanych i złożonych własności ewolucji cząstki.

```
# przykładowa trajektoria (górny wykres)
# wraz z krzywą fazową (dolny wykres)
var('x y z')
T = srange(0, 150*pi, 0.01)
sol=desolve_odeint( vector([y, x-x^3-0.26*y+0.3*cos(z), 1]), [0.1, 0.1, 0], T, [x, y, z])
t = line(zip(T, sol[:, 0]), figsize=(12, 4), axes_labels=["t$", "x(t)$"], frame=1, axes=0)
b = line(zip(sol[:, 0], sol[:, 1]), figsize=(12, 4), axes_labels=["x(t)$", "v(t)$"], frame=0)
html.table([[t], [b]])
```

2.2.5 Ruch periodyczny o okresie 1

W modelu występują 3 bezwymiarowe parametry: współczynnik tarcia γ , amplituda zewnętrznej siły A oraz częstość drgań ω_0 siły periodycznej w czasie. Poniżej pokażemy kilka charakterystycznych trajektorii układu. Zaczniemy od prostej periodycznej ewolucji, ruchu okresowego o tzw. okresie 1.

Założmy następujące wartości parametrów:

$$\gamma = 0.15, \quad A = 0.3, \quad \omega_0 = 1\tag{2.20}$$

W tym przypadku obserwujemy regularny ruch. Jeżeli nieco zaburzymy warunki początkowe, to nowy ruch jest także regularny (trzeba być ostrożnym, gdy mówimy “nieco zaburzymy”).

```
# wykresy dla przypadku z tłumieniem
var('x y z')
x0, y0, z0 = 0.1, 0.1, 0
kolor = 'green'
#
# siła
F = x-x^3
```

```

V = -integrate(F,x)
#
# tarcie: parametr gamma
g = 0.1
A = 0.3
w = 1
#
# układ różniczkowych równań ruchu
dx = y
dy = F - g*y + A*cos(z)
dz = w
#
# numeryczne rozwiązanie równań ruchu
T = srange(0,30*pi,0.01)
num = desolve_odeint(vector([dx,dy,dz]), [x0,y0,z0], T, [x,y,z])
#
# wykresy funkcji
xmin = 1.5
lt = plot(V, (x,-xmin,xmin), figsize=4)
lt += point((x0,V(x=x0)), color=kolor, size=50, axes_labels=['$x$', '$V(x)$'])
lb = list_plot(zip(num[:,0],num[:,1]), plotjoined=1, color=kolor, axes_labels=['$x(t)$', '$y(t)$'])
rt = list_plot(zip(T,num[:,0].tolist()), plotjoined=1, color=kolor, axes_labels=['$t$', '$x(t)$'])
rb = list_plot(zip(T,num[:,1].tolist()), plotjoined=1, color=kolor, axes_labels=['$t$', '$y(t)$'])
#
html("""Układ równań różniczkowych
$\dot{x} = %s$
$\dot{y} = %s$
$\dot{z} = %s$
z warunkami początkowymi
$(x_0,y_0,z_0) = (%.2f,%.2f,%.2f)$
""%(dx,dy,dz,x0,y0,z0))
html.table([[lt,rt],[lb,rb]])

```

Przyjrzyjmy się teraz dwóm trajektoriom startującym z bliskich warunków początkowych. Rozpatrzmy ich początkową i asymptotyczną (dla długich czasów) ewolucję.

```

# wykresy dla przypadku z tłumieniem
var('x y z')
x01, y01, z01 = 0.1,0.1,0
x02, y02, z02 = 0.11,0.1,0
#
# siła
F = x-x^3
V = -integrate(F,x)
#
# tarcie: parametr gamma
g = 0.1
A = 0.3
w = 1
#
# układ różniczkowych równań ruchu
dx = y
dy = F - g*y + A*cos(z)
dz = w
#
# numeryczne rozwiązanie równań ruchu
T = srange(0,200*pi,0.01)
num1 = desolve_odeint(vector([dx,dy,dz]), [x01,y01,z01], T, [x,y,z])

```

```

num2 = desolve_odeint(vector([dx,dy,dz]), [x02,y02,z02], T, [x,y,z])
#
lnum = int(len(num1[:,0])/10)
trans1 = num1[:lnum]
asymp1 = num1[-lnum:]
trans2 = num2[:lnum]
asymp2 = num2[-lnum:]
#
# wykresy funkcji
lt = list_plot(zip(trans1[:,0],trans1[:,1]), plotjoined=1, color='green', axes_labels=['x','y'])
lt += list_plot(zip(trans2[:,0],trans2[:,1]), plotjoined=1, color='red')
rt = list_plot(zip(T[:lnum],trans1[:,0].tolist()), plotjoined=1, color='green', axes_labels=['t','x'])
rt += list_plot(zip(T[:lnum],trans2[:,0].tolist()), plotjoined=1, color='red')
lb = list_plot(zip(asymp1[:,0],asymp1[:,1]), plotjoined=1, color='green', axes_labels=['x','y'])
lb += list_plot(zip(asymp2[:,0],asymp2[:,1]), plotjoined=0, color='red')
rb = list_plot(zip(T[-lnum:],asymp1[:,0].tolist()), plotjoined=1, color='green', axes_labels=['t','x'])
rb += list_plot(zip(T[-lnum:],asymp2[:,0].tolist()), plotjoined=1, color='red')
#
html("""Układ równań różniczkowych
 $\dot{x} = x^2$ 
 $\dot{y} = y^2$ 
 $\dot{z} = z^2$ 
z różnymi warunkami początkowymi
<span style="color:green;"> $x_{01}, y_{01}, z_{01} = (0.2, 0.2, 0.2)$ </span>
<span style="color:red;"> $x_{02}, y_{02}, z_{02} = (0.2, 0.2, 0.2)$ </span>
""")
html.table([lt,rt],[lb,rb])

```

Na dwóch górnych diagramach przedstawioną reżim krótkich czasów. Ponieważ 2 warunki początkowe nieco się różnią, więc początkowa ewolucja nieco się różni. Kolor czerwony i zielony jest rozróżnialny na prawym górnym rysunku pokazującym ewolucję $x(t)$ dla krótkich czasów. Jeżeli przyjrzymy się reżimowi długich czasów (dwa dolne diagramy) to zauważymy duże podobieństwo w ewolucji: krzywe fazowe są zamknięte więc jest to prosty ruch periodyczny, przypominający nieco zdeformowaną funkcję typu $\sin(\alpha t)$ czy też $\cos(\alpha t)$. Jest to funkcja okresowa z charakterystycznym jednym jedynym okresem T . Dlatego mówimy, że jest ruch periodyczny o okresie 1. Dwie krzywe $x(t)$ na dolnym prawym rysunku nie są rozróżnialne.

Można zrobić doświadczenie numeryczne i wybierać różne warunki początkowe. Zobaczymy, że trajektorie dążą do tego samego okresowego rozwiązania, są przyciągane do tego okresowego rozwiązania. Innymi słowy, ta krzywa fazowa o okresie 1 jest ATRAKTOREM. Atraktor ten nazywa się periodycznym atraktorem o okresie 1 lub 1-okresowym atraktorem. Można by postawić pytanie: jak wygląda basen przyciągania dla tego atraktora. Aby dać odpowiedź na to pytanie należy zbadać numerycznie np. kwadrat warunków początkowych (x_0, y_0) i wybrać te warunki początkowe które dążą do powyższej krzywej fazowej o okresie 1. Okazuje się, że basen przyciągania jest “porządnym” zbiorem, którego brzeg jest gładką krzywą, podobnie jak w przypadku zilustrowanym powyżej dla układu tylko z tarcie, bez siły okresowej.

2.2.6 Ruch periodyczny o okresie 3

Założmy następujące wartości parametrów:

$$\gamma = 0.22, \quad A = 0.3, \quad \omega_0 = 1 \quad (2.21)$$

W tym przypadku obserwujemy także periodyczny ruch, ale nieco bardziej skomplikowany. Nie jest to prosty periodyczny ruch, ale tzw. ruch o okresie 3, tzn. teraz okres jest 3 razy dłuższy niż w poprzednim przypadku.

```
# wykresy dla przypadku z tłumieniem
var('x y z')
x0, y0, z0 = 0.1, 0.1, 0
kolor = 'red'
#
# siła
F = x-x^3
V = -integrate(F,x)
#
# tarcie: parametr gamma
g = 0.22
A = 0.3
w = 1
#
# układ różniczkowych równań ruchu
dx = y
dy = F - g*y + A*cos(z)
dz = w
#
# numeryczne rozwiązanie równań ruchu
T = srange(0, 20*pi, 0.01)
num = desolve_odeint(vector([dx, dy, dz]), [x0, y0, z0], T, [x, y, z])
#
# wykresy funkcji
xmin = 1.5
lt = plot(V, (x, -xmin, xmin), figsize=4)
lt += point((x0, V(x=x0)), color=kolor, size=50, axes_labels=['$x$', '$V(x)$'])
lb = list_plot(zip(num[:, 0], num[:, 1]), plotjoined=1, color=kolor, axes_labels=['$x(t)$', '$y(t)$'])
rt = list_plot(zip(T, num[:, 0].tolist()), plotjoined=1, color=kolor, axes_labels=['$t$', '$x(t)$'])
rb = list_plot(zip(T, num[:, 1].tolist()), plotjoined=1, color=kolor, axes_labels=['$t$', '$y(t)$'])
#
html("""Układ równań różniczkowych
$\dot{x} = %s$
$\dot{y} = %s$
$\dot{z} = %s$
z warunkami początkowymi
$(x_0, y_0, z_0) = (%.2f, %.2f, %.2f)$
""%(dx, dy, dz, x0, y0, z0))
html.table([[lt, rt], [lb, rb]])
```

I znów zobaczymy, jak początkowa ewolucja różni się od tej po długim czasie.

```
# wykresy dla przypadku z tłumieniem
var('x y z')
x01, y01, z01 = 0.10, 0.1, 0
x02, y02, z02 = 0.11, 0.1, 0
#
# siła
F = x-x^3
V = -integrate(F,x)
#
# tarcie: parametr gamma
g = 0.22
A = 0.3
```

```

w = 1
#
# układ różniczkowych równań ruchu
dx = y
dy = F - g*y + A*cos(z)
dz = w
#
# numeryczne rozwiązanie równań ruchu
T = srange(0,200*pi,0.01)
num1 = desolve_odeint(vector([dx,dy,dz]), [x01,y01,z01], T, [x,y,z])
num2 = desolve_odeint(vector([dx,dy,dz]), [x02,y02,z02], T, [x,y,z])
#
lnum = int(len(num1[:,0])/10)
trans1 = num1[:lnum]
asympt1 = num1[-lnum:]
trans2 = num2[:lnum]
asympt2 = num2[-lnum:]
#
# wykresy funkcji
lt = list_plot(zip(trans1[:,0],trans1[:,1]), plotjoined=1, color='green', axes_labels=['x','y'])
lt += list_plot(zip(trans2[:,0],trans2[:,1]), plotjoined=1, color='red')
rt = list_plot(zip(T[:lnum],trans1[:,0].tolist()), plotjoined=1, color='green', axes_labels=['T','x'])
rt += list_plot(zip(T[:lnum],trans2[:,0].tolist()), plotjoined=1, color='red')
lb = list_plot(zip(asympt1[:,0],asympt1[:,1]), plotjoined=1, color='green', axes_labels=['x','y'])
lb += list_plot(zip(asympt2[:,0],asympt2[:,1]), plotjoined=1, color='red')
rb = list_plot(zip(T[-lnum:],asympt1[:,0].tolist()), plotjoined=1, color='green', axes_labels=['T','x'])
rb += list_plot(zip(T[-lnum:],asympt2[:,0].tolist()), plotjoined=1, color='red')
#
html("""Układ równań różniczkowych
 $\dot{x} = y$ 
 $\dot{y} = F - g y + A \cos(z)$ 
 $\dot{z} = w$ 
z różnymi warunkami początkowymi
<span style="color:green;"> $(x_{01}, y_{01}, z_{01}) = (0.25, 0.3, 1)$ </span>
<span style="color:red;"> $(x_{02}, y_{02}, z_{02}) = (0.25, 0.3, 1)$ </span>
""")
html.table([[lt,rt],[lb,rb]])

```

Dla długich czasów, krzywe fazowe są zamknięte, ale nie są to krzywe typu zdeformowana elipsa. To są krzywe z 2 pętelkami. Tym niemniej, ruch jest periodyczny.

Podobnie jak poprzednim przypadkiem, można zrobić doświadczenie numeryczne i wybierać różne warunki początkowe. Zobaczymy, że wiele trajektorii dąży do tej samej okresowej orbity, są one przyciągane do tej zamkniętej krzywej fazowej. Innymi słowy, ta krzywa fazowa o okresie 3 jest ATRAKTOREM. Atraktor ten nazywa się periodycznym atraktorem o okresie 3 lub 3-okresowym atraktorem. Basen przyciągania dla tego atraktora na płaszczyźnie warunków początkowych (x_0, y_0) jest "porządnym" zbiorem o wymiarze 2 (czyli kawałek płaszczyzny), którego brzeg jest gładką krzywą.

2.2.7 Ruch chaotyczny

Założmy następujące wartości parametrów:

$$\gamma = 0.25, \quad A = 0.3, \quad \omega_0 = 1 \quad (2.22)$$

W tym przypadku obserwujemy ruch, który wydaje się być wyjątkowo nieregularny, chaotyczny.

```
# wykresy dla przypadku chaotycznego
var('x y z')
x0, y0, z0 = 0.1, 0.1, 0
kolor = 'firebrick'
#
# siła
F = x-x^3
V = -integrate(F,x)
#
# tarcie: parametr gamma
g = 0.25
A = 0.3
w = 1
#
# układ różniczkowych równań ruchu
dx = y
dy = F - g*y + A*cos(z)
dz = w
#
# numeryczne rozwiązanie równań ruchu
T = srange(0, 50*pi, 0.01)
num = desolve_odeint(vector([dx, dy, dz]), [x0, y0, z0], T, [x, y, z])
#
# wykresy funkcji
xmin = 1.5
lt = plot(V, (x, -xmin, xmin), figsize=4)
lt += point((x0, V(x=x0)), color=kolor, size=50, axes_labels=['$x$', '$V(x)$'])
lb = list_plot(zip(num[:, 0], num[:, 1]), plotjoined=1, color=kolor, axes_labels=['$x(t)$', '$y(t)$'])
rt = list_plot(zip(T, num[:, 0].tolist()), plotjoined=1, color=kolor, axes_labels=['$t$', '$x(t)$'])
rb = list_plot(zip(T, num[:, 1].tolist()), plotjoined=1, color=kolor, axes_labels=['$t$', '$y(t)$'])
#
html("""Układ równań różniczkowych
 $\dot{x} =$  %s$
 $\dot{y} =$  %s$
 $\dot{z} =$  %s$
z warunkami początkowymi
 $(x_0, y_0, z_0) = (%.2f, %.2f, %.2f)$ 
"""%(dx, dy, dz, x0, y0, z0))
html.table([[lt, rt], [lb, rb]])
```

Zobaczmy, jak tym razem ewoluują rozwiązania o 2 bliskich warunkach początkowych.

```
var('x y z')
x01, y01, z01 = 0.1, 0.1, 0
x02, y02, z02 = 0.11, 0.1, 0
#
# siła
F = x-x^3
V = -integrate(F,x)
#
# tarcie: parametr gamma
g = 0.25
A = 0.3
w = 1
#
# układ różniczkowych równań ruchu
```

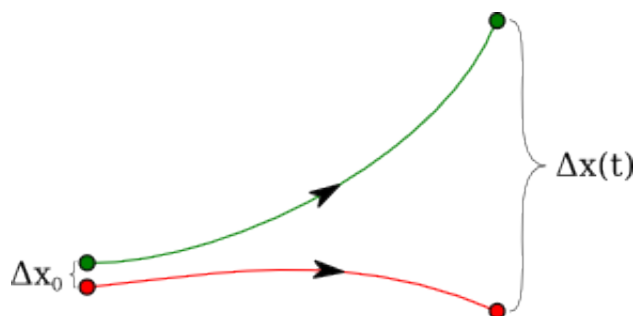


```

dx = y
dy = F - g*y + A*cos(z)
dz = w
#
# numeryczne rozwiązanie równań ruchu
T = srange(0,200*pi,0.01)
num1 = desolve_odeint(vector([dx,dy,dz]), [x01,y01,z01], T, [x,y,z])
num2 = desolve_odeint(vector([dx,dy,dz]), [x02,y02,z02], T, [x,y,z])
#
lnum = int(len(num1[:,0])/10)
trans1 = num1[:lnum]
asympt1 = num1[-lnum:]
trans2 = num2[:lnum]
asympt2 = num2[-lnum:]
#
# wykresy funkcji
lt = list_plot(zip(trans1[:,0],trans1[:,1]), plotjoined=1, color='green', axes_labels=['x', 'y'])
lt += list_plot(zip(trans2[:,0],trans2[:,1]), plotjoined=1, color='red')
rt = list_plot(zip(T[:lnum],trans1[:,0].tolist()), plotjoined=1, color='green', axes_labels=['t', 'x'])
rt += list_plot(zip(T[:lnum],trans2[:,0].tolist()), plotjoined=1, color='red')
lb = list_plot(zip(asympt1[:,0],asympt1[:,1]), plotjoined=1, color='green', axes_labels=['x', 'y'])
lb += list_plot(zip(asympt2[:,0],asympt2[:,1]), plotjoined=1, color='red')
rb = list_plot(zip(T[-lnum:],asympt1[:,0].tolist()), plotjoined=1, color='green', axes_labels=['t', 'x'])
rb += list_plot(zip(T[-lnum:],asympt2[:,0].tolist()), plotjoined=1, color='red')
#
html("""Układ równań różniczkowych
 $\dot{x} = y$ 
 $\dot{y} = F - g*y + A*\cos(z)$ 
 $\dot{z} = w$ 
z różnymi warunkami początkowymi
> $(x_{01}, y_{01}, z_{01}) = (x_0, y_0, z_0)$ 
> $(x_{02}, y_{02}, z_{02}) = (x_0, y_0, z_0)$ 
""")
html.table([[lt,rt],[lb,rb]])

```

Początkowa ewolucja dwóch rozwiązań jest nierozróżnialna (ponieważ 2 warunki początkowe są bardzo blisko siebie). Po pewnym charakterystycznym czasie, zwanym czasem Lapunowa, trajektorie zaczynają różnić się coraz bardziej, zaczynają rozbiegać się: patrz trajektoria czerwona i zielona na dolnym prawym rysunku.



Rysunek 2.3: Schematyczne trajektorie w reżimie chaotycznym.

W reżimie chaotycznym, te dwie trajektorie oddalają się od siebie w eksponencjalnie szybkim tempie określonym przez zależność:

$$|x_1(t) - x_2(t)| = |x_1(0) - x_2(0)|e^{\lambda t}, \quad \lambda > 0 \quad (2.23)$$

lub

$$|\Delta x(t)| = |\Delta x_0|e^{\lambda t}, \quad \lambda > 0 \quad (2.24)$$

gdzie λ nazywa się wykładnikiem Lapunowa.

Różnice w ewolucji stają się zbyt duże i pojawia się dylemat: która trajektoria jest właściwa, skoro nasza aparatura nie rozróżnia bliskich warunków początkowych. Determinizm staje się złudnym. Nie możemy przewidywać właściwej ewolucji układu.

Przedstawiony powyżej reżim chaotyczny nie jest jedyny. W układzie istnieje wiele takich wartości parametrów (γ, A, ω) , dla których pojawia się ruch chaotyczny. Należy nadmienić, że dla długich czasów wiele trajektorii generowanych przez różne warunki początkowe zachowuje się bardzo podobnie, wiele trajektorii jest przyciąganych. Tu także istnieje atraktor i jego basen przyciągania. Jednakże ten atraktor jest dziwny: jego wymiar nie jest liczbą całkowitą i atraktor jest fraktalem. Dlatego nazywa się dziwnym atraktorem. Brzeg basenu przyciągania tego atraktora też ma dziwną strukturę i jego wymiar jest fraktalny.

Zadania

- Niech $\gamma = 0.1$, $\omega_0 = 1.4$, $(x_0, y_0, z_0) = (-0.5, -0.2, 0)$. Zmieniaj parametr $A = 0.1, 0.32, 0.338, 0.35$.
Obserwuj scenariusz podwojenia okresu:
 - pojawia się atraktor periodyczny o okresie 1.
 - pojawia się atraktor periodyczny o okresie 2.
 - pojawia się atraktor periodyczny o okresie 4.
 - pojawia się atraktor periodyczny o okresie 8 (trudno trafić).
 - pojawia się ruch nieregularny, chaotyczny.
- Zbadaj zachowanie się układu dla następujących wartości parametrów: $\gamma = 1.35 - 1.38$, $A = 1$, $\omega_0 = 1$, $(x_0, y_0, z_0) = (0.0, 0.5, 0)$.
- To samo dla wartości $\gamma = 0.5$, $A = 0.34875$, $\omega_0 = 1$, $(x_0, y_0, z_0) = (0, 0, 0)$

2.3 Scenariusz przejścia do chaosu

Zmieniając parametry układu oraz warunki początkowe, możemy sterować własnościami ewolucji czasowej. Widzieliśmy, że istnieją rozwiązania periodyczne. Może to być prosty ruch periodyczny charakteryzujący się jednym charakterystycznym okresem T (lub częstotliwością). Mogą to być ruchy periodyczne bardziej skomplikowane: o okresie 2, 3, 4, itd. Zauważmy, że ruch periodyczny o okresie 3 powtarza się po czasie 3 razy dłuższym niż ruch o okresie 1. Dlatego też regularność ruchu można zaobserwować po czasie 3 razy dłuższym. Ruch periodyczny o okresie 20 powtarza się po czasie 20 razy dłuższym niż ruch o okresie 1. Dlatego też regularność ruchu jest obserwowana po czasie 20 razy dłuższym. Ruch periodyczny o okresie 2000 powtarza się po czasie 2000 razy dłuższym niż ruch o okresie 1. Dlatego też regularność ruchu może być rozpoznana po czasie 200 razy dłuższym. Zwiększając periodyczność ruchu aż do nieskończoności, zauważamy że regularność ruchu powtarza się po nieskończonym czasie, czyli ruch staje się nieregularny dla obserwatora. Trajektoria wygląda tak, jakby to był ruch przypadkowy, losowy, chaotyczny. Ruch jest ciągle deterministyczny, ale skomplikowany,

niepowtarzalny, nieregularny. W niektórych przypadkach układ jest wyjątkowo wrażliwy na warunki początkowe: dla dwóch różnych, ale bardzo mało różniących się warunków początkowych, odpowiadające im trajektorie z czasem zaczynają się różnić i odbiegać od siebie. Jeżeli zmniejszymy odległość między warunkami początkowymi, to czas po jakim można rozróżnić 2 trajektorii wydłuża się, ale prędzej czy później, trajektorie zaczynają się rozbiegać. Z praktycznego punktu widzenia, warunki początkowe można zadawać ze skończoną dokładnością, ale nie z zerową dokładnością, tak jak to się zakłada w twierdzeniach matematycznych. Dlatego też w reżimie, w którym układ jest czuły na warunki początkowe, w praktyce niepewność warunków początkowych powoduje niepewność ewolucji czasowej. Można to sprecyzować w matematycznym sensie w następujący sposób:

Niech $x(t)$ będzie trajektorią z warunkiem początkowym $x(0)$, a $X(t)$ będzie trajektorią z warunkiem początkowym $X(0)$. Niech dwa warunki początkowe różnią się o małą wielkość:

$$|X(0) - x(0)| = \epsilon_0 \quad (2.25)$$

gdzie $|\dots|$ oznacza odległość przy zadanej metryce. Jeżeli różnią się o taką wielkość lub mniejszą, wówczas są one dla nas nierozróżnialne. Traktujemy je jako takie same w ramach błędu pomiarowego. Pytamy, jaka jest różnica

$$|X(t) - x(t)| = \epsilon(t) \quad (2.26)$$

po czasie t spowodowana niepewnością warunków początkowych ϵ_0 . Wrażliwość na warunki początkowe oznacza, że 2 trajektorie oddalają się od siebie w bardzo szybkim, eksponencjalnym tempie:

$$\epsilon(t) = \epsilon_0 e^{\lambda t} \quad (2.27)$$

gdzie λ nazywa się wykładnikiem Lapunowa. Jeżeli $\lambda < 0$ to dwie bliskie sobie trajektorie nie oddalają się od siebie. Z kolei jeżeli $\lambda > 0$ to trajektorie rozbiegają się.

Niech δ będzie dokładnością naszej aparatury do rozróżniania trajektorii. Jeżeli $\epsilon(t) > \delta$, czyli gdy dwie trajektorie stają się dla nas rozróżnialne, to nie możemy przewidzieć dalszej ewolucji układu: która trajektoria jest właściwa, $x(t)$ czy $X(t)$? Po jakim czasie nasze przewidywania tracą sens? Po takim czasie, że

$$\epsilon(t) = \epsilon_0 e^{\lambda t} > \delta \quad (2.28)$$

czyli po czasie

$$t > \frac{1}{\lambda} \ln \left[\frac{\delta}{\epsilon_0} \right] \quad (2.29)$$

Założmy, że stan początkowy możemy zadać z dokładnością $\epsilon_0 = 10^{-8}$, a tolerancja $\delta = 10^{-5}$ jest dla nas satysfakcjonująca (choć jest to 3 rzędy wielkości gorzej niż dla warunku początkowego). Jak długo nasze przewidywania są akceptowalne:

$$t_1 \approx \frac{1}{\lambda} \ln \left[\frac{\delta}{\epsilon_0} \right] = \frac{1}{\lambda} \ln \left[\frac{10^{-5}}{10^{-8}} \right] = \frac{1}{\lambda} \ln [10^3] = \frac{3}{\lambda} \ln 10 \quad (2.30)$$

Założmy, że ktoś jest w stanie przygotować stan początkowy ze znacznie lepszą dokładnością, a mianowicie 1000 razy lepiej, tzn. $\epsilon_0 = 10^{-11}$. O ile dłużej możemy przewidywać ewolucję układu:

$$t_2 \approx \frac{1}{\lambda} \ln \left[\frac{10^{-5}}{10^{-11}} \right] = \frac{1}{\lambda} \ln [10^6] = \frac{6}{\lambda} \ln 10 = 2t_1 \quad (2.31)$$

To jest zaledwie 2 razy dłuższy czas!! Widać, że gdy układ jest w reżimie chaotycznym, przewidywalność czasowa jest bardzo ograniczona. Zwiększanie dokładności wyznaczania warunków początkowych 1000-krotnie powoduje wydłużenie czasu przewidywalności zaledwie 2 razy. To jest właśnie problem z prognozą pogody. Możemy zwiększać sieć punktów pomiarowych, a i tak przewidywania pogody są rozsądne zaledwie na kilka dni do przodu.

Problem, czy układ wykazuje własności chaotyczne czy nie, nie jest łatwy do stwierdzenia. Ponieważ układ równań różniczkowych zwykle nie można analitycznie rozwiązać, trzeba bazować na metodach komputerowych. Z jednej strony układ jest czuły na warunki początkowe, z drugiej strony sama metoda numeryczna i obliczenia komputerowe obciążone są błędami, których nie można wyeliminować. Może zdarzyć się, że to nie własność układu a artefakty komputerowe wytwarzają złudzenie chaosu. Trzeba na to być czułym. Obecnie istnieją dobre programy komputerowe uwzględniające niedoskonałości o których mowa. Ponadto istnieją kilka charakterystyk, które mają specyficzne własności dla układów chaotycznych. Oto te charakterystyki:

1. Wykładniki Lapunowa λ_i
2. Widmo (spektrum) mocy $P(\omega)$
3. Funkcja korelacyjna $C(\tau)$
4. Cięcie Poincarego
5. Entropia Kołmogorowa K

Badanie wszystkich charakterystyk jest uciążliwe i czasochłonne, ale eliminuje możliwość pomyłki w stwierdzeniu chaotyczności. Przedstawimy główne cechy tych wielkości jakie występują w reżimie chaotycznym i niechaotycznym.

2.3.1 Scenariusz podwojenia okresu

Przedstawimy teraz standardowy scenariusz przejścia do chaosu, który nazywa się przejściem do chaosu poprzez podwojenie okresu. Jest uniwersalny scenariusz, występujący zarówno w układach z ciągłym czasem jaki i w układach dyskretnych. Został potwierdzony w wielu eksperymentach na różnorodnych układach fizycznych.

```
# wykresy dla przypadku z tłumieniem
var('x y z')
x0, y0, z0 = -0.5, -0.1, 0
kolor = 'green'
# siła
F = x-x^3
V = -integrate(F, x)
# tarcie: parametr gamma
g = 0.5
w = 1
```

```

#punkty bifurkacji: 0.34357; 0.35506; 0.35785; 0.35846; ostatni 0.3586
Akeys = ['$a_1$', '$a_2$', '$a_3$', '$a_4$']
Aval = [0.325, 0.354, 0.357, 0.358]
A = dict(zip(Akeys, Aval))
p = A
for a in A.keys():
    # układ różniczkowych równań ruchu
    dx = y
    dy = F - g*y + A[a]*cos(z)
    dz = w
    # numeryczne rozwiązanie równań ruchu
    T = srange(0, 100*pi, 0.01)
    num = desolve_odeint(vector([dx, dy, dz]), [x0, y0, z0], T, [x, y, z])
    figsize = [12, 3] if a == '$a_4$' else 3.5
    start, stop = int(len(num[:, 0])*0.8), len(num[:, 0])
    p[a] = list_plot(zip(num[:, 0][start:stop], num[:, 1][start:stop]), plotjoined=1, \
        color=kolor, axes_labels=['$x(t)$', '$v(t)$'], legend_label='s=%f'%(a, A[a]), figsize=
#
html("""Układ równań różniczkowych
$\dot{x} = %s$
$\dot{y} = %s$
$\dot{z} = %s$
z warunkami początkowymi
$(x_0, y_0, z_0) = (%.2f, %.2f, %.2f)$
""%(dx, dy, dz, x0, y0, z0))
html.table([p['$a_1$'], p['$a_2$'], p['$a_3$']])
p['$a_4$'].show()

```

2.3.2 Wykładniki Lapunowa

Dla rozpatrywanego układu oscylatora Duffinga przestrzeń fazowa jest 3-wymiarowa. Dlatego też w rzeczywistości są 3 wykładniki Lapunowa, a nie 1 jak powiedzieliśmy powyżej. Aby wyjaśnić ten problem, musimy rozważyć zbiór warunków początkowych, które tworzą kulę K w badanej przestrzeni fazowej. Jeżeli będziemy iterować równania dla $x(t), y(t), z(t)$ startując z wszystkich warunków początkowych w kuli K , to zbiór punktów zawartych początkowo w kuli zmieni swój kształt. Kula już nie będzie kulą. Prędkość z jaką kula ulega deformacji we wszystkich 3 kierunkach (x, y, z) w przestrzeni fazowej jest określona przez 3 wykładniki Lapunowa $\lambda_1, \lambda_2, \lambda_3$. Jeżeli badany układ jest chaotyczny, to zazwyczaj kula powiększa się w jednym kierunku, a maleje w dwóch pozostałych przyjmując kształt elipsoidy. W takim wypadku możemy zdefiniować trzy wykładniki Lapunowa mierzące deformacje elipsoidy w trzech wzajemnie prostopadłych kierunkach. Ilość wykładników Lapunowa jest więc zależna od wymiaru układu. Są one jednym z kryteriów chaotyczności ruchu. Jeżeli elipsoida w jednym kierunku rozciąga się, wielkość jej osi w tym kierunku rośnie i wykładnik Lapunowa jest dodatni. W kierunkach, w których osie elipsoidy maleją, wykładniki Lapunowa są ujemne.

Dwie trajektorie leżące początkowo blisko siebie propagują w czasie w odległości $l(t) \propto e^{\lambda_1 t}$, powierzchnia S zmienia się w tempie $S(t) \propto e^{(\lambda_1 + \lambda_2)t}$, a objętość M zmienia się w tempie $M(t) \propto e^{(\lambda_1 + \lambda_2 + \lambda_3)t}$. W reżimie chaotycznym co najmniej jeden z wykładników Lapunowa jest dodatni. Oznacza to, że w przestrzeni fazowej trajektorie rozbiegają się w jednym kierunku. Jeżeli wszystkie 3 wykładniki są ujemne, układ jest w reżimie regularnym (periodycznym, quasi-periodycznym). Nie ma metod analitycznych pozwalających obliczyć wykładniki Lapunowa. Metody numeryczne też nie są proste. W literaturze można znaleźć algorytmy

służące do wyznaczania $\lambda_1, \lambda_2, \lambda_3$.

W przypadku oscylatora Duffinga można otrzymać cząstkowe informacje o wykładnikach Lapunowa.

1. Trzecie równanie dla pomocniczej zmiennej z można rozwiązać otrzymując funkcję

$$z(t) = \omega t + c \quad (2.32)$$

Z pewnością dwie bliskie sobie trajektorie $z_1(t) = \omega t + c_1$ oraz $z_2(t) = \omega t + c_2$ dla chwili $t = 0$ nie rozbiegają się exponencjalnie ponieważ

$$|z_1(t) - z_2(t)| = |c_1 - c_2| \quad (2.33)$$

Dlatego też jeden z wykładników wynosi zero, np.

$$\lambda_2 = 0 \quad (2.34)$$

2. Przypomnijmy w tym miejscu, że oscylator Duffinga jest opisany przez układ równań

$$\begin{aligned} \dot{x} &= F_1 = y, & x(0) &= x_0, \\ \dot{y} &= F_2 = x - x^3 - \gamma y + A \cos z, & y(0) &= y_0, \\ \dot{z} &= F_3 = \omega, & z(0) &= 0. \end{aligned} \quad (2.35)$$

Zbadajmy, jak zmienia się w czasie objętość fazowa układu. W tym celu musimy obliczyć dywergencję pola wektorowego

$$\operatorname{div} \vec{F} = \frac{\partial F_1}{\partial x} + \frac{\partial F_2}{\partial y} + \frac{\partial F_3}{\partial z} = -\gamma < 0 \quad (2.36)$$

Oznacza to, że objętość fazowa w przestrzeni 3-wymiarowej maleje w tempie (zobacz paragraf o układach dysypatywnych)

$$M(t) \propto e^{-\gamma t} \quad (2.37)$$

Z drugiej strony, jak powiedzieliśmy powyżej,

$$M(t) \propto e^{(\lambda_1 + \lambda_2 + \lambda_3)t} \quad (2.38)$$

Wynika stąd, że suma wszystkich wykładników jest stała i wynosi

$$\lambda_1 + \lambda_2 + \lambda_3 = -\gamma < 0 \quad (2.39)$$

czyli tylko stała tłumienia γ określa tempo malenia objętości fazowej. Ponieważ $\lambda_2 = 0$, otrzymujemy interesujący związek pomiędzy dwoma pozostałymi wykładnikami:

$$\lambda_1 + \lambda_3 = -\gamma \quad (2.40)$$

W reżimie chaotycznym jeden z wykładników jest dodatni, np. $\lambda_1 > 0$ oraz drugi wykładnik musi być ujemny, np. $\lambda_3 < 0$. Mamy obecnie następujące informacje o wykładnikach Lapunowa dla oscylatora Duffinga:

$$\lambda_1 > \lambda_2 > \lambda_3, \quad \lambda_1 > 0, \quad \lambda_2 = 0, \quad \lambda_3 < 0, \quad M(t) = M(0)e^{(\lambda_1 + \lambda_2 + \lambda_3)t} = M(0)e^{-\gamma t} \quad (2.41)$$

Zwracamy uwagę na to, że elipsoida w 3-wymiarowej przestrzeni fazowej rozciąga się w jednym kierunku, kurczy się w drugim kierunku i nie zmienia się w trzecim kierunku oraz objętość elipsoidy cały czas maleje. Tak to wygląda w reżimie chaotycznym. W reżimie nie-chaotycznym: elipsoida kurczy się w jednym kierunku, kurczy się w drugim kierunku i nie zmienia się w trzecim kierunku oraz objętość elipsoidy cały czas maleje. Atraktory, które pokazywaliśmy poprzednio, istnieją w 3-wymiarowej przestrzeni fazowej, ale ponieważ objętość fazowa cały czas maleje, wymiar atraktorów musi być mniejszy od 3. W reżimie nie-chaotycznym, n-okresowe atraktory (krzywe) mają wymiar 1. Atraktory w reżimie chaotycznym mają wymiar większy niż 1, ale mniejszy niż 3. Kaplana i Yorke (1979) postawili hipotezę, że istnieje związek pomiędzy wymiarem fraktalnym atraktora D_A a wykładnikami Lapunowa. Relacja ta ma postać:

$$D_A = 2 + \frac{\lambda_1}{|\lambda_3|} > 2 \quad (2.42)$$

Jeżeli analizujemy wymiar atraktora w odwzorowaniu Poincarego (na płaszczyźnie), to wymiar ten jest o 1 mniejszy:

$$d_A = D_A - 1 \quad (2.43)$$

Do dzisiaj jest to tylko hipoteza, choć w wielu przypadkach potwierdzona przez eksperymenty numeryczne.

2.3.3 Widmo mocy

Jest to kolejna wielkość, która może być indykatorem chaotycznego zachowania się układu deterministycznego. Pojęcie widma mocy jest dobrze ugruntowane w teorii sygnałów, traktowanych jako nośnik informacji. W ogólności sygnały mogą być deterministyczne (jak w naszym przypadku) i losowe (stochastyczne). W sensie inżynierskim, sygnał to dowolna funkcja czasu. Jako modele sygnałów wprowadza się również wielkości nazywane dystrybucjami (funkcjami uogólnionymi). Tylko nieliczne proste sygnały można opisać formułami matematycznymi. Większość sygnałów, z jakimi spotykamy się w praktyce, ma przebieg na tyle złożony i nieregularny, że ich bezpośredni opis jako funkcji czasu jest kłopotliwy. Dlatego też należy posługiwać się ich różnego rodzaju reprezentacjami. Reprezentacja sygnału stanowi pewien rodzaj jego symbolicznego opisu, niekiedy o znacznym stopniu abstrakcji. Jej istotą jest to, że zawiera ona pełną informację o sygnale, choć zwykle wyrażoną w innym języku, niż bezpośredni język w terminach funkcji czasu. Oznacza to, że znając sygnał, możemy jednoznacznie wyznaczyć jego reprezentację, znając zaś tę reprezentację – odtworzyć jednoznacznie sygnał. Istnieje wiele sposobów reprezentacji sygnałów. Jednym z nich jest analiza furierowska za pomocą transformat Fouriera lub szeregów Fouriera.

Przypomnijmy pojęcie transformacji Fouriera funkcji lub dystrybucji. W najprostszym ujęciu transformatą Fouriera $\hat{f}(\omega)$ funkcji $f(t)$ nazywamy całkę

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} e^{i\omega t} f(t) dt \quad (2.44)$$

gdzie ω jest dowolną liczbą rzeczywistą.

Ponieważ nas interesuje ewolucja sygnału czasowego $f(t) = (x(t), y(t), z(t), \dots)$ dla czasów $t > 0$, zdefiniujemy nieco inaczej transformatę Fouriera jako graniczną wartość całki:

$$\hat{f}(\omega) = \lim_{T \rightarrow \infty} \int_0^T e^{i\omega t} f(t) dt \quad (2.45)$$

W praktyce obliczeń komputerowych nigdy nie wykonujemy dokładnej granicy $T \rightarrow \infty$, ale rozpatrujemy dostatecznie długi czas, gdy pojawia się stan ustalony i efekty przejściowe w ewolucji zanikają. Ze względu na występowanie funkcji podcałkowej $e^{i\omega t}$, transformata Fouriera jest funkcją zespoloną. Dlatego też bada się funkcję rzeczywistą w postaci

$$P(\omega) = |\hat{f}(\omega)|^2 \quad (2.46)$$

Nazywa się ona widmem mocy sygnału czasowego $f(t)$. W pewnych przypadkach, faktycznie jest to wielkość fizyczna mająca interpretację mocy, a liczba ω 'jest częstotliwością, która jest wielkością dodatnią, $\omega > 0$ '. W dalszym ciągu przyjmiemy to założenie o dodatniości "częstotliwości". W ogólności, jej związek z mocą (w sensie fizycznym) jest luźny. To widmo mocy jest zdefiniowane inaczej niż w teorii stacjonarnych procesów stochastycznych: tam jest to transformacja Fouriera funkcji korelacyjnej $C(t)$ procesu stochastycznego.

Aby wyrobić sobie intuicję o własnościach transformaty Fouriera i widma mocy, wystarczy rozpatrzyć kilka przypadków funkcji $f(t)$.

Przypadek 1 Jedna harmonika (fala monochromatyczna)

$$f_1(t) = A \cos(\Omega t), \quad \hat{f}_1(\omega) = A \int_0^\infty e^{i\omega t} \cos(\Omega t) dt = \frac{\pi}{2} A \delta(\omega - \Omega) \quad (2.47)$$

Transformatą Fouriera jest delta Diraca δ , czyli w widmie mocy pojawia się jeden pik (który w praktyce jest zawsze skończony).

Przypadek 2 Kilka harmonik

$$f_2(t) = \sum_{k=1}^n A_k \cos(\Omega_k t), \quad \hat{f}_2(\omega) = \sum_{k=1}^n A_k \int_0^\infty e^{i\omega t} \cos(\Omega_k t) dt = \frac{\pi}{2} \sum_{k=1}^n A_k \delta(\omega - \Omega_k) \quad (2.48)$$

Transformatą Fouriera jest suma przesuniętych delt Diraca δ , czyli w widmie mocy pojawia się szereg pików (które w praktyce są zawsze skończone).

Zauważmy, że dla tak zdefiniowanych transformacji Fouriera nie istnieje widmo mocy, ponieważ w ścisłym sensie matematycznym nie istnieje $\delta^2(\omega - \Omega)$ dla delty Diraca. Jednak nie chodzi o precyzję matematyczną, ale o to że pojawia się pik, który nigdy nie jest nieskończony jak w delcie Diraca. My jednak potrzebujemy praktycznej metody sprawdzania chaotyczności procesu i zwykle sygnał próbujemy dla dyskretnych wartości czasu t . Dlatego też należy wykorzystać aparat Dyskretnej Transformacji Fouriera, która skończony ciąg sygnału

$$\{x_0, x_1, x_2, \dots, x_{N-1}\} \quad (2.49)$$

przekształca w skończony ciąg amplitud

$$\{A_0, A_1, A_2, \dots, A_{N-1}\} \quad (2.50)$$

odpowiednich harmonik poprzez relacje:

$$A_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N}, \quad x_n = \frac{1}{N} \sum_{k=0}^{N-1} A_k e^{2\pi i k n / N} \quad (2.51)$$

Dla odpowiednio dużej liczby N (w praktyce rzędu 100), zgodność pomiędzy transformatą Fouriera a Dyskretną Transformatą Fouriera jest zadziwiająco dobra.

2.3.4 Funkcja korelacyjna

Jeżeli badamy deterministyczny proces, nie zawsze jest sens mówić o wartości średniej, w takim sensie jak w teorii procesów stochastycznych lub na wykładach z fizyki statystycznej: średniowanie po realizacjach lub po zespole statystycznym. Ale jeżeli proces deterministyczny jest ergodyczny (trudne pojęcie!), to średnia wartość jest dobrze określona i średnia po zespole jest równoważna średniej po czasie. Jeżeli dodatkowo proces jest stacjonarny, to można zdefiniować funkcję korelacyjną $C(\tau)$ dla procesu deterministycznego. W naszym przypadku: dla położenia lub prędkości, jest ona zdefiniowana przez relacje:

$$C(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [x(t+\tau) - \langle x(t+\tau) \rangle][x(t) - \langle x(t) \rangle] dt, \quad \langle x(t) \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T x(t) dt \quad (2.52)$$

2.3.5 Odwzorowanie (cięcie) Poincarego

Odwzorowanie Poincarego jest innym przedstawieniem dynamiki układu. Najprościej jest to wytłumaczyć na przykładzie oscylatora Duffinga. Jego przestrzeń fazowa jest 3-wymiarowa. Ruch w trzecim wymiarze jest jednostajny, $z(t) = \omega_0 t$. Rzut orbity na płaszczyznę (x, y) jest przedstawiony w postaci krzywych fazowych w poprzednich częściach książki.

Jak widać, we wszystkich przykładach krzywe fazowe na płaszczyźnie są ograniczone na pewnym obszarze (x, y) . We wszystkich rozpatrywanych przypadkach ruch wydaje się być prawie-periodyczny: układ ciągle powraca w te same obszary. Można zbudować następujące przedstawienie tego ruchu.

Okres siły periodycznej wynosi

$$T = \frac{2\pi}{\omega_0} \quad (2.53)$$

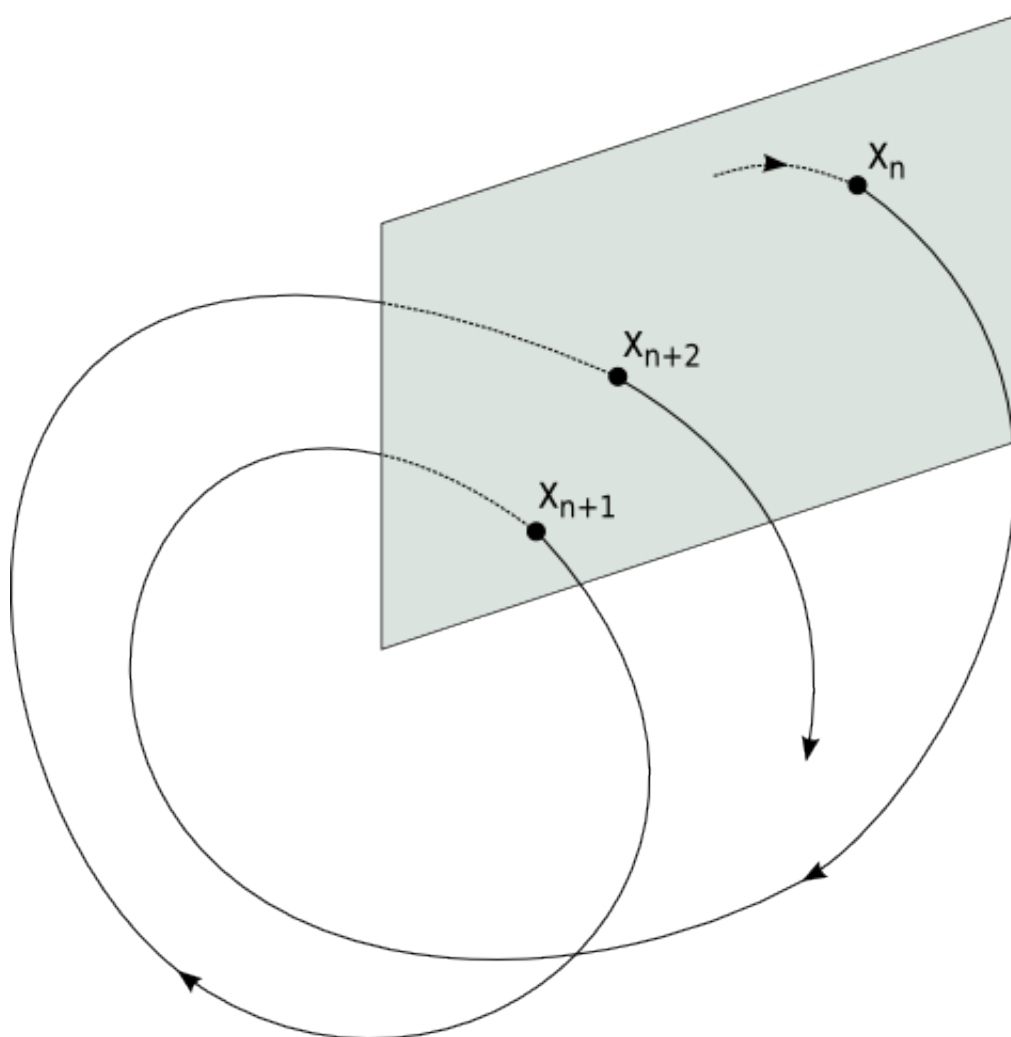
Wprowadzamy dyskretny czas

$$t_n = nT, \quad n = 1, 2, 3, \dots \quad (2.54)$$

Zapisujemy położenie i prędkość cząstki w dyskretnych chwilach czasu:

$$x_n = x(t_n), \quad y_n = y(t_n), \quad x(0) = x_0, \quad y(0) = y_0 \quad (2.55)$$

Współrzędne tych punktów nanosimy na płaszczyznę. Otrzymujemy odwzorowanie które nazywamy odwzorowaniem Poincarego. Obrazowo mówiąc można w 3-wymiarowej przestrzeni



Rysunek 2.4: Konstrukcja cięcia Poincarego.

fazowej wprowadzić płaszczyznę, tak aby nigdzie nie była styczna do trajektorii i była transwersalna do trajektorii (ściślej mówiąc do potoku fazowego), czyli aby trajektoria przecinała płaszczyznę, a nie była równoległa do niej (nie omijała jej).

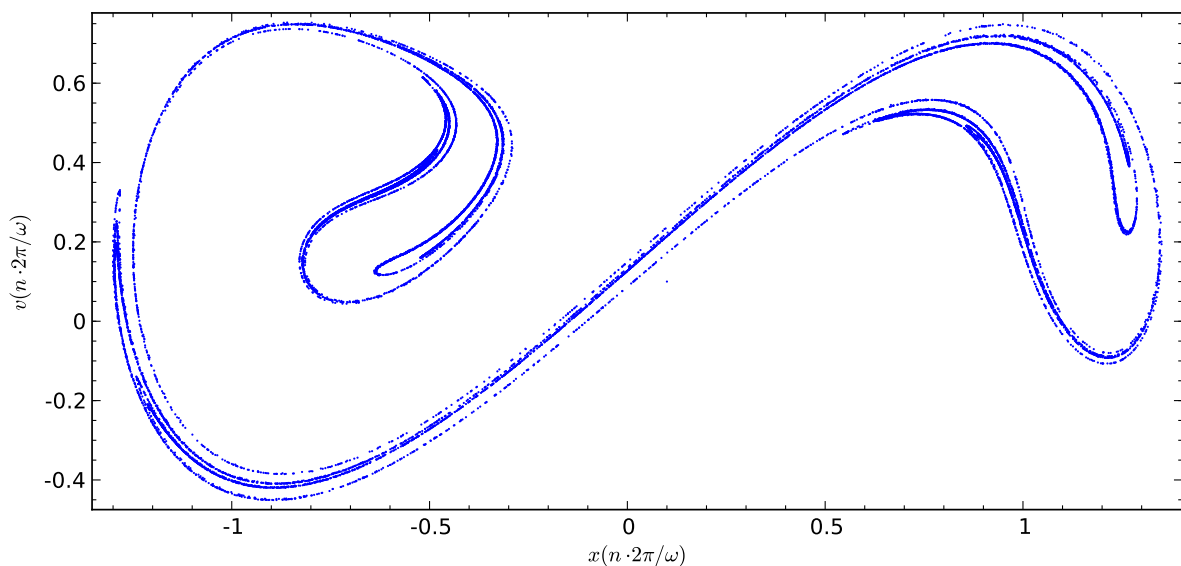
Odwzorowanie Poincarego to przyporządkowanie:

$$x_{n+1} = \mathcal{G}(x_n) \quad (2.56)$$

Jawna konstrukcja tego odwzorowania z wyjściowego układu równań różniczkowych jest możliwa tylko w bardzo specjalnych przypadkach. W przypadku oscylatora Duffinga, nie można otrzymać jawnej postaci tego odwzorowania. Jedynie użycie komputera pozwala na graficzne przedstawienie funkcji \mathcal{G} .

Jakie wnioski płyną z takiego przedstawienia.

1. Gdyby trajektoria była krzywą zamkniętą w kształcie elipsy (atraktor o okresie 1) to na cięciu Poincarego otrzymalibyśmy 1 punkt.
2. Gdyby trajektoria była atraktorem o okresie 2 to na cięciu Poincarego otrzymalibyśmy 2 punkty.
3. Gdyby trajektoria była chaotyczna, to za każdym razem przebiega przez inne punkty płaszczyzny i tworzy zbiór składający się z nieskończenie wielu punktów. Poniżej pokazano takie odwzorowanie dla oscylatora Duffinga.



Rysunek 2.5: Dziwny atraktor.

Jeżeli jesteśmy w stanie zbudować graficznie przedstawienie Poincarego danego układu dynamicznego z ciągłym czasem, wówczas możemy rozpoznać takie reżimy które są “podejrzane” o własności chaotyczne. Numerycznie nie powinno nastręczać to większych problemów. Jeżeli znamy ω_0 bądź okres powrotu do obliczenia cięcia to wystarczy wykorzystać poniższy kod Sage. Zwracamy jedynie uwagę na to, że odpowiednio “gęsty” obraz uzyskamy dla bardzo długich przebiegów (dużych T).

```
var('x y z')
# parametry układu równań różniczkowych
```

```
a, g = 0.3, 0.26
# częstotliwość (do obliczania cięcia Poincarego)
w0 = 1
# wartości początkowe
x0, y0, z0 = 0.1, 0.1, 0
#układ równań różniczkowych
dx = y
dy = x - x**3 - g*y + a*cos(z)
dz = w0
#krok co jaki wypełniać się ma nasza lista
#rozwiązań ustawiamy równy okresowi
h = 2.0*pi/w0
###
#symulacje
###
T = 10000
listT = srange(0,T,float(h), include_endpoint=True)
sol = desolve_odeint(vector([dx, dy, dz]), [x0, y0, z0], listT, [x,y,z])
#i sam rysunek cięcia
points(zip(sol[:,0],sol[:,1]), figsize=(8,4), axes_labels=["$x(n\cdot 2 \pi/\omega)$", "$y(n\cdot 2 \pi/\omega)$"])
```

2.3.6 Przykłady chaosu w Naturze

Należy odróżnić procesy chaotyczne od procesów losowych. Procesy chaotyczne są deterministyczne, a procesy stochastyczne są procesami losowymi. Procesy chaotyczne są badane przez matematyków, fizyków, chemików, biologów, socjologów, meteorologów, astrofizyków, w teorii informacji i neuronauce. We wszystkich tych gałęziach nauki, występują deterministyczne modele wykazujące własności chaotyczne. Od lat 60-tych XX wieku opublikowano tysiące prac na temat układów chaotycznych. Matematycy mówią, że prawie wszystkie układy dynamiczne są chaotyczne, a tylko nieliczne układy nie wykazują tej własności. Matematycy dowodzą, że przestrzeń fazowa układu modelowanego przez autonomiczny układ równań różniczkowych musi być co najmniej 3-wymiarowa, aby istniał chaos. Dla układów dyskretnych nie ma takich ograniczeń: jedno równanie rekurencyjne $x_{n+1} = f(x_n)$ także wykazuje własności chaotyczne.

Poniżej podajemy kilka przykładów rzeczywistych zjawisk wykazujących własności chaotyczne.

1. Dynamika cieczy i turbulencja
2. Lasery
3. Układy elektroniczne
4. Plasma
5. Reakcje chemiczne

Na stronie internetowej Wikipedii z hasłem Chaos Theory można znaleźć dalsze przykłady oraz podstawowe prace na ten temat. Na zakończenie tej części książki musimy wspomnieć o człowieku, który to wszystko zapoczątkował w 1961 roku. Był to Edward Lorenz, matematyk i meteorolog amerykański, który analizował jeden z najprostszych modeli pozwalających przewidywać pogodę. To z jego nazwiskiem związany jest “efekt motyla” obrazujący niezwykłą czułość dynamiki na zaburzenia warunków początkowych: czy ruch motyla w Brazylii może

spowodować tornado w Teksasie (ściśle rzecz ujmując to Philip Merilees zasugerował Lorenzowi taki tytuł wykładu podczas posiedzenia American Association for the Advancement of Science w 1972 roku). W tym obrazowym powiedzeniu zawarta jest istota chaosu: Motyl poprzez swój lot zaburza lokalnie ruch powietrza. Ten zaburzony ruch powietrza narasta i powoduje coraz to większe zmiany pogodowe, zmienia radykalnie “trajektorię” doprowadzając do tornado, które pojawi się nad Teksasem. Czy faktycznie motyl może być taki groźny?

Dodatek numeryczny

3.1 Liczby losowe

Uwaga: W tym kursie do numerycznych realizacji używać będziemy pakietu Sage. Jest to dostępny dla każdego program typu open-source bazujący na języku Python.

3.1.1 Liczby losowe i pseudolosowe

Intuicyjnie dość dobrze rozumiemy co oznacza termin *liczba losowa*. Każdy z nas choć raz w życiu podrzucił monetę do góry po to, by “ślepy los” zdecydował za niego o jakimś wyborze (jeżeli w ten sposób zdecydowaliście o wyborze studiów, to szczerze mówiąc - gratuluję). Oczywiście na monecie nie ma żadnych liczb, ale można sobie potraktować reszkę (R) jako 0 a orła (O) jako 1 (co bardzo dobrze reprezentuje fałsz i prawdę lub niemożliwe i pewne zdarzenie w teorii prawdopodobieństwa). Teraz już możemy sobie podrzucać monetę i na kartce papieru zapisywać kolejne wylosowane (wyrzucone) przez nas liczby

0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1,

co odpowiada oczywiście wyrzuceniu kolejno

R, O, R, R, R, R, O, R, O, O, R, R, O, R, O.

W naszym przypadku zapiszemy sobie te liczby od razu do listy w notatniku Sage.

```
rzuty = [0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1]
```

Mamy teraz je dostępne pod zmienną `rzuty`. Do prostych zagadnień, gdzie potrzebne jest nam kilka, czy nawet kilkanaście takich liczb, bez problemu możemy poradzić sobie rzucając monetą. Jeżeli potrzebujemy zdecydować o wyborze pomiędzy trzema możliwościami możemy użyć sześcienną kość do gry i przykładowo wybrać wynik poprzez działanie modulo 3 ($\text{mod}3$). Tym razem dostaniemy trzy możliwe liczby 0, 1, 2

```
# rzuty kością
5, 3, 6, 5, 6, 6, 5, 3, 2, 4, 3, 1, 1, 6, 1,

# modulo 3
2, 0, 0, 2, 0, 0, 2, 0, 2, 1, 0, 1, 1, 0, 1.
```

Sytuacja robi się jednak nieco bardziej skomplikowana, gdy będziemy potrzebować tysiąc, milion czy bilion takich liczb. Jeżeli nawet grupa 10 studentów była by w stanie wyrzucić monetą tysiąc losowych zer i jedynek w pół godziny (włączając w to zapisywanie w liście Sage lub nawet na kartce papieru) to uzyskanie miliona liczb jest praktycznie nie do zrobienia w ten sposób. problem pojawia się też w momencie, gdy chcielibyśmy mieć liczby naturalne z zakresu np: 0 – 10, czy w końcu losowe liczby zmiennoprzecinkowe. Metody chałupnicze w tym momencie się kończą.

Z pomocą może przyjść nam komputer. Obecnie znakomita większość języków programowania (przynajmniej tych realnie wykorzystywanych¹) posiada w swoich standardowych bibliotekach funkcje (metody, klasy) umożliwiające wygenerowanie liczby (pseudo)losowej z przedziału $[0, 1)$ lub też $[0, \text{RAND_MAX}]$, gdzie ów `RAND_MAX` to stała zależna od architektury komputera, kompilatora i bibliotek.

W Sage liczby losowe uzyskuje się poprzez funkcję `random()`. Zwraca ona liczbę losową z przedziału $[0.0, 1.0)$. Wykorzystując proste wyrażenie listowe możemy przypisać do listy `N` liczb losowych.

```
N = 1000
lista = [random() for i in xrange(N)]
```

Inna funkcja `randint(a, b)`, zwraca liczby całkowite z przedziału $[a, b]$. Czyli symulacja rzutu monetą może być zrealizowana poprzez

```
rzut_moneta = [randint(0,1) for i in xrange(N)]
```

Zadanie 2.2.1 Zamodeluj w Sage rzut kością. Wygeneruj listę 1000 liczb odzwierciedlających 1000 rzutów symetryczną sześcienną kością do gry. Wynik zapisz w zmiennej `rzut_kostka`.

Matematycznie rzecz biorąc liczbę losową można utożsamić z wartością jaką przybiera pewna zmienna losowa ξ . Możemy napisać, że dla procesu jakim jest rzut kością zmienna losowa ξ może przybierać wartości 0 lub 1. Matematyczne konsekwencje poznaliście już na wykładzie [Procesy i zjawiska losowe](#), tutaj zajmiemy się znacznie szerzej generowaniem liczb losowych i wykorzystaniem ich właśnie do realizacji procesów losowych, ze szczególnym uwzględnieniem zastosowania dla rynków finansowych, czy w ogólności w modelach ekonomicznych.

No koniec tego rozdziału musimy sobie powiedzieć jasno: program komputerowy bazujący na deterministycznym generatorze liczb losowych może wygenerować tylko i wyłącznie liczby pseudolosowe, czyli takie, które tylko imitują prawdziwe liczby czysto losowe. Te ostatnie osiągalne są tylko procesie rzeczywistym. Możemy jednak za pomocą takich generatorów uzyskać ciąg liczb (bitów), który pod pewnymi względami będzie nierozróżnialny od ciągu uzyskanego z prawdziwie losowego źródła (np: z rzutu rzeczywistą kością).

¹ Generator liczb pseudolosowych można napisać nawet dla tak egzotycznych języków jak `Brainf*ck`.

3.1.2 Generatory liczb

Generator liczb losowych (RNG, z ang. random number generator) lub nieco bardziej ściśle *generator zdarzeń losowych* (REG, z ang. random event generator) to układ produkujący losowy ciąg elementów binarnych (bitów) najczęściej ułożony w postaci szeregu liczb losowych. Z punktu widzenia sposobu generowania liczb losowych wyróżniamy generatory sprzętowe (fizyczne, rzeczywiste) i programowe.

Generatory sprzętowe

TRNG (z ang. True RNG) - działające na zasadzie obrazowania właściwości i parametrów fizycznego procesu stochastycznego. Może to być ów rzut kością, monetą, wybieranie karty z talii kart itp. Wykorzystywać można też: efekt fotoelektryczny, szum termiczny, szum śrutowy, proces zaniku radioaktywnego...

Generatory programowe

PRNG, (z ang. Pseudo RNG) - działające na zasadzie deterministycznego obliczania ciągu liczb, które wyglądają jak liczby losowe. Algorytmy realizujące PRNG istnieją już ponad pół wieku i są obecnie zaimplementowane dla większości języków programowania. Na podstawie początkowej wartości nazywanej ziarnem czy zarodkiem (z ang. seed) oblicza kolejne wartości. Obie prezentowane funkcje Sage (`random` i `randint`) korzystają właśnie z jednego z takich algorytmów, zwanego *Mersenne Twister*. Jest to obecnie chyba najbardziej popularny algorytm opracowany w 1997 roku. Np. Matlab/GNU Octave też wykorzystuje ten algorytm. Jest on stosunkowo skomplikowany i może być trudny do realizacji, dlatego też omówimy sobie dużo prostszy, liniowy generator i omówimy jego zalety i (przede wszystkim) wady.

Programowe generowanie liczb losowych ² oparte jest na rekurencji

$$x_i = f(x_{i-1}, x_{i-2}, \dots, x_{i-k}),$$

czy w nieco bardziej zwartej formie

$$x_i = f(x_{i-1}).$$

Sekwencje te będą w oczywisty sposób deterministyczne. Problem polega na wygenerowaniu liczb których własności bardzo dobrze przypominają główne własności liczb prawdziwie losowych. Dodatkowo sekwencje liczb pseudolosowych będą powtarzały się co pewien okres, więc dość istotne jest aby generator takich liczb posiadał ów okres jak najdłuższy.

Liniowy generator kongruencyjny

LCG (linear congruential generator) wyznaczony jest przez metodę rekurencyjną

$$X_{n+1} = (aX_n + c) \mod m.$$

² Od tej chwili będziemy zawsze pisać *liczba losowa* a mieć na myśli *liczbę pseudolosową*, chyba, że napisane zostanie explicite, że mówimy o rzeczywistych liczbach losowych.

Stan początkowy to wartość ziarna (załążka). Nie jest on zbytnio bezpieczny - istnieją techniki identyfikacji parametrów modelu na podstawie obserwacji wyników. Dla niektórych parametrów jest prawie losowy a dla innych dość szybko staje się okresowy. W powyższej definicji x_0 to ziarno (załążek), a mnożnik, c przesunięcie a $m \in \mathbb{Z}$ nazywamy modułem. Dwie liczby nazywamy kongruentnymi (przystającymi) modulo m jeżeli ich różnica jest podzielna przez m . Jeżeli $0 \leq a < m$ oraz $a \equiv b \pmod{m}$ wtedy a nazywamy resztą $b \pmod{m}$. Liczbę a można łatwo obliczyć z

$$a = b - \lfloor b/m \rfloor \times m$$

gdzie funkcja podłoga (z ang. floor) $\lfloor \cdot \rfloor$ oblicza największą liczbą całkowitą mniejszą od \cdot .

Jeżeli weźmiemy $c = 0$ dostaniemy multiplikatywny generator kongruencyjny. Jeżeli chodzi o moduł, to typowymi wartościami będą potęgi 2^k , a wartościami tych potęg będą typowe wielkości maszynowe dla przechowywania liczb całkowitych. Tak było przynajmniej dla wczesnych realizacji takiego generatora, co związane było z możliwością łatwej redukcji modulo poprzez wykorzystanie przepełnienia w stałopozycyjnej reprezentacji liczb w operacji mnożenia (w ciele liczb całkowitych) ax_i . W operacjach stałoprzecinkowych pierwszy bit reprezentuje znak, wobec czego w wyniku takiego mnożenia zamiast liczb z zakresu $[0, 2^{32} - 1)$ dostaniemy liczby z zakresu $[-2^{31} + 1, 2^{31} - 1]$. W ogólności wykonując operacje na liczbach większych od $2^{31} - 1$ jako wynik zachowujemy tylko bity niskiego rzędu.

Mnożnik a wybierać trzeba w taki sposób, aby LCG miał jak najdłuższy okres. Na 32-bitowych maszynach popularnymi wartościami początkowo były $m = 2^{32}$ i $a = 65539$. Jako, że dzisiejsze komputery są na tyle wydajne, by przeprowadzać redukcję modulo bardzo wydajnie, wiele ówczesnych implementacji generatora wykorzystuje operacje zmiennoprzecinkowe o zwiększonej precyzji. Inne wartości $a = 1099087573, 2396548189, 3934873077, 2304580733$ również produkują porządne sekwencje liczb losowych.

Innym dobrym wyborem dla m jest podstawienie dużej liczby pierwszej p . Wtedy okresem LCG będzie $p-1$ jeżeli tylko mnożnik ustawimy jako jego pierwiastek pierwotny. Szczególnie ważne wydają się być liczby pierwsze postaci $2^p - 1$, nazywane liczbami Mersenne'a. Na maszynach 32-bitowych popularnym wyborem bywa para $m = 2^{31} - 1$ i jej pierwiastek pierwotny $a = 7^5 = 16807$.

Implementacja LCG w Sage nie powinna nastręczać zbytnich problemów.

```
def myLCG(x, a=1664525, b=1013904223, m=2**32):
    return mod(a*x+b, m)
```

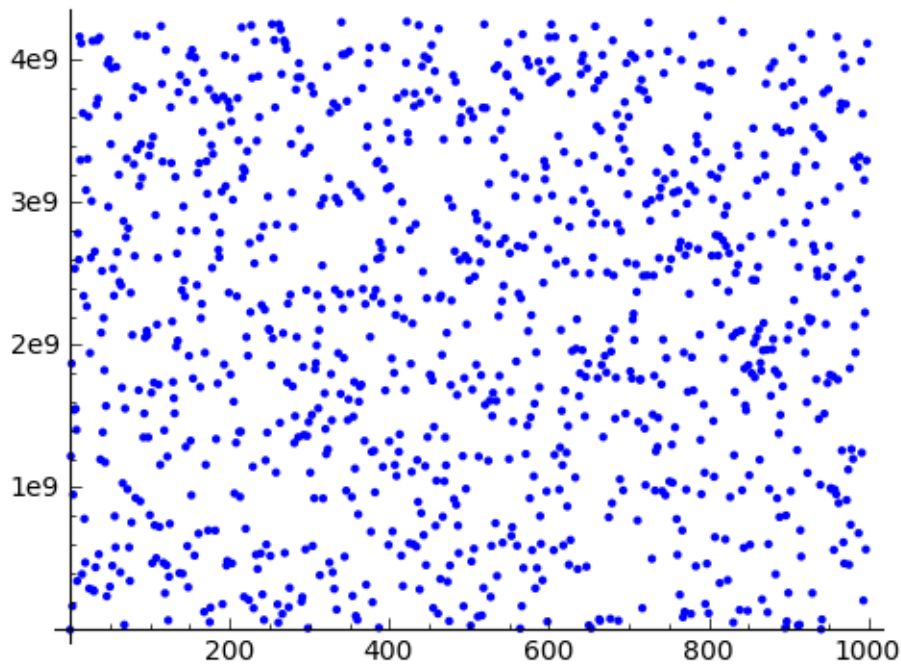
Możemy teraz wygenerować N liczb używając LCG i zmagazynować je w pythonowskiej liście.

```
def get_from_LCG(n=1, seed=123):
    ret = [seed]
    for i in xrange(n-1):
        ret.append(myLCG(ret[i]))
    return ret
```

```
lcg_list = get_from_LCG(N)
```

Powinniśmy dostać rysunek podobny do tego poniżej.

Jak widać, program generuje liczby losowe z zakresu $[0, m)$.



Rysunek 3.1: 1000 liczb losowych wygenerowanych generatorem liniowym LCG

Rejestr przesuwający z liniowym sprzężeniem zwrotnym

TBA

Generator Wichmanna – Hilla

TBA

Mersenne Twister

TBA

W dalszej części wykładu (a raczej ćwiczeń) będziemy bazować na domyślnym generowaniu liczb losowych w Sage. Posłużymy nam do tego wspomnianą już funkcją `random()` zwracającą liczbę pseudolosową o rozkładzie jednorodnym na odcinku $[0,1)$ (co często oznaczane jest poprzez $U(0,1)$).

$$U(0,1) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{poza} \end{cases}$$

Aby uzyskać liczbę z przedziału $[0,12.76)$ wystarczy po prostu pomnożyć liczbę zwracaną przez `random()` przez prawą granicę

```
random() * 12.76
```

a żeby uzyskać listę 123 liczb z przedziału $[-13.3, 33.1)$ należy wykonać

```
[random()*(33.1+13.3) - 13.3 for i in xrange(123)]
```

W ogólności do wygenerowania listy N liczb losowych z przedziału $[A,B)$ należy użyć polecenia

```
N = 100
A = -10
B = 20
[random()*(B-A) + A for i in xrange(N)]
```

Zadanie 2.2.2 Zmodyfikuj definicję `mylcg` tak, aby funkcja zwracała liczby losowe z przedziału $[0,1)$.

Zadanie 2.2.3 LCG zdefiniowany tak jak powyżej produkuje stosunkowo dobre liczby losowe (prace naukowe nad tym stosunkowo prostym generatorem trwają do dzisiaj, dowodzone są coraz to inne okresy bazujące na wyborze różnych zestawów parametrów a, c, m). Naszym zadaniem będzie natomiast zepsucie takiego generatora. Proszę znaleźć (numerycznie) 4 zestawy parametrów definiujących LCG takich, aby okres generatora był krótki. Wykreśl w Sage 4 rysunki LCG(N) (dla powiedzmy $N=1000$) dla owych parametrów. Powinieneś zauważyć regularność.

3.1.3 Generowanie liczb losowych o zadanym rozkładzie

Jako, że już dysponujemy generatorem liczb losowych o rozkładzie jednostajnym na odcinku $[0,1) = U(0,1)$ to możemy pokusić się o wygenerowanie liczb losowych o różnych innych rozkładach prawdopodobieństwa. Znane jest kilka metod generowania takich liczb. Wszystkie przedstawione tutaj będą opierały się na tym, że umiemy generować liczby z rozkładem $U(0,1)$. Szczególną uwagę poświęcimy generowaniu liczb z rozkładem $N(0,1)$. Jest to standardowy zapis oznaczający rozkład normalny (Gaussa) o średniej równej 0 i odchyleniu standardowym równym 1. Zanim omówimy pierwszą metodę, wcześniej zdefiniujemy sobie pojęcie *histogramu*. Będzie nam on potrzebny do wizualizacji rozkładów (czy raczej ich gęstości) z wygenerowanych liczb losowych.

Histogram

Wikipedia definiuje histogram jako jeden z graficznych sposobów przedstawiania rozkładu empirycznego cechy. Konstruuje się go jako szereg prostokątów odpowiadających liczebności elementów wpadających do określonego przedziału klasowego. Szerokości przedziałów klasowych mogą mieć stałe lub zmienne długości. W bardziej matematycznym sensie histogram to funkcja zliczająca ilość obserwacji pasujących do oddzielnych przedziałów klasowych. Jeżeli n oznacza liczbę wszystkich obserwacji, a k to liczba przedziałów, wtedy histogram m_i spełnia następujący warunek

$$n = \sum_{i=1}^k m_i$$

Ideę histogramu najlepiej zrozumieć na przykładzie. Mamy następującą listę liczb

```
l = [1, -3, -5, -1, -3, 1, 5, 1, 3, -3, 4, 2, 4, -1, 4, 5, -2, 4, 3, -4]
```

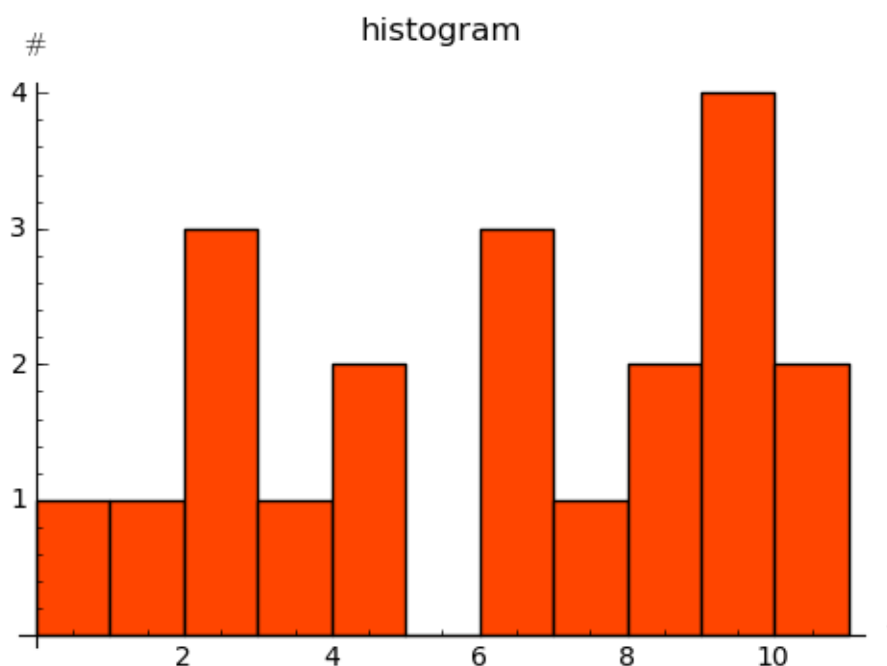
Budując histogram na początku musimy ustalić szerokość przedziału. Zaczniemy od łatwiejszej wersji: niech szerokość będzie stała. Najlepiej podzielić ową listę na przedziały zawierające liczby całkowite. W zasadzie wystarczy zliczać ile jest poszczególnych liczb całkowitych w liście `l`. Zróbmy to. Widzimy, że mamy

-5	-4	-3	-2	-1	0	1	2	3	4	5
1	1	3	1	2	0	3	1	2	4	2

W zasadzie mamy już nasz histogram. Jeżeli posumujemy ilość elementów listy (`len(l)`), oraz obliczymy n zobaczymy, że dostaniemy tą samą liczbę (=20). Pozostaje narysować ów histogram. Na odciętej musimy odłożyć przedziały klasowe a na rzędnej liczebności danego przedziału. Przyjęło się rysować histogram używając słupków. Sage na chwilę obecną posiada kilka metod narysowania takiego histogramu. Jeżeli nie zależy nam na poprawnym opisanu odciętej (np: chcemy tylko zobaczyć kształt histogramu), wystarczy napisać

```
h = [l.count(i) for i in range(-5,6)]
bar_chart(h, width=1, color="orangered").show(axes_labels=['$i$', '$\#$'], title=
```

Co pozwoli nam wygenerować taki rysunek:



Rysunek 3.2: Prosty wykres liczebności, gdzie wykorzystaliśmy funkcję `bar_chart()`.

Nie jest to prawdziwy histogram, bowiem odłożone na osi OY liczebności powinny odpowiadać rzeczywistym wartościom (przedziałom). Możemy skorzystać z pakietu `Time Series` dostępnego w Sage. Wystarczy prosta komenda aby uzyskać dostęp do wielu statystycznych funkcji typowych dla analizy szeregu czasowego.

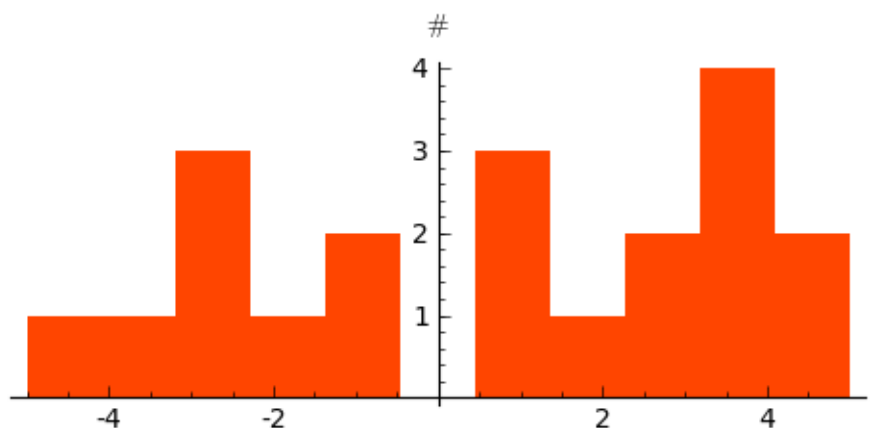
```
v = finance.TimeSeries(l)
```

I teraz aby obliczyć histogram dla 10 równych przedziałów (od minimalnej do maksymalnej wartości występującej w liście `l`), wystarczy napisać

```
v.histogram(bins=11)
```

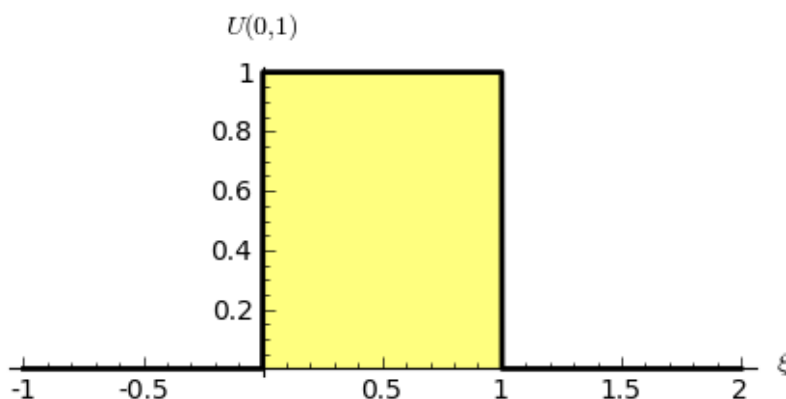
a aby narysować jego wykres

```
v.plot_histogram(bins=11, normalize=0, color="orangered", axes_labels=['$i$', '$\n$'])
```



Rysunek 3.3: Histogram dla listy `l` uzyskany z wykorzystaniem pakietu `TimeSeries`

Oczywiście całą procedurę można powtórzyć dla liczb zmiennoprzecinkowych (rzeczywistych, wymiernych). W tym wypadku należałoby oczywiście policzyć ile posiadanych liczb wpada do zdefiniowanych “pudełek”. Zobaczmy drugi przykład, gdzie obliczymy i narysujemy w Sage histogram dla dziesięciu tysięcy liczb z $U(0,1)$. Powinniśmy dostać



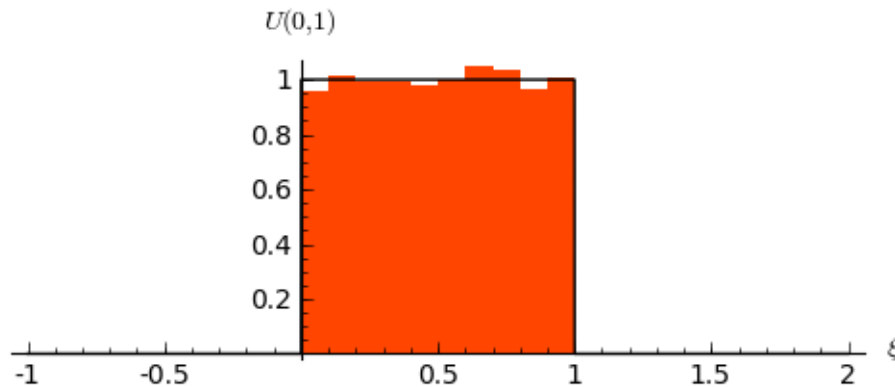
Rysunek 3.4: Wykres rozkładu $U(0,1)$

Przykład 2 Wygenerujemy 10000 liczb losowych a następnie dla nich obliczymy i narysujemy histogram.

```
N = 10000
u01 = [random() for i in xrange(N)]
fu01 = lambda x: 0 if x < 0 or x > 1 else 1
v = finance.TimeSeries([random() for i in xrange(N)])
```

```
plot1 = plot(fu01, (-1,2), thickness=1, color="black")
plot2 = v.plot_histogram(bins=10, color="orangered")
(plot1 + plot2).show(axes_labels=[r'$\xi$', r'$U(0,1)$'])
```

Ostatnia linia wyrysuje nam obie funkcje na jednym wykresie. Zachęcamy czytelnika do pok eksperymentowania z powyższym kodem - można zmienić liczbę prób N i łatwo zobaczyć jak histogram zaczyna oddalać się od teoretycznego rozkładu dla małych N i jak zbliża się dla dużych. Można też zobaczyć jak ilość przedziałów (parametr `bins`) wpływa na otrzymany histogram.



Rysunek 3.5: Wykres rozkładu $U(0,1)$ + histogram miliona prób.

Metoda inwersyjna

Każdy rozkład prawdopodobieństwa może być jednoznacznie scharakteryzowany poprzez pewną funkcję rzeczywistą zwaną **dystrybuantą**.

Dystrybuanta Niech \mathbb{P} będzie rozkładem prawdopodobieństwa. Funkcję $\mathbb{F} : \mathbb{R} \rightarrow \mathbb{R}$ daną wzorem

$$\mathbb{F}(\xi) = \mathbb{P}((-\infty, \xi))$$

nazywamy dystrybuantą rozkładu \mathbb{P} .

W metodzie inwersyjnej żądany rozkład o dystrybuancie \mathbb{F} uzyskuje się poprzez przekształcenie zmiennej losowej o rozkładzie $U(0,1)$ za pomocą funkcji odwrotnej do \mathbb{F} .

Twierdzenie Załóżmy, że dystrybuanta \mathbb{F} jest ściśle rosnąca. Jeśli zmienna losowa u ma rozkład $U(0,1)$ to $\mathbb{F}^{-1}(u)$ ma dystrybuantę \mathbb{F} .

Dowód TBA

Algorytm wykorzystujący powyższe twierdzenie jest bardzo prosty i wygląda następująco:

1. Generujemy liczbę $u \in U(0,1)$.
2. Przekształcamy u stosując

$$x = \mathbb{F}^{-1}(u)$$

Wynikowa liczba losowa x posiada żądany rozkład \mathbb{P} .

Oczywiście skuteczność tej metody zależy bezpośrednio od tego czy możemy łatwo obliczyć \mathbb{F}^{-1} . Jeżeli tak - jest to najprostsza znana metoda generowania liczb losowych z danym rozkładem. Do rozkładów, do których można zastosować tą metodę należą wszystkie rozkłady, których dystrybucja znana jest jawnie *oraz* można ją łatwo odwrócić. O takich rozkładach powiemy sobie niżej.

Rozkład wykładniczy

Przejdźmy wreszcie do generowania liczb losowych z rozkładem innym niż $U(0,1)$. Na początek weźmy jeden z najbardziej powszechnych, czy popularnych rozkładów prawdopodobieństwa - **rozkład wykładniczy**. Gęstość takiego rozkładu dana jest wzorem

$$f(\xi) = \lambda e^{-\lambda \xi}$$

Jak łatwo policzyć, dystrybucja $F(x)$ wynosi

$$F(x) = \int_{-\infty}^x f(\xi) d\xi = -e^{-\lambda \xi} \Big|_{-\infty}^x = -e^{-\lambda x} + 1,$$

a jej odwrotność

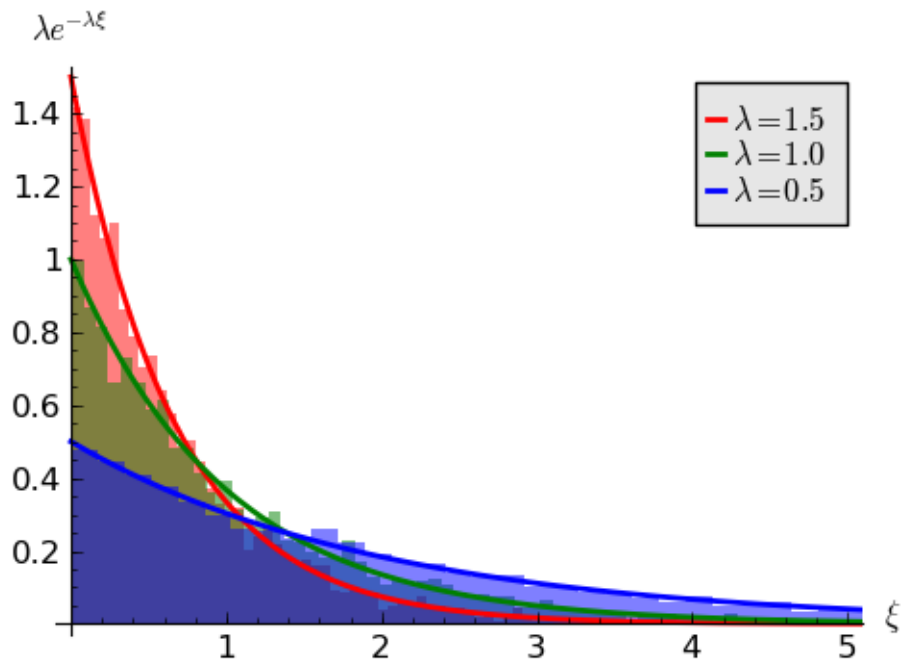
$$F^{-1}(u) = -\frac{1}{\lambda} \ln(1 - u).$$

Spróbujmy wygenerować 5000 liczb o rozkładzie wykładniczym. Następnie obliczymy sobie histogram, unormujemy go i porównamy z teoretycznym rozkładem dla kilku wartości $\lambda = 0.5, 1, 1.5$.

```
f(xi, a) = a * exp(-a * xi)
F(u, a) = -log(1-u)/a
N = 5000
kolor = ["red", "green", "blue"]
parlist = [1.5, 1, 0.5]
p = []
for par in parlist:
    lista = [F(random(), par) for i in xrange(N)]
    v = finance.TimeSeries(lista)
    P = v.plot_histogram(bins=100, color=kolor[parlist.index(par)], alpha=0.5)
    P.set_aspect_ratio("automatic")
    p.append(P)
    p.append(plot(f(xi,par), 0, max(lista), thickness=2, color=kolor[parlist.index(par)],
        legend_label=r'\xi, \lambda = %.1f$'%par))
sum(p).show(xmin=0, xmax=5, figsize=5, axes_labels=[r'\xi', r'\lambda e^{-\lambda \xi}'])
```

Jak widać na rysunku liczby losowe przekształcone metodą inwersji w oparciu o odwrotność dystrybucyjną, dość dobrze odwzorowują rozkład wykładniczy. Lepszy wynik można oczywiście uzyskać zwiększając parametry N oraz bins .

Alternatywnie można wykorzystać przekształcenie bazujące na spostrzeżeniu, że liczby $1 - u$ oraz $u \in U(0,1)$ posiadają ten sam rozkład $U(0,1)$.



Rysunek 3.6: Wykres gęstości rozkładu wykładniczego oraz histogram z 5000 prób liczb losowych dla trzech wartości parametru λ .

Rozkład Cauchy'ego

Rozkład ten dany jest wzorem

$$f(\xi) = \frac{\sigma}{\pi(\xi^2 + \sigma^2)}$$

Odwrotność dystrybuanty powyższego rozkładu wynosi

$$F^{-1}(u) = \sigma \tan \left[\pi \left(u - \frac{1}{2} \right) \right].$$

Stosując proste i bardzo naturalne przekształcenie oryginalnej zmiennej $v = 1/2 - u$ dostajemy nieco prostsze wyrażenie

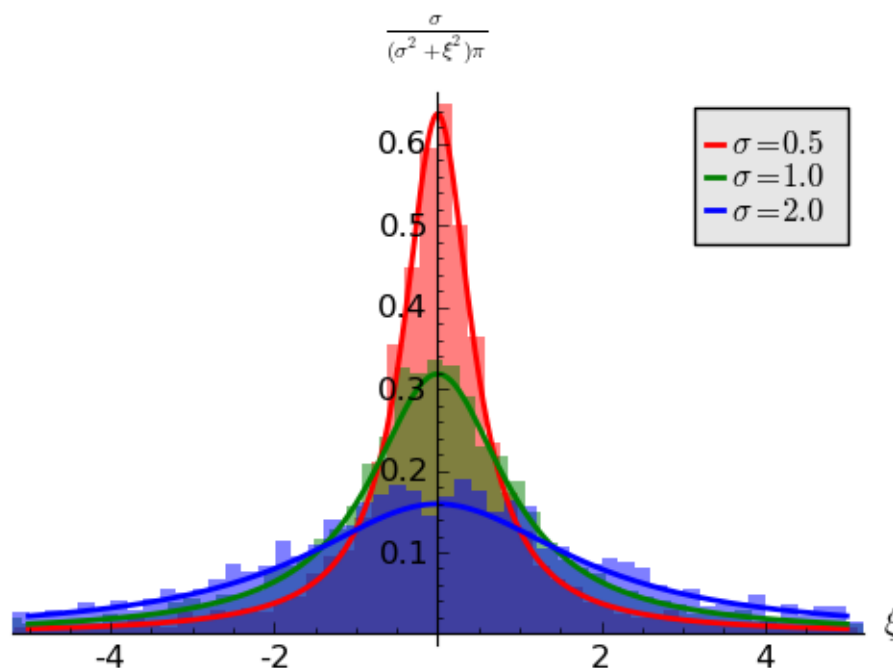
$$F^{-1}(v) = \sigma \tan(\pi v).$$

Stosując podobne metody jak w poprzednim rozdziale możemy sprawdzić, czy powyższe przekształcenie generuje liczby z odpowiednim rozkładem.

Rozkład logistyczny

Rozkład ten dany jest wzorem

$$f(\xi) = \frac{1}{2 + e^{\xi} + e^{-\xi}}$$



Rysunek 3.7: Wykres gęstości rozkładu Cauchy'ego oraz histogram z 5000 prób liczb losowych dla trzech wartości parametru $\sigma = 0.5, 1, 2$.

Odwrotność dystrybuanty powyższego rozkładu wynosi

$$F^{-1}(u) = \ln \frac{u}{1-u}.$$

Stosując podobne metody jak w poprzednim rozdziale możemy sprawdzić, czy powyższe przekształcenie generuje liczby z odpowiednim rozkładem.

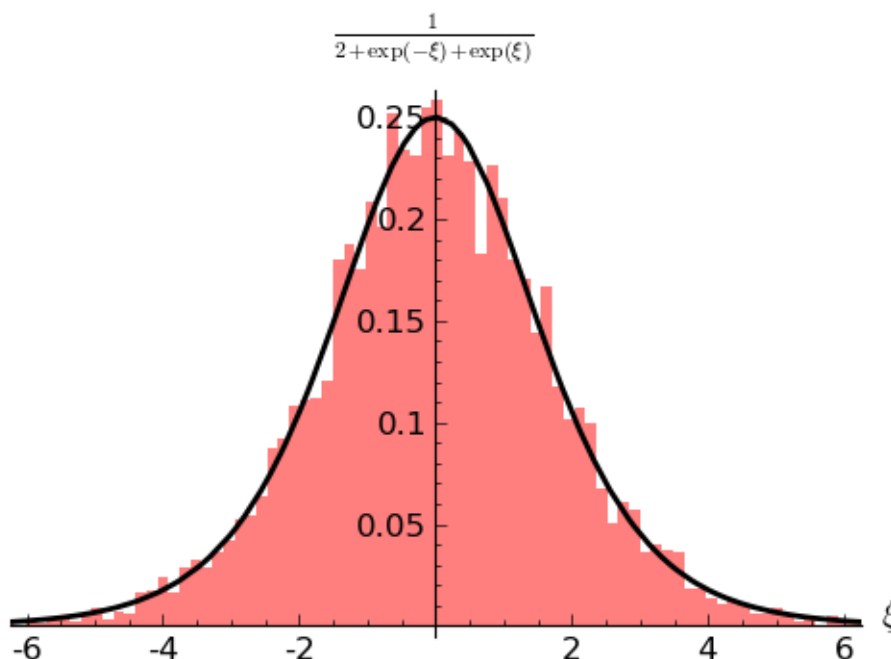
Zadanie 2.3.1 Wygeneruj 200000 liczb losowych z rozkładem Pareto. Narysuj ich unormowany histogram oraz funkcję gęstości. Porównaj obie funkcje zmieniając wszystkie parametry rozkładu.

Zadanie 2.3.2. Wygeneruj 1000 liczb losowych z rozkładem trójkątnym. Narysuj ich unormowany histogram oraz funkcję gęstości. Porównaj obie funkcje zmieniając wszystkie parametry rozkładu.

3.2 Równania różniczkowe zwyczajne

3.2.1 Metoda Eulera

Jest to bodaj najprostszy sposób na numeryczne rozwiązywanie równań różniczkowych zwyczajnych. Technicznie to metoda pierwszego rzędu. Bazuje na prostej interpretacji definicji



Rysunek 3.8: Wykres gęstości rozkładu logistycznego oraz histogram z 200000 prób liczb losowych.

pochodnej. Rozpatrujemy równanie postaci

$$\frac{dy}{ds} = y' = f(y, s),$$

z zadanymi warunkami początkowymi $(sx(0), y(0)) = (x_0, y_0)$. Stosując przekształcenie

$$\frac{dy}{ds} = \lim_{\Delta s \rightarrow 0} \frac{\Delta y}{\Delta s}$$

Na chwilę zapomnijmy o tej granicy. Mamy

$$\frac{\Delta y}{\Delta s} = \frac{y(s + \Delta s) - y(s)}{\Delta s}$$

stawiając krok “czasowy” $\Delta x = h$ (jak to zwyczajowo w symulacjach komputerowych) dostajemy

$$\begin{aligned} \frac{y(s + h) - y(s)}{h} &\simeq f(y, s), \\ y(s + h) &\simeq y(s) + f(y, s)h. \end{aligned}$$

Oczywiście pełna równość zachodzi tylko w granicy $h \rightarrow 0$. Na potrzeby numeryczne jednak nie musimy się tym przejmować. Należy mieć tylko świadomość, że zmniejszanie kroku czasowego zbliża nas do wyniku dokładnego (zazwyczaj). Jako, że w iteracjach “czasowych” wykorzystujemy stały krok h , więc w danej, powiedzmy i -tej iteracji czas rzeczywisty zastąpiony zostanie poprzez $s_i = i \cdot h$, możemy równie dobrze wprowadzić indeksowanie nie po czasie s ale po zmiennej iteracyjnej i

$$\begin{aligned} y(s = ih) &= y(s_i) = y_i, \quad s_i = ih, \\ y(ih + h) &= y(h(i + 1)) = y(ih) + f(y(ih), ih)h, \\ y_{i+1} &= y_i + f(y_i, s_i)h. \end{aligned}$$

Dostając tzw. schemat Eulera. Innymi metodami wyprowadzenia tego prostego schematu będzie użycie rozwinięcia Taylora lub wycałkowanie równania różnicowego od s_0 do $s_0 + h$ za pomocą metody prostokątów (używając tylko jednego prostokąta na całym przedziale całkowania).

Zadanie D1.1 Uzyskaj schemat Eulera z rozwinięcia Taylora.

Zadanie D1.2 Uzyskaj schemat Eulera całkując równanie różniczkowe.

Dokładność metody Eulera mocno zależy od wyboru kroku całkowania h . Musimy też na początku zadać warunki startowe (początkowe) iteracji, zupełnie jak podczas rozwiązywania równań metodami analitycznymi. Dla przykładu obliczmy numerycznie ruch przetłumionego sprężyny o współczynniku sprężystości k .

$$\dot{x}(t) = -kx(t).$$

Rozwiązaniem jest zanik eksponencjalny $x(t) = x_0 \exp(-kt)$. Wybierzemy sobie warunek początkowy $x_0 = 1$ i współczynnik sprężystości $k = 0.1$. Policzmy N iteracji i narysujmy wykres. Na początku napiszemy dyskretną wersję równania ruchu

$$x_{i+1} = x_i - kx_i h = x_i(1 - kh).$$

Teraz kod Sage

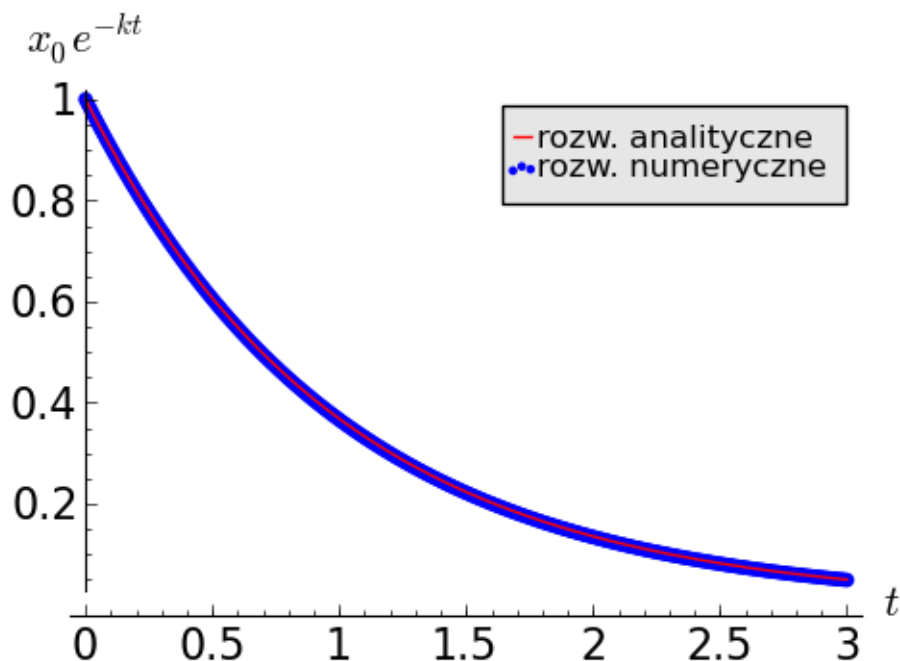
```
# liczba iteracji rownania
N = 300
# czas początkowy
t = 0
# skok czasowy
h = 0.01
# parametr rownania (sprezystosc)
k = 1
# wartosc początkowa x(t=0)
x_0 = 1
#inicializacja listy poprzez wartosc początkowa
lista_x = [x_0]
for i in xrange(N):
    lista_x.append(lista_x[i] * (1 - k*h))
```

wynik (rozwiązanie numeryczne równania) trzymany jest w liście `lista_x`. Wyrzysujmy ją standardowo, razem z wynikiem analitycznym.

Pomimo użycia prostej metody (pierwszego rzędu), wykresy wyglądają identycznie. No, ale czy na pewno jest tak pięknie? Poprawność metody możemy łatwo zbadać obliczając błędy względny (E_w) i bezwzględny (E_b).

$$E_b = \bar{y} - y,$$
$$E_w = \frac{\bar{y} - y}{\bar{y}} = \frac{E_b}{\bar{y}}.$$

gdzie y to wielkość obliczona algorytmem a \bar{y} to dokładna wielkość analityczna. Dla jasności - nie interesują nas w tym przypadku znaki błędów a jedynie ich wartość bezwzględna (tu proszę zwrócić uwagę na nomenklaturę, żeby nie pomylić wartości bezwzględnej z błędem



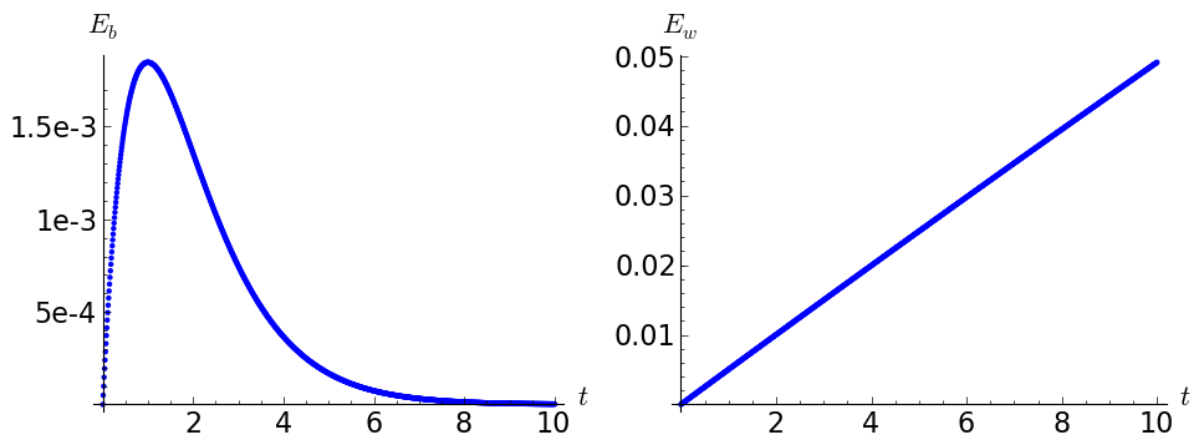
Rysunek 3.9: Rozwiązanie analityczne i numeryczne równania $\dot{x} = -kx$.

bezwzględny). Dlatego najczęściej oblicza się nie E_w a $|E_w|$. Wymaga to drobnej korekty powyższego kodu

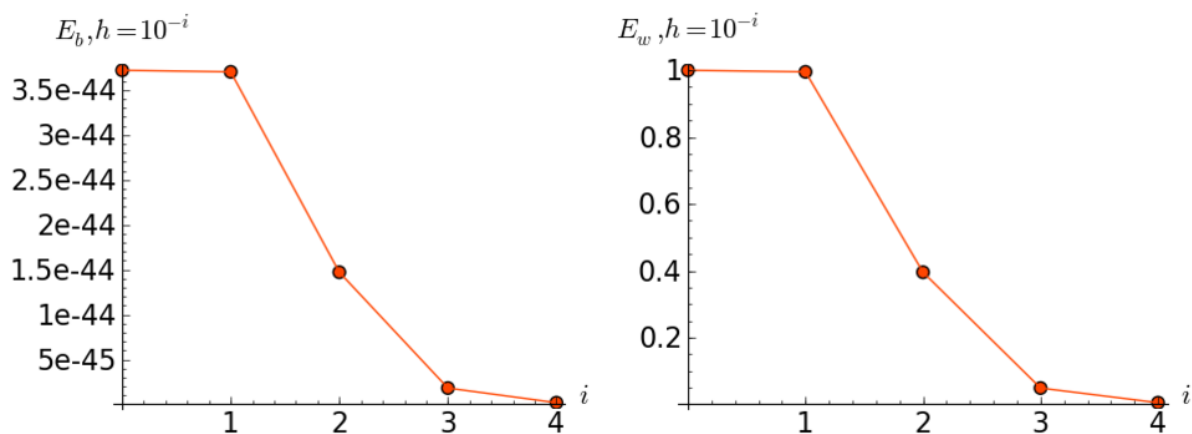
```
N, t, h, k, x_0 = 1000, 0, 0.01, 1, 1
g(s) = x_0*exp(-k*s)
lista_x = [x_0]
Eb = [g(0) - x_0]
Ew = [Ew[0]/g(0)]
for i in xrange(1,N):
    lista_x.append(lista_x[i-1] * (1 - k*h))
    Eb.append(abs(g(i*h) - lista_x[i]))
    Ew.append(Eb[i]/g(i*h))
list_plot(zip([i*h for i in xrange(N+1)],Eb)).show()
list_plot(zip([i*h for i in xrange(N+1)],Ew)).show()
```

Spójrzmy. Na pierwszym wykresie odłożony mamy błąd bezwzględny. Widzimy, że dla krótkich czasów odbiega on od wartości analitycznej by dla większych czasów zmaleć do zera. Mogą być tego 2 powody: (i) różnica pomiędzy obiema wartościami maleje zo zera lub (ii) obie wartości maleją do zera, więc ich różnica też. Jako, że funkcja jest eksponencjalna, dużo bardziej prawdopodobny jest ten drugi scenariusz. Aby zobaczyć, czy błąd rośnie z ilością iteracji (w czasie) wykreślimy błąd względny. Mówi on nam o stosunku błędu bezwzględnego do wartości analitycznej (rysunek po prawej). Tu jak widać rośnie on wraz z czasem, z czego możemy wywnioskować, że wraz z ilością iteracji coraz mniej dokładnie obliczamy wartość y .

Najprostsza metoda poprawienia jakości rozwiązań jest zmniejszenie kroku całkowania. Zależności pozostaną podobne, zmniejszy się jednak wartość błędów w danej chwili czasowej.



Rysunek 3.10: Błąd bezwzględny (po lewej) i błąd względny (po prawej).



Rysunek 3.11: Błąd bezwzględny (po lewej) i błąd względny (po prawej) po czasie $t = 10$ dla różnych kroków czasowych $h = 1, 0.1, 0.01, 0.001, 0.0001$.

W tabeli zawarto wartości błędów bezwzględnego i względnego dla różnych wielkości kroku czasowego symulacji, po osiągnięciu czasu końcowego $T_E = 10$. Widać, że pomimo, że za każdym zmniejszeniem kroku zwiększała się ilość kroków czasowych, dokładność obliczeń rosła - malał zarówno błąd bezwzględny jak i względny.

h	N	E_b	E_w
1	100	3.7210^{-44}	1
0.1	1 000	3.7010^{-44}	0.995
0.01	10 000	1.4710^{-44}	0.396
0.001	100 000	1.8210^{-45}	0.0488
0.0001	1 000 000	1.8610^{-46}	0.0499

Przejdźmy teraz do rozwiązywania równania różniczkowego wyższego stopnia. Znów posłużymy się przykładem oscylatora harmonicznego. Tym razem rozwiążemy równanie Newtona dla punktowej cząstki o masie m w potencjale $U(x) = -kx^2/2$. Pomińmy siły tarcia. Również w spokoju zostawimy wymuszenie.

$$m\ddot{x}(t) = -kx(t).$$

Równanie to posiada znane **analityczne rozwiązanie**. Oznaczając $\omega_0^2 = k/m$ dostajemy

$$x(t) = A \sin(\omega_0 t) + B \cos(\omega_0 t).$$

Stałe A , B (amplitudy) zależne są od wyboru warunków początkowych. Spróbujmy numerycznie rozwiązać równanie ruchu tak, aby pokazać zgodność z rozwiązaniem. Aby napisać schemat Eulera dla równania drugiego stopnia najpierw trzeba przepisać równanie do układu równań na x i $v = \dot{x}$.

$$\begin{aligned}\dot{x}(t) &= v(t), \\ \dot{v}(t) &= -\frac{k}{m}x(t) = -\omega_0^2 x(t).\end{aligned}$$

Teraz wystarczy zdyskretyzować te równania, tak samo jak robiliśmy to z równaniem pierwszego rzędu.

$$\begin{aligned}x_{i+1} &= x_i + v_i h, \\ v_{i+1} &= v_i - \omega_0^2 x_i h.\end{aligned}$$

Po ustaleniu warunków początkowych $x(0) = x_0$ oraz $v(0) = v_0$ możemy rozpocząć normalną procedurę symulacji - wybieramy krok czasowy h , ustalamy parametry równania i do dzieła.

```
h = 0.01 # skok
N = 100 # liczba krokow

x0 = 1
v0 = 0
omega0 = 1

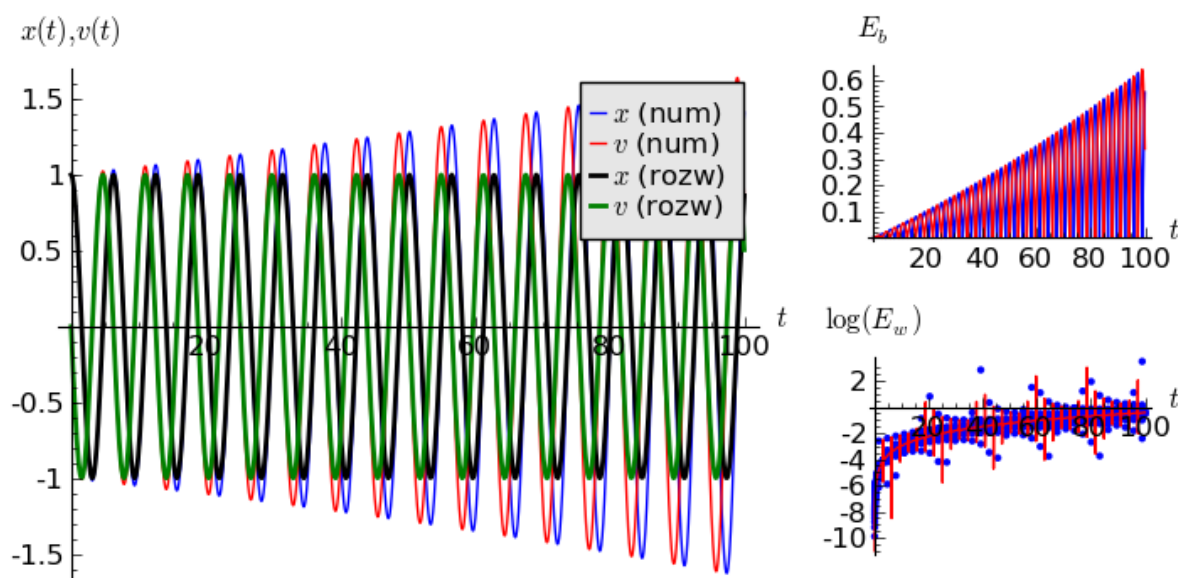
lista_x = [x0]
lista_v = [v0]
lista_t = [0]
for i in xrange(N) :
```

```
lista_x.append(lista_x[i] + lista_v[i] * h)
lista_v.append(lista_v[i] - omega0^2 * lista_x[i] * h)
lista_t.append(lista_t[i] + h)
```

Wykorzystamy też Sage do obliczenia rozwiązania analitycznego dla naszego zagadnienia.

```
var('t x omega x_0 v_0')
x = function('x', t)
assume(omega>0)
eq = diff(x, t, 2) + omega^2 * x == 0
solx = desolve(eq, x, ivar=t, ics=[0, x_0, v_0])
solv = diff(solx,t)
```

Teraz możemy zobaczyć jak dokładna jest metoda Eulera w przypadku równań wyższych rzędów. Poniżej znajdziecie wykres rozwiązań dla $h=0.01$ i 10000 kroków.



Rysunek 3.12: Porównanie numerycznego ($x(t)$ linia niebieska, $v(t)$ linia czerwona) i analitycznego ($x(t)$ linia czarna, $v(t)$ linia zielona) rozwiązania zagadnienia oscylatora harmonicznego. Jak widać odchylenia od rozwiązań dokładnych zaczynają być znaczące już dla kilku kroków symulacji. Błąd bezwzględny widnieje na prawym górnym panelu; błąd względny wykreślony jest na prawym dolnym panelu w skali logarytmicznej dla lepszej czytelności. Parametry użyte dla powyższej symulacji: $x_0 = 1$, $v_0 = 0$, $\omega_0 = 1$, $h = 0.01$, $N = 10000$.

Inną, aczkolwiek trudniejszą metodą będzie użycie algorytmów wyższego rzędu takich jak schemat Rungego-Kutty (2-giego, 4-tego i wyższych rzędów). Dociekliwy student może zajrzeć [tutaj](#).

Zadanie D1.3 Przeprowadź podobną symulację dla innych wartości h . Wykreśl zależność błędów względnego i bezwzględnego w funkcji wartości h . Błędy badaj po rzeczywistym czasie symulacji $T_E = 100$.

Zadanie D1.4 Rozwiązać numerycznie równania

- $\dot{x}(t) = -kx^3$

2. $\dot{x}(t) = F$
3. $m\ddot{x}(t) = mg$
4. $\ddot{r}(t) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right], r > 0$
5. $m\ddot{x}(t) = -\gamma\dot{x}(t) - kx(t)$
6. $m\ddot{x}(t) = -kx(t) - \gamma\dot{x}(t) + a \cos(\omega t)$

Dla każdego przypadku (A-F) należy

1. narysować x (dla D-F również v) w funkcji t (opisać osie),
2. odpowiedzieć na pytanie: z jakim ruchem mamy do czynienia [dla jakich parametrów równania ruch jest cykliczny (periodyczny), dla jakich rozwiązanie jest stałe (niezmienne w czasie)...],
3. znaleźć błąd względny i bezwzględny, wykreślić w funkcji czasu.

Pytania:

1. Czym różni się przypadek E od F?
2. Co opisuje potencjał w D? Jakie ma zastosowanie w fizyce?

3.3 Stochastyczne równania różniczkowe

Stochastyczne równania różniczkowe (SDE, od angielskiej nazwy Stochastic differential equations) są obecnie uważane za standardowe narzędzie wykorzystywane do analizy niektórych wielkości opisujących dynamikę rynków finansowych. Do tych wielkości należą ceny aktywów, stopy procentowe czy ich pochodne. W przeciwieństwie do zwyczajnych równań różniczkowych, które posiadają jendoznaczne rozwiązanie, rozwiązaniami SDE są ciągłe w czasie procesy stochastyczne. Metody komputerowe wykorzystywane do analizy SDE bazują na klasycznych metodach wykorzystywanych do rozwiązywania tradycyjnych, deterministycznych równań różniczkowych, są jednak uogólnione tak, aby radzić sobie z procesami losowymi.

Zestaw zmiennych losowych X_t indeksowanych liczbami rzeczywistymi t nazywamy procesem losowym ciągłym (ze względu na czas). Każda *realizacja* procesu losowego to przypadkowa wartość zmiennej losowej X_t dla każdego t , jest więc funkcją czasu. Co ciekawe, *każda* deterministyczna funkcja $f(t)$ może być uważana za proces stochastyczny, którego wariancja znika.

Najbardziej znanym przykładem procesu losowego szeroko występującego w modelach fizyki, chemii ale i rynków finansowych jest *proces Wienera* $W(t) = W_t$, ciągły proces stochastyczny posiadający następujące własności

1. jest to proces rzeczywisty,
2. startuje z zera ($W_0 = 0$),
3. ma stacjonarne i niezależne przyrosty na nieprzekrywających się przedziałach,

4. jest procesem Gaussa o zerowej wartości średniej $\langle W_t - W_s \rangle = 0$ i wariancji przyrostów $\langle [W_t - W_s]^2 \rangle = 2D(t - s)$,
5. proces Wienera może być reprezentowany ciągłymi trajektoriami.

Wynika z tego, że dla każdej różnicy czasów $t - s$ zmienna losowa $W_t - W_s$ jest zmienną losową gaussowską o zerowej wartości średniej i wariancji $2D(t - s)$. Więc jego rozkład prawdopodobieństwa ma postać

$$f_{W_t - W_s}(x) = \frac{1}{\sqrt{2\pi D(t - s)}} \exp \left[-\frac{x^2}{4D(t - s)} \right].$$

Proces taki może być wyprowadzony jako proces graniczny błędzenia przypadkowego. Wystarczy tylko zbadać granicę dla której wielkość skoku i czas pomiędzy skokami będą maleć do zera. Tak zdefiniowanym procesem posługujemy się zwyczajowo, gdy podczas analizy problemu pojawia się jakaś nieregularna siła czy zaburzenie którego nie możemy opisać równaniami deterministycznymi.

Typowe dla rynków finansowych *równanie dyfuzji* może być modelowane przez równanie różniczkowe posiadające część deterministyczną zwaną **dryfem** oraz część losową zwaną **dyfuzją**. Ta ostatnia jest bardzo często reprezentowana właśnie przez proces Wienera. Możemy sobie napisać ogólne równanie

$$dX = a(t, X)dt + b(t, X)dW_t.$$

Jest to postać różniczkowa. W zwykłych równaniach różniczkowych zazwyczaj stosujemy pochodne dx/dt . W tym przypadku postać różniczkowa ma większy sens, jako, że wiele interesujących nas procesów losowych (jak ruch Browna) są procesami ciągłymi aczkolwiek nie są różniczkowalne. Powyższe równanie nabiera większego sensu pod znakiem całki

$$X(t) = X(0) + \int_0^t a(s, y)ds + \int_0^t b(s, y)dW_s.$$

Ostatni wyraz z prawej zwany jest całką Ito.

3.3.1 Równanie Blacka-Scholesa

Jednym z bardziej znanych, historycznym już równaniem stochastycznym, jest równanie opisujące geometryczny ruch Browna

$$dX = \mu X dt + \sigma X dW_t,$$

gdzie μ, σ to wielkości stałe. Równanie to jest jednym z podstawowych elementów modelu wyceny opcji Blacka-Scholesa. Teoria ta została nagrodzona Nagrodą Nobla z ekonomii w roku 1997, a opracowana przez absolwenta fizyki i doktora matematyki Fischera Blacka oraz ekonomistę Myrona Scholesa. Teoria Blacka-Scholesa pozwala na wycenę wartości tzw. finansowych instrumentów pochodnych, czyli opcji, oraz służy do optymalizacji “bezpiecznego” portfela inwestycyjnego.

Pomimo tego, że równanie to wygląda na proste, żeby nie powiedzieć zbyt proste na to by opisywać jakkolwiek rzeczywistość na rynkach finansowych, ma ono olbrzymie znaczenie,

jako, że może być rozwiązane ściśle dając wynikowy wzór z którego możemy wyliczyć zmianę cen prostych opcji. Jak już powiedzieliśmy, rozwiązanie jest geometrycznym ruchem Browna

$$X(t) = X_0 \exp \left[\left(\mu - \frac{\sigma^2}{2} \right) t + \sigma dW_t \right].$$

Dzięki zamkniętej postaci rozwiązania mamy możliwość testowania metod numerycznych, które przedstawimy poniżej.

3.3.2 Schemat Eulera-Maruyamy

Najprostszą metodą numerycznego rozwiązywania równań różniczkowych zwyczajnych jest metoda Eulera. Bazuje ona np. na rozwinięciu Tylora w pierwszym rzędzie przybliżenia. Stochastycznym analogiem tej metody jest metoda Eulera-Maruyamy.

Będziemy chcieli podać przybliżone rozwiązanie ogólnej postaci SDE na przedziale czasowym $t \in [t_0, t_E]$. Na początku zdyskretyzujemy sobie ów przedział czasowy, ustalając na siatkę N punktów

$$t_0 < t_1 < t_2 < \dots < t_{N-2} < t_E.$$

Dążymy do tego, aby na tej siatce znaleźć przybliżone wartości zmiennej X . Oznaczmy je

$$w_0 < w_1 < w_2 < \dots < w_{N-2} < w_E.$$

Są to oczywiście przybliżone rozwiązania zmiennej x dla odpowiednich czasów z powyższej siatki $\{t_i\}$. Zakładając wartość początkową dla SDE $X(t_0) = X_0$ możemy pokusić się o rozwiązanie numeryczne w następującej postaci

$$\begin{aligned} w_0 &= X_0 \\ w_{i+1} &= w_i + a(t_i, w_i) \Delta t_{i+1} + b(t_i, w_i) \Delta W_{i+1} \\ \Delta t_{i+1} &= t_{i+1} - t_i \\ \Delta W_{i+1} &= W(t_{i+1}) - W(t_i). \end{aligned}$$

Kluczową sprawą w tym punkcie jest problem: jak zamodelować ΔW_i ? Mając do dyspozycji generator liczb losowych z rozkładem $N(0, 1)$ każdą losową liczbę ΔW_i obliczamy ze wzoru

$$\Delta W_i = \sqrt{\Delta t_i} z_i,$$

gdzie z_i jest losowana właśnie z $N(0, 1)$. Aby scałkować proces stochastyczny użyjemy formuły na przyrost procesu Wienera

$$\int_{t_i}^{t_{i+1}} \Gamma(t) dt = \int_{t_i}^{t_{i+1}} dW(t) = W(t_{i+1}) - W(t_i)$$

Z definicji procesu Wienera wiemy, że jest on procesem Gaussa o zerowej średniej i wariancji liniowej w czasie $\langle [W(t_{i+1}) - W(t_i)]^2 \rangle = 2D\Delta t_i$, co daje nam w sensie średnio-kwadratowym $\Delta W \propto \sqrt{\langle [\Delta W(t)]^2 \rangle} = \sqrt{\Delta t_i}$. Scałkowanie procesu Wienera prowadzi do

$$\int_{t_i}^{t_{i+1}} dW_t = \sqrt{\Delta t_i} z_i.$$

Jeżeli założymy sobie, że krok czasowy (odległości na siatce rozwiązań) jest stały i wynosi $\Delta t_i = h$ możemy napisać schemat explicite

$$w_{i+1} = w_i + ha(t_i, w_i) + \sqrt{hb}(t_i, w_i)z_i.$$

Jako, że każdy zestaw wartości $\{w_i\}, i = 0, \dots, E$ wyprodukowany przez powyższą formułę będzie przybliżonym rozwiązaniem procesu losowego, to i każda realizacja (każdy zestaw) będzie również losowa, a co za tym idzie - każda realizacja procesu będzie inna.

3.3.3 Schemat Millsteina

Dodaje on poprawkę do poprzedniego rozwiązania, powodując, że schemat staje się schematem pierwszego rzędu w sensie silnym. Dany jest on wzorem iteracyjnym

$$w_0 = X_0$$

$$w_{i+1} = w_i + a(w_i, t_i)h - \frac{h}{2}b(x_i, t_i)\frac{\partial b'}{\partial x}(w_i, t_i)(z_i^2 - 1) + \sqrt{hb}(w_i, t_i)z_i.$$

Obie metody (Millsteina i Eulera-Maruyamy) redukują się do tego samego schematu gdy część losowa nie jest zależna od zmiennej x . Jeżeli zależność istnieje, schemat Millsteina będzie szybciej zbieżny od schematu EM.

3.4 Przykłady całkowania procesów stochastycznych

3.4.1 Proces dyfuzji

Jest to prawdopodobnie najprostszy proces stochastyczny wykorzystujący biały szum gaussowski jako proces losowy. Przez matematyków nazywany jest po prostu procesem Wienera ponieważ prawa strona równania ruchu zawiera tylko i wyłącznie ów proces. Z drugiej strony jest obok procesu Poissona najważniejszym procesem losowym na bazie którego można zdefiniować całą rodzinę procesów losowych o ciągłych realizacjach. Równanie to można przedstawić używając równania Ito

$$dx(t) = \sqrt{2D}dW(t).$$

Realizacja jest funkcją ciągłą, ale nigdzie nieróżniczkowalną (jako że pochodna procesu Wienera nie istnieje). Za pomocą znanego już schematu Eulera-Maruyamy (EM) możemy sobie wygenerować pojedynczą realizację takiego procesu. Parametr D reguluje natężenie szumu.

$$x_0 = 0$$

$$x_{i+1} = x_i + \sqrt{2hD}N(0, 1).$$

Wiemy, że dla procesu Wienera $W(0) = 0$, wystartujemy więc z $x(0) = 0$. Weźmy krok $h = 0.01$ i 5000 kroków czasowych. Dla przejrzystości weźmiemy natężenie szumu $D = 1$. Jako, że wiemy jak generować zmienne z rozkładem $N(0, 1)$ użyjemy sobie “symbolicznego” oznaczenia na funkcję zwracającą takie zmienne. Funkcję taką nazwiemy `std_norm`. Konkretna realizacja takiej funkcji może odbywać np: poprzez algorytm Boxa-Mullera. Funkcja ta będzie przy wywołaniu zwracała jedną liczbę losową z $N(0, 1)$.

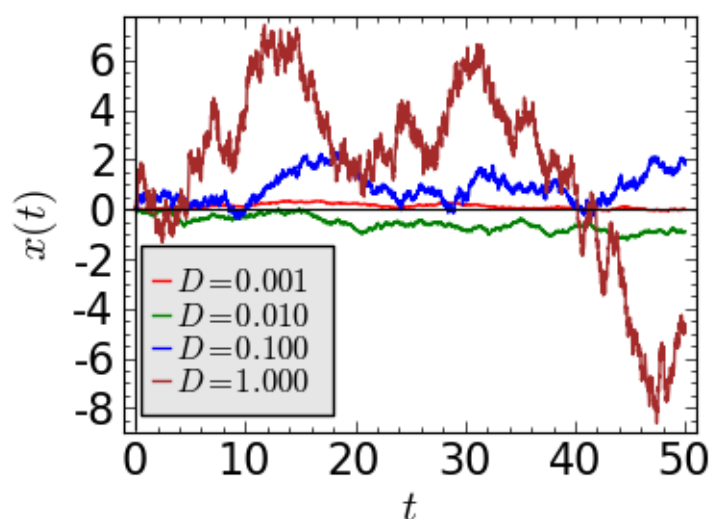
```

h = 0.01
N = 5000
x0 = 0
D = 1

x = [x0]
for i in xrange(1,N):
    n01 = std_norm()
    x.append(x[i-1] + sqrt(2*h*D) * n01)

```

Teraz narysujmy sobie takie realizacje dla kilku różnych wartości parametru D .



Rysunek 3.13: Proces dyfuzji dla kilku różnych wartości parametru D .

Na pierwszy rzut oka trajektorie (czy realizacje, przebiegi...) wyglądają kompletnie inaczej. Dla małych wartości D krzywe są bardziej regularne niż dla tych parametryzowanych przez większe wartości D , dla których to wykres jest mocno poszarpany i nieregularny. Jeżeli jednak narysowalibyśmy je osobno, nie oznaczając osi, identyfikacja byłaby niemożliwa - nie widzimy bowiem relacji pomiędzy wartościami (przyrostami).

Jako, że rozwiązanie równania dyfuzji znane jest od dawien dawna, możemy potestować na ile dokładnie rozwiązujemy nasze równanie. Oczywiście jedyne co możemy określić, to rozkład gęstości prawdopodobieństwa. Zakładając warunki początkowe

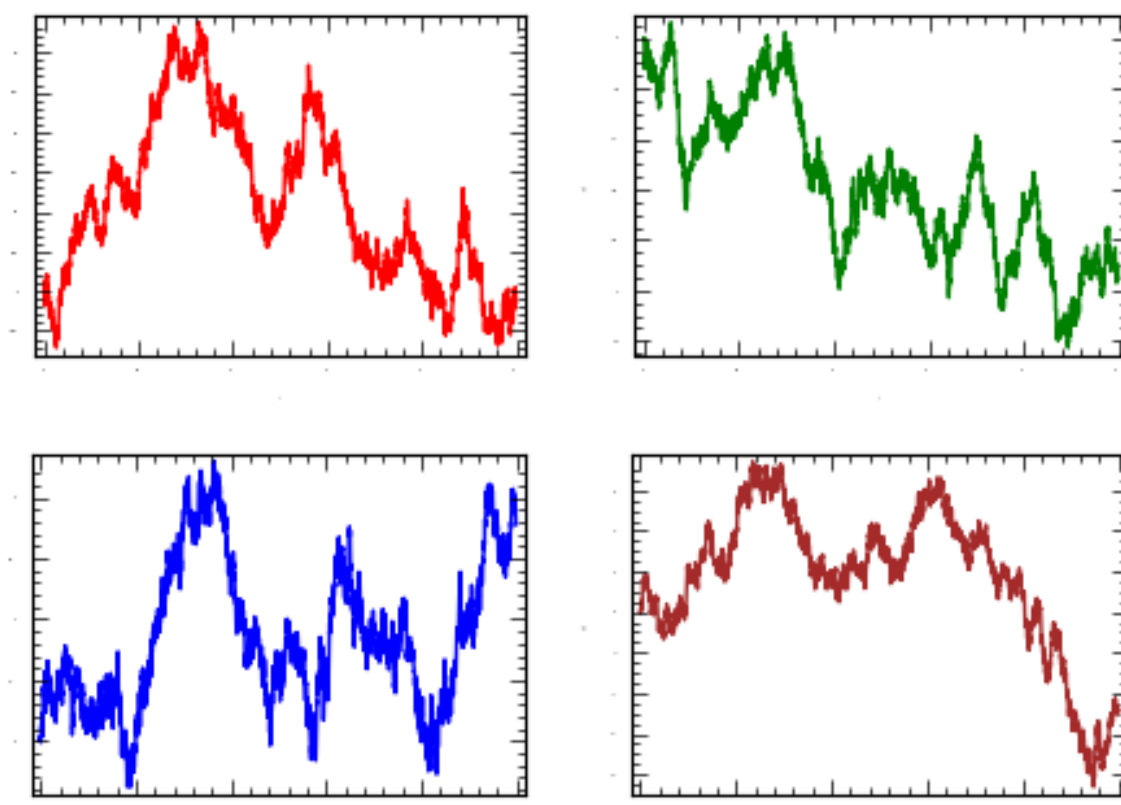
$$P(x, 0) = \delta(x)$$

co oznacza, że cały zespół cząstek podlegających dyfuzji wystartuje z $x(0) = 0$, możemy podać odpowiedź, bazując na równaniu dyfuzji

$$\frac{\partial P(x, t)}{\partial t} = D \frac{\partial^2 P(x, t)}{\partial x^2}.$$

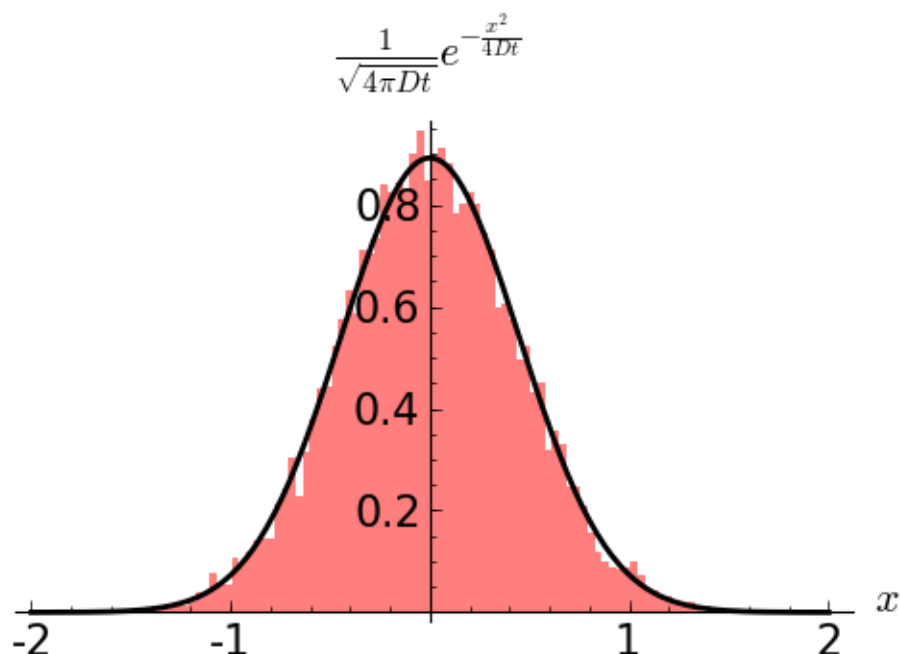
Wynikiem jest rozkład Gaussa postaci

$$P(x, t) = \frac{1}{\sqrt{4\pi Dt}} \exp \left[-\frac{x^2}{4Dt} \right].$$



Rysunek 3.14: Proces dyfuzji dla kilku różnych wartości parametru D . Są to te same przebiegi co w poprzednim wykresie. Kolory tu użyte odpowiadają kolorom z wykresu poprzedniego.

w którym wariancja rośnie liniowo z czasem a średnia jest równa zero. Zobaczmy, czy jesteśmy w stanie zweryfikować powyższy wzór numerycznie. Postaramy się znaleźć histogram pozycji 10000 cząstek po 100 krokach i porównamy go z powyższym wynikiem. Założymy sobie $D = 0.1$ i krok czasowy $h = 0.01$. Oznacza to, że po 100 krokach symulacji osiągniemy rzeczywisty czas równy 1.



Rysunek 3.15: Proces dyfuzji po 100 krokach symulacji dla $D = 0.1$, $h = 0.01$, $N = 100$ a co za tym idzie $t = Nh$.

Zadanie 6.1.1 Wykonaj w Sage powyższy rysunek. Metodę generowania histogramów znajdziesz w pierwszej części skryptu. Listę położeń, którą należy podać do histogramu wygeneruj w podobny sposób jak na początku tego rozdziału.

3.4.2 Dyfuzja ze stałym dryfem

Proces taki otrzymuje się bezpośrednio jako graniczny przypadek niesymetrycznego błędzenia przypadkowego (polecam poczytać [ten podrozdział](#)). Równanie opisujące ten proces ma postać zbliżoną do poprzedniego, bogatsze jest jednak o dodatkowy czynnik, zwany dryfem

$$\frac{\partial P(x, t)}{\partial t} = -V \frac{\partial P(x, t)}{\partial x} + D \frac{\partial^2 P(x, t)}{\partial x^2}.$$

Rozwiązaniem równania dyfuzji z dryfem, z takim samym warunkiem początkowym jak poprzednio (wszystkie cząstki, bądź realizacje zaczynają z tego samego położenia $x(0) = 0$ jest następująca funkcja

$$P(x, t) = \frac{1}{\sqrt{4\pi Dt}} \exp \left[-\frac{(x - Vt)^2}{4Dt} \right].$$

Jest to funkcja Gaussa opisująca zmienne losowe normalne, a dwa pierwsze momenty wynoszą odpowiednio $\xi(t) = Vt$ oraz $\sigma_\xi^2 = 2Dt$. Łatwo zauważyć, że zarówno średnia jak i wariancja

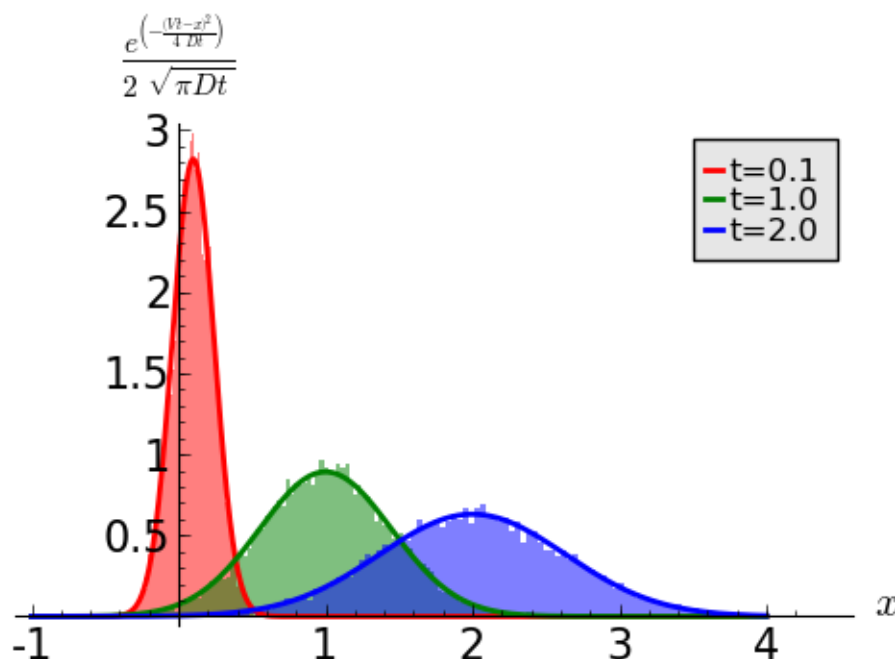
zależne są liniowo od czasu. Ponadto wariancja jest identyczna jak w procesie dyfuzji bez dryftu. Dryft ów stały można z punktu widzenia fizyki rozumieć jako stałą siłę przyłożoną do cząstki (coś na kształt cząstki umieszczonej na równi pochyłej) - położenie cząstki rośnie liniowo z czasem (jak w ruchu jednostajnie przyspieszonym), ale fluktuacje rosną w czasie jak pierwiastek \sqrt{t} .

Podobną analizę numeryczną jak poprzednio możemy przeprowadzić i tutaj. Tym razem, wykreślimy sobie stroboskopowo histogramy położenia po kilku krokach: $N = 10, 100, 200$. Po lekkiej modyfikacji numeryczny schemat EM będzie wyglądał tak

```
h = 0.01
N = 5000
x0 = 0
V = 1
D = 1

x = [x0]
for i in xrange(1, N):
    n01 = std_norm()
    x.append(x[i-1] + V*h + sqrt(2*h*D) * n01)
```

Teraz wystarczy zobaczyć, czy histogramy położenia po czasie $t=0.1, 1, 2$ będą odpowiadały obliczonej powyżej funkcji rozkładu.



Rysunek 3.16: Proces dyfuzji ze stałym dryftem po 10, 100 i 200 krokach symulacji dla $D = 0.1$, $h = 0.01$, a co za tym idzie $t = 0.1, 1, 2$.

Możemy policzyć sobie teraz średnie, odchylenie standardowe oraz błędy względny i bezwzględny przybliżenia dokładnych rozwiązań procesu dyfuzji z dryftem.

czas		teoria	symulacje	E_b	E_w [%]
$t=0.1$ std	średnia 0.1414	0.10 0.1413	0.09972 0.00007544	0.0002766 0.05335	0.2766
$t=1$ std	średnia 0.4472	1.0 0.444	1.005 0.003026	0.005345 0.6766	0.5345
$t=2$ std	średnia 0.6324	2.0 0.6362	2.001 0.003708	0.001460 0.5863	0.07302

Jak widzimy błędy bezwzględne dochodzą do około pół punktu procentowego różnicy dla 10000 realizacji. Zwiększenie próby spowoduje jeszcze lepsze dopasowanie, zmniejszenie spowoduje większe odchylenia od wartości rzeczywistych.

Zadanie 6.1.2 Oblicz błędy przybliżenia rozwiązania problemu dyfuzji ze stałym dryftem dla 10, 100, 500 i 1000 różnych realizacji. Zestawienia podaj w tabeli.

3.4.3 Proces Ornsteina-Uhlenbecka

3.4.4 Równanie Blacka-Scholesa

3.5 Lapunov

3.6 Bifurkacje

3.7 ARMA

Analizując dane pochodzące z szeroko pojętych źródeł rynków finansowych dość często korzysta się z Modeli Autoregresyjnych ze Średnią Kroczącą. Angielska nazwa takich modeli to Autoregressive Moving Average Models i stąd nazwa ARMA. Zwykle procedura polega na pewnej analizie posiadanych danych i dopasowaniu do nich parametrów takiego modelu. Takim zagadnieniem zajmuje się [Analiza Szeregów Czasowych](#). Szeregów dostarcza “samo życie”, czyli np: giełda. Jako, że ten kurs ma na celu modelowanie rynków, czyli ma *generować* takie szeregi, pobawimy się teraz w symulowanie rynku.

Model ARMA to nic innego jak układ równań różnicowych ze stałymi współczynnikami.