# *Zbiór zadań z modelowania matematycznego w środowisku Wolfram Mathematica* [1].

*Konrad Kisiel & Grzegorz Siudem*

*13 marca 2017*

[1] Złożono w klasie `tufte-handout`, `www.ctan.org/pkg/tufte-latex`

Przedstawiamy zbiór zadań i problemów związanych z dynamiką modeli matematycznych. Wersja ciągle robocza. Będziemy wdzięczni za wszelkie uwagi.

## *Spis treści*

*Materiały wprowadzające* 2

*Definicje i oznaczenia* 2

*Mathematica* 6

*Wymagania formalne* 8

*Obroty* 9

*Plan badań* 9

*Pytania i problemy* 9

*Rodzina kwadratowa* 13

*Plan badań* 13

*Pytania i problemy* 14

*Ergodyczność* 18

*Plan badań* 18

*Pytania i problemy* 19

*Równanie Van der Pola* 24

*Plan badań* 24

*Pytania i problemy* 25

## *Materiały wprowadzające*

Poniżej zamieszczamy garść najważniejszych definicji stosowanych podczas zajęć, krótki opis środowiska Wolfram Mathematica, a na zakończenie, opisujemy formalne wymagania, jakie będziemy stawiać wynikom prac domowych.

Zwracamy uwagę, że każda pojawiająca się w tekście nazwa mathematicznej funkcji (np. `D[]`) jest hiperłączem, po kliknięciu w które można przeczytać dokumentację tej funkcji na stronie reference.wolfram.com/language/

**Zalecamy przeczytać cały rozdział przed wykonywaniem zadań.**

### *Definicje i oznaczenia*

Niech $f\colon X \to X$ będzie pewną funkcją, a $X$ (w zasadzie dowolnym) zbiorem. Mówiąc o dynamice zadanej przez funkcję $f$ mamy na myśli iteracje tej funkcji, czyli jej wielokrotne złożenia

$$f^n(x) = \underbrace{f \circ f \circ \cdots \circ f}_{n \text{ razy}}(x) = \underbrace{f(f(f(\dots f}_{n \text{ razy}}(x)\dots))).$$

Aby lepiej opisywać takie złożenia stosujemy wyjaśnijmy poniższe terminy.

W Mathematice złożenie funkcji otrzymamy dzięki funkcji `Nest[]`.

**Definicja 1** **Orbitą** *funkcji $f$ w punkcie $x \in X$ nazywamy zbiór*

$$\mathcal{O}(f,x) = \{x, f(x), f^2(x), f^3(x), \dots\}.$$

*Gdy nie prowadzi to do nieporozumień orbitą będziemy także nazywali skończone podzbiory zbioru $\mathcal{O}(f,x)$ postaci*

$$\{x, f(x), f^2(x), f^3(x), \dots, f^N(x)\}.$$

Dzięki temu będziemy mogli pisać o tym, że funkcja `NestList[]` generuje orbity.

**Definicja 2** *Orbitę $\mathcal{O}(f,x)$ nazywamy* **okresową** *o okresie $k$ jeśli*

$$\forall_{n\in\mathbb{N}} \;\; f^{k+n}(x) = f^n(x).$$

**Definicja 3** *Gdy $k$ w poprzedniej definicji przyjmuje wartość $k = 1$, wówczas punkt $x$ nazywamy* **punktem stałym**, *gdyż*

$$f(x) = x.$$

W poszukiwaniu punktów stałych pomoże funkcja `FixedPoint[]`.

**Definicja 4** *Punkt $x_p \in X$ nazywamy* **punktem preokresowym** *jeśli $\mathcal{O}(f, f(x_p))$ jest orbitą okresową.*

W kolejnych definicjach zakładamy, że $(X, \mathcal{B}, \mu)$ jest przestrzenią z miarą.

**Definicja 5** *Mówimy, że* **przekształcenie** $T : X \to X$ **zachowuje miarę** *$\mu$ gdy*

$$\forall B \in \mathcal{B}, \qquad \mu\left(T^{-1}(B)\right) = \mu(B).$$

**Definicja 6** *Przekształcenie T nazwiemy* **ergodycznym** *jeśli zachowuje miarę i ponadto spełnia warunek*

$$\forall B \in \mathcal{B}, \qquad T^{-1}(B) = B \implies (\mu(B) = 0 \quad \vee \quad \mu(X \setminus B) = 0).$$

**Twierdzenie 1 (Twierdzenie Birkhoffa)** *Niech $T : X \to X$ będzie przekształceniem ergodycznym na przestrzeni $(X, \mathcal{B}, \mu)$. Załóżmy dodatkowo, że $\mu(X) < \infty$. Wówczas dla dowolnej funkcji $f \in L^1(\mu)$ zachodzi zbieżność:*

$$\frac{1}{n}\sum_{k=0}^{n-1} f\left(T^{(k)}(x_0)\right) \to \frac{1}{\mu(X)} \int_X f(x) \mathrm{d}\mu(x), \quad \textit{dla} \quad \mu\textit{-p.w.} \quad x_0 \in X.$$

Definicje i twierdzenia związane z teorią procesów stochastycznych przytaczamy za podręcznikiem Iwanika i Misiewicz [1].

**Definicja 7** **Procesem stochastycznym** $\{X_t,\ t \in \mathbb{T}\}$ *nazywamy taką funkcję $(t, \omega) \to X_t(\omega)$, $\omega \in \Omega$, $t \in \mathbb{T}$, że $X_t(\omega)$ jest zmienną losową dla każdego ustalonego $t \in \mathbb{T}$.*

**Definicja 8** *Zbiór wszystkich możliwych wartości procesu stochastycznego $\{X_t,\ t \in \mathbb{T}\}$ $\mathcal{S} = \{X_t(\omega) : t \in \mathbb{T}, \omega \in \Omega\}$ nazywamy* **przestrzenią stanów** *procesu lub jego* **przestrzenią fazową**.

Będziemy rozważali co najwyżej przeliczalne przestrzenie stanów.

**Definicja 9 (Warunek Markowa)** *Proces stochastyczny $\{X_n,\ n \in \mathbb{N}\}$ o wartościach w $\mathcal{S}_m = \{1, ..., m\}$, $m \in \mathbb{N}$ nazywamy* **łańcuchem Markowa** *lub* **procesem Markowa**, *jeśli dla dowolnego $n \in \mathbb{N}$ i dowolnego wyboru stanów $k_0, k_1, ..., k_n \in \mathcal{S}_m$ zachodzi poniższa własność*

$$\mathbb{P}(X_n = k_n | X_{n-1} = k_{n-1}, ..., X_0 = k_0) = \mathbb{P}(X_n = k_n | X_{n-1} = k_{n-1}).$$

Równanie to bywa także nazywane warunkiem Markowa lub warunkiem braku pamięci.

**Definicja 10** *Łańcuch Markowa $\{X_n,\ n \in \mathbb{N}\}$ jest* **jednorodny** *lub równoważnie* **ma stacjonarne prawdopodobieństwa przejścia**, *jeśli dla dowolnych stanów $i, j \in \mathcal{S}_m$, $m \in \mathbb{N}$ i dowolnego $n \in \mathbb{N}$ zachodzi równość*

$$\mathbb{P}(X_n = j | X_{n-1} = i) = \mathbb{P}(X_1 = j | X_0 = i).$$

W dalszej części rozważane będą jedynie łańcuchy Markowa o stacjonarnym prawdopodobieństwie przejścia.

**Definicja 11** *Prawdopodobieństwa $p_{ji} := \mathbb{P}(X_1 = j | X_0 = i)$, dla $i, j \in \mathcal{S}_m$, $m \in \mathbb{N}$, nazywamy* **prawdopodobieństwami przejścia w jednym kroku**, *a utworzoną z nich macierz*

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1m} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ p_{m1} & p_{m2} & p_{m3} & \cdots & p_{mm} \end{bmatrix},$$

**macierzą przejścia w jednym kroku**.

**Uwaga 1** *Przyjęto tutaj notację inną od zawartej w cytowanych podręcznikach do procesów stochastycznych (m. in. Iwanik i Misiewicz [1]), gdzie autorzy zapisywali prawdopodobieństwo przejścia w jednym kroku jako* $\tilde{p}_{ij} := \mathbb{P}\left(X_1 = j | X_0 = i\right)$, *co powodowało, że w ich ujęciu macierz przejścia w jednym kroku była transpozycją macierzy, według defiinicji 11 -* $\tilde{P}^T = P$.

**Uwaga 2** *Ujęcie teorii procesów Markowa w formalizm teorii układów dynamicznych często okazuje się pomocne nie tylko pojęciowo, ale też ze względów rachunkowych (porównaj rozdział trzeci i piąty w podręczniku Lasoty i Mackeya [2]). Wówczas macierz przejścia P nazywa się operatorem Markowa. Znane z teorii układów dynamicznych pojęcie asymptotycznej stabilności wiąże się wówczas ściśle z ergodycznością łańcuchów Markowa. W dalszej części termin macierz przejścia P będzie stosowany zamiennie z terminem operator Markowa P.*

Samo pojęcie można zdefiiniować też abstrakcyjnie, ale w rozważanym przypadku wystarczy utożsamiać operator Markowa z operatorem generującym proces Markowa, czyli z macierzą przejścia w jednym kroku.

Z aplikacyjnego punktu widzenia bardzo ważne jest pytanie, jak działa proces Markowa, gdy iterowany jest wielokrotnie[2]. Odpowiedzi na pytanie to udziela poniższe twierdzenie.

[2] To właśnie te wielokrotne iteracje są przyczyną, dla której formalizm teorii układów dynamicznych jest tu z powodzeniem stosowany.

**Twierdzenie 2** *Jeżeli proces Markowa P jest iterowany n-krotnie to odpowiada mu wtedy proces Markowa o macierzy przejścia w jednym kroku równej* $Q = P^n$, *przy czym* $q_{ij}(n) = [P^n]_{ij}$, *jest prawdopodobieństwem przejścia w jednym kroku dla procesu będącego n-krotną iteracją procesu wyjściowego.*

**Definicja 12** *Macierz* $P \in \mathbb{M}^{m\times m}\left([0,1]\right)$, $m \in \mathbb{N}$, $P = [p_{ij}]_{i,j=1,\ldots,m}$ *spełniającą poniższe założenia*

- $p_{ij} \geqslant 0$ *dla dowolnych* $i$, $j \in \mathcal{S}_m$,
- $\sum\limits_{i=1}^{m} p_{ij} = 1$,

*nazywamy* **macierzą stochastyczną**.

Widać, że przyjęta defiinicja 11 macierzy $P$ powoduje, że macierzą stochastyczną nazywamy macierz stochastyczną kolumnowo (tzn. w kolumnach sumującą się do 1). Macierze stochastyczne wierszowo są właściwe przy transponowanej defiinicji $P$ (porównaj uwaga 1). Macierz będącą jednocześnie stochastyczną kolumnowo i wierszowo nazywamy **macierzą podwójnie stochastyczną**.

Drugą, po macierzy przejścia w jednym kroku, wygodną metodą reprezentacji procesu Markowa są diagramy przejścia w jednym kroku.

**Definicja 13** **Diagramem przejścia w jednym kroku** *procesu Markowa* $\{X_n,\ n \in \mathbb{N}\}$ *nazywamy graf skierowany, w którym wierzchołkami są elementy przestrzeni fazowej, a łuki łączą dwa stany, o ile prawdopodobieństwo przejścia między nimi w jednym kroku jest niezerowe.*

Każdy ze stanów należący do przestrzeni fazowej $\mathcal{S}_m$ dla procesu Markowa może zostać scharakteryzowany ze względu na możliwości przejścia między nim a innymi stanami. Poniżej zaprezentowano stosowaną później charakteryzację stanów.

**Definicja 14** *Stan* $i \in \mathcal{S}_m$ *jest* **pochłaniający** *jeśli* $p_{ii} = 1$.

Intuicyjnie stan pochłaniający to taki stan, z którego nie można wyjść.

**Definicja 15** *Stan* $i \in \mathcal{S}_m$ *jest* **osiągalny** *ze stanu* $j \in \mathcal{S}_m$ *jeśli istnieje takie* $n \in \mathbb{N}$, *że* $p_{ij}(n) > 0$. *Będzie to oznaczane przez* $j \to i$.

Jeden stan jest dla drugiego osiągalny, o ile można w toku iteracji przejść z jednego do drugiego.

**Definicja 16** *Stany* $i, j \in \mathcal{S}_m$ **wzajemnie się komunikują** *jeśli* $j \to i$ *oraz* $i \to j$.

Komunikowanie się jest relacją równoważności (symetryczną, zwrotną i przechodnią), dzieli więc wszystkie stany procesu Markowa na klasy abstrakcji - klasy stanów komunikujące się między sobą.

**Definicja 17** *Łańcuch Markowa nazywamy* **nieprzywiedlnym**, *gdy wszystkie jego stany wzajemnie się komunikują.*

Nieprzywiedlność oznacza, że wszystkie stany należą do tej samej klasy abstrakcji względem relacji komunikowania się.

**Definicja 18** **Okresem** *stanu* $i \in \mathcal{S}_m$ *nazywamy liczbę*

$$o(i) = NWD(n \in \mathbb{N} : p_{ii}(n) > 0),$$

*czyli największy wspólny dzielnik zbioru takich* $n$, *że powrót do stanu* $i$ *jest możliwy po* $n$ *krokach.*

*Stan* $i$ *jest* **okresowy** *jeśli* $o(i) > 1$ *i* **nieokresowy** *gdy* $o(i) = 1$.

**Definicja 19** *Nieprzywiedlny łańcuch Markowa jest* **okresowy** *jeśli wszystkie jego stany są okresowe o tym samym okresie* $d > 1$. *W przeciwnym przypadku mówimy, że łańcuch jest* **nieokresowy**.

Defiinicja 19 jest dobrze określona, co wynika z poniższego twierdzenia, przytoczonego za podręcznikiem Iwanika i Misiewicz [1].

**Twierdzenie 3** *W nieprzywiedlnym łańcuchu Markowa wszystkie stany mają ten sam okres.*

**Definicja 20** *Rozkład początkowy* $\pi_i = \mathbb{P}(X_0 = i)$, $i \in \mathcal{S}_m$ *łańcucha Markowa* $\{X_n, n \in \mathbb{N}\}$ *jest* **stacjonarny** *lub* **niezmienniczy** *jeśli zachodzi równanie*

$$\pi = P\pi.$$

Rozkład stacjonarny jest więc punktem stałym operatora $P$.

**Definicja 21** *Jednorodny łańcuch Markowa jest* $\{X_n,\ n \in \mathbb{N}\}$ **ergodyczny**, *jeśli dla każdego* $i \in \mathcal{S}_m$ *istnieją i nie zależą od* $j$ *następujące granice*

$$q_i = \lim_{n\to\infty} p_{ij}(n) > 0,$$

*oraz* $\sum_{i=1}^{m} q_i = 1$. *Otrzymany rozkład* $\mathbf{q} = (q_i)_{i=1,\dots,m}$ *nazywamy* **ergodycznym**.

Ergodyczność w języku operatorów Markowa (porównaj rozdział 5.6 w podręczniku Lasoty i Mackeya [2]) jest równoważna istnieniu asymptotycznie stabilnego rozkładu prawdopodobieństwa dla operatora (macierzy) $P$.

**Twierdzenie 4** *Każdy rozkład ergodyczny dla pewnego łańcucha Markowa jest też rozkładem stacjonarnym tego łańcucha.*

**Uwaga 3** *Implikacja z twierdzenia 4 w ogólności nie daje się odwrócić. Twierdzenie odwrotne nie musi być prawdziwe. Łańcuch Markowa może posiadać więcej niż jeden rozkład stacjonarny (porównaj twierdzenie 5), a, zgodnie z definicją 21, rozkład ergodyczny wyznaczony jest jednoznacznie. Na podstawie twierdzenia 5 każdy łańcuch Markowa posiada co najmniej jeden rozkład stacjonarny, ale co do istnienia rozkładu ergodycznego dla dowolnego procesu nie ma gwarancji. (porównaj twierdzenie 6).*

**Twierdzenie 5** *Dla każdego procesu Markowa o skończonej liczbie stanów istnieje co najmniej jeden rozkład stacjonarny. Wszystkie rozkłady stacjonarne (jako wektory z przestrzeni* $\mathbb{R}^m$*) należą do podprzestrzeni rozpiętej przez wektory własne macierzy* $P$ *odpowiadające wartości własnej* 1.

Jest to tak naprawdę cecha każdej macierzy stochastycznej, dla której 1 jest na pewno wartością własną.

**Twierdzenie 6** *Rozważmy nieokresowy i nieprzywiedlny proces Markowa o skończonej liczbie stanów. Wówczas łańcuch ten posiada dokładnie jeden rozkład stacjonarny* $\pi$. *Co więcej rozkład* $\pi$ *jest też rozkładem ergodycznym tego procesu.*

Jeśli stanów byłoby przeliczalnie wiele wówczas należy jeszcze założyć, że każdy stan jest dodatnio powracający, czyli, że prawdopodobieństwo powrotu do każdego stanu w skończonym czasie jest większe niż 0.

*Mathematica*

Podstawową jednostką składni w środowisku Mathematica jest komórka. Komórki mogą przyjmować kilka typów, ale dwa najważniejsze z nich to `In[]`, czyli wprowadzany przez użtkownika skrypt oraz `Out[]`, czyli wynik działania `In[]`. Aby wykonać komórkę, w której aktualnie znajduje się kursor należy wcisnąć `[Shift]+[Enter]` lub prawy `[Enter]`.

Komórki wyróżnione są wewnątrz notatnika ramką po prawej stronie.

Zachęcamy do sprawdzenia opisywanej składni już teraz. Można to zrobić nawet bez dostępu do wydziałowych licencji, używając aplikacji Wolfram Programming Lab.

Nazwy wbudowanych funkcji w języku Wolfram zawsze zaczynają się wielką literą według wzoru

```
NazwaFunkcji[arg1, arg2, ...]
```

Zauważmy, że te nazwy dość dokładnie precyzują co dana funkcja robi, np. `Series[]` rozwija w szereg, a `Solve[]` rozwiązuje równania. Podstawowa znajomość języka angielskiego pozwala zatem na dość dokładne przewidywanie nazwy funkcji, której funkcjonalność chcielibyśmy uzyskać.

gdzie argumenty funkcji przekazywane są w kwadratowych nawiasach `[]`. Najważniejszym sposobem poznawania funkcjonalności Mathematiki jest jej bogata pomoc. Jest to także podstawowe narzędzie pracy w tym środowisku. Pomoc wywołuje się klawiszem `[F1]` wewnątrz aktywnego notatnika, lub, nawet bez aktywnej aplikacji, można sprawdzić jej internetową wersję.

Dostępną, przypomnijmy, pod adresem reference.wolfram.com/language/

Jako ćwiczenie wprowadzające do środowiska Wolfram Mathematica proponujemy sprawdzić w dokumentacji do czego mogą służyć następujące funkcje

Zwracamy uwagę na bogactwo przykładów dostępnych w plikach pomocy.

- `Plot[]`
- `Integrate[]`,
- `Animate[]`,
- `Manipulate[]`,
- `Table[]`,
- `Graphics[]`,
- `D[]`,
- `ListPlot[]`,
- `Simplify[]`.

Jak można zauważyć, podczas analizy powyższych przykładów, podstawową strukturą danych w Mathematice jest lista. Lista to nic innego jak pewien uporządkowany zbiór pewnych (niekoniecznie tego samego typu!) elementów. Poniżej zamieszczamy przykłady list

Jaki wyniki dają te kody źródłowe?

```
{a, b, c, 1, 2, 3}

{{1,0},{0,1}}

Table[1/i,{i,10}]
```

Lista list jest sposobem zapisu macierzy w środowisku Mathematica - po więcej szczegółów odsyłamy do dokumentacji (zwłaszcza `Symbolic & Numeric Computation`→`Matrices and Linear Algebra`).

Funkcja `Table[]` jest jedną z częściej używanych, podczas pracy z Mathematiką. Sugerujemy przejrzeć jej dokumentację.

Poniższe kody źródłowe są przykładami wykorzystania środowiska Mathematica do tworzenia grafiki. Sugerujemy je skopiować do notatnika i sprawdzić co daje ich wykonanie. Przejrzenie dokumentacji wykorzystanych funkcji również będzie pouczające.

```
Graphics[Table[Circle[RandomReal[{-10, 10}, {2}]], {80}]]

Graphics[{Blue, FilledCurve[Line[RandomReal[{0, 10}, {10,
2}]]], Red, FilledCurve[Line[RandomReal[{0, 10}, {20, 2}]]],
Green, FilledCurve[Line[RandomReal[{0, 10}, {100, 2}]]]}]
```

Polecamy także wypróbować działania funkcji `Disk[]`, `Polygon[]`, a także możliwości manipulowania obiektami graficznymi (kolory, grubości linii etc.), przykłady Czytelnik znajdzie oczywiście w dokumentacji `Graphics[]`.

*Wymagania formalne*

Prawidłowo przygotowana praca domowa powinna spełniać następujące wymagania

- Po pierwsze **musi być zgodna z** obowiązującym, umieszczonym na stronie przedmiotu, **szablonem odpowiedzi**. W szczególności, nie może zajmować więcej niż jednej kartki A4.
- Kody źródłowe powinny być czytelne i poprawne.
- Wszystkie liczby i cyfry występujące na wykresach i rysunkach powinny być w rozmiarach zbliżonych do czcionki tekstu oraz kroju, umożliwiającym ich wygodne odczytanie.
- Wszystkie krzywe (w tym osie współrzędnych i wykresy) występujące na rysunkach powinny być pogrubione tak, aby być czytelne i odróżniać się od reszty rysunku.
- Osie wykresów powinny być podpisane.

# *Obroty*

Eppur si muove — Galileo Galilei

Celem ćwiczeń będzie zbadanie podstawowych dynamicznych własności obrotu na okręgu ($\mathcal{S}^1 = [0, 2\pi]/(0 = 2\pi)$), czyli funkcji

$$f_\theta : \mathcal{S}^1 \to \mathcal{S}^1, \quad f_\theta(x) = (x + \theta) \mod 2\pi.$$

Na przykładzie tych badań poznamy też podstawową funkcjonalność środowiska Wolfram Mathematica.

**Uwaga:** Przed wykonywaniem zadań sugerowane jest przeczytanie pierwszego rozdziału niniejszego skryptu.

## *Plan badań*

1. Implementacja funkcji $f_\theta$ (definicja funkcji, definicja zmiennej). — `:=` vs. `=`
2. Rysowanie wykresu funkcji $f_\theta$. — `Plot[]`
3. Rysowanie wykresu orbity $f_\theta$ (naiwne). — `ListPlot[]`
4. Łączenie wykresów. — `GraphicsGrid[]`, `Show[]`
5. Dyskusja zależności $f_\theta(x)$ od $\theta$ i $x$. — `Manipulate[]`
6. Iterowanie funkcji $f_\theta$. — `Nest[]`
7. Generowanie orbity $f_\theta$. — `NestList[]` vs. `Table[]`
8. Analiza orbity funkcji $f_\theta$ i dyskusja jej zależności od $\theta$. — `Histogram[]` vs. `HistogramList[]`

## *Pytania i problemy*

$\clubsuit_1$ Wyjaśnij jak działa i, rysując wykres funkcji sinus, podaj przykład użycia funkcji `Plot[]`.

$\clubsuit_2$ Wyjaśnij jak działa i, rysując wykres listy `{3,1,4,1,5,9}`, podaj przykład użycia funkcji `ListPlot[]`.

$\clubsuit_3$ Wyjaśnij jak działa i, generując iteracje funkcji $h(x) = 2x$, podaj przykład użycia funkcji `Nest[]`.

$\clubsuit_4$ Wyjaśnij jak działa i, generując orbitę funkcji sinus, podaj przykład użycia funkcji `NestList[]`.

$\clubsuit_5$ Wyjaśnij jak działa i, wyznaczając histogram listy liczb pseudolosowych , podaj przykład użycia funkcji `Histogram[]`.

Potrzebną listę najprościej wygenerować komendą `li=RandomReal[{0,1},{100}]`

$\clubsuit_6$ Powtórz zadanie $\clubsuit_5$, ale wykorzystując funkcje `HistogramList[]` i `ListPlot[]`.

Tutaj, dla różnorodności proponujemy listę `Exp/@li`.

$\clubsuit_7$ Jak w środowisku Mathematica definiuje się funkcje? Zdefiniuj własną funkcję $h(x) = |x|$.

♣$_8$ Jak w środowisku Mathematica definiuje się zmienne? Podaj przykłady co najmniej 5 typów zmiennych.

♣$_9$ Jak stworzyć listę? Jak zmienić dokładnie jeden jej element?

♣$_{10}$ Wyjaśnij jak działa i, generując macierz jednostkową $5 \times 5$, podaj przykład użycia funkcji `Table[]`.

♣$_{11}$ Wygeneruj na dwa sposoby (raz iterując liczby całkowite, raz zmiennoprzecinkowe) listę ze współrzędnymi wykresu funkcji sinus na zbiorze $[0, 2\pi]$.

Zwróćmy uwagę, że jest to lista par postaci `{{a,Sin[a]},{b,Sin[b]},...}`

♣$_{12}$ Jak dodać więcej niz jeden wykres na `Plot[]`? Podaj przykład z funkcjami sinus i kosinus.

♣$_{13}$ Jak dodać więcej niz jeden wykres na `ListPlot[]`? Podaj przykład z listami `{1,2,3,4}` i `{{4,5},{3,2},{2,1},{1,0}}`.

♣$_{14}$ Jak dodać kropki na wykresie typu `Plot[]`?

Sugrowanym przykładem mogą być punkty `{{0.1,0.15},{0.2,0.19},{0.3,0.32},{0.4,0.39},{0.5,0.51},{0.6,0.59}}` i przybliżająca je funkcja $h(x) = x$.

♣$_{15}$ Powtórz zadanie ♣$_{14}$, ale tym razem dodając linię ciągłą na wykresie typu `ListPlot[]`.

♣$_{16}$ Wyjaśnij jak działa funkcja `Manipulate[]`. Zilustruj jej działanie przykładem.

♣$_{17}$ Na przykładzie funkcji $h(x, y) = e^x/y$, $x, y \in [0, 1]$ wyjaśnij jak inicjować zmienne w bloku `Manipulate[]` *typowymi wartościami* i do jakich nieporozumień może prowadzić brak takiej inicjacji.

♣$_{18}$ Jak pogrubić krzywą na wykresie typu `Plot[]` i zmienić jej kolor?

W zadaniach ♣$_{18}$ − ♣$_{23}$ sugerujemy uważnie przeczytać sekcję `Options` w dokumentacji funkcji `Plot[]` lub `ListPlot[]`.

♣$_{19}$ Jak pogrubic kropki na wykresie typu `ListPlot[]` i zmienić ich kolor?

♣$_{20}$ Jak dodać do wykresu (dowolnego typu) linie siatki?

♣$_{21}$ Jak zmienić na wykresie (dowolnego typu) zakres na osi odciętych i rzędnych?

♣$_{22}$ Jak na wykresie (dowolnego typu) opisać osie i zmienic ich styl (np. kolor, krój czcionki bądź jej rozmiar)?

♣$_{23}$ Jak do wykresu (dowolnego typu) dodać legendę?

♣$_{24}$ Opisz co najmniej 3 typy komórek (np. `In`, `Out`, `Text`).

♣$_{25}$ Wyjaśnij czym różnią się w działaniu np. funkcji `Table[]` liczniki `{i,1,10,1}` i `{i,{1,10,1}}`.

♣26 Czym różnią się przypisania `=` i `:=`? Zilustruj tę różnicę wykorzystując funkcję `RandomInteger[]`.

♣27 Wyjaśnij jak działa funkcja `Show[]`. Podaj przykład jej wykorzystania.

**Uwaga:** Pamiętaj aby w rozwiązaniach ♠ zastosować opcje zwiększające czytelność (a niekiedy nawet nadające sens) wykresów. Są to opcje takie jak: `PlotRange`, `AxesOrigin`, `PlotStyle`, `PlotLegends` itp.

♠1 Narysuj wykres pierwszych 10 elementów orbity funkcji $h(x) = x^3$ w $x_0 = 0.5$.

♠2 Wykorzystując warunkową instrukcję `If[]` napisz własną implementację funkcji `Mod[]`.

♠3 Narysuj wykres pierwszych 10 elementów orbity funkcji $h(x) = \sqrt{x}$ w $x_0 = 0.5$.

♠4 Narysuj histogram pierwszych 100 elementów orbity funkcji $h(x) = \sqrt{x}$ w $x_0 = 0.5$.

Dobierz odpowiednie rozmiary przedziałów (binów).

♠5 Narysuj wykresy trzech pierwszych iteracji funkcji sinus na zbiorze $[0, 2\pi]$.

♠6 Narysuj histogram pierwszych 100 elementów orbity funkcji $h(x) = x^3$ w $x_0 = 0.5$.

Dobierz odpowiednie rozmiary przedziałów (binów).

♠7 Wygeneruj listę pierwszych 20 liczb pierwszych.

♠8 Narysuj wykres pierwszych 5 elementów orbity funkcji $h(x) = e^x$ w $x_0 = 0.05$.

Zamiast funkcji `ListPlot[]` wykorzystaj `ListLogPlot[]`.

♠9 Narysuj histogram pierwszych 100 elementów orbity funkcji $h(x) = \sin x$ w $x_0 = \pi/2$.

Dobierz odpowiednie rozmiary przedziałów (binów).

♠10 Narysuj wykres drugiej iteracji funkcji wykładniczej na $[0, 1]$.

Zamiast funkcji `Plot[]` wykorzystaj `LogPlot[]`.

♠11 Narysuj wykres pierwszych 50 elementów orbity funkcji $h(x) = 3x(1-x)$ w $x_0 = 0.27$.

♠12 Narysuj wykres czwartej iteracji funkcji $h(x) = 3x(1-x)$ na zbiorze $[0, 1]$.

♠13 Wygeneruj listę pierwszych 20 cyfr w rozwinięciu dziesiętnym liczby $\pi$.

♠14 Narysuj wykres pierwszych 10 elementów orbity funkcji $h(x) = \sin x$ w $x_0 = \pi/2$.

♠15 Narysuj wykres trzeciej iteracji funkcji $h(x) = \sin(e^x)$ na $[0, 1]$.

♠16 Wygeneruj listę pierwszych 20 cyfr w rozwinięciu dziesiętnym liczby $e$.

♠17 Narysuj histogram pierwszych 100 elementów orbity funkcji $h(x) = 3x(1 - x)$ w $x_0 = 0.27$.

♠18 Wygeneruj listę pierwszych 20 cyfr w rozwinięciu dziesiętnym liczby $\sqrt{2}$.

♠19 Wygeneruj listę wykresów pierwszych 5 iteracji funkcji sinus na zbiorze $[0, 2\pi]$.

♠20 Wygeneruj macierz $5 \times 5$, o współrzędnych będących sumą numeru wiersza i kolumny.

♠21 Zbadaj zależność wykresu funkcji $h_C(x) = e^{-Cx^2}$ od parametru $C \in [0, 2]$.

Polecamy tutaj funkcję `Manipulate[]`.

♠22 Wyznacz wszystkie macierze $2 \times 2$ o elementach ze zbioru $\{0, 1\}$.

♠23 Zbadaj dynamikę funkcji $h_C(x) = Cx$, w zależności od wartości $C$.

Polecamy tutaj funkcję `Manipulate[]`.

♠24 Narysuj wykres funkcji $\zeta$ Riemanna.

♠25 Rozwiąż graficznie równanie $e^x + x = 0$.

♠26 Wyznacz macierz permutacji dla permutacji $\{3, 5, 4, 2, 1\}$.

♠27 Wyznacz trzy wybrane orbity dla całkowitoliczbowej funkcji z otwartego problemu $3n + 1$ danej wzorem

$$h(n) = \begin{cases} \frac{n}{2} & \text{gdy } n \text{ jest parzysta} \\ 3n + 1 & \text{gdy } n \text{ jest nieparzysta} \end{cases}.$$

## *Rodzina kwadratowa*

A W DODATKU, DAJĘ SŁOWO, MAM RODZINĘ WYJĄTKOWĄ!
J. BRZECHWA

Celem ćwiczeń będzie numeryczne zbadanie dynamiki rozwoju populacji bakterii, modelowane przez rodzinę odwzorowań kwadratowych

$$f_\lambda(x) = \lambda x(1-x), \qquad x \in [0,1],$$

gdzie $\lambda \in [1, 4]$ to parametr.

### *Plan badań*

**1.** Rodzina kwadratowa jako uproszczony model populacji bakterii.

Ustalmy pojemność środowiska $M > 0$, liczbę $y_0 \in [0, M]$ jako początkową ilość bakterii oraz wielkość $x_0 := y_0/M$.

Dlaczego naiwny model $\hat{f}_\lambda(x) = \lambda x$ jest niefizyczny?

- Niech populacji zmienia się zgodnie z $f_\lambda(x) = \lambda x(1-x)$,
- Dokonaj interpretacji biologicznej rekurencji postaci

$$x_{n+1} = f_\lambda(x_n).$$

**2.** Dla ustalonych wartości $x_0$ i $\lambda$ wygeneruj 30 pierwszych elementów orbity $\mathcal{O}(f_\lambda, x_0)$.

Przy definiowaniu $f_\lambda[x\_] :=$ przydatna może okazać się funkcja `N[]`. Dlaczego?

**3.** Przedstaw w formie graficznej uzyskaną orbitę.

**4.** Przy użyciu funkcji `Manipulate[]` przedyskutuj zależność dynamiki od $\lambda$ i $x_0$.

Proszę pamiętać też aby uwzględnić możliwość zmiany długości obserwowanej orbity

**5.** Narysuj kolejne iteracje funkcji $f_\lambda(x)$. Przedyskutuj zależność kolejnych iteracji od parametru $\lambda$ oraz zinterpretuj uzyskane wyniki.

Pomocne może być naniesienie na badany wykres funkcji $h(x) = x$

**6.** Diagram bifurkacyjny.

Dla wybranego (dobrze dobranego) punktu startowego $x_0$ wygeneruj pierwsze $n$ elementów orbity $\mathcal{O}(f_\lambda, x_0)$. Dysponując szeregami czasowymi dla różnych wartości $\lambda$ nanieś ich „zachowania asymptotyczne" dla kolejnych wartości $\lambda$ na wykres aby otrzymać poszukiwany diagram bifurkacyjny.

Na diagramie bifurkacyjnym należy umieścić tylko punkty powyżej pewnej liczby iteracji $n_0$.

**7. Dodatkowe**
Wyestymuj miarę niezmienniczą dla $f_4$, w tym celu:

- Wylosuj wektor $m$ liczb pseudolosowych z rozkładu równomiernego na $[0, 1]$.

- Na każdej ze współrzędnych rozważanego wektora wywołaj $n$-tą iterację funkcji $f_4$ Zadziałaj na każdej współrzędnej tego wektora $n$ razy rozważaną funkcją.

Polecamy do tego celu `Nest[]` i `Map[]`.

- Przy użyciu funkcji `HistogramList[]` utwórz z tej listy odpowiednią listę histogramu - sugerujemy samemu popracować nad optymalnym doborem liczby przedziałów i ich szerokości na jakie chcemy podzielić nasze dane.

Warto zapoznać się z opcją `PDF` dla funkcji `HistogramList[]` - będzie to pomocne.

- Stosując funkcje `Partition[]` i `Riffle[]` tak przekształć uzyskaną listę aby można było narysować ją z wykorzystaniem funkcji `ListPlot[]`.
- Porównaj uzyskany wynik z wynikiem dokładnym:

$$h(x) = \frac{1}{\pi\sqrt{x(1-x)}}.$$

*Pytania i problemy*

♣1 Wyjaśnij jak działa funkcja `NestWhile[]`. Podaj przykład jej zastosowania.

♣2 Wyjaśnij jak działa funkcja `NestWhileList[]`. Podaj przykład jej zastosowania.

♣3 Wyjaśnij jak działa funkcja `FixedPoint[]` . Podaj przykład jej zastosowania.

♣4 Wyjaśnij jak działa funkcja `Partition[]`. Podaj przykład jej zastosowania.

♣5 Wyjaśnij jak działa funkcja `Riffle[]`. Podaj przykład jej zastosowania.

♣6 Wyjaśnij jak działa funkcja `Join[]`. Podaj przykład jej zastosowania.

♣7 Wyjaśnij jak działa funkcja `Flatten[]`. Podaj przykład jej zastosowania.

♣8 Wyjaśnij jak działa funkcja `Reverse[]`. Podaj przykład jej zastosowania.

♣9 Wyjaśnij jak działa funkcja `Animate[]`. Podaj przykład jej zastosowania.

♣10 Czym różnią się porównania `==` (`Equal`) i `===` (`SameQ`)?

♣11 Jak działa funkcja `GraphicsGrid[]`? Podaj przykład jej zastosowania.

♣$_{12}$ Jak działa funkcja `GraphicsRow[]`? Podaj przykład jej zastosowania.

♣$_{13}$ Jak działa funkcja `GraphicsColumn[]`? Podaj przykład jej zastosowania.

♣$_{14}$ Wyjaśnij jak działa funkcja `Range[]`. Podaj przykład jej zastosowania.

♣$_{15}$ Wyjaśnij jak działa funkcja `Directive[]`. Podaj przykład jej zastosowania.

♣$_{16}$ Wyjaśnij jak działa funkcja `Integrate[]`. Podaj przykład jej zastosowania.

♣$_{17}$ Wyjaśnij jak działa funkcja `D[]`. Podaj przykład jej zastosowania.

♣$_{18}$ Wyjaśnij jak działa funkcja `Solve[]`. Podaj przykład jej zastosowania.

♣$_{19}$ Wyjaśnij jak działa funkcja `NSolve[]`. Podaj przykład jej zastosowania.

♣$_{20}$ Wyjaśnij na przykładach czym różni się sześć różnych liczników opisanych w dokumentacji funkcji `Manipulate[]`.

♣$_{21}$ Wyjaśnij jak działa funkcja `Sort[]`. Podaj przykład jej zastosowania.

♣$_{22}$ Wyjaśnij jak działa, a następnie wykorzystaj w napisaniu własnej funkcji, procedurę `Function[]` w wersji `Function[x,body]`.

♣$_{23}$ Wyjaśnij jak działa, a następnie wykorzystaj w napisaniu własnej funkcji, procedurę `Function[]` w wersji `Function[F[#]]`.
Co oznacza `#` (`Slot`)?

♣$_{24}$ Wyjaśnij jak działa, a następnie wykorzystaj w napisaniu własnej funkcji, procedurę `F[#]&`. Jakie znaczenie mają `#` (`Slot`) i `&` (`Function`)?

♣$_{25}$ Jakie wartości może przyjmować i jak działa (m. in. w `Plot[]` i `ListPlot[]`) dyrektywa `Filling`?

♣$_{26}$ Jakie wartości może przyjmować i jak działa (w `ListPlot[]`) dyrektywa `Joined`?

♣$_{27}$ Jakie wartości może przyjmować i jak działa (w `ListPlot[]`) dyrektywa `Mesh`?

**Uwaga:** Pamiętaj aby w rozwiązaniach ♠ zastosować opcje zwiększające czytelność (a niekiedy nawet nadające sens) wykresów. Są to opcje takie jak: `PlotRange`, `AxesOrigin`, `PlotStyle`, `PlotLegends` itp.

♠$_{1}$ Analizując wykres orbity wyznacz (empirycznie) wartość $\lambda$ przy której zachodzi pierwsza bifurkacja w rodzinie kwadratowej.

♠$_2$ Porównaj na wykresie, w zależności od $n \in \{1, 2, \ldots, 25\}$, czasy wykonywania komend `Nest[3 # (1 - #) &, 4/10, n]` oraz `Nest[3 # (1 - #) &, 0.4, n]`. Wyjaśnij różnice.

Do zliczania czasu wykonania funkcji polecamy komendę `AbsoluteTiming[]`.

♠$_3$ Analizując wykres orbity wyznacz (empirycznie) wartość $\lambda$ przy której, dla rodziny kwadratowej, pojawia się pierwsza orbita o okresie 4.

♠$_4$ Analizując $f_{3.5}^4$ Przy pomocy funkcji `Solve[]` wyznacz wartości orbity okresowej o okresach 2 i 4 oraz punkty stałe. Sprawdź wszystkie wyniki bezpośrednio z definicji (wywołując funkcję $f_{3.5}$ odpowiednią liczbę razy).

♠$_5$ Powtórz wykres diagramu bifurkacyjnego z zajęć w lepszej rozdzielczości dla $\lambda \in (1, 3)$. Dopasuj krzywą do obserwowanych punktów.

Można wyliczyć jej wzór, analizując postać funkcji $f_\lambda$, odszukać go w literaturze albo dopasować numerycznie np. przy pomocy funkcji `Fit[]` lub `FindFit[]`.

♠$_6$ Wykonując operacje na listach wyznacz wartość $\lambda$ dla pierwszej bifurkacji podwajania okresu.

♠$_7$ Analizując $f_{3.1}^2$ Przy pomocy funkcji `Solve[]` wyznacz wartości orbity okresowej o okresie 2 oraz punkty stałe. Sprawdź wszystkie wyniki bezpośrednio z definicji (wywołując funkcję $f_{3.1}$ odpowiednią liczbę razy). Narysuj wykres zależności punktów orbity o okresie 2 w funkcji $\lambda \in (3, 3.5)$.

♠$_8$ Wykonaj animację (względem parametru $\lambda$) histogramu orbity funkcji $f_\lambda$. W raporcie zamieść kod źródłowy i wybrane, reprezentatywne, klatki tej animacji, nie mniej niż 10.

W zadaniu tym bardzo ważne jest odpowiednie dobranie długości orbity i liczby binów. Zbyt małe zniweczą spodziewany efekt.

♠$_9$ Dodając strzałki do wspólnego wykresu funkcji $f_\lambda$ i $h(x) = x$ zilustruj pierwsze trzy iteracje metodą graficzną. Zamieść wyniki dla czterech wartości $\lambda$ generujących odmienną dynamikę.

♠$_{10}$ Wyznacz wartość stałej $a$, aby funkcja $h(x) = aI_{[0,\pi]}(x)\sin(x)$ była gęstością rozkładu prawdopodobieństwa. Narysuj jej wykres.

♠$_{11}$ Wychodząc od listy danej funkcją `Range[200]` i stosując operacje na listach, wygeneruj listę par postaci `{n,n+1}`, gdzie $n = 0, 2, 4, \ldots, 200$.

♠$_{12}$ Powtórz wykres diagramu bifurkacyjnego z zajęć w lepszej rozdzielczości dla $\lambda \in (3, 3.7)$. Odczytaj z wykresu możliwie najdokładniej wartości $\lambda$ odpowiadające trzem pierwszym bifurkacjom podwojenia okresu.

♠$_{13}$ Analizując wykres orbity wyznacz (empirycznie) wartość $\lambda$ przy której, dla rodziny kwadratowej, pojawia się pierwsza orbita o okresie 8.

♠$_{14}$ Odczytaj z wykresu wartości 4 kolejnych bifurkacji, a następnie narysuj histogramy orbity przed i po każdej bifurkacji.

Jeśli $\lambda_b$ to punkt bifurkacji to przed oznacza $\lambda_b - \varepsilon$, a po $\lambda_b + \varepsilon$ np. dla $\varepsilon = 0.001$.

♠$_{15}$ Korzystając z licznika postaci `{λ,3,4,0.01}` wygeneruj listę orbit funkcji $f_\lambda$, następnie usuń dublujące się elementy i narysuj wykres liczności orbity w funkcji $\lambda$.

W zadaniach ♠$_{15}$, ♠$_1$6 należy odnosić się do kolejnych elementów orbity (co najmniej 200), które występują powyżej pewnej dostatecznie dużej liczby iteracji $n_0$.

♠$_{16}$ Wykonując operacje na listach wyznacz wartość $\lambda$ dla drugiej bifurkacji podwajania okresu.

♠$_{17}$ Wyznacz wartość stałej $a$, aby funkcja $h(x) = a\,(1/2 - |x - 1/2|)$ określona na zbiorze $x \in [0,1]$ była gęstością rozkładu prawdopodobieństwa. Narysuj jej wykres.

♠$_{18}$ Na diagramie bifurkacyjnym z zajęć nanieś strzałki wskazujące miejsca pierwszych trzech bifurkacji.

♠$_{19}$ Narysuj wykres zależności pola ograniczonego wykresem funkcji $f_\lambda^5$ w funkcji $\lambda \in [1,4]$.

♠$_{20}$ Powtórz wykres diagramu bifurkacyjnego z zajęć w lepszej rozdzielczości dla $\lambda \in (3.7,4)$.

♠$_{21}$ Usuń dublujące się punkty z diagramu bifurkacyjnego z zajęć.

♠$_{22}$ Wykonując operacje na listach wyznacz wartość $\lambda$ dla trzeciej bifurkacji podwajania okresu.

♠$_{23}$ Wychodząc od listy danej funkcją `Range[200]` i stosując operacje na listach, wygeneruj listę liczb mniejszych niż 200 i podzielnych przez 11.

♠$_{24}$ Oblicz pole pod wykresem diagramu bifurkacyjnego dla $\lambda \in [0,3]$.

♠$_{25}$ Wyznacz wartość stałej $a$, aby funkcja $h(x) = \frac{a}{\sqrt{xe^x}}$ określona na zbiorze $x \in (0,+\infty)$ była gęstością rozkładu prawdopodobieństwa. Narysuj jej wykres.

♠$_{26}$ Odczytaj z diagramu bifurkacyjnego krotności orbit pojawiających się w pierwszych 4 bifurkacjach.

♠$_{27}$ Narysuj wykres zależności pola ograniczonego wykresem funkcji $f_4^n$ w funkcji $n = 1, 2, \ldots 10$.

## *Ergodyczność*

### Wstrząśnięte, nie zmieszane — James Bond

#### *Plan badań*

1. Korzystając z twierdzenia Birkhoffa i wyników uzyskanych na pierwszych zajęciach dla obrotu na okręgu (o kąt niewspółmierny do $\pi$) oblicz numerycznie poniższe całki. Porównaj uzyskane wyniki z wartością dokładną.

   Można użyć funkcji `Integrate[]`.

   a) $\int_0^{2\pi} \sin(x)dx$,

   b) $\int_0^{2\pi} \cos^2(x)dx$.

2. Zaproponuj różne procesy Markowa, posiadające co najmniej 5 stanów, które są:

   Polecamy porównać działanie `DiscreteMarkovProcess[]` z iterowaniem macierzy.

   a) nieokresowe,

   b) nieprzywiedlne,

   c) ergodyczne.

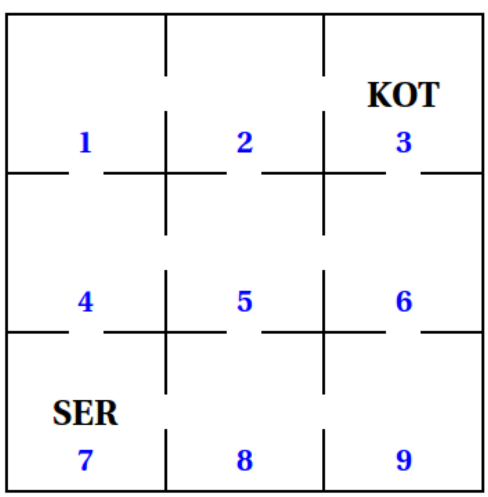
   Dla każdego z tych procesów wyznacz macierz przejścia w jednym kroku i narysuj diagram przejścia.

   Przydatna będzie tu funkcja `WeightedAdjacencyGraph[]`.

3. Dla procesu ergodycznego z punktu 2 znajdź stacjonarny rozkład prawdopodobieństwa. Badając macierz przejścia pokaż, że rozkład ten jest jednoznaczny.

   Przydatne mogą być funkcje `Eigensystem[]`, `Eigenvalues[]`, `Eigenvectors[]`

4. (Na podstawie [1]) Rozważ szczura umieszczonego w labiryncie:



Nasz badany szczur będzie błądził po labiryncie do czasu aż trafi do klatki z serem, albo do klatki z kotem. Zapisz macierz przejścia odpowiedniego procesu Markowa, znajdź prawdopodobieństwa osiągnięcia każdego ze stanów pochłaniających pod warunkiem, że zaczynaliśmy z odpowiedniego stanu początkowego. .

warto skorzystać z symetrii labiryntu

5. Wykonaj symulację szczura z punktu 3. Porównaj wyniki symulacji z obliczeniami analitycznymi.

przydatna będzie funkcja `NestWhile[]`

6. **(Dodatkowe)** Odwzorowanie z rodziny logistycznej $f_4(x)$ jest ergodyczne. Oblicz wartość oczekiwaną $\mathbb{E}\sqrt{X}$ jako wartość oczekiwaną rozkładu danego przez miarę niezmienniczą

$$h(x) = \frac{1}{\pi\sqrt{x(1-x)}}$$

oraz poprzez iterację odwzorowania $f_4(x)$.

### *Pytania i problemy*

$\clubsuit_1$ Wyjaśnij w jaki sposób można obliczyć $n$-tą potęgę macierzy.

$\clubsuit_2$ Wyjaśnij jak działa funkcja `Eigenvalues[]` oraz zilustruj to na przykładzie.

$\clubsuit_3$ Wyjaśnij jak działa funkcja `Eigenvectors[]` oraz zilustruj na przykładzie.

$\clubsuit_4$ Wyjaśnij jak działa funkcja `Eigensystem[]` oraz zilustruj na przykładzie.

$\clubsuit_5$ Jaka jest różnica między funkcjami `Map[]` i `Apply[]`.

$\clubsuit_6$ Wyjaśnij oraz zilustruj na przykładzie działanie funkcji `NIntegrate[]`.

$\clubsuit_7$ Wyjaśnij oraz zilustruj na przykładzie działanie funkcji `WeightedAdjacencyGraph[]`.

$\clubsuit_8$ Wyjaśnij jak wykorzystać funkcję `ArrayPlot[]` do narysowania szachownicy $4 \times 4$.

$\clubsuit_9$ Wyjaśnij oraz zilustruj na przykładzie działanie funkcji `MatrixForm[]`.

$\clubsuit_{10}$ Wyjaśnij dlaczego kod: `MatrixForm`$[\{1,1\}] == \{2,2\}$ nie zwraa wartości logicznej `True`.

$\clubsuit_{11}$ Wyjaśnij jak zmienić kolor wybranej (tylko jednej) krawędzi w grafie generowanym za pomocą funkcji `WeightedAdjacencyGraph[]`.

$\clubsuit_{12}$ Niech $n \in \mathbb{N}$, parzysta. W jaki sposób w grafie o $n$ wierzchołkach generowanym za pomocą funkcji `WeightedAdjacencyGraph[]` pokolorować parzyste wierzchołki na zielono a nieparzyste na żółto?

♣13 Wyjaśnij oraz zilustruj na przykładzie działanie funkcji `Transpose[]`.

♣14 Wyjaśnij jak przy pomocy funkcji `AdjacencyGraph[]` wygenerować graf pełny o 5 wierzchołkach.

♣15 Wyjaśnij i zilustruj przykładem działanie funkcji `Total[]`.

♣16 Czym różnią się od siebie i co dokładnie robią notacje @, @@, @@@, /@?

♣17 Wyjaśnij oraz zilustruj przykładem działanie funkcji `Graph[]`.

♣18 Wyjaśnij oraz zilustruj przykładem działanie funkcji `Max[]` i `Min[]`.

♣19 Wyjaśnij oraz zilustruj przykładem działanie dyrektywy `GraphLayout`.

♣20 Wyjaśnij oraz zilustruj przykładem działanie dyrektywy `VertexyStyle`

♣21 Wyjaśnij oraz zilustruj przykładem działanie dyrektywy `EdgeStyle`

♣22 Wyjaśnij oraz zilustruj przykładem działanie dyrektywy `DiscreteMarkovProcess`.

♣23 Czym różnią się funkcje `DirectedEdge[]` i `UndirectedEdge[]`?

♣24 Wyjaśnij oraz zilustruj przykładem działanie dyrektywy `MarkovProcessProperties[]`.

♣25 Wyjaśnij oraz zilustruj przykładem działanie funkcji `Mean[]`

♣26 Wyjaśnij oraz zilustruj przykładem działanie funkcji `SparseArray[]`

♣27 Jak z macierzy rzadkiej (`SparseArray[]`) uzyskać zwykłą macierz?

**Uwaga:** Pamiętaj aby w rozwiązaniach ♠ zastosować opcje zwiększające czytelność (a niekiedy nawet nadające sens) wykresów. Są to opcje takie jak: `PlotRange`, `AxesOrigin`, `PlotStyle`, `PlotLegends` itp.

♠1 Zaproponuj proces Markowa o co najmniej 6 stanach posiadający stany o różnym okresie, tak aby graf reprezentujący ten proces był spójny (jako graf **nieskierowany**). Dla zaproponowanego procesu:

- narysuj diagram przejścia w jednym kroku,
- narysuj macierz przejścia w jednym kroku,
- znajdź wszystkie stacjonarne rozkłady prawdopodobieństwa.

♠2 Zaproponuj proces Markowa o okresie 2 posiadający co najmniej 6 stanów, tak aby graf reprezentujący ten proces był spójny (jako graf **skierowany**). Dla zaproponowanego procesu:

- narysuj diagram przejścia w jednym kroku,
- narysuj macierz przejścia w jednym kroku,
- znajdź wszystkie stacjonarne rozkłady prawdopodobieństwa.

♠3 Wykorzystując funkcję `Eigensystem[]` znajdź wszystkie stacjonarne rozkłady prawdopodobieństwa dla grafu zadanego macierzą przejścia **(Rozumianej w myśl definicji 11)**

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0.5 & 0 & 0 & 1 \\ 0 & 0 & 0.5 & 0.5 & 0 \end{pmatrix}.$$

♠4 Powtórz zadanie ♠3 dla macierzy przejścia:

$$\begin{pmatrix} 1 & 0 & 0 & 0.3 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 1 \\ 0 & 0 & 0 & 0.2 & 0 \end{pmatrix}$$

♠5 Wykorzystując funkcje `AdjacencyGraph[]`, `Table[]`, `If[]` napisz program, który dla zadanego $n \in \mathbb{N}$ wygeneruje graf pełny o $n$ wierzchołkach.

♠6 Wykorzystując funkcje `Graph[]`, `Table[]`, `If[]` napisz program, który dla zadanego $n \in \mathbb{N}$ wygeneruje graf pełny o $n$ wierzchołkach.

♠7 Napisz funkcję, który, wykorzystując funkcję `Total[]` sprawdza czy podana w jej argumencie macierz jest macierzą przejścia w jednym kroku (w myśl definicji 11).

♠8 Napisz funkcję, który, wykorzystując funkcję `Plus[]` sprawdza czy podana w jej argumencie macierz jest macierzą przejścia w jednym kroku (w myśl definicji 11).

♠9 Napisz funkcję która dla zadanego $n \in \mathbb{N}$ zwracać będzie przy pomocy `ArrayPlot[]` graficzną reprezentację macierzy $n \times n$ przedstawiającą pikselowe koło, czyli figurę, w której małymi kwadratami przybliża się z coraz większą dokładnością (wraz ze wzrostem wartości $n$) wnętrze dwuwymiarowej kuli.

♠10 Wykonaj zadanie ♠9 przybliżając wnętrze dwuwymiarowej figury danej następująco:

$$P = \left\{ (x, y) \in \mathbb{R}^2 : 4 + 2x < x^2 + y^2 < 9 \right\}.$$

♠11 Wykonaj zadanie ♠9 przybliżając wnętrze trójkąta równobocznego.

♠$_{12}$ Wykonaj dwie animacje $P^n$ (wypisanej jako macierz), w funkcji $n = 1, \ldots, 10$, gdzie $P$ to macierz przejścia w jednym kroku dla ergodycznego i okresowego procesu Markowa

♠$_{13}$ Stwórz rzadką macierz $100 \times 100$, której elementy to 1 dla współrzędnych będących liczbami pierwszymi i 0 dla pozostałych.

W raporcie wystarczy podać kod źródłowy i wycinek macierzy np. $10 \times 10$.

♠$_{14}$ Stosując funkcje `Apply[]` i `Reverse[]` przygotuj kod, który pozwoli zmienić kierunek skierowanych krawędzi w wybranym grafie skierowanym.

♠$_{15}$ Stwórz rzadką macierz $10 \times 10$, której $(n, k)$-ty elementy to $k$-ta cyfra po przecinku liczby $\pi^n$.

♠$_{16}$ Napisz program, który dla zadanego $n \in \mathbb{N}_+$ zawraca graf będący siatką kwadratową. Zapisz jego macierz sąsiedztwa.

Do wygenerowania grafu polecam użyć funkcji `Graph[]`.

♠$_{17}$ Napisz skrypt, który tworzy w losowy sposób macierz przejścia procesu Markowa o 20 stanach. Rozstrzygnij jaki typ dynamiki prezentuje.

♠$_{18}$ Dla wybranego procesu ergodycznego $P$ o co najmniej 5 stanach i losowej wartości początkowej $x_0$ narysuj wykres $n(\epsilon)$ zależności minimalnej liczby iteracji $n$, dla których $|P^n x_0 - P^\infty x_0|_2 < \epsilon$.

♠$_{19}$ Posługując się funkcją `GridGraph[]` wyznacz macierz przejścia w jednym kroku dla błądzenia losowego po siatce kwadratowej.

♠$_{20}$ Mając daną macierz przejścia w jednym kroku (definiuje ona w sposób jednoznaczny pewien graf) napisz funkcję zwracającą listę stopni jego wierzchołków. Zilustruj na przykładzie działanie tej funkcji.

♠$_{21}$ Mając do dyspozycji wybrany graf nieskierowany zadany za pomocą funkcji `Graph[]` napisz funkcję zwracającą listę stopni jego wierzchołków. Zilustruj działanie tej funkcji na przykładzie.

♠$_{22}$ Dla wybranego procesu ergodycznego $P$ i losowej wartości początkowej $x_0$ narysuj wykres $\epsilon(n)$ zależności $\epsilon = |P^n x_0|_2$ od liczby iteracji $n$.

♠$_{23}$ Wykorzystując funckję `AbsoluteTiming[]` Porównaj czasy iteracji wybranej macierzy z zajęć przy użyciu funkcji `MatrixPower[]` i `Nest[]`. Dla obu metod narysuj wykres zależności czasu od liczby iteracji.

♠$_{24}$ Przy pomocy funkcji `SparseArray[]` wygeneruj macierz sąsiedztwa w siatce kwadratowej $10 \times 10$.

♠25 Dla ogólnej postaci macierzy przejścia w jednym kroku dwustanowego procesu Markowa

$$\begin{pmatrix} p & 1-q \\ 1-p & q \end{pmatrix},$$

gdzie $p, q \in [0,1]$, wyznacz symboliczny wzór na rozkład stacjonarny. Wyznacz warunek na $p$ i $q$ kiedy ta macierz jest podwójnie stochastyczna, wypisz wzór na rozkład stacjonarny dla tego przypadku.

♠26 Dla rozkładu prawdopodobieństwa danego przez $(p, 1-p)$, gdzie $p \in [0,1]$, przy pomocy funkcji `D[]` znajdź i określ ekstremum entropii danej przez $\mathcal{H}(p_1, p_2, \ldots, p_n) = -\sum_{k=1}^{n} -p_k \log p_k$.

♠27 Dla wybranego ergodycznego łańcucha Markowa o liczbie stanów co najmniej 10 narysuj wykres pierwszej współrzędnej w funkcji liczby iteracji (co najmniej 20).

## *Równanie Van der Pola*

ZAPISAŁEM SIĘ DO TAKICH, CO TYM INTERESEM KRĘCĄ,
MÓWIĄ, ŻE SIĘ TO OPŁACA I POMAGA W STUDIOWANIU!

— J.KACZMARSKI

Celem zajęć będzie zbadanie dynamiki zadanej przez następujące równanie różniczkowe

$$\ddot{x} + f'(x)\,\dot{x} + ax = 0, \tag{1}$$

gdzie $a > 0$, a $f : \mathbb{R} \to \mathbb{R}$ jest nieparzysta i odpowiednio gładka. Równanie (1), będące podtypem tak zwanych równań Liénarda (patrz [3]), rozważymy dla ustalonej funkcji $f$ danej wzorem

$$f(x) = -\mu\left(x - \frac{x^3}{3}\right), \quad \text{dla} \quad \mu \in \mathbb{R}, \tag{2}$$

Dla której równanie (1) przyjmuje postać

$$\ddot{x} - \mu\left(1 - x^2\right)\dot{x} + ax = 0, \qquad a > 0,\ \mu \in \mathbb{R}. \tag{3}$$

Równanie (1) pojawia się przy opisie natężenia prądu występującego w obwodzie RLC, gdy rezystor posiada nieliniową charakterystykę prądowo-napięciową. Obwód z rezystorem nazywanym *diodą tunelową* można w przybliżeniu opisać równaniem (3), które nazywane jest *równaniem van der Pola* (więcej informacji można znaleźć w [4]).

### *Plan badań*

1. Uproszczenie $a$ w (1) poprzez zauważenie, że parametr ten można dowolnie zmieniać skalując czas. Tym samym jeżeli interesuje nas dynamika trajektorii to możemy położyć $a = 1$.
   **Rozwiązanie:** Pokazujemy, że rozwiązania równań

   $$\ddot{x} + \frac{1}{\sqrt{a}} f'(x)\dot{x} + x = 0,$$
   $$\ddot{x} + f'(x)\dot{x} + ax = 0$$

   są izomorficzne z dokładnością do przeskalowania czasu. Dzięki temu uzyskamy informacje, iż ich portrety fazowe również są jedynie przeskalowane.

2. Pokazujemy, że znak przy funkcji $f'(x)$ nie jest ważny bo odpowiada tylko za odwrócenie czasu. Oznacza to, że jeżeli poznamy dynamikę równania (3) dla $a = 1$ i $\mu > 0$ to jesteśmy w stanie wnioskować o dynamice dla dowolnych $a > 0$ i $\mu \in \mathbb{R} \setminus \{0\}$.

**Uwaga:** Przypadek $\mu = 0$ trzeba rozważyć oddzielnie jako trywialny.

3. Korzystamy z poprzednich wyników oraz zapisujemy (3) w postaci układu równań różniczkowych zwyczajnych

   $$\begin{cases} \dot{x} = \mu\left(x - \frac{x^3}{3}\right) + y, \\ \dot{y} = -x. \end{cases} \qquad \mu > 0 \tag{4}$$

4. Przy użyciu funkcji `Solve[]` znajdź rozwiązania stacjonarne układu (4). Następnie przy użyciu funkcji `D[]` dokonaj linearyzacji (4). Zauważ, że dla każdej funkcji $f(x)$ nieparzystej możemy znaleźć punkty stacjonarne.

5. Badamy stabilność zlinearyzowanego punktu równowagi za pomocą funkcji `Eigenvalues[]` albo `Eigensystem[]`. Następnie opierając się na twierdzeniu Grobmana-Hartmana analizujemy stabilność punktu równowagi układu (3).

**Uwaga:** Dla $\mu = 0$ nie można stosować twierdzenia G-H.

6. Przy ustalonej wartości $\mu = 2$:

   - rysujemy portret fazowy dla układu (4),
   - wykorzystujemy `NDSolve[]` aby rozwiązać (4) z ustalonymi warunkami początkowymi $(x(0), y(0)) = (2, 3)$ i prezentujemy rozwiązanie w postaci dwóch funkcji $(x(t), y(t))$ na jednym wykresie (z legendą),
   - dla danych z poprzedniego punktu rysujemy trajektorię (najpierw na oddzielnym wykresie a potem na wykresie portretu fazowego).

7. Zestawienie w jednym bloku `Manipluate[]` wykresów portretu fazowego z zaznaczonym (czerwoną kropką) punktem stacjonarnym oraz z dwoma zaznaczonymi trajektoriami, tak aby:

   - umożliwić zmianę wartości $\mu > 0$ w zakresie $[0, 5]$,
   - umożliwić wpisanie punktów startowych $(x_1(0), y_1(0))$ oraz $(x_2(0), y_2(0))$ dla dwóch różnych trajektorii,
   - umożliwić, za pomocą przycisków, wybór górnej granicy czasów $t_1$, $t_2$ dla których rozważane są trajektorie z poprzedniego punktu (niech $t_1, t_2 \in \{1, 2, 4, 8, 16, 32\}$),

8. Przy wykorzystaniu manipulate'a z punktu **7.** zbadanie zachowania trajektorii układu (3). Następnie podsumowanie otrzymanych wyników dla oscylatora van der Pola i sformułowanie hipotezy o granicznym zachowaniu trajektorii niezerowych.

9. **(Dodatkowe)** Za pomocą bloku `Manipulate[]` z punktu **7.** zbadać inne funkcje nieparzyste $f(x)$ zamiast ograniczać się do rozważanej (2). Z ciekawych funkcji polecam:

$$f(x) = \mu x - \sin(x), \quad \text{dla} \quad \mu \in \left[\frac{1}{100}, \frac{1}{10}\right].$$

## *Pytania i problemy*

$\clubsuit_1$ Jak, przy pomocy funkcji `D[]` wyznaczyć macierz pochodnych funkcji dwóch zmiennych?

♣2 Wyjaśnij dlaczego kod:

```
a = MatrixForm[{{1, 0}, {0, 1}}];
a.a
```

nie zwraca wyniku $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

♣3 Wyjaśnij oraz zilustruj przykładem działanie funkji `MatrixForm[]`.

♣4 Wyjaśnij oraz zilustruj przykładem działanie funkcji `Eigenvalues[]`.

♣5 Wyjaśnij oraz zilustruj przykładem działanie funkcji `Eigensystem[]`.

♣6 Wyjaśnij oraz zilustruj przykładem działanie funkjic `Eigenvectors[]`.

♣7 Wyjaśnij czym różnią się następujące działania na macierzach: . (kropka) i $*$ (gwiazdka).

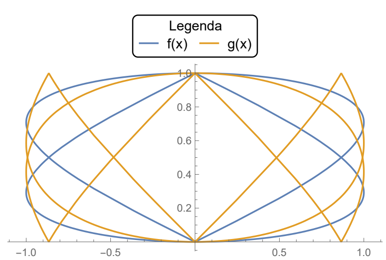
♣8 Wyjaśnij oraz zilustruj przykładem działanie funkcji `ParametricPlot[]`.

♣9 Wyjaśnij w jaki sposób wymusić zakres wyświetlanych osi dla polecenia `ParametricPlot[]`.

♣10 Wyjaśnij w jaki sposób do wykresu danego kodem

```
ParametricPlot[{{Sin[2t], Mod[Sin[t],1]},
        {Sin[2t], Mod[2 Sin[t], 1]}}, {t, 0, 2 Pi}]
```

dodać legendę tak aby efekt wyglądał następująco:




♣11 Wyjaśnij oraz zilustruj przykładem działanie funkcji `VectorPlot[]`.

♣12 Wyjaśnij jak zmienić wielkosć wektorów generowanych przy pomocy `VectorPlot[]`.

♣13 Wyjaśnij jak działa opcja `StreamPoints` możliwa do ustawienia przy wykresach generowanych za pomocą funkcji `VectorPlot[]`.

♣14 Wyjaśnij oraz zilustruj przykładem działanie funkcji `Solve[]`.

♣15 Wyjaśnij oraz zilustruj przykładem działanie funkcji `NSolve[]`.

♣16 Jaka jest różnica między funkcjami `Solve[]` a `NSolve[]`? Podaj przykład ją ilustrujący.

♣17 Wyjaśnij oraz zilustruj przykładem działanie funkcji `DSolve[]`.

♣18 Wyjaśnij w jaki sposób odwołujemy się do rozwiązań równań algebraicznych otrzymanych za pomocą funkcji `Solve[]`.

♣19 Wyjaśnij oraz zilustruj przykładem działanie funkcji `NDSolve[]`.

♣20 Jaka jest różnica między funkcjami `DSolve[]` a `NDSolve[]`? Podaj przykład ją ilustrujący.

♣21 Wyjaśnij w jaki sposób odwołujemy się do rozwiązań RRZ otrzymanych za pomocą funkcji `NDSolve[]`.

♣22 Wyjaśnij oraz zilustruj przykładem działanie funkcji `ReplaceAll[]`.

♣23 Wyjaśnij oraz zilustruj przykładem działanie funkcji `Evaluate[]`.

♣24 Wyjaśnij w jaki sposób za pomocą funkcji `Plot[]` otrzymać wykres funkcji `Series[Cos[x], {x, 0, 4}]`.

Oczywiście aby uzyskać wykres należy opuścić resztę Peano.

♣25 Wyjaśnij oraz zilustruj przykładem działanie funkcji `Series[]`.

♣26 Wyjaśnij oraz zilustruj przykładem działanie funkcji `FullSimplify[]`.

♣27 Wyjaśnij oraz zilustruj przykładem działanie funkcji `Expand[]`.

**Uwaga:** Pamiętaj aby w rozwiązaniach ♠ zastosować opcje zwiększające czytelność (a niekiedy nawet nadające sens) wykresów. Są to opcje takie jak: `PlotRange`, `AxesOrigin`, `PlotStyle`, `PlotLegends` itp.

♠1 Narysuj portret fazowy układu

$$\begin{cases} \dot{x} = -x + \frac{2}{\pi}\operatorname{Arctg}(30\pi x), \\ \dot{y} = -y + \frac{2}{\pi}\operatorname{Arctg}(30\pi y), \end{cases} \qquad (x, y) \in [-2, 2]^2.$$

♠2 Narysuj portret fazowy układu

$$\begin{cases} \dot{x} = y \sin(x), \\ \dot{y} = x \cos(y), \end{cases} \qquad (x, y) \in [-2\pi, 2\pi]^2.$$

♠3 Narysuj portret fazowy układu

$$\begin{cases} \dot{x} = \sin(y), \\ \dot{y} = \sin(x), \end{cases} \qquad (x, y) \in [-2\pi, 2\pi]^2.$$

♠4 Narysuj portret fazowy oraz opisz dynamikę zadaną przez układ:

$$\begin{cases} \dot{x} = \begin{cases} x, & x^2+y^2>1, \\ 0, & x^2+y^2=0, \\ -4x, & x^2+y^2<1, \end{cases} \\ \dot{y} = \begin{cases} y, & x^2+y^2>1, \\ 0, & x^2+y^2=0, \\ -4y, & x^2+y^2<1, \end{cases} \end{cases} \qquad (x,y) \in [-1,1]^2.$$

♠5 Znajdź punkty stacjonarne układu

$$\begin{cases} \dot{x} = x^2 - 2xy + y^2, \\ \dot{y} = y^2 - 3x - 3y + 8. \end{cases}$$

♠6 Znajdź punkty stacjonarne układu

$$\begin{cases} \dot{x} = \sin(y), \\ \dot{y} = y^2 + x^2 - 4. \end{cases}$$

♠7 Znajdź punkty stacjonarne układu

$$\begin{cases} \dot{x} = xyz + \sqrt{2}z, \\ \dot{y} = x^2 - y^2 + z^2, \\ \dot{z} = z - x. \end{cases}$$

♠8 Znajdź punkty stacjonarne układu

$$\begin{cases} \dot{x} = e^y + x + y, \\ \dot{y} = xy. \end{cases}$$

♠9 Rozważ układ:

$$\begin{cases} \dot{x} = x^2 - 2xy + y^2, \\ \dot{y} = y^2 - 3x - 3y + 8. \end{cases}$$

Narysuj rozwiązanie tego układu w czasie $t \in [0, 50]$ startujące z punktu $(x(0), y(0)) = (1, 3)$. Na wykresie zamieść legendę.

♠10 Rozważ układ:

$$\begin{cases} \dot{x} = y + \frac{2}{\pi}\operatorname{Arctg}(30\pi x), \\ \dot{y} = -x + \frac{2}{\pi}\operatorname{Arctg}(30\pi y). \end{cases}$$

Narysuj rozwiązanie tego układu w czasie $t \in [0, 50]$ startujące z punktu $(x(0), y(0)) = (1, 3)$. Na wykresie zamieść legendę.

♠11 Rozważ układ:

$$\begin{cases} \dot{x} = \sin(y), \\ \dot{y} = \cos^2(x). \end{cases}$$

Narysuj rozwiązanie tego układu w czasie $t \in [0, 50]$ startujące z punktu $(x(0), y(0)) = (0, 0)$. Na wykresie zamieść legendę.

♠12 Rozważ układ:

$$\begin{cases} \dot{x} = z + y + \frac{2}{\pi}\mathrm{Arctg}(30\pi x), \\ \dot{y} = -x + \frac{2}{\pi}\mathrm{Arctg}(30\pi y), \\ \dot{z} = t/25 \end{cases}$$

Narysuj rozwiązanie tego układu w czasie $t \in [0, 70]$ startujące z punktu $(x(0), y(0), z(0)) = (1, 2, 3)$. Na wykresie zamieść legendę.

♠13 Na portrecie fazowym układu

$$\begin{cases} \dot{x} = x^2 - 2xy + y^2, \\ \dot{y} = y^2 - 3x - 3y + 8. \end{cases}$$

umieść trajektorię, dla czasu $t \in [0, 50]$, startującą z punktu $(x(0), y(0)) = (1, 3)$.

Przypominamy, że w zadaniach ♠13-♠16 trzeba szczególnie zadbać o czytelność wykresu tak aby trajektorie były dobrze widoczne. Jeżeli ktoś decyduje się na czarno-biały wydruk to proszę też to uwzględnić przy dobieraniu odcieni szarości

♠14 Na portrecie fazowym układu

$$\begin{cases} \dot{x} = y + \frac{2}{\pi}\mathrm{Arctg}(30\pi x), \\ \dot{y} = -x + \frac{2}{\pi}\mathrm{Arctg}(30\pi y). \end{cases}$$

umieść trajektorię, dla czasu $t \in [0, 50]$, startującą z punktu $(x(0), y(0)) = (1, 3)$.

♠15 Na portrecie fazowym układu

$$\begin{cases} \dot{x} = \sin(y), \\ \dot{y} = \cos^2(x), \end{cases}$$

umieść dwie trajektorie. Jedną startującą z punktu $(x(0), y(0)) = (0, 1)$ a drugą z punktu $(x(0), y(0)) = (0, 0)$. Rozważ czas $t \in [0, 15]$.

♠16 Na portrecie fazowym układu

$$\begin{cases} \dot{x} = y + z + \frac{2}{\pi}\mathrm{Arctg}(30\pi x), \\ \dot{y} = -x + \frac{2}{\pi}\mathrm{Arctg}(30\pi y), \\ \dot{z} = 1, \end{cases}$$

umieść trajektorię, dla czasu $t \in [0, 50]$, startującą z punktu $(x(0), y(0), z(0)) = (1, 2, 3)$.

W zadaniu ♠16 zalecany jest kolorowy wydruk (bardzo ułatwi uzyskanie czytelności wykresu).

♠17 Znajdź rozwiązanie ogólne równania:

$$x''(t) + 2x'(t) + x(t) = \cos(t).$$

W zadaniach ♠17–♠20 przydatna może okazać się funkcja `Simplify[]`.

♠18 Znajdź rozwiązanie ogólne równania:

$$x''(t) + 2x'(t) + x(t) = \sin(t)\cos(t).$$

♠19 Znajdź rozwiązanie ogólne równania:

$$x''(t) + 2x'(t) + x(t) = e^t.$$

♠20 Znajdź rozwiązanie ogólne równania:

$$x''(t) + 2x'(t) + x(t) = te^t.$$

♠21 Przy użyciu funkcji `ParametricPlot[]` narysuj spiralę Archimedesa.

♠22 Przy użyciu funkcji `ParametricPlot[]` narysuj cykloidę.

♠23 Przy użyciu funkcji `ParametricPlot[]` narysuj kardioidę.

♠24 Przy użyciu funkcji `ParametricPlot[]` narysuj czterolistną koniczynę (Quadrifolium).

♠25 Rozwiąż równanie $x''(t) = -k^2 x(t)$, przewidując jego rozwiązanie w postaci funkcji analitycznej $x(t) = \sum_{n=0}^{\infty} a_n t^n$, przekształcając równanie różniczkowe na równanie rekurencyjne na $a_n$ i rozwiązując je (np. funkcją `RSolve[]`) aby następnie zsumować otrzymany szereg, otrzymując ogólne rozwiązanie wyjściowego równana.

♠26 Rozwiąż równanie $x''(t) = k^2 x(t)$, przewidując jego rozwiązanie w postaci funkcji analitycznej $x(t) = \sum_{n=0}^{\infty} a_n t^n$, przekształcając równanie różniczkowe na równanie rekurencyjne na $a_n$ i rozwiązując je (np. funkcją `RSolve[]`) aby następnie zsumować otrzymany szereg, otrzymując ogólne rozwiązanie wyjściowego równana.

♠27 Rozwiąż równanie $\ddot{x} = -k^2 x$, a następnie zapisz rozwiązanie w postaci wykładniczej i trygonometrycznej (wykorzystując w tym celu wbudowane funkcje środowiska Wolfram Mathematica.

## *Literatura*

[1] A. Iwanik, J.K. Misiewicz. *Wykłady z procesów stochastycznych z zadaniami: Część pierwsza – Procesy Markowa*. Oficyna Wydawnicza Uniwersytetu Zielonogórskiego, 2009.

[2] A. Lasota, M. C. Mackey. *Chaos, Fractals, and Noise. Stochastic Aspects of Dynamics*. Springer-Verlag, 1994.

[3] A. Palczewski. *Równania różniczkowe zwyczajne*. Wydawnictwo Naukowo-Techniczne, 2004.

[4] L. Wang. *TReliable design of tunnel diode and resonant tunnelling diode based microwave sources*. PhD thesis, The School of Engineering University of Glasgow, 2011.