KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Matematyka na UWr – studia pełne możliwości

Pracownia graficzna wizualizacji obiektów matematycznych

program zajęć na rok 2012

Dariusz Buraczewski
Tomasz Elsner

Wrocław, styczeń 2012

Spis treści

Pracownia 1

Wprowadzenie do programu *Mathematica*

Przydatne polecenia

`Sin[x], Cos[x], Tan[x], Cot[x]`	funkcje trygonometryczne
`Sqrt[x], Log[x], x^n, Pi, E`	$\sqrt{x}$, $\log x$, x^n, π, e
`Factor[x^3+x]`	rozłóż wielomian $x^3 + x$ na czynniki
`Expand[(a+b)^7]`	rozwiń wielomian
`Solve[{2x+3y==4, 5x+6y==7},{x,y}]`	rozwiąż układ równań z niewiadomymi x i y
`NSolve[x^5-3x+1==0, x]`	rozwiąż numerycznie równanie z niewiadomą x
`FindRoot[f[x]==g[x],{x,5}]`	znajdź numerycznie rozwiązanie w pobliżu $x = 5$
`...  // N`	podaj wartość numeryczną (przybliżoną)
`a = 3`	zdefiniuj stałą $a = 3$
`f[x_] = 5x+Sin[x]`	zdefiniuj funkcję $f(x) = 5x + \sin x$
`Clear[a]`	wyczyść przypisanie stałej/funkcji a
`D[f[x],{x,2}]`	oblicz drugą pochodną funkcji $f(x)$ zmiennej x
`Limit[f[x], x->Infinity]`	$\lim_{x\to\infty} f(x)$
`Plot[f[x],{x,0,1}]`	narysuj wykres $y = f(x)$ dla $0 \le x \le 1$

Wprowadzenie

Przykład 1. *Rozwiąż równanie* $x^3 - 2x + 1 = 0$. *Następnie znajdź wartość numeryczną pierwiastków.*

```
Solve[x^3 - 2x + 1 == 0]
% // N
```

Przykład 2. *Znajdź wartość* $f''(\frac{1}{4})$, *gdzie* $f(x) = \dfrac{x+4}{(x^2+5x+7)^4}$.

```
f[x_] = (x + 4)/(x^2 + 5 x + 7)^4;
g[x_] = D[f[x], {x, 2}]
g[1/4]
g[1/4] // N
```

Przykład 3. *Znajdź pierwsze 3 nieujemne rozwiązania równania* $\tan x = x$.

```
Solve[x == Tan[x], x]
Plot[{Tan[x], x}, {x, 0, 10}]
FindRoot[x == Tan[x], {x, 0}]
FindRoot[x == Tan[x], {x, 4}]
FindRoot[x == Tan[x], {x, 8}]
FindRoot[x == Tan[x], {x, 7}]
```

Zadania

Zadanie 1. *Oblicz*

$$\frac{\sin\frac{\pi}{12}}{\sqrt{\tan 1+\frac{3}{4}}}$$

Znajdź wartość numeryczną tego wyrażenia.

Zadanie 2. *Narysuj wykres funkcji* $f(x)=x^x$ *i znajdź* $\lim_{x\to 0} f(x)$.

Zadanie 3. *Narysuj we wspólnym układzie współrzędnych wykresy funkcji* $\sin x$, $\sin 2x$ *i* $\sin 3x$.

Zadanie 4. *Niech*

$$f(x)=\frac{x\log x}{x^3+3x-1}$$

Oblicz $f''(2)$.

Zadanie 5. *Znajdź trzy pierwsze nieujemne rozwiązania równania* $x+\log x=\tan x$.

Zadanie 6. *Znajdź wszystkie punkty, w których funkcje* $f(x)=x+2$ *i* $g(x)=x^2-2\sin x+3$ *się przecinają.*

Pracownia 2

Wykresy funkcji

Przydatne polecenia

`Limit[f[x], x->0, Direction->1]`	$\lim_{x\to 0^+} f(x)$
`Limit[f[x], x->0, Direction->-1]`	$\lim_{x\to 0^-} f(x)$
`g[x_]=D[f[x],x]`	zdefiniuj funkcję $g(x) := f'(x)$
`Plot[{f[x],g[x]},{x,0,1},`	narysuj wspólny wykres $f(x)$ i $g(x)$ dla $0 \le x \le 1$ ...
`  PlotRange->{{0,Pi},{0,1}}]`	... ustal wielkość rysunku na $0 \le x \le \pi$, $0 \le y \le 1$ (opcjonalne)
`Animate[Plot[...],{a,0,5},`	wykonuj animację wykresu dla parametru $0 \le a \le 5$ ...
`  AnimationDirection->Backward]`	... zmniejszając wartość parametru (opcjonalne)
`Manipulate[Plot[...],{a,0,5}]`	pozwól ręcznie zmieniać wartość parametru $0 \le a \le 5$

Wprowadzenie

Przykład 1. *Narysuj wykres funkcji* $\sin ax$ *dla różnych wartości* a*. Wprowadź możliwość manipulacji wartością parametru.*

```
Manipulate[Plot[Sin[a*x], {x, 0, 2 Pi}], {a, 1, 10}]
```

Przykład 2. *Narysuj animację pokazującą drgającą strunę opisywaną funkcją* $f(x,t) = \cos 5t \sin 2x$ *dla* $0 \le x \le 2\pi$.

```
Animate[Plot[Cos[5 t] Sin[2 x], {x, 0, 2 Pi},
  PlotRange -> {{0, 2 Pi}, {-1, 1}}], {t, 0, Infinity}]
```

Zadania

Zadanie 1. *Narysuj wykres funkcji* $\sin ax + \sin bx$ *dla różnych wartości* a *i* b*. Wprowadź możliwość manipulowania wartościami parametrów* a *i* b *(użyj opcji* `Manipulate`*).*

Zadanie 2. *Dla każdej z funkcji* $f(x) = \sin\frac{1}{x}$, $g(x) = x\sin\frac{1}{x}$, $h(x) = x^2\sin\frac{1}{x}$*:*

(a) *narysuj wykres i przyjrzyj się (pomniejszając obserwowany obszar opcją* `PlotRange`*) jej zachowaniu w pobliżu zera;*

(b) *oblicz granicę w zerze;*

(c) *oblicz pochodną;*

(d) *narysuj wykres pochodnej;*

(e) *oblicz granicę pochodnej w zerze.*

Zadanie 3. *Sprawdź na ile dobrym przybliżeniem funkcji trygonometrycznych dla małych kątów x są:* $\sin x \cong x$, $\cos x \cong 1$, $\tan x \cong x$, *tzn.*

(a) *dla każdej z par narysuj na wspólnym wykresie funkcję trygonometryczną oraz przybliżającą ją funkcję liniową,*

(b) *przy pomocy wykresów funkcji* $x - \sin x$, $1 - \cos x$ *i* $\tan x - x$ *oceń dla jakich kątów wspomniane przybliżenie daje dokładność 2 cyfr po przecinku,*

(c) *sprawdź dla jakich kątów błąd przybliżenia nie przekracza 1%.*

Zadanie 4. *Narysuj animację pokazującą styczną jako granicę siecznych dla dowolnie wybranej funkcji, tzn.*

(a) *wybierz funkcję, np.* $f(x) = \sin x$ *i narysuj jej wykres,*

(b) *wybierz punkt* x_0 *i narysuj (na jednym rysunku z wykresem* $f(x)$*) prostą przechodzącą przez* $(x_0, f(x_0))$ *i* $(x_0 + h, f(x_0 + h))$,

(c) *wykonaj animację, która będzie zmniejszać* h *od 1 do 0.*

Rozwiązania

Zadanie 1

```
Manipulate[Plot[Sin[a*x] + Sin[b*x], {x, 0, 2 Pi}], {a, 1, 10}, {b, 1, 10}]
```

Zadanie 2

```
f[x_] = x^2 Sin[1/x];
Limit[f[x], x -> 0]
g[x_] = D[f[x], x];
Plot[g[x], {x, -1, 1}]
Limit[g[x], x -> 0]
```

Zadanie 3

```
Plot[{x - Sin[x], 0.005}, {x, 0, Pi/2}, PlotRange -> {{0, Pi/2}, {0, 0.01}}]
```

Zadanie 4

```
Animate[Plot[{Sin[x], Sin[1] + (Sin[1 + h] - Sin[1])/h*(x - 1)}, {x, -Pi, Pi},
  PlotRange -> {{-Pi, Pi}, {-2, 2}}], {h, 0, 1}, AnimationDirection -> Backward]
```

Pracownia 3

Szeregi

Przydatne polecenia

`Table[f[n],{n,25}]`	wypisz ciąg $f(1), f(2), \ldots, f(25)$
`Table[f[n],{n,10,20}]`	wypisz ciąg $f(10), f(11), \ldots, f(20)$
`Prime[n]`	n-ta liczba pierwsza
`Binomial[n,k]`	symbol Newtona $\binom{n}{k}$
`Sum[k^2, {k,3,n}]`	$\sum_{k=3}^{n} k^2$
`Sum[k^3, {k,n}]`	$\sum_{k=1}^{n} k^3$ (1 jest domyślnym dolnym indeksem)
`Sum[1/n^3, {n,4,Infinity}]`	$\sum_{n=4}^{\infty} \frac{1}{n^3}$
`Sum[x^n, {n,1,Infinity},`	oblicz sumę szeregu $\sum_{n=1}^{\infty} x^n$ ...
`GenerateConditions->True]`	... i podaj warunek zbieżności
`SumConvergence[1/n^2, n]`	sprawdź czy szereg $\sum_n \frac{1}{n^2}$ jest zbieżny
`Reduce[n^2+m*n+m^3>=0,{m,n}]`	rozwiąż nierówność z niewiadomymi m i n
`Reduce[x^2+y^3==x,{x,y}]`	rozwiąż równanie z niewiadomymi x i y
`... // N`	oblicz przybliżoną (numeryczną) wartość wyrażenia
`N[...,5]`	wartość numeryczną z dokładnością do 5 cyfr znaczących
`Pi, E`	stałe π i e
`Simplify[%]`	uprość ostatnie wyrażenie
`Expand[(a+b)^7]`	rozwiń wielomian

Wprowadzenie

Przykład 1. *Oblicz* $\sum_{k=1}^{\infty} \frac{k}{2^k}$.

```
Sum[k/2^k,{k,1,Infinity}]
```

Przykład 2. *Sprawdź czy prawdziwy jest wzór* $1 \cdot 2^1 + 2 \cdot 2^2 + \cdots + n \cdot 2^n = (n-1) \cdot 2^{n+1} + 2$.

```
Sum[k*2^k, {k, 1, n}] == (n - 1) 2^(n + 1) + 2
Simplify[%]
```

Przykład 3. *Wypisz 25 początkowych liczb pierwszych. Oblicz ich sumę.*

```
Table[Prime[i], {i, 25}]
Sum[Prime[i], {i, 25}]
```

Zadania

Zadanie 1. *Oblicz*

$$\sum_{k=1}^{\infty} \frac{1}{k^2}, \qquad \sum_{k=1}^{\infty} \frac{1}{k^3}$$

Zadanie 2. *Oblicz*

$$\sum_{k=0}^{n} \binom{n}{k}$$

Zadanie 3. *Sprawdź dla jakich wartości k poniższe szeregi są zbieżne:*

$$\sum \frac{1}{n^k}, \qquad \sum \frac{1}{n(\log n)^k}, \qquad \sum \frac{1}{n \log n (\log\log n)^k}$$

Zadanie 4. *Sprawdź czy prawdziwy jest wzór* $1^3 + 2^3 + \cdots + n^3 = (1 + 2 + \cdots + n)^2$.

Zadanie 5. *Oblicz*

$$\frac{1}{3} + \frac{1}{15} + \frac{1}{35} + \frac{1}{63} + \cdots + \frac{1}{4n^2 - 1}$$

Zadanie 6. *Oblicz promień zbieżności szeregu potęgowego*

$$\sum \frac{4^{n+5} x^{3n+7}}{n 6^{2n}}$$

Zadanie 7. *Przypomnijmy, że stała Eulera jest zdefiniowana jako granica ciągu* $\sum_{k=1}^{n} \frac{1}{k} - \log n$. *Oblicz z jak największą dokładnością potrafisz wartość tej stałej. Porównaj twój wynik z rzeczywistą wartością (znajdź ją w Internecie lub Mathematice).*

Zadanie 8.

(a) *Znajdź wzór na sumę k-tych potęg kolejnych liczb naturalnych, tj.* $1^k + 2^k + \cdots + n^k$, *dla* $k = 1, 2, 3, 4$.

(b) *Przyjrzyj się otrzymanym wynikom. Zauważ, że zawsze otrzymujesz wielomian zmiennej n i postaw hipotezę opisującą asymptotyczne zachowanie tego wielomianu (dla dużych n).*

(c) *Sprawdź na ile dobre przybliżenie wyniku otrzymujesz stosując przybliżony wzór z punktu 2.*

Zadanie 9. *Niech $\pi(n)$ oznacza liczbę liczb pierwszych nie większych niż n, np. $f(10) = 4$. Co sądzisz o następującej hipotezie: dla dużych wartości n zachodzi $\pi(n) \cong \frac{n}{\log n}$?*

Zadanie 10. *Istnieje wiele wzrorów na liczbę π. Oto kilka z nich:*

$$\frac{\pi}{4} = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

$$\frac{\pi}{2} = \prod_{n=1}^{\infty} \frac{4n^2}{4n^2 - 1}$$

$$\frac{\pi^2}{6} = \sum_{n=1}^{\infty} \frac{1}{n^2}$$

$$\pi = \sum_{n=0}^{\infty} \frac{1}{16^n} \left(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6} \right)$$

$$\frac{1}{\pi} = 12 \sum_{n=0}^{\infty} \frac{(-1)^n (6n)!(13\,591\,409 + 545\,140\,134n)}{(3n)!(n!)^3 640\,320^{3n+3/2}}$$

Który z nich uważasz za najlepszy do wyznaczenia wartości π?

Rozwiązania

Zadanie 1

```
Sum[1/k^2, {k, 1, Infinity}]
Sum[1/k^3, {k, 1, Infinity}]
```

Zadanie 2

```
Sum[Binomial[n, k], {k, 0, n}]
```

Zadanie 3

```
SumConvergence[1/n^k, n]
SumConvergence[1/(n*Log[n]^k), n]
SumConvergence[1/(n*Log[n](Log[Log[n]])^k), n]
```

Zadanie 4

```
Sum[k^3, {k, 1, n}] == Sum[k, {k, 1, n}]^2
```

Zadanie 5

```
Sum[1/(4 k^2- 1 ), {k, 1, n}]
```

Zadanie 6

```
SumConvergence[4^(n + 5) x^(3n + 7)/(n*6^(2n)), n]
```

Zadanie 7

```
N[Sum[1/k, {k, 1, 100000}] - Log[100000], 10]
N[EulerGamma, 10]
```

Zadanie 8

```
Table[Expand[Sum[m^k, {m, 1, n}]], {k, 1, 4}]
```

Zadanie 9

```
Table[k Log[Prime[k]]/Prime[k] // N, {k, 1, 1000000000, 10000000}]
```

Zadanie 10

```
Table[Floor[Log[10, Abs[
  N[Sum[1/16^n(4/(8n + 1) - 2/(8n + 4)-1/(8n + 5)-1/(8n + 6)), {n, 0, m}], 1000]
    - N[Pi, 1000]]]], {m, 50}]
Table[Floor[Log[10, Abs[
  N[1/(12Sum[((-1)^k(6k)!(13591409 + 545140134k))/((3k)!(k!)^3640320^(3k + 3/2)),
    {k, 0, n}]), 1000] - N[Pi, 1000]]]], {n, 50}]
```

Pracownia 4

Liniowe rekurencje

Przydatne polecenia

`{{a,b},{c,d}}`	macierz $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$
`{a,b}`	wektor (a, b)
`A.B`	iloczyn macierzy A i B
`A.v`	iloczyn macierzy A i wektora v
`MatrixPower[A,n]`	potęga macierzy A^n
`A // MatrixForm`	zapisz macierz A w przyjaznej postaci graficznej
`... // N`	wartość numeryczną (przybliżoną) wyrażenia
`Simplify[...]`	przedstaw wyrażenie w najprostszej postaci
`Limit[f[n], n->Infinity]`	oblicz $\lim_{n\to\infty} f(n)$
`f[n_]=3n+1`	zdefiniuj funkcję $f(n) = 3n + 1$
`Clear[f]`	usuń z pamięci definicję funkcji lub zmiennej f

Wprowadzenie

Przykład 1. *Ciąg Fibonacciego F_n jest zdefiniowany rekurencyjnie w następujący sposób:*

$$\begin{cases} F_1 = 1 \\ F_2 = 1 \\ F_n = F_{n-1} + F_{n-2} \ \textit{dla}\ n \geq 3 \end{cases}$$

(a) *Oblicz wyrazy ciągu F_{100}, F_{200}, F_{1000}.*

(b) *Znajdź ogólny wzór na n-ty wyraz ciągu F_n.*

(c) *Oblicz granicę $\lim_{n\to\infty} F_{n+1}/F_n$.*

Zdefiniujmy wektor $v_n = \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix}$ oraz macierz $A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$. Wówczas $v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Zauważmy, że

$$A \cdot v_n = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_{n+1} \\ F_n \end{pmatrix} = \begin{pmatrix} F_{n+1} + F_n \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = v_{n+1}$$

czyli $Av_n = v_{n+1}$. Stąd dostajemy $v_n = Av_{n-1} = A^2 v_{n-2} = \cdots = A^{n-1} v_1$. Podsumowując, n-ty wyraz ciągu Fibonacciego F_n jest równy drugiej współrzędnej wektora $A^{n-1} v_1$:

```
A={{1,1},{1,0}};
F[n_]=(MatrixPower[A,n-1].{1,1})[[2]]
Simplify[%]
Limit[F[n+1]/F[n], n->Infinity]
```

Zadania

Zadanie 1. *Ciąg G_n jest zdefiniowany następująco:*

$$\begin{cases} G_1 = 4 \\ G_2 = 7 \\ G_n = G_{n-1} + G_{n-2} \ \textit{dla}\ n \geq 3 \end{cases}$$

(a) *Oblicz wyrazy ciągu G_{100}, $G_{1\,000}$, $G_{10\,000}$.*

(b) *Znajdź ogólny wzór na n-ty wyraz ciągu G_n.*

(c) *Oblicz granicę $\lim_{n\to\infty} G_{n+1}/G_n$.*

Zadanie 2. *Ciąg H_n jest zdefiniowany następująco:*

$$\begin{cases} H_1 = 1 \\ H_2 = 1 \\ H_n = 4H_{n-1} - 3H_{n-2} \ \textit{dla}\ n \geq 3 \end{cases}$$

(a) *Znajdź ogólny wzór na n-ty wyraz ciągu H_n.*

(b) *Oblicz granicę $\lim_{n\to\infty} H_{n+1}/H_n$.*

Zadanie 3. *Porównaj wyniki otrzymane w przykładzie* 1(c) *oraz w zadaniach* 1(c) *i* 2(b). *Który element rekurencji wpływa, a który nie wpływa na asymptotyczne zachowanie ilorazu kolejnych wyrazów ciągu? Postaw hipotezę i przetestuj ją dla kilku przykładów.*

Zadanie 4. *Znajdź ogólny wzór na n-ty wyraz ciągu zdefiniowanego następująco:*

$$\begin{cases} a_1 = 2 \\ a_2 = 1 \\ a_3 = 4 \\ a_n = 3a_{n-1} - 4a_{n-2} + 2a_{n-3} \ \textit{dla}\ n \geq 4 \end{cases}$$

Rozwiązania

Zadanie 1

```
A={{1,1},{1,0}};
G[n_]=(MatrixPower[A,n-1].{4,7})[[2]]
Simplify[%]
Limit[G[n+1]/G[n], n->Infinity]
```

Zadanie 2

```
A={{4,-3},{1,0}};
H[n_]=(MatrixPower[A,n-1].{1,1})[[2]]
Simplify[%]
Limit[H[n+1]/H[n], n->Infinity]
```

Zadanie 4

```
A={{3,-4,2},{1,0,0},{0,1,0}};
a[n_]=(MatrixPower[A,n-1].{2,1,4})[[3]]
Simplify[%]
```

Pracownia 5

Błądzenie losowe

Przydatne polecenia

`{{a,b},{c,d}}`	macierz $\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$
`{a,b}`	wektor (a, b)
`A.B`	iloczyn macierzy A i B
`A.v`	iloczyn macierzy A i wektora v
`MatrixPower[A,n]`	A^n
`A // MatrixForm`	zapisz macierz A w przyjaznej postaci graficznej
`...  // N`	wartość numeryczną (przybliżoną) wyrażenia
`Simplify[...]`	przedstaw wyrażenie w najprostszej postaci
`Round[x, 0.001]`	zaokrąglij x do trzeciej cyfry po przecinku
`f[n_]=3n+1`	zdefiniuj funkcję $f(n) = 3n + 1$
`Clear[f]`	usuń z pamięci definicję funkcji lub zmiennej f
`Table[f[n],{n,6,10}]`	wypisz ciąg liczb: $f(6), f(7), \ldots, f(10)$

Wprowadzenie

Przykład 1. *Do domu prowadzą cztery schodki, oznaczmy je A, B, C, D. Początkowo znajdujemy się na stopniu A (najniższym). Po schodkach wchodzimy wielokrotnie rzucając monetą i za każdym razem wykonując krok uzależniony od wyniku rzutu:*

- *jeśli wypadnie reszka, to schodzimy 1 schodek w dół, a jeśli wypadnie orzeł, to wchodzimy 1 schodek do góry, z tym że:*
- *jeśli znajdujemy się na schodku A (najniższym), to niezależnie od wyniku wchodzimy 1 schodek w górę (w dół nie da się zejść),*
- *jeśli znajdujemy się na schodku D (najwyższym), to odpoczywamy i nigdzie dalej już nie chodzimy.*

Jakie jest prawdopodobieństwo, że wejdziemy na najwyższy schodek w nie więcej niż 10 krokach? A nie więcej niż 20 krokach? A ile kroków musimy zrobić, aby prawdopodobieństwo wejścia na najwyższy schodek wyniosło przynajmniej 90%? Przynajmniej 95%? Przynajmniej 99%?

Oznaczmy:

a_n – prawdopodobieństwo, że po n krokach będziemy znajdowali się na stopniu A
b_n – prawdopodobieństwo, że po n krokach będziemy znajdowali się na stopniu B
c_n – prawdopodobieństwo, że po n krokach będziemy znajdowali się na stopniu C
d_n – prawdopodobieństwo, że po n krokach będziemy znajdowali się na stopniu D

Oczywiście $a_0 = 1$ i $b_0 = c_0 = d_0 = 0$. Zauważmy teraz, że $a_{n+1} = \frac{1}{2}b_n$ (w $(n+1)$-szym kroku znajdziemy się na stopniu A tylko wtedy, gdy w n-tym kroku będziemy na stopniu B i wyrzucimy reszkę). Podobnie obliczamy:

$$\begin{aligned} a_{n+1} &= \tfrac{1}{2}b_n \\ b_{n+1} &= a_n + \tfrac{1}{2}c_n \\ c_{n+1} &= \tfrac{1}{2}b_n \\ d_{n+1} &= \tfrac{1}{2}c_n + d_n \end{aligned}$$

Jeśli wprowadzimy teraz wektor

$$v_n = \begin{pmatrix} a_n \\ b_n \\ c_n \\ d_n \end{pmatrix}$$

to (podobnie jak na poprzednich zajęciach) zauważymy, że $v_{n+1} = Av_n$, gdzie

$$A = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 1 \end{pmatrix}$$

skąd dostajemy $v_n = A^n v_0$, gdzie $v_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$. W zadaniu szukamy wartości d_n, która jest ostatnią (czwartą) współrzędną wektora $A^n v_0$.

```
A={{0,.5,0,0},{1,0,.5,0},{0,.5,0,0},{0,0,.5,1}};
d[n_]=(MatrixPower[A,n].{1,0,0,0})[[4]] // N
d[10]; d[20]
Table[d[n],{n,1,40}]
```

Przykład 2. *Jakie jest prawdopodobieństwo, że rzucając 10 razy monetą wyrzucimy (w którymkolwiek momencie) 3 orły pod rząd?*

To zadanie, wbrew pozorom, jest bardzo podobne do poprzedniego. Spróbujmy opisywać schodki etykietami $\ldots R$, $\ldots RO$, $\ldots ROO$, $\ldots ROOO$, zamiast A, B, C, D. Należy to rozumieć tak: jeśli znajdujemy się na schodku $\ldots ROO$, to ostatnie trzy rzuty dały nam wynik R, O i O. Stajemy na najniższym schodku i rzucamy monetą. Jeśli wyrzucimy reszkę, zostajemy na najniższym schodku, jeśli orła – wchodzimy schodek wyżej. Na każdym kolejnym schodku, wyrzucenie orła oznacza krok do góry, zaś wyrzucenie reszki (uwaga!), spadek na sam dół. Jeśli znajdziemy się na najwyższym schodku, nigdzie już nie idziemy (wyrzuciliśmy już 3 orły pod rząd). W zadaniu pytanie jest o prawdopodobieństwo, że po 10 krokach znajdziemy się na najwyższym schodku.

```
A={{.5,.5,.5,0},{.5,0,0,0},{0,.5,0,0},{0,0,.5,1}};
d[n_]=(MatrixPower[A,n].{1,0,0,0})[[4]];
Round[d[10],0.001]
```

Zadania

Zadanie 1. *Rozwiąż problem postawiony w Przykładzie 1 w sytuacji, gdy schodków jest pięć.*

Zadanie 2. *Jakie jest prawdopodobieństwo, że rzucając monetą 20 razy wyrzucimy 4 orły pod rząd? A ile razy trzeba rzucić, by prawdopodobieństwo wyrzucenia 4 orłów pod rząd wyniosło przynajmniej 90%? A 95%? A 99%?*

Zadanie 3. *Jakie jest prawdopodobieństwo, że rzucając kostką do gry 100 razy, zdarzy nam się wyrzucić 5 szóstek z rzędu?*

Zadanie 4. *Na stole leży 10 kamyków. Rzucamy kostką do gry i jeśli wyrzucimy 1, 2, 3 lub 4 oczka, to zabieramy ze stołu taką liczbę kamyków (jeśli na stole jest mniej kamyków niż liczba wyrzuconych oczek, to zabieramy wszystkie kamyki). Jeśli wyrzucimy 5 lub 6 oczek – tracimy wszystkie zebrane kamyki (odkładamy je z powrotem na stół). Gra kończy się, gdy zbierzemy wszystkie 10 kamyków. Jakie jest prawdopodobieństwo, że gra zakończy się w nie więcej niż 10 ruchach? 20 ruchach? 100 ruchach?*

Zadanie 5. *Zdefiniuj funkcję $p(n)$ obliczającą prawdopodobieństwo zakończenia gry z poprzedniego zadania w* ***dokładnie*** *n ruchach. Wartość oczekiwaną liczby ruchów, w których kończy się gra (tzn. średnią liczbę ruchów, po których kończy się gra) obliczamy według wzoru*

$$\sum_{n=1}^{\infty} n \cdot p(n)$$

Oblicz wartość oczekiwaną liczby ruchów po których kończy się gra z poprzedniego zadania. Jeśli polecenie `Limit` *nie zadziała, użyj polecenia* `Table` *do wypisania odpowiednio dużej liczby sum częściowych.*

Zadanie 6. *Do dziesięciu pojemników ustawionych w rzędzie wrzucamy losowo jedną kulę, białą lub czarną. Prawdopodobieństwo wrzucenia kuli białej do danego pojemnika wynosi $2/5$, a czarnej $3/5$. Oblicz prawdopodobieństwo, że w trzech kolejnych pojemnikach znajdują się kule czarne.*

Zadanie 7. *Gracze A i B rozgrywają ze sobą kolejne partie. Prawdopodobieństwo wygrania pojedynczej partii przez gracza A wynosi $2/3$, a przez gracza B, $1/3$. Gracz, który wygra 3 kolejne partie wygrywa całą rozgrywkę. Oblicz prawdopodobieństwo, że ostatecznym zwycięzcą zostanie gracz B.*

Rozwiązania

Zadanie 1

```
A={{0,.5,0,0,0},{1,0,.5,0,0},{0,.5,0,.5,0},{0,0,.5,0,0},{0,0,0,.5,1}};
e[n_]=MatrixPower[A,n][[5,1]];
Table[Round[e[n],0.001],{n,50}]
```

Zadanie 2

```
A={{.5,.5,.5,.5,0},{.5,0,0,0,0},{0,.5,0,0,0},{0,0,.5,0,0},{0,0,0,.5,1}};
e[n_]=MatrixPower[A,n][[5,1]];
Table[Round[e[n],0.001],{n,50}]
```

Zadanie 3

```
p=1/6; q=5/6;
A={{q,q,q,q,q,0},{p,0,0,0,0,0},{0,p,0,0,0,0},
  {0,0,p,0,0,0},{0,0,0,p,0,0},{0,0,0,0,p,1}};
x[n_]=MatrixPower[A,n][[6,1]];
Round[x[100],0.001]
```

Zadanie 4

```
A=1/6{
  {2,2,2,2,2,2,2,2,2,2,0},
  {1,0,0,0,0,0,0,0,0,0,0},
  {1,1,0,0,0,0,0,0,0,0,0},
  {1,1,1,0,0,0,0,0,0,0,0},
  {1,1,1,1,0,0,0,0,0,0,0},
  {0,1,1,1,1,0,0,0,0,0,0},
  {0,0,1,1,1,1,0,0,0,0,0},
  {0,0,0,1,1,1,1,0,0,0,0},
  {0,0,0,0,1,1,1,1,0,0,0},
  {0,0,0,0,0,1,1,1,1,0,0},
  {0,0,0,0,0,0,1,2,3,4,6}}
MatrixPower[A,10][[11,1]]//N
```

Zadanie 5

```
p[n_]=(MatrixPower[A,n]-MatrixPower[n-1,1])[[11,1]];
Table[Sum[p[k]*k,{k,1,n}],{n,1,100}] // N
```

Zadanie 6

```
A=1/5{
  {2,3,0,0},
  {2,0,3,0},
  {2,0,0,3},
  {0,0,0,5}}
MatrixPower[A,10][[1,4]]//N
```

Pracownia 6

Kredyt i emerytura

Przydatne polecenia

`Manipulate[(1+p)^n,`	manipuluj wyrażeniem $(1+p)^n$ ...
`  {p,2,5},{n,25,35}]`	... dla $2 \leq p \leq 5$ i $25 \leq n \leq 35$
`Plot[f[t], {t,10,20},`	narysuj wykres funkcji $f(t)$ dla $10 \leq t \leq 20$ ...
`  PlotRange->{{1,10},{-3,3}}]`	... ustal rozmiar rysunku $1 \leq x \leq 10$ i $-3 \leq y \leq 3$ (opcjonalne)

Wprowadzenie

Jeśli spłacamy w ratach zaciągnięty w banku kredyt, to każda rata składa się z dwóch części: *raty odsetkowej*, czyli odsetek od bieżącego zadłużenia za ostatni miesiąc i *raty kapitałowej*, czyli spłaty części kredytu. Ponieważ każdego miesiąca nasz dług jest mniejszy, więc rata kredytu powinna maleć. Jednak zazwyczaj bank, dla wygody kredytobiorcy, ustala stałą kwotę raty na cały okres kredytowania[1]. Jak obliczyć wielkość takiej raty?

Przyjmijmy, że zaciągamy kredyt na kwotę K zł oprocentowany na $p\%$ w skali roku i będziemy go spłacać w n równych ratach, po r zł miesięcznie. Wobec tego każdego miesiąca nasze zadłużenie powiększy się o odsetki za ostatni miesiąc, czyli zwiększy się $(1+q)$ razy (gdzie $q = \frac{p}{1200}$), a następnie zmniejszy o kwotę r zapłaconej raty. Zadłużenie w kolejnych miesiącach będzie więc wynosiło:

Miesiąc	Zadłużenie
0	K
1	$K \cdot (1+q) - r$
2	$(K \cdot (1+q) - r) \cdot (1+q) - r = K \cdot (1+q)^2 - r \cdot ((1+q)+1)$
3	$(K \cdot (1+q)^2 - r \cdot ((1+q)+1) \cdot (1+q) - r = K \cdot (1+q)^3 - r \cdot ((1+q)^2 + (1+q) + 1)$
...	...
n	$K \cdot (1+q)^n - r \cdot ((1+q)^{n-1} + \cdots + (1+q)^2 + (1+q) + 1)$

Wysokość raty r powinna być tak ustalona, by wyrażenie w ostatnim wierszu było równe zero (tzn. po zapłacie n-tej raty kredyt będzie całkowicie spłacony). Stosując wzór na sumę szeregu geometrycznego nietrudno wyprowadzić wzór na wysokość raty r w zależności od kwoty kredytu K, liczby rat n i oprocentowania $p\%$ w skali roku.

[1] Ponieważ oprocentowanie kredytu jest zmienne, więc wielkość raty może ulegać zmianie, gdy zmienia się oprocentowanie kredytu.

Zadania

Zadanie 1. *Bank udziela ci kredytu hipotecznego na kwotę 350.000 zł, na 30 lat, oprocentowanego na 6,5% w skali roku.*

(a) *Oblicz miesięczną tego ratę kredytu hipotecznego.*

(b) *Oblicz o ile zmieni się rata kredytu, jeśli okres kredytowania wydłużymy do 35 lat lub 40 lat albo skrócimy do 25 lat. Użyj opcji* `Manipulate`.

(c) *Oblicz jak zmieni się rata kredytu, jeśli stopa procentowa zwiększy się o 0,5 pkt. procentowego, o 1 pkt. procentowy, o 1,5 pkt. procentowego. Użyj opcji* `Manipulate`.

Zadanie 2.

(a) *Narysuj wykres wysokości raty kredytu na 100.000 zł o okresie kredytowania 30 lat jako funkcję stopy procentowej.*

(b) *Narysuj wykres wysokości raty kredytu na 100.000 zł oprocentowanego na 6% w skali roku jako funkcję długości okresu kredytowania.*

Zadanie 3. *Przez 30 lat wpłacasz co miesiąc 100 zł do funduszu emerytalnego. średnia roczna realna stopa zwrotu z inwestycji w jednostki tego funduszu to 2%. Wyprowadź wzór na zgromadzony kapitał (w podobny sposób jak wzór na ratę kredytu) i oblicz jaki kapitał zgromadzisz po 30 latach. Przy pomocy opcji* `Manipulate` *sprawdź jak zmiana realnej stopy zwrotu wpływa na zgromadzony kapitał.*

Zadanie 4. *Zrób symulację pokazującą wpływ wieku emerytalnego na wysokość twojej emerytury. W tym celu przyjmij następujące założenia:*

- *pracujesz nieprzerwanie od ukończenia studiów do osiągnięcia wieku emerytalnego,*
- *twoje realne wynagrodzenie jest stałe przez cały okres pracy (założenie to uwzględniamy przyjmując zerową inflację i stałe wynagrodzenie),*
- *składka emerytalna to 19,52% wynagrodzenia brutto,*
- *realna stopa zwrotu z inwestycji funduszu emerytalnego (lub realna stopa waloryzacji kapitału dokonywanej przez ZUS) mieści się w przedziale* $[0\%, 5\%]$ *(użyj opcji* `Manipulate`*),*
- *wiek emerytalny to 60 lat, 65 lat lub 67 lat.*

Podobnie jak w poprzednim zadaniu oblicz kapitał emerytalny zgromadzony w momencie przejścia na emeryturę a następnie podziel go przez oczekiwaną dalszą długość trwania życia[2] *wyrażoną w miesiącach. Otrzymasz w ten sposób szacunkową wysokość emerytury. Wykonaj symulację osobno dla kobiet i dla mężczyzn. Odpowiedz na pytanie jak duży wzrost szacowanej emerytury (w procentach) spowoduje podwyższenie wieku emerytalnego dla kobiet z 60 lat do 67 lat i dla mężczyzn z 65 lat do 67 lat.*

[2] Dane znajdziesz na stronie GUS, w publikacji `http://www.stat.gov.pl/cps/rde/xbcr/gus/PUBL_trwanie_zycia_2007.pdf`

Rozwiązania

Zadanie 1

```
K=350000;
Manipulate[q=p/1200; K*q (1+q)^(12n)/((1+q)^(12n)-1) // N, {p,3,10}, {n,20,40}]
```

Zadanie 2a

```
Clear[p,q];
K=100000; n=360;
q[p_]=p/1200;
Plot[K*q[p] (1+q[p])^n/((1+q[p])^n-1), {p,0,10}]
```

Zadanie 2b

```
Clear[n];
K=100000; q=6/1200;
Plot[K*q(1+q)^n/((1+q)^n-1), {n,10,50}, PlotRange->{{10,50},{0,10000}}]
```

Zadanie 3

```
r=100; n=360;
Manipulate[r*((1+(q/1200))^n-1)/(q/1200),{q,0.1,5}]
```

Zadanie 4

Oczekiwana dalsza długość trwania życia:
60-letnia kobieta: **23 lata**,
65-letni mężczyzna: **14,6 lat**,
67-letnia kobieta: **17,3 lat**,
67-letni mężczyzna: **13,4 lat**.

```
r=.1952*3646;
n=(65-24)*12;
K[p_]=r*((1+(p/1200))^n-1)/(p/1200);
Manipulate[K/(14.6*12),{p,0.1,5}]
```

Pracownia 7

Pętle i instrukcje warunkowe

Przydatne polecenia

`Do[Print[k], {k,5,12}]`	wykonaj `Print[k]` dla $k = 5,\ 6,\ 7, \ldots,\ 12$
`Print[k]`	wydrukuj wartość zmiennej k
`While[n<5, Print[n]]`	wykonuj polecenie `Print[k]` dopóki $n < 5$
`n++`	zwiększ zmienną n o 1
`If[x==3, 1, 0]`	jeżeli $x = 3$ zwróć wartość 1, w przeciwnym razie zwróć wartość 0
`If[x<5 && x>2, 1, 0]`	jeżeli $x < 5$ i $x > 2$ ...
`If[x>5 || x<2, 1, 0]`	jeżeli $x > 5$ lub $x < 2$ ...
`If[x!=2, 1, 0]`	jeżeli $x \neq 2$ ...
`Prime[n]`	n-ta liczba pierwsza
`PrimeQ[p]`	czy p jest liczbą pierwszą
`!PrimeQ[p]`	czy p **nie** jest liczbą pierwszą
`Divisible[n,m]`	czy n dzieli się przez n
`GCD[m,n]`	$NWD(m, n)$
`Sum[f[n],{n,1,10}]`	$\sum_{k=1}^{n} f(n)$

Wprowadzenie

Przykład 1. *Wypisz wszystkie liczby pierwsze z przedziału* $[120, 150]$.

```
Do[If[PrimeQ[n],Print[n]],{n,120,150}]
```

Przykład 2. *Oblicz sumę wszystkich liczb pierwszych mniejszych od 100.*

Są trzy różne sposoby rozwiązania tego zadania:

```
Sum[If[PrimeQ[n],n],{n,1,100}]

suma=0;  Do[If[PrimeQ[n], suma=suma+n],{n,1,100}]
suma

suma=0;  For[n=1, n<=100, n++, If[PrimeQ[n], suma=suma+n]]
suma
```

Przykład 3. *Oblicz ile jest liczb trzycyfrowych niepodzielnych przez 3 ani przez 7.*

```
Sum[If[!Divisible[n,3]&&!Divisible[n,7],n,0],{n,100,999}]
```

Przykład 4. *Oblicz sumę wszystkich ułamków nieskracalnych mniejszych od 1, których licznik i mianownik nie przekraczają 5.*

```
suma = 0;
Do[suma=suma+If[GCD[k,j]==1,k/j,0],{k,1,5},{j,k+1,5}]
suma
```

Zadania

Zadanie 1. *Oblicz* 10! *używając (a) pętli* `Do`*, (b) pętli* `For`*, (c)* `While`*.*

Zadanie 2. *Oblicz sumę wszystkich liczb naturalnych z przedziału* $[1, 100]$ *niepodzielnych przez 7.*

Zadanie 3. *Oblicz sumę wszystkich liczb naturalnych z przedziału* $[1, 100]$ *podzielnych przez 3 lub przez 5. Wypisz wszystkie te liczby.*

Zadanie 4. *Znajdź liczbę liczb naturalnych w przedziale* $[1, 1000]$*, które nie są podzielne przez 2, przez 3, ani przez 7. Nie wypisuj tych liczb.*

Zadanie 5. *Wypisz kwadraty wszystkich liczb pierwszych z przedziału* $[1, 20]$.

Zadanie 6. *Oblicz sumę wszystkich ułamków postaci* $\frac{p}{q}$*, gdzie* $p < q$ *oraz* p *i* q *są liczbami pierwszymi mniejszymi od 20.*

Zadanie 7. *Znajdź najmniejszą (a) czterocyfrową, (b) pięciocyfrową, (c) sześciocyfrową liczbę pierwszą.*

Zadanie 8. *Liczby bliźniacze to dwie liczby pierwsze różniące się o 2, np. 3 i 5 lub 41 i 43.*

(a) *Wypisz pierwsze 20 par liczb bliźniaczych.*

(b) *Wypisz wszystkie pary 4-cyfrowych liczb bliźniaczych.*

(c) *Zdefiniuj funkcję, która znajduje parę najmniejszych liczb bliźniaczych większych od* n *(wykorzystaj polecenie* `Module`*). Użyj jej do znalezienia pary najmniejszych liczb bliźniaczych większych niż* 10^{60}.

Zadanie 9. *Zdefiniuj funkcję, która dla zadanego* n *znajduje najmniejsze liczby czworacze większe od* n *(czyli czwórki liczb* p*,* $p+2$*,* $p+6$*,* $p+8$ *takie, że wszystkie cztery są liczbami pierwszymi). Wykorzystaj ją do znalezienia najmniejszych liczb czworaczych większych niż* 10^{15}*. Zastanów się dlaczego chcemy by* p*,* $p+2$*,* $p+6$ *i* $p+8$ *były pierwsze, nie zaś* p*,* $p+2$*,* $p+4$ *i* $p+6$.

Zadanie 10. *Liczba 2520 to najmniejsza liczba podzielna przez każdą z liczb od 1 do 10. Jaka jest najmniejsza liczba naturalna podzielna przez każdą z liczb od 1 do 13? A przez każdą z liczb od 1 do 20? do 30? do 50?*

Rozwiązania

Zadanie 2

```
Sum[If[!Divisible[n,7],n,0],{n,1,100}]
```

Zadanie 3

```
Sum[If[Divisible[n,3]||Divisible[n,5],n,0],{n,1,100}]
```

Zadanie 4

```
Sum[If[!Divisible[n,2]&&!Divisible[n,3]&&!Divisible[n,7],1,0],{n,1,1000}]
```

Zadanie 5

```
Do[If[PrimeQ[n],Print[n^2]],{n,1,20}]
```

Zadanie 7

```
n=1000;
While[!PrimeQ[n],n++];
n
```

Zadanie 8b

```
Do[If[PrimeQ[n]&&PrimeQ[n+2],Print[n," ",n+2]],{n,1000,9997}]
```

Zadanie 8c

```
bliz[n0_]:=Module[{n=n0}, sukces=0;
  i=n;
  While[sukces==0,
  If[PrimeQ[i]&&PrimeQ[i+2],Print[i,"  ",i+2]; sukces++]; i++]]
bliz[10^60]
```

Zadanie 9

```
czwor[n0_]:=Module[{n=n0}, suk=0;
  i=n;
  While[suk==0,
  If[PrimeQ[i]&&PrimeQ[i+2]&&PrimeQ[i+6]&&PrimeQ[i+8],
  Print[i,"  ",i+2,"  ",i+4,"  ",i+6]; suk++]; i++]]
czwor[10^15]
```

Pracownia 8

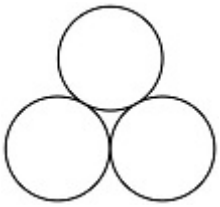
Grafika

Przydatne polecenia

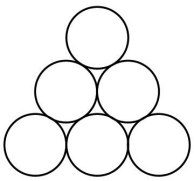
`Graphics[ ... ]`	utwórz dwuwymiarowy obiekt graficzny
`Circle[{x,y}, r]`	utwórz okrąg o środku w punkcie (x, y) i promieniu r
`Show[Graphics[Circle[{0,0}, 2]]]`	narysuj okrąg o promieniu 2
`Disk[{x,y}, r]`	utwórz koło (zamalowane) o środku (x, y) i promieniu r
`Point[{x,y}]`	utwórz punkt w punkcie (x, y)
`Line[{{x1,y1}, {x2,y2}, ...}]`	narysuj linię łączącą punkty $(x1, y1)$, $(x2, y2)$, ...
`Rectangle[{x1,y1}, {x2,y2}]`	utwórz wypełniony prostokąt taki, że punkty $(x1, y1)$ i $(x2, y2)$ są końcami jednej z przekątnych
`Polygon[{{x1,y1}, {x2,y2}, ...}]`	utwórz wypełniony wielokąt, którego wierzchołkami są punkty $(x1, y1)$, $(x2, y2)$, ...
`Text['abc', {x,y}]`	napisz *abc* tak, że środek napisu jest w punkcie (x, y).

Zadania

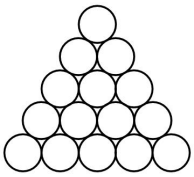
Zadanie 1. *Narysuj poniższy rysunek*



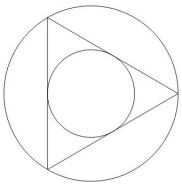
Zadanie 2. *Narysuj poniższy rysunek*



Zadanie 3. *Narysuj poniższy rysunek*



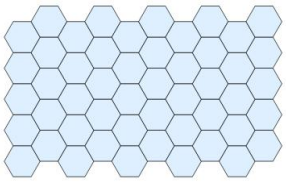
Zadanie 4. *Narysuj poniższy rysunek*



Zadanie 5. *Narysuj siedmiokąt foremny. Wskazówka: możesz utworzyć wielokąt, którego wierzchołki leżą na okręgu jednostkowym, wówczas współrzędne jego wierzchołków można wyrazić przy pomocy funkcji sinus i cosinus.*

Zadanie 6. *Utwórz manipular, który rysuje n-kąty foremne dla $3 \leq n \leq 30$. Zaobserwuj jak kolejne n-kąty przybliżają okrąg.*

Zadanie 7. *Narysuj parkietaż płaszczyzny złożony z sześciokątów:*



Rozwiązania

Zadanie 1

```
  Show[Graphics[{Thickness[0.01],
   Circle[{0, 0}, 1/2], Circle[{1, 0}, 1/2],
   Circle[{1/2, Sqrt[3]/2}, 1/2]
   }]]
```

Zadanie 2

```
pa[1] = {0, 0}; pa[2] = {1, 0}; pa[3] = {1/2, Sqrt[3]/2};
pa[4] = (pa[1] + pa[2])/2; pa[5] = (pa[1] + pa[3])/2;
pa[6] = (pa[2] + pa[3])/2;
Show[Graphics[{Thickness[0.01],
   Table[Circle[pa[n], 1/4], {n, 1, 6}]
   }]]
```

Zadanie 3

```
pa[1] = {0, 0}; pa[2] = {1, 0}; pa[3] = {1/2, Sqrt[3]/2};
pa[4] = (pa[1] + pa[2])/2; pa[5] = (pa[1] + pa[3])/2;
pa[6] = (pa[2] + pa[3])/2;
pa[7] = (pa[1] + pa[4])/2; pa[8] = (pa[2] + pa[4])/2;
pa[9] = (pa[1] + pa[5])/2;
pa[10] = (pa[5] + pa[3])/2; pa[11] = (pa[2] + pa[6])/2;
pa[12] = (pa[6] + pa[3])/2;
pa[13] = (pa[4] + pa[5])/2; pa[14] = (pa[4] + pa[6])/2;
pa[15] = (pa[5] + pa[6])/2;

Show[Graphics[{Thickness[0.01],
   Table[Circle[pa[n], 1/8], {n, 1, 15}]
   }]]
```

Zadanie 4

```
Show[Graphics[{
   Line[{{1, 0}, {Cos[2 Pi/3], Sin[2 Pi/3]}, {Cos[4 Pi/3],
      Sin[4 Pi/3]}, {1, 0}}],
   Circle[{0, 0}, 1],
   Circle[{0, 0}, 1/2]},
  AspectRatio -> 1]]
```

Zadanie 5

```
Show[Graphics[{LightGreen, Polygon[Table[{Cos[2 Pi k/7], Sin[2 Pi k/ 7]}, {k, 1, 7}]]}]]
```

Zadanie 6

```
Manipulate[
 Show[Graphics[{Orange, PlotRange -> {{-1, 1}, {-1, 1}},
    Polygon[Table[{Cos[2 Pi k/n], Sin[2 Pi k/ n]}, {k, 1, n}]]}]],
    {n, 3, 30, 1}]
```

Zadanie 7

```
h[x_, y_] :=
  Polygon[Table[{Cos[2 Pi k/6] + x, Sin[2 Pi k/6] + y}, {k, 6}]];
Graphics[{EdgeForm[Opacity[.7]], LightBlue,
  Table[h[3 i + 3 ((-1)^j + 1)/4, Sqrt[3]/2 j], {i, 5}, {j, 10}]}]
```

Pracownia 9

Krzywe stożkowe

Przydatne polecenia

`Graphics[Point[{1,2}]]`	narysuj punkt $(1,2)$
`Graphics[Point[{{1,2},{1,3}}]]`	narysuj punkty $(1,2)$ i $(1,3)$
`Graphics[Line[{{1,2},{1,3}}]]`	narysuj odcinek łączący punkty $(1,2)$ i $(1,3)$
`ContourPlot[x^2+y^2==4,`	narysuj krzywą $x^2+y^2=4$ ...
`{x,-3,3},{y,-3,3},`	... w obszarze $-3 \leq x, y \leq 3$
`AspectRatio->Automatic]`	... ustal równe jednostki na obu osiach (opcjonalne)
`Show[Graphics[...],ContourPlot[...]]`	narysuj we wspólnym układzie współrzędnych
`Abs[x]`	wartość bezwzględna $\|x\|$

Wprowadzenie

Krzywe stożkowe (elipsa, hiperbola, parabola) można zdefiniować na kilka sposobów:

Definicja 1 Krzywa stożkowa to zbiór punktów (x, y) spełniających równanie kwadratowe[1], tzn.

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

Definicja 2 Krzywa stożkowa to zbiór punktów X o tej własności, że stosunek odległości X od ustalonego punktu F (ogniska) do odległości X od ustalonej prostej l (kierownicy) jest stały i równy $\epsilon > 0$ (mimośród):

$$|XF| : d(X, l) = \epsilon$$

Definicja 3 Elipsa to zbiór punktów X, dla których suma odległości od ustalonych punktów F_1 i F_2 (ognisk) jest stała:

$$|XF_1| + |XF_2| = 2a$$

Hiperbola to zbiór punktów X, dla których różnica odległości od ognisk jest stała:

$$\big||XF_1| - |XF_2|\big| = 2a$$

Definicja 4 Krzywa stożkowa to przekrój nieskończonego stożka płaszczyzną nieprzechodzącą przez wierzchołek stożka.

[1] Tak naprawdę oprócz krzywej stożkowej równaniem takim możemy opisać zbiór pusty, punkt, prostą lub dwie proste.

Zadania

Zadanie 1. *Narysuj krzywą opisaną równaniem* $ax^2 + bxy + cx^2 = 1$ *i sprawdź jak zachowuje się ona przy manipulowaniu parametrami* a, b, c.

Zadanie 2. *(Elipsa) Zaznacz dwa punkty* A *i* B *i narysuj krzywą złożoną ze wszystkich punktów* X, *dla których* $|XA|+|XB| = 2a$. *Wprowadź możliwość manipulacji parametrem* a *oraz zmiany odległości punktów* A *i* B.

Zadanie 3. *(Hiperbola) Zaznacz dwa punkty* A *i* B *i narysuj krzywą złożoną ze wszystkich punktów* X, *dla których* $||XA| - |XB|| = 2a$. *Wprowadź możliwość manipulacji parametrem* a *oraz zmiany odległości punktów* A *i* B.

Zadanie 4. *Zaznacz punkt* F, *prostą* l *i narysuj krzywą stożkową o ognisku* F, *kierownicy* l *i mimośrodzie* ϵ. *Wprowadź możliwość manipulowania wielkością mimośrodu. Sprawdź jaką krzywą dostajemy w przypadkach* $\epsilon < 1$, $\epsilon = 1$, $\epsilon > 1$.

Zadanie 5. *(dla chętnych) Narysuj nieskończony stożek oraz przekrój tego stożka płaszczyzną. Wprowadź możliwość manipulacji kątem nachylenia płaszczyzny. Pokaż, że możesz otrzymać w ten sposób każdy z trzech rodzajów krzywych: elipsę, hiperbolę i parabolę.*

Rozwiązania

Zadanie 1

```
Manipulate[ContourPlot[a*x^2+b*x*y+c*x^2==1,{x,-10,10},{y,-10,10}],
  {a,-1,1},{b,-1,1},{c,-1,1}]
```

Zadanie 2

```
Manipulate[Show[
  ContourPlot[Sqrt[(x+c)^2+y^2]+Sqrt[(x-c)^2+y^2]==2a,{x,-1,1},{y,-1,1},
    AspectRatio->Automatic],
  Graphics[Point[{{-c,0},{c,0}}]]],
{c,0,1},{a,0,1}]
```

Zadanie 3

```
Manipulate[Show[
  ContourPlot[Abs[Sqrt[(x+c)^2+y^2]-Sqrt[(x-c)^2+y^2]]==2a,{x,-1,1},{y,-1,1},
    AspectRatio->Automatic],
  Graphics[Point[{{-c,0},{c,0}}]]],
{c,0,1},{a,0,1}]
```

Zadanie 4

```
Manipulate[Show[
  ContourPlot[Sqrt[(x-1)^2+y^2]/Abs[x]==e,{x,-5,5},{y,-5,5},
    AspectRatio->Automatic],
  Graphics[Point[{1,0}]],
  Graphics[Line[{{0,-5},{0,5}}]]],
{e,0,2}]
```

Zadanie 5

```
Manipulate[
  Show[ContourPlot3D[{x^2+y^2==z^2, a*x+y-3 z==-15},
    {x,-10,10}, {y,-10,10}, {z,-10,10},
    Mesh->None, ContourStyle->{{Yellow, Opacity[0.8]}, {Green, Opacity[0.8]}}]],
{a, -5, 5}]
```

Pracownia 10

Cykloida

Przydatne polecenia

`Animate[...,{t,0,5}]`	zrób animację dla parametru $0 \leq t \leq 5$
`ParametricPlot[{x[t],y[t]},`	narysuj krzywą zadaną parametrycznie $(x(t), y(t))$ ...
`{t,0,1},`	... dla wartości parametru $0 \leq t \leq 1$ ...
`PlotRange->{{a,b},{c,d}},`	... ustal układ współrzędnych $a \leq x \leq b$, $c \leq y \leq d$ (opcjonalne) ...
`Axes->False]`	... nie wyświetlaj osi współrzędnych (opcjonalne)
`Graphics[Circle[{x,y},r]]`	narysuj okrąg o środku (x, y) i promieniu r
`Graphics[Disk[{x,y},r]]`	narysuj koło o środku (x, y) i promieniu r
`Graphics[Line[{{a,b},{c,d}}]]`	narysuj odcinek łączący punkty (a, b) i (c, d)
`Graphics[Rectangle[{a,b},{c,d}]]`	narysuj prostokąt o przekątnej mającej końce (a, b) i (c, d)
`Graphics[Point[{x,y}]]`	narysuj punkt (x, y)
`Graphics[Text[ąaa",{a,b}]`	wydrukuj tekst "aaa" wyśrodkowany względem punktu (a, b)
`Show[Graphics[...],`	wyświetl we wspólnym układzie współrzędnych
`ParametricPlot[...]]`	

Wprowadzenie

Cykloida to krzywa zakreślana przez punkt leżący na brzegu okręgu, który toczy się (bez poślizgu) po prostej. Naszym celem będzie wyprowadzić postać parametryczną tej krzywej, a następnie ją narysować.

Załóżmy, że koło ma promień r i ma prędkość obrotową 1 (rad/s). Wobec tego środek koła ma prędkość liniową r (koło przejedzie w czasie t odległość tr). Ustalmy układ współrzędnych tak, by koło jechało po osi OX, początkowo jego środek znajdował się na osi OY, a punkt P, którego ślad rysujemy miał współrzędne $(0, 0)$.

Ruch punktu na brzegu okręgu można opisać jako złożenie ruchu postępowego środka koła i ruchu obrotowego punktu P względem środka koła. Stąd:

$$f(t) = s(t) + p(t)$$

gdzie $s(t) = (rt, r)$ jest położeniem środka koła w chwili t, zaś $p(t) = A(-t) \cdot (0, -r)^T$ jest położeniem punktu P względem środka koła. Macierz $A(t) = \begin{pmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{pmatrix}$ jest macierzą obrotu o kąt t wokół punktu $(0, 0)$.

Hypocykloida to krzywa zdefiniowana podobnie jak cykloida, tyle że okrąg (o promieniu r) toczy się bez poślizgu po zewnętrzu okręgu o promieniu R. Przyjmijmy, że koło toczy się z prędkością 1 rad/s (względem "drogi"). Wówczas w czasie t przejedzie drogę tr po okręgu o promieniu R, tzn. środek

porusza się po okręgu o promieniu $R + r$ z prędkością kątową r/R (rad/s). Z kolei punkt na brzegu koła obraca się względem środka koła z prędkością kątową $1+r/R$ (koło obraca się z prędkością kątową 1 względem drogi, zaś punkt styczności z drogą obraca się z prędkością kątową r/R względem środka koła). Położenie punktu P opisujemy więc funkcją:

$$f(t) = s(t) + p(t)$$

gdzie $s(t) = A(\frac{r}{R}t) \cdot (R + r, 0)^T$ oraz $p(t) = A(t(1 + \frac{r}{R}) \cdot (-r, 0)^T$.

Epicykloida różni się od hypocykloidy jedynie tym, że mniejszy okrąg toczy się we wnętrzu większego.

Zadania

Zadanie 1. *Narysuj cykloidę. Następnie wykonaj animację pokazującą jak powstaje cykloida, tzn. narysuj toczący się okrąg, jego promień poprowadzony do punktu rysującego krzywą, ślad tego punktu oraz prostą, po której okrąg się toczy.*

Zadanie 2. *Wykonaj animację rysującą hypocykloidę dla różnych wymiernych wartości ilorazu* $k = R/r$.

Zadanie 3. *Napisz parametryczne równanie epicykloidy i wykonaj animację rysującą epicykloidę dla różnych wymiernych wartości ilorazu* $k = R/r$.

Zadanie 4. *Narysuj krzywą zakreślaną przez punkt leżący we wnętrzu koła w każdej z sytuacji przedstawionej w zadaniach 1, 2 i 3.*

Zadanie 5. *Utwórz animację kwadratu "toczącego się" po prostej.*

Rozwiązania

Zadanie 1

```
A[t_]={{Cos[t],-Sin[t]}, {Sin[t],Cos[t]}};
s[t_]={t,1};
p[t_]=A[-t].{0,-1};
Animate[Show[
  ParametricPlot[s[t]+p[t], {t,0,T},
    PlotRange -> {{0,4Pi}, {-1,3}}, Axes->False],
  Graphics[{Circle[s[T],1], Line[{s[T],s[T]+p[T]}],
    Line[{{0,0},{4 Pi,0}}], Point[{s[T],s[T]+p[T]}]}]],
{T,0,4Pi}]
```

Zadanie 2

```
k=2.5;
Clear[A,s,p];
A[t_]={{Cos[t],-Sin[t]}, {Sin[t],Cos[t]}};
s[t_]=A[t/k].{k+1,0};
p[t_]=A[t(1+1/k)].{-1,0};
Animate[Show[
  ParametricPlot[s[t] + p[t], {t, 0, T},
  PlotRange -> {{-k-3,k+3}, {-k-3,k+3}}, Axes -> False],
  Graphics[{Circle[s[T], 1], Line[{s[T], s[T] + p[T]}],
  Circle[{0, 0}, k], Point[{s[T], s[T] + p[T]}]}]], {T, 0, 4 Pi*k}]
```

Zadanie 3

```
k=4;
Clear[A,s,p];
A[t_]={{Cos[t],-Sin[t]}, {Sin[t],Cos[t]}};
s[t_]=A[-t/k].{k-1,0};
p[t_]=A[t(1-1/k)].{1,0};
Animate[Show[
  ParametricPlot[s[t] + p[t], {t, 0, T},
  PlotRange -> {{-k-3,k+3}, {-k-3,k+3}}, Axes -> False],
  Graphics[{Circle[s[T], 1], Line[{s[T], s[T] + p[T]}],
  Circle[{0, 0}, k], Point[{s[T], s[T] + p[T]}]}]], {T, 0, 4 Pi*k}]
```

Pracownia 11

Grafika 3D

Przydatne polecenia

`ContourPlot3D[2x+3y+z==1,`	narysuj powierzchnię o równaniu $2x + 3y + z = 1$ ...
`{x,0,1},{y,0,1},{z,0,1},`	... w obszarze $0 \leq x, y, z \leq 1$
`Mesh->None,`	... nie rysuj siatki (opcjonalne)
`ContourStyle->{Yellow,Opacity[0.8]},`	... ustaw kolor i przezroczystość (opcjonalne)
`Axes->None]`	... nie rysuj osi układu współrzędnych (opcjonalne)
`Graphics3D[Sphere[{a,b,c},r]]`	narysuj sferę o środku (a, b, c) i promieniu r

Zadania

Zadanie 1. *Narysuj zbiór punktów (x, y, z) spełniających równanie*

$$ax^2 + by^2 + cz^2 + dxy + eyz + fzx = 0$$

i wprowadź możliwość manipulacji parametrami. Opisz jakie powierzchnie otrzymałeś (elipsoida, paraboloida, hiperboloida jedno- lub dwupowłokowa itd.).

Zadanie 2. *Narysuj nieskończony stożek oraz przekrój tego stożka płaszczyzną. Wprowadź możliwość manipulacji kątem nachylenia płaszczyzny. Pokaż, że możesz otrzymać w ten sposób każdy z trzech rodzajów krzywych: elipsę, hiperbolę i parabolę.*

Zadanie 3. *Narysuj przekrój sześcianu płaszczyzną z możliwością manipulacji położeniem i nachyleniem tej płaszczyzny. Ustal jaki wielokąt można otrzymać jako przekrój sześcianu płaszczyzną (wypisz wszystkie możliwości). Czy można otrzymać pięciokąt foremny? inny pięciokąt? sześciokąt foremny? siedmiokąt?*

Zadanie 4. *Ustal na ile najwięcej części można podzielić przestrzeń 4 płaszczyznami. W tym celu narysuj 4 płaszczyzny takie, by nie miały one punktu wspólnego, ale by każde trzy miały punkt wspólny i policz ile kawałków przestrzeni otrzymałeś.*

Zadanie 5. *Narysuj 4 sfery o jednakowym promieniu, które są parami styczne (piramidka z 4 sfer). Narysuj piramidkę złożoną z 10 sfer (trzypoziomową).*

Rozwiązania

Zadanie 1

```
Manipulate[CountourPlot3D[a*x^2+b*y^2+c*z^2+d*z*y+e*y*z+f*z*x==0,
  {x,-10,10}, {y,-10,10}, {z,-10,10}],
  {a,-1,1}, {b,-1,1}, {c, -1,1}, {d,-1,1}, {e,-1,1}, {f,-1,1}]
```

Zadanie 2

```
Manipulate[
  Show[ContourPlot3D[{x^2+y^2==z^2, a*x+y-3 z==-15},
    {x,-10,10}, {y,-10,10}, {z,-10,10},
    Mesh->None, ContourStyle->{{Yellow, Opacity[0.8]}, {Green, Opacity[0.8]}}]],
{a, -5, 5}]
```

Zadanie 3

```
Manipulate[
  ContourPlot3D[a*x+b*y+c*z+d==0, {x,-1,1}, {y,-1,1}, {z,-1,1},
    Mesh->None, Axes->None],
{a,-1,1}, {b,-1,1}, {c,-1,1}, {d,-1,1}]
```

Zadanie 4

```
ContourPlot3D[{x+y-z==0, x-y+z==0, -x+y+z==0, x+y+z==1},
  {x,-2,2}, {y,-2,2}, {z,-2,2}, Mesh->None]
```

Zadanie 5

```
Show[Graphics3D[Sphere[{-1,0,0},1]],
  Graphics3D[Sphere[{1,0,0},1]],
  Graphics3D[Sphere[{0,Sqrt[3],0},1]],
  Graphics3D[Sphere[{0,0,2Sqrt[2/3]},1]]]
```

Pracownia 12

Pocisk balistyczny I

Przydatne polecenia

`ParametricPlot[{x[t],y[t]},`	narysuj krzywą zadaną parametrycznie $(x(t), y(t))$ ...
`{t,0,1},`	... dla wartości parametru $0 \leq t \leq 1$
`PlotRange->{{a,b},{c,d}},`	... ustal układ współrzędnych $a \leq x \leq b$, $c \leq y \leq d$ (opcjonalne)
`AspectRatio->Automatic,`	... ustal jednakową skalę na obu osiach (opcjonalne)
`PlotLabel->ToString[a]]`	... opisz rysunek podając wartość zmiennej a (opcjonalne)
`Show[Graphics[...],`	narysuj we wspólnym układzie współrzędnych
`ParametricPlot[...]]`	
`Tan[x Degree]`	oblicz $\tan x$ dla x wyrażonego w stopniach
`Solve[x^2+4x-3==0,x]`	rozwiąż równanie z niewiadomą x
`x1=x/.Solve[x^2+4x-3==0,x][[1]]`	oznacz przez x_1 pierwsze rozwiązanie równania

Wprowadzenie

Ruch pocisku wystrzelonego z punktu $p = (x_0, y_0)$ z prędkością początkową $\vec{v} = (v_x, v_y)$ można opisać jako złożenie ruchu poziomego (jednostajnego) i ruchu pionowego (jednostajnie przyspieszonego). Jeśli zaniedbamy opór powietrza, to równania takiego ruchu mają postać:

$$x(t) = x_0 + v_x \cdot t$$

$$y(t) = y_0 + v_y \cdot t - \frac{gt^2}{2}$$

Jeśli wystrzelimy pocisk z punktu $(0, 0)$ z armaty znajdującej się na równinie, to rozwiązując równanie $y(t) = 0$ nietrudno wyliczyć moment T, w którym pocisk uderzy w ziemię.

Zadania

Zadanie 1. *Przyjmując prędkość wylotową kuli armatniej na poziomie 150 m/s narysuj trajektorię lotu kuli i ustal odległość w jakiej kula uderzy w ziemię, jeśli:*

(a) *armata stoi na równinie i strzela pod kątem* $30°$ *do poziomu,*

(b) *armata stoi u podnóża wzniesienia o nachyleniu* $10°$ *i strzela pod kątem* $40°$ *do poziomu,*

(c) *armata stoi na szczycie wzniesienia o nachyleniu* $10°$ *i strzela pod kątem* $20°$ *do poziomu.*

Zadanie 2. *Ustal "doświadczalnie" maksymalny zasięg armaty w sytuacjach 1(a), 1(b) i 1(c), zmieniając kąt nachylenia armaty. Dla jakiego kąta nachylenia otrzymujemy maksymalny zasięg?*

Zadanie 3. *Wykonaj animację lecącej kuli armatniej, tzn. narysuj armatę (odcinek lub prostokąt), umożliwiając zmianę kąta jej nachylenia, lecącą kulę (punkt lub koło) oraz ślad zakreślany przez lecącą kulę dla każdej z sytuacji 1(a), 1(b), 1(c).*

Rozwiązania

Zadanie 1

```
a=40;
b=10;
x[t_]=150Cos[a Degree]*t;
y[t_]=150Sin[a Degree]*t-9.81t^2/2;
T=t/.Solve[y[t]==x[t]*Tan[b Degree],t][[2]]
Show[ParametricPlot[{x[t],y[t]},{t,0,T},
    AspectRatio->Automatic, PlotLabel->ToString[x[T]]],
  Graphics[Line[{{0,0},{3000,3000*Tan[b Degree]}}]]]
```

Zadanie 2

```
b=10;
x[t_,a_]=150Cos[a Degree]*t;
y[t_,a_]=150Sin[a Degree]*t-9.81t^2/2;
T[a_]=t/.Solve[y[t,a]==x[t,a]*Tan[b Degree],t][[2]];
Manipulate[
Show[ParametricPlot[{x[t,a],y[t,a]},{t,0,T[a]},
    AspectRatio->Automatic, PlotRange->{{0,2000},{0,1000}},
    PlotLabel->ToString[x[T[a],a]]],
  Graphics[Line[{{0,0},{3000,3000*Tan[b Degree]}}]]],{a,11,89}]
```

Zadanie 3

```
b=10;
x[t_,a_]=150Cos[a Degree]*t;
y[t_,a_]=150Sin[a Degree]*t-9.81t^2/2;
T[a_]=t/.Solve[y[t,a]==x[t,a]*Tan[b Degree],t][[2]];
Manipulate[
  Animate[
  Show[ParametricPlot[{x[s,a],y[s,a]},{s,0,t},
      AspectRatio->Automatic, PlotRange->{{0,2000},{0,1000}}],
    Graphics[Line[{{0,0},{3000,3000*Tan[b Degree]}}]],
    Graphics[Point[{x[t,a],y[t,a]}]]],
  {t,0,T[a]}], {a,11,89}]
```

Pracownia 13

Pocisk balistyczny II

Przydatne polecenia

`s=NDSolve[`	zapisz w zmiennej s rozwiązanie...
`  {x'[t]==2x[t]+3, x[0]==5},`	... równania różniczkowego $x'(t) = 2x(t) + 3$, $x(0) = 5$
`  x, {t,0,10}]`	... dla funkcji x w przedziale $0 \leq t \leq 10$
`Evaluate[x[t]/.s]`	funkcja $x(t)$ zapamiętana w rozwiązaniu s
`Norm[{a,b}]`	długość wektora (a, b)

Wprowadzenie

Na poprzednich zajęciach rysując trajektorię kuli armatniej zaniedbywaliśmy opór powietrza. Tym razem wykonamy dokładniejszą symulację, uwzględniając ten czynnik. Siła oporu powietrza jest zawsze skierowana przeciwnie do kierunku ruchu i ma wartość:

$$|F_d| = \frac{1}{2} \cdot c_d \cdot \rho \cdot A \cdot v^2$$

gdzie ρ jest gęstością powietrza, A polem przekroju kuli, c_d współczynnikiem oporu zależnym od kształtu lecącego obiektu (dla kuli $c_d = 0,47$), zaś v jest prędkością kuli.

Aby wykonać animację ruchu kuli (tzn. położenie w chwili t) potrzebujemy ustalić położenie początkowe, prędkość początkową oraz prędkość w chwili t. Dla ustalenia prędkości wystarczy znać przyspieszenie kuli w chwili t. Rozpatrzmy osobno ruch w poziomie i w pionie. Oznaczmy przez $(x(t), y(t))$ położenie w chwili t, natomiast przez $\vec{v}(t) = (v_x(t), v_y(t))$ i $\vec{a}(t) = (a_x(t), a_y(t))$ wektory prędkości i przyspieszenia w chwili t. Wówczas:

$$a_x(t) = -\frac{c_d \rho A}{2m} \cdot \sqrt{v_x^2(t) + v_y^2(t)} \cdot v_x(t)$$
$$a_y(t) = -\frac{c_d \rho A}{2m} \cdot \sqrt{v_x^2(t) + v_y^2(t)} \cdot v_y(t) - g$$

Ponieważ $a_x(t) = v_x'(t)$ i $v_x(t) = x'(t)$ i podobnie dla y, więc możemy to przepisać w postaci układu sześciu równań (dwa z nich to równania różniczkowe, a pozostałe cztery to warunki początkowe) pokazanego na następnej stronie.

$$
\begin{aligned}
x''(t) &= -\frac{c_d \rho A}{2m} \cdot \sqrt{(x'(t))^2 + (y'(t))^2} \cdot x'(t) \\
y''(t) &= -\frac{c_d \rho A}{2m} \cdot \sqrt{(x'(t))^2 + (y'(t))^2} \cdot y'(t) - g \\
x'(0) &= v_0 \cos \phi \\
y'(0) &= v_0 \sin \phi \\
x(0) &= 0 \\
y(0) &= 0
\end{aligned}
$$

Aby wykonać symulację lotu kuli należy rozwiązać powyższy układ równań różniczkowych.

Zadania

Zadanie 1. *Wykonaj symulację lotu 20-kilogramowej żelaznej kuli armatniej uwzględniając siłę oporu powietrza. Przyjmij współczynnik oporu dla kuli* $c_d = 0,47$. *Gęstość powietrza i gęstość żelaza znajdź w Internecie. Przyjmij, że armata stoi na równinie, a prędkość wylotowa kuli armatniej jest równa 150 m/s.*

(a) *Narysuj trajektorię ruchu kuli dla armaty strzelającej pod kątem* $30°$ *do poziomu. Znajdź zasięg armaty i porównaj otrzymany wynik z wynikiem podobnej symulacji nie uwzględniającej oporu powietrza.*

(b) *Wprowadź możliwość manipulacji kątem nachylenia armaty. Ustal maksymalny zasięg armaty i dla jakiego kąta nachylenia jest on osiągany.*

(c) *Wykonaj animację lecącej kuli armatniej.*

Rozwiązania

Zadanie 1

```
c=0.47; rho=1.2; m=20;
R=((20/7700)/(4Pi/3))^(1/3);
k=-.5c*rho*Pi*R^2/m;
Manipulate[
  s=NDSolve[{x''[t]==k*Norm[x'[t],y'[t]]*x'[t],
    y''[t]==k*Norm[x'[t],y'[t]]*y'[t]-9.81,
    x'[0]==150Cos[a Degree], y'[0]==150Sin[a Degree], x[0]==0, y[0]==0},
    {x,y},{t,0,15}]
  ParametricPlot[Evaluate[{x[t],y[t]}/.s],{t,0,15}],
{a,1,89}]
```

Pracownia 14

Grawitacja i prędkość ucieczki

Przydatne polecenia

`s=NDSolve[`	zapisz w zmiennej s rozwiązanie...
`{x'[t]==2x[t]+3, x[0]==5},`	... równania różniczkowego $x'(t) = 2x(t) + 3$, $x(0) = 5$
`x, {t,0,10},`	... dla funkcji x w przedziale $0 \leq t \leq 10$
`MaxSteps->10^6]`	... zwiększ liczbę kroków (dla dużych t, jeśli wyrzuca błąd)
`Evaluate[x[t]/.s]`	funkcja $x(t)$ zapamiętana w rozwiązaniu s
`Norm[{a,b}]`	oblicz długość wektora (a, b)
`Graphics[Disk[{a,b},r]`	narysuj koło o środku (a, b) i promieniu r

Wprowadzenie

Siła przyciągania grawitacyjnego ziemi ma wartość

$$|F_g| = \frac{GMm}{R^2}$$

gdzie G jest stałą grawitacyjną, M masą Ziemi, m masą ciała, zaś R odległością od środka kuli ziemskiej. Wobec tego przyspieszenie ziemskie ma wartość

$$g(R) = \frac{GM}{R^2}$$

(przy powierzchni ziemi mamy $g = 9.81m/s2$). Oznaczając przez $r(t)$ położenie ciała w chwili t, przyspieszenie tego ciała będzie wektorem równym:

$$a(t) = -\frac{GM}{|r(t)|^3} \cdot r(t)$$

Wyobraźmy sobie rakietę balistyczną wystrzeloną poziomo (tzn. równolegle do powierzchni ziemi) na wysokości 200 km nad ziemią. Ruch rakiety można opisać następującym układem równań:

$$r''(t) = -\frac{GM}{|r(t)|^3} \cdot r(t)$$

$$r'(0) = (v, 0)$$

$$r(0) = (0, R_z + 200)$$

gdzie początek układu współrzędnych umieściliśmy w środku Ziemi, przez R_z oznaczamy promień Ziemi, zaś przyjęte jednostki to km, s i kg.

Zadania

Zadanie 1. *Narysuj tor ruchu rakiety balistycznej wystrzelonej równolegle do powierzchni ziemi na wysokości 200 km nad ziemią z prędkością v km/s. W tym celu sprawdź w Internecie wartość G, M i R_z (pamiętaj o niestandardowych jednostkach: km, s, kg). Wprowadź możliwość manipulowania prędkością v. Następnie:*

- *znajdź minimalną prędkość, dla której rakieta nie spadnie, ale wejdzie na orbitę Ziemi (jest to tzw. pierwsza prędkość kosmiczna),*
- *znajdź minimalną prędkość, dla której rakieta opuści orbitę Ziemi (jest to tzw. druga prędkość kosmiczna).*

Zadanie 2. *Wyznacz "doświadczalnie" pierwszą i drugą prędkość kosmiczną dla Księżyca oraz dla Merkurego, Marsa i Wenus. Potrzebne dane (promień i masa) znajdź w Internecie.*

Rozwiązania

Zadanie 1

```
G=6.673*10^(-20); M=5.97*10^24;
k=-G*M;
Manipulate[
  s=NDSolve[{r''[t]==k*r[t]/(Norm[r[t]]^3),
    r'[0]=={v,0}, r[0]=={0,6371+200}}, r, {t,0,100000}, MaxSteps->10^6];
  Show[ParametricPlot[Evaluate[r[t]/.s],{t,0,100000},
    PlotRange->{{-20000,20000},{-20000,20000}}],Graphics[Disk[{0,0},6371]]],
{v,1,15}]
```

Pracownia 15

Listy I

Przydatne polecenia

`lista+3`	dodaj 3 do każdego elementu listy
`lista*2`	pomnóż przez 2 każdy element listy
`Table[nk, {n,1,10}, {k,1,10}]`	utwórz macierz (listę list)
`... // TableForm`	zapisz w postaci tabeli
`... // MatrixForm`	zapisz w postaci macierzy
`lista[[4]]`	czwarty element listy
`Length[lista]`	podaj długość (liczbę elementów) listy
`Count[lista, element]`	oblicz ile razy dany element występuje na liście
`Position[lista, element]`	podaj numery pozycji, na których występuje dany element
`MemberQ[lista, element]`	sprawdź, czy podany element jest na liście
`FreeQ[lista, element]`	sprawdź czy podany element nie występuje na liście
`TableForm[lista]`	wypisz listę w formie tablicy
`ColumnForm[lista]`	wypisz listę w formie macierzowej
`Sort[lista]`	posortuj listę
`Union[lista]`	połącz listy, posortuj i usuń elementy powtarzające się
`Reverse[lista]`	odwróć kolejność elementów listy
`Range[5]`	lista $\{1, 2, 3, 4, 5\}$
`Range[4,7]`	lista $\{4, 5, 6, 7\}$
`Range[4,12,2]`	lista $\{4, 6, 8, 10, 12\}$
`Apply[f,{2,3}]`	oblicz wartość funkcji $f(2, 3)$
`Max[a,b]`	$\max\{a, b\}$
`Plus[a,b,c]`	$a + b + c$
`Times[a,b,c]`	$a \cdot b \cdot c$
`IntegerDigits[51231]`	utwórz listę cyfr liczby, tu: $\{5, 1, 2, 3, 1, \}$
`FromDigits[{5,1,2,3,1}]`	utwórz liczbę z listy cyfr, tu: 51231
`FactorInteger[n]`	rozłóż n na czynniki pierwsze (utwórz listę dzielników pierwszych z wykładnikami)

Zadania

Zadanie 1. *Skonstruuj listę zawierającą* $1!, 2!, \ldots, 10!$ *na co najmniej 2 sposoby.*

Zadanie 2. *Napisz funkcję, która pobiera listę, a zwraca listę, w której każdy element jest podniesiony do kwadratu.*

Zadanie 3. *Wypisz wszystkie liczby podzielne przez 3 lub 7 i mniejsze niż 100*

Zadanie 4. *Napisz funkcję, która dla zadanej liczby zwraca jej największy dzielnik pierwszy. Jaki jest największy dzielnik pierwszy liczby 600851475143?*

Zadanie 5. *Liczba (lub słowo) jest palindromiczna jeżeli czytana wprost i wspak jest taka sama, np. 121, 4554. Napisz funkcję, która sprawdza czy dana liczba jest palindromiczna.*

Zadanie 6. *Oblicz sumę* $\sqrt{1} + \sqrt{2} + \cdots + \sqrt{100}$.

Zadanie 7. *Z ilu cyfr składa się liczba 1000! ?*

Zadanie 8. *Znajdź sumę cyfr liczby 100 000!*

Zadanie 9. *Napisz funkcję* $f(k)$, *która oblicza sumę kwadratów cyfr liczby* k.

Zadanie 10. *Oblicz sumę cyfr setnej liczby Fibonacciego.*

Zadanie 11. *Znajdź największą liczbę będącą iloczynem dwóch liczb trzycyfrowych, która jest palindromiczna.*

Zadanie 12. *Znajdź wszystkie liczby równe sumie sześcianów swoich cyfr (wykaż, że takie liczby nie mogą być "za duże" i ułóż program przeszukujący).*

Zadanie 13. *Czy istnieją liczby* a, b, c, d *mniejsze niż 1000 i parami różne takie, że* $a^3 + b^3 = c^3 + d^3$*?*

Zadanie 14. *Liczby* a, b, c *nazywamy trójką pitagorejską, jeżeli* $a^2 + b^2 = c^2$. *Wypisz wszystkie trójki pitagorejskie składające się z liczb mniejszych niż 1000.*

Rozwiązania

Zadanie 2

```
sqr[lista_] = lista^2
```

Zadanie 3

```
Union[Range[3, 100, 3], Range[7, 100, 7]]
```

Zadanie 4

```
larr[n_] := FactorInteger[n][[Length[FactorInteger[n]], 1]]
larr[600851475143]
```

Zadanie 5

```
Pal[n_] := n == FromDigits[Reverse[IntegerDigits[n]]]
```

Zadanie 6

```
Apply[Plus, Sqrt[Range[1, 100]]] // N
```

Zadanie 7

```
Length[IntegerDigits[1000!]]
```

Zadanie 8

```
Apply[Plus, IntegerDigits[1000000!]]
```

Zadanie 9

```
f[k_] := Apply[Plus, IntegerDigits[k]^2]
```

Zadanie 11

```
lista = {};
Do[If[Pal[i j], AppendTo[lista, i j]], {i, 100, 999}, {j, i, 999}]
Max[lista]
```

Pracownia 16

Listy II

Przydatne polecenia

`AppendTo[lst,a]`	dopisz a do listy *lst* i wynik podstaw pod *lst*
`Apply[f,{a,b,c}]`	oblicz $f[a,b,c]$
`Apply[Plus,{a,b,c}]`	oblicz $a+b+c$
`Map[f,{a,b,c}]`	oblicz $\{f[a],f[b],f[c]\}$
`Union[lista1,lista2,..]`	oblicz sumę zbiorów (list) i posortuj otrzymany wynik
`Complement[lista1, lista2, ...]`	wypisz elementy pierwszej listy, które nie znajdują się na kolejnych listach
`Intersection[lista1, lista2, ...]`	oblicz przekrój wszystkich list i posortuj
`Rest[lst]`	usuwa z listy *lst* pierwszy element
`RotateLeft[lst]`	przesuwa cyklicznie wszystkie elementy listy *lst* w lewo
`FreeQ[lista1,a]`	zwraca *True* jeżeli a nie występuje na liście i zwraca *False* jeżeli a występuje na liście
`PrimeQ[n]`	sprawdza, czy n jest liczbą pierwszą
`Divisors[n]`	zwraca wszystkie dzielniki n
`FactorInteger[n]`	oblicza rozkład n na czynniki pierwsze
`Factorial[n]`	liczy $n!$
`Print[k]`	pisze k na ekranie

Wprowadzenie

Przykład 1. *Oblicz sumę* $\frac{1}{\sqrt{1}}+\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{3}}+..+\frac{1}{\sqrt{50}}$

```
Apply[Plus, Map[Sqrt, Range[50]]] // N
```

Przykład 2. *Oblicz sumę* $1^2+2^2+..+100^2$.

```
kw[k_] := k^2
Apply[Plus, Map[kw, Range[100]]]
```

Przykład 3. *Napisz funkcję, która dla zadanej liczby n zwraca listę wszystkich liczb pierwszych dzielących n*

```
f1[{x_, _}] := x
f2[n_] := Map[f1, FactorInteger[n]]
```

Zadania

Zadanie 1. *Napisz funkcję, która dla zadanej liczby pierwszej n, liczy liczbę dzielników pierwszych liczby n. Sprawdź jej działanie dla liczb: 12, 122, 3456, 4353456. (Wskazówka: FactorInteger, Length)*

Zadanie 2. *Utwórz listę pierwszych 30 liczb nieparzystych, a następnie dla każdego* $k \leq 30$ *oblicz sumę pierwszych k liczb nieparzystych.*

Zadanie 3. *Utwórz listę pierwszych 30 liczb trójkątnych, tzn. liczb postaci* $\frac{n(n+1)}{2}$. *Dla każdego* $k \leq 30$ *oblicz sumę pierwszych k liczb trójkątnych.*

Zadanie 4. *Liczba jest doskonała jeżeli jest równa sumie wszystkich swoich dzielników różnych od niej samej, np.* $6 = 1 + 2 + 3$, $28 = 1 + 2 + 4 + 7 + 14$. *Wypisz wszystkie liczby doskonałe mniejsze od 10000. Znajdź liczbę doskonałą w przedziale [33500000, 33600000]*

Zadanie 5. *Znajdź wszystkie liczby mniejsze niż 1000, które są zarówno liczbami Fibonacciego jak i liczbami pierwszymi. Rozwiąż to zadanie dwoma sposobami:*

- *utwórz listę liczb pierwszych oraz listę liczb Fibonacciego mniejszych niż 1000 i oblicz ich przekrój;*
- *przy pomocy pętli (While) oblicz kolejne liczby Fibonacciego i te które są pierwsze (PrimeQ) dołącz do pomocniczej listy (AppendTo).*

Zadanie 6. *Utwórz listę wszystkich liczb naturalnych mniejszych niż 71, które nie są liczbami pierwszymi (wskazówka: możesz skorzystać z faktu, że liczba 71 jest 21 liczbą pierwszą, a następnie użyć poleceń Range, Complement, Prime).*

Zadanie 7. *Liczba 145 ma następującą własność :* $1! + 4! + 5! = 1 + 24 + 120 = 145$. *Znajdź kolejną liczbę, która jest równa sumie silni swoich cyfr*

Zadanie 8. *Znajdź wszystkie liczby, które są równe sumie piątych potęg swoich cyfr, np.* $4151 = 4^5 + 1^5 + 5^5 + 1^5 = 1024 + 1 + 3125 + 1$.

Zadanie 9. *Liczba 9801 ma następującą własność :* $9801 = (98 + 01)^2$. *Znajdź wszystkie liczby czterocyfrowe o tej własności (tzn. równe kwadratowi sumy swoich dwucyfrowych części). Użyj funkcji IntegerDigits[liczba, podstawa].*

Zadanie 10. *Liczby zaprzyjaźnione to para różnych liczb naturalnych, takich że suma dzielników każdej z tych liczb równa się drugiej (nie uwzględniając tych dwóch liczb jako dzielników), np.* $220 = 1 + 2 + 4 + 71 + 142$ *(dzielniki 284),* $284 = 1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110$ *(dzielniki 220). Znajdź wszystkie pary liczb zaprzyjaźnionych mniejszych od 100000.*

Zadanie 11. *Utwórz listę 1000 najmniejszych liczb pierwszych złożonych wyłącznie z liczb nieparzystych : 1, 3, 5, 7, 9 (wskazówka: While, IntegerDigits, FreeQ, AppendTo).*

Zadanie 12. *Utwórz listę 50 najmniejszych liczb pierwszych składających się wyłącznie z liczb : 1 i 3 ale takich, że obie te cyfry muszą w niej wystąpić.*

Zadanie 13. *Liczba 4231 jest największą liczbą pierwszą złożoną dokładnie z liczb 1, 2, 3, 4. Znajdź największą liczbę pierwszą złożoną dokładnie z liczb 1, 2, 3, 4, 5, 6, 7.*

Zadanie 14. *Znajdź wszystkie pary liczb m i n takie że suma sześcianów cyfr liczby m jest równa jest n, a suma sześcianów cyfr liczby n jest równa m.*

Zadanie 15. *Liczbę 1201 możemy przedstawić w postaci* $x^2 + ny^2$ *na szesnaście sposobów. Znajdź je wszystkie (oczywiście x, y, n mają być liczbami naturalnymi).*

Zadanie 16. *Do danej liczby dodajemy tę, która powstaje przez napisanie jej cyfr w odwrotnym porządku, np.*$1999 + 9991 = 11990$. *Do tak otrzymanej liczby stosujemy ponownie tę samą operację:* $11990 + 9911 = 21901$ *i tak dalej. Po kilku następnych iteracjach otrzymamy liczbę palindromiczną 712217. Dla każdej początkowej liczby n zawsze dojdziemy do liczby palindromicznej. Napisz procedurę, która zlicza, za którym to będzie razem i jaką liczbę otrzymamy.*

Zadanie 17. *Istnieją liczby naturalne, które są kwadratami i zawierają każdą z 10 cyfr dokładnie jeden raz (oczywiście są one dziesięciocyfrowe). Najmniejsza z nich to* $1026753849 = 32043^2$*, zaś największa to* $9814072356 = 99066^2$*. Znajdź wszystkie pozostałe.*

Zadanie 18. *Dawno temu, podczas wojny rzymsko-żydowskiej Józef Flawiusz wraz z grupą 41 żydowskich powstańców został otoczony przez Rzymian w jaskini! Woląc samobójstwo od poddania powstańcy postanowili utworzyć krąg i zabijać co trzecią osobę, aż nikt nie pozostanie przy życiu! Jednak Flawiusz wraz z przyjacielem nie zgadzali się na to bezsensowne samobójstwo i szybko wyliczył, gdzie powinni stanąć on i jego przyjaciel aby jako ostatni zostali tylko oni! Znajdź te miejsca. Wskazówka: RotateLeft, Rest.*

Rozwiązania

Zadanie 1

```
dzielniki[n_] := Length[FactorInteger[n]]
```

Zadanie 2

```
tr = Table[2 n - 1, {n, 1, 30}];
suma = 0;
For[i = 1, i <= 30, i++, suma = suma + tr[[i]]; Print[suma]]
```

Zadanie 4

```
doskonala[n_] := n == Apply[Plus, Divisors[n]] - n
For[i = 1, i <= 10000, i++, If[doskonala[i], Print[i]]]
For[i = 33500000, i <= 33600000, i++, If[doskonala[i], Print[i]]]
```

Zadanie 7

```
For[i = 146, i <= 100000, i++,
    If[i == Apply[Plus, Map[Factorial, IntegerDigits[i]]], Print[i]]]
```

Zadanie 8

```
pot[n_] := n^5
For[i = 2, i <= 200000, i++,
 If[i == Apply[Plus, Map[pot, IntegerDigits[i]]], Print[i]]]
```

Zadanie 9

```
For[i = 1000, i <= 9999, i++, l = IntegerDigits[i, 100];
 If[i == (l[[1]] + l[[2]])^2, Print[i]]]
```

Zadanie 10

```
zap = {}; For[i = 1, i <= 100000, i++, j = Apply[Plus, Divisors[i]] - i;
  If[i == Apply[Plus, Divisors[j]] - j && i < j, AppendTo[zap, {i, j}]]]
zap
```

Zadanie 11

```
i = 1;
sukces = 0; lis = {};
While[sukces <= 10000, lista = IntegerDigits[Prime[i]];
 If[FreeQ[lista, 2] && FreeQ[lista, 4] && FreeQ[lista, 6] &&
   FreeQ[lista, 8], AppendTo[lis, Prime[i]]; sukces++]; i++]
lis
```

Zadanie 12

```
i = 1;
sukces = 0; lis = {};
While[sukces <= 50, lista = Union[IntegerDigits[Prime[i]]];
 If[{1, 3} == lista, AppendTo[lis, Prime[i]]; sukces++]; i++]
lis
```

Zadanie 13

```
sukces = 0; i = 7777777;
While[sukces == 0 && i > 0,
 If[PrimeQ[i] && Union[IntegerDigits[i]] == {1, 2, 3, 4, 5, 6, 7},
  sukces = 1; Print[i]]; i--]
```

Pracownia 17

Fraktale

Przydatne polecenia

`RandomInteger[{1, 3}]`	wylosuj jedną z liczb 1,2,3 (z równym prawdopodobieństwem)
`ListPlot[{0,0},{0.2,0.3}]`	zaznacz w układzie współrzędnych punkty
`Switch[wyr,1,pol1,2,pol2,...]`	oblicz wyrażenie *wyr*, jeżeli jest ono 1, wykonaj polecenie *pol1*, jeżeli 2, to wykonaj *polecenie2* itd.
`A.B`	oblicz iloczyn macierzy A i B

Wprowadzenie

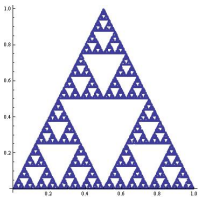
Gra w chaos

Przykład 1. *Mamy zadany trójkąt o wierzchołkach A, B i C. Wybierzmy dowolny punkt x_0 leżący wewnątrz trójkąta. Następnie rzucamy kostką. Jeżeli wypadnie 1 lub 2 to wybieramy wierzchołek A, jeżeli 3 lub 4 to B, a jeżeli 5 lub 6 to C. Oznaczmy przez x_1 punkt leżący na środku odcinka o końcach w x_0 i wylosowanym punkcie. Ponownie losujemy jeden z wierzchołków trójkąta i definiujemy x_2 jako środek odcinka łączącego x_1 i wylosowany punkt, itd.*

Narysuj pierwsze 100, 500, 5000, 20000 otrzymanych punktów.

```
PA = {0, 0};PB = {1, 0};PC = {1/2, 1}; X[0] = {1/2, 1/2};
For[i = 1, i <= 20000, i++, X[i] = Switch[RandomInteger[{1, 3}],
    1, 1/2 (X[i - 1] + PA),
    2, 1/2 (X[i - 1] + PB),
    3, 1/2 (X[i - 1] + PC)];]
ListPlot[Table[X[i], {i, 20000}], AspectRatio -> 1]
```

W wyniku otrzymamy trójkąt Sierpińskiego



IFS

Przekształcenie afiniczne T płaszczyzny jest to odwzorowanie postaci:

$$T(x) = Ax + b, \qquad x \in \mathbb{R}^2,$$

gdzie A jest odwzorowaniem liniowym $\mathbb{R}^2$, czyli macierzą 2x2, a $b \in \mathbb{R}^2$. Przekształcenie T działa więc na punkcie x, najpierw nakładając macierz A, a następnie przesuwając wynik o b (translacja). Najprostszymi przykładami odwzorowań liniowych są obroty, skalowanie, odbicia.

Gra w chaos jest szczególnym przypadkiem tzw. metody IFS (*iteracyjny system funkcji*, z jęz. ang. iterated function system). Metoda ta jest następująca. Mając zadany ciąg odwzorowań afinicznych $T_1, .., T_k$ oraz punkt początkowy x_0 losujemy liczbę J ze zbioru $1, .., k$ i definiujemy $x_1 = T_J(x_0)$, a następnie powtarzamy wielokrotnie tą czynność.

Zauważmy, że jest to rzeczywiście uogólnienie gry w chaos. Załóżmy, że wierzchołki trójkąta mają współrzędne:

$$P_1 = (0,0),\ P_2 = (1,0),\ P_3 = (1/2,1).$$

Wówczas n-ty krok gry w chaos polega na tym, że losujemy jeden z wierzchołków P_J i obliczamy

$$x_{n+1} = \frac{1}{2}(x_n + P_J).$$

Zdefiniujmy trzy odwzorowania afiniczne

$$\begin{aligned} T_1(x) &= \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \\ T_2(x) &= \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} x + \begin{pmatrix} 1/2 \\ 0 \end{pmatrix}, \\ T_3(x) &= \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix} x + \begin{pmatrix} 1/4 \\ 1/2 \end{pmatrix}. \end{aligned}$$

Łatwo sprawdzić, że $T_i(x) = \frac{1}{2}(x + P_i)$, więc metoda IFS jest tak naprawdę grą w chaos.

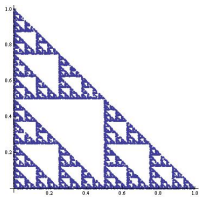
Oczywiście jeżeli odwzorowania T_i są dowolne to może się zdarzyć, że otrzymany ciąg punktów nie posiada ciekawych własności, np może uciekać do nieskończoności. Zakłada się jednak, że odwzorowania T_i zachowują kwadrat K o wierzchołkach $(0,0), (1,0), (1,1), (0,1)$ (wybór konkretnego kwadratu jest sprawą umowną i nieraz wygodniej jest ustalić inny kwadrat). Łatwo sprawdzić, czy odwzorowanie posiada tą własność. Mianowicie odwzorowania afiniczne przekształcają odcinki na odcinki. Wystarczy więc policzyć jak działa T_i na każdym z wierzchołków kwadratu K. Jeżeli obraz każdego z nich jest zawarty wewnątrz K, to oznacza, ze czworokąt rozpięty na tych punktach, będą obrazem K jest również zawarty w K.

Następnie pokazuje się, że jeżeli odwzorowania T_i przekształcają kwadrat K w siebie, to generowany metodą IFS zbiór punktów jest zbieżny do pewnego zbioru W (nie będziemy tu dokładnie definiować w jakim sensie jest ta zbieżność). Zbiór ten jest fraktalem, a więc figurą samopodobną. Ponadto dla każdego odwzorowania T_i, $T_i(W) \subset W$, ale zbiór W spełnia mocniejszą własność:

$$W = T_1(W) \cup T_2(W) \cup .. \cup T_k(W).$$

Powyższą własność łatwo sprawdzić dla trójkąta Sierpińskiego. Mianowicie zauważmy, że trójkąt Sierpińskiego składa się z trzech mniejszych trójkątów i każdy z nich można otrzymać zmniejszając dwukrotnie wyjściowy obiekt i przesuwając go o odpowiedni wektor, i tak też zostały powyżej zdefiniowane odwzorowania T_1, T_2, T_3.

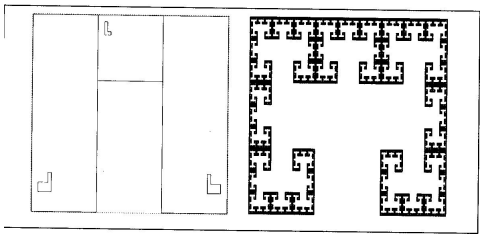
Przykład 2. *Narysuj modyfikację trójkąta Sierpińskiego.*



Zauważmy, że podobnie jak w poprzednim przypadku trójkąt składa się z trzech mniejszych identycznych trójkątów. Każdy z nich można otrzymać zmniejszając wyjściowy zbiór o połowę, a więc nakładając na niego macierz $A = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$, a następnie przesuwając o wektor odpowiednio $(0,0)$, $(1/2,0)$, $(0,1/2)$. Te trzy odwzorowania definiują odpowiedni system IFS.

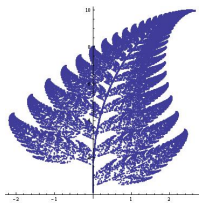
```
A = {{1/2, 0},{0, 1/2}}; B1 = {0, 0}; B2 = {1/2, 0}; B3 = {0, 1/2};
X[0] = {1/2, 1/2};
For[i = 1, i <= 10000, i++, X[i] = Switch[RandomInteger[{1, 3}],
    1, A.X[i - 1] + B1,
    2, A.X[i - 1] + B2,
    3, A.X[i - 1] + B3];]
ListPlot[Table[X[i], {i, 10000}], AspectRatio -> 1]
```

Przykład 3. *Wyznacz odwzorowania afiniczne zaznaczone po lewej stronie rysunku (każdy prostokąt powstał przez odpowiednie przekształcenie kwadratu z literą L w lewym górnym rogu) i narysuj labirynt Cantora.*



```
A1 = {{1/3, 0},{0, 1/3}};B1 = {1/3, 2/3};
A2 = {{0, -1},{1, 0}}.{{1, 0},{0, 1/3}}; B2 = {1/3, 0};
A3 = {{-1, 0},{0, 1}}.A2; B3 = {2/3, 0};
X[0] = {1/2, 1/2};
For[i = 1, i <= 40000, i++, X[i] = Switch[RandomInteger[{1, 3}],
    1, A1.X[i - 1] + B1,
    2, A2.X[i - 1] + B2,
    3, A3.X[i - 1] + B3];]
ListPlot[Table[X[i], {i, 40000}], AspectRatio -> 1]
```

Jednym z najbardziej znanych przykładów fraktali powstałych metodą IFS jest paprotka Barnsleya



Powstała ona przy pomocy poniższego kodu. Zauważmy, że odpowiedni system IFS składa się z czterech odwzorowań afinicznych. W odróżnieniu jednak od poprzednich przykładów nie są one losowane jednostajnie, a prawdopodobieństwami odpowiednio $\frac{1}{100}$, $\frac{85}{100}$, $\frac{7}{100}$, $\frac{7}{100}$,

```
A1 = {{0, 0},{0, 0.16}}; B1 = {0, 0};
A2 = {{0.85, 0.04},{-0.04, 0.85}}; B2 = {0, 1.6};
A3 = {{0.2, -0.26},{0.23, 0.22}}; B3 = {0, 1.6};
A4 = {{-0.15, 0.28},{0.26, 0.24}}; B4 = {0, 0.44};
X[0] = {0, 0};
For[i = 1, i <= 40000, i++, j = RandomInteger[{1, 100}];
 X[i] = If[j == 1, A1.X[i - 1] + B1,
   If[j <= 86, A2.X[i - 1] + B2,
    If[j <= 93, A3.X[i - 1] + B3, A4.X[i - 1] + B4 ]]]]
ListPlot[Table[X[i], {i, 40000}], AspectRatio -> 1]
```

Fraktalne własności trójkąta Pascala

Przykład 4. *Zaznacz pozycje w pierwszych 256 wierszach trójkąta Pascala, na których występują liczby parzyste.*

```
n = 256;
p = Table[0, {n}, {n}];
For[i = 1, i <= n, i++, p[[i, i]] = 1];
For[i = 1, i <= n, i++, p[[i, 1]] = 1];
For[i = 3, i <= n, i++,
  For[j = 2, j < i, j++,
   p[[i, j]] = Mod[p[[i - 1, j - 1]] + p[[i - 1, j]], 2]]];
punkty = {};
For[i = 1, i <= n, i++,
 For[j = 1, j <= i, j++,
  If[p[[i, j]] == 1,
   AppendTo[punkty, {1/2 - i/(2 n) + j/n, 1 - i/n}]]]]
ListPlot[punkty]
```

Zadania

Zadanie 1. *Rozpatrzmy grę w chaos, ale tym razem dla pięciokąta. Tzn. startujemy w dowolnego punktu leżącego wewnątrz pięciokąta x_0. Losujemy liczbę od 1 do 5 i w zależności od wylosowanej liczby wybieramy odpowiedni wierzchołek. Tym razem jednak kolejny punkt nie jest środkiem odcinka łączącego x_0 z wylosowanym wierzchołkiem, ale dzieli go w stosunku* $1/2$ *(tzn. x_0 liczymy z wagą 1/3, a wylosowany wierzchołek z wagą 2/3). Narysuj pierwsze 10000 punktów.*

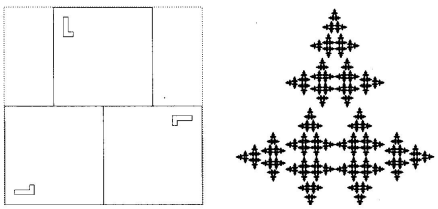
Zadanie 2. *Jak wyżej, ale tym razem dzielimy odcinki w stosunku* $3/5$.

Zadanie 3. *Napisz grę w chaos dla sześciokąta foremnego. W jakim stosunku należy dzielić kolejne odcinki, aby otrzymany ciekawy rysunek?*

Zadanie 4. *Napisz grę w chaos dla ośmiokąta foremnego.*

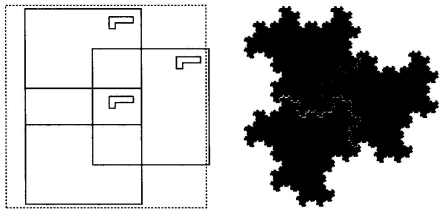
Zadanie 5. *Narysuj poniższe fraktale metodą IFS*[1]

Bliźniacza choinka

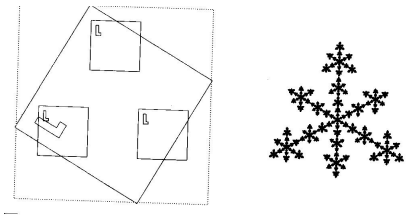


[1]Grafiki zostały skopiowane z książki: H.O.Peitgen, H.Jürgens, D.Saupe, *Granice Chaosu. Fraktale*

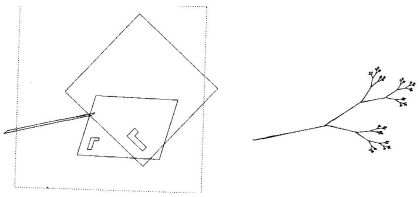
Smok



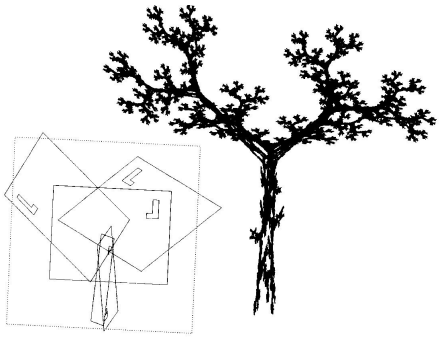
Kryształek



Gałązka



Drzewo



Zadanie 6. *Zaznacz pozycje w pierwszych 70 wierszach trójkąta Pascala, na których występują liczby podzielne przez 3.*

Zadanie 7. *Zaznacz w trójkącie Pascala jedynie te punkty, które*

a) zostawiają resztę 2 modulo 3 (100 wierszy);

b) zostawiają resztę 3 modulo 5 (100 wierszy);

c) są podzielne przez 11 (200 wierszy);

Rozwiązania

Zadanie 1

```
PA = {1/6, 0};PB = {5/6, 0};PC = {1, 2/3};PD = {1/2, 1};PE = {0, 2/3};
X[0] = {1/2, 1/2};
For[i = 1, i <= 20000, i++, X[i] = Switch[RandomInteger[{1, 5}],
    1, 1/3 X[i - 1] + (2 PA)/3,
    2, 1/3 X[i - 1] + (2 PB)/3,
    3, 1/3 X[i - 1] + (2 PC)/3,
    4, 1/3 X[i - 1] + (2 PD)/3,
    5, 1/3 X[i - 1] + (2 PE)/3];]
ListPlot[Table[X[i], {i, 20000}], AspectRatio -> 1]
```

Zadanie 6

```
n = 70;
p = Table[0, {n}, {n}];
For[i = 1, i <= n, i++, p[[i, i]] = 1];
For[i = 1, i <= n, i++, p[[i, 1]] = 1];
For[i = 3, i <= n, i++,
  For[j = 2, j < i, j++,
   p[[i, j]] = Mod[p[[i - 1, j - 1]] + p[[i - 1, j]], 3]]];
punkty = {};
For[i = 1, i <= n, i++,
 For[j = 1, j <= i, j++,
  If[p[[i, j]] == 0,
   AppendTo[punkty, {1/2 - i/(2 n) + j/n, 1 - i/n}]]]]
ListPlot[punkty]
```

Pracownia 18

Gra w życie

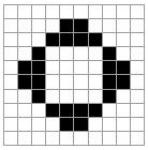
Wprowadzenie

Reguły *gry w życie* są bardzo proste. Dzielimy płaszczyznę na kwadratowe komórki. Każda komórka ma ośmiu sąsiadów, są to komórki przylegające do niej bokami i rogami. Komórka może znajdować się w jednym ze stanów, jest *martwa* lub *żywa*. Następnie w każdym z kolejnych kroków komórka może zmienić swój stan wg następujących reguł:

- żywa komórka pozostanie żywa jeżeli dokładnie dwie lub trzy komórki spośród jej sąsiadów są żywe, inaczej komórka umrze (z samotności lub tłoku);
- martwa komórka ożyje jeżeli jest otoczona dokładnie trzema sąsiadami.

Okazuje się w zależności od początkowego stanu komórek zachowanie układu może być zupełnie różne i często prowadzi do zdumiewających efektów.

Przykład 1. *Zagraj w życie zaczynając od poniższego stanu*



```
k = {{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 1, 0, 0, 0},
    {0, 0, 1, 0, 0, 0, 0, 1, 0, 0},
    {0, 1, 1, 0, 0, 0, 0, 1, 1, 0},
    {0, 1, 1, 0, 0, 0, 0, 1, 1, 0},
    {0, 0, 1, 0, 0, 0, 0, 1, 0, 0},
    {0, 0, 0, 1, 1, 1, 1, 0, 0, 0},
    {0, 0, 0, 0, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0}};
lista = {k};
For[m = 2, m <= 50, m++,
 kold = k;
 For[i = 1, i <= 10, i++, For[j = 1, j <= 10, j++, zywe = 0;
   If[i > 1 && j > 1 && kold[[i - 1, j - 1]] == 1, zywe++];
   If[i > 1 && kold[[i - 1, j]] == 1, zywe++];
   If[i > 1 && j < 10 && kold[[i - 1, j + 1]] == 1, zywe++];
```

```
   If[ j > 1 && kold[[i, j - 1]] == 1, zywe++];
   If[ j < 10 && kold[[i, j + 1]] == 1, zywe++];
   If[i < 10 && j > 1 && kold[[i + 1, j - 1]] == 1, zywe++];
   If[i < 10 && kold[[i + 1, j]] == 1, zywe++];
   If[i < 10 && j < 10 && kold[[i + 1, j + 1]] == 1, zywe++];
   If[kold[[i, j]] == 1,
    If[zywe == 2 || zywe == 3, kold[[i, j]] = 1, k[[i, j]] = 0],
    If[zywe == 3, k[[i, j]] = 1, k[[i, j]] = 0]]
   ]];
 AppendTo[lista, k];
 ]
Manipulate[ArrayPlot[lista[[m]], Mesh -> True], {m, 1, 20, 1}]
```

Zadania

Zadanie 1. *Znajdź w Internecie przykłady gry w życie (ang. Game of Life) i przetestuj je korzystając z powyższego kodu (np. pl.wikipedia.org/wiki/Gra_w_życie , en.wikipedia.org/wiki/Conway's_Game_of_Life)*

Zadanie 2. *Utwórz losową macierz o rozmiarze* 20×20 *(*50×50*) składającą się z zer i jedynek i zagraj w życie.*

Zadanie 3. *(Reguła jeden z ośmiu) Rozpatrzmy następujący wariant gry w życie:*

- *komórka żywa zawsze zostaje żywa;*
- *komórka martwa ożywia sie, gdy dokładnie jeden jej sąsiad jest żywy.*

Utwórz dużą macierz np. 30×30 *lub* 50×50 *i następnie*

a) zacznij układu w którym dokładnie jedna komórka znajdująca się w samym środku macierzy jest żywa. Narysuj obraz otrzymany po 20, 30 krokach.

b) przetestuj różne możliwości układu początkowego.

Zadanie 4. *(Reguła większości) Rozważmy kolejny wariant gry w życie: jeżeli 5 lub więcej komórek spośród 9 sąsiadujących ze sobą komórek (włączając komórkę środkową) jest żywych, to ta komórka stanie się, lub pozostanie żywa. Wystartuj z losowej macierzy* 50×50 *(*100×100*) i wykonaj około 20-30 kroków. Rozkład żywych komórek powinien się ustabilizować.*

Zadanie 5. *(Reguła parzystości) Tym razem stan komórki zależy od niej samej (C) oraz od sąsiadów: północnego (N), południowego (S), wschodniego (E) i zachodniego (W). Jeżeli suma C + N + S + E + W (komórka żywa ma wartość 1, a martwa 0)jest nieparzysta, to komórka jest żywa, jeżeli parzysta to komórka jest martwa.*

a) Zagraj w życie rozpoczynając od jednej żywej komórki umieszczonej w lewym dolnym rogu.

b) Zagraj w życie od stanu w którym żywe komórki tworzą kwadrat 4x4 umieszczony w samym środku obszaru.

Rozwiązania

Zadanie 2

```
n = 50;
kk = Table[RandomInteger[], {n}, {n}];
lista = {kk};
For[m = 2, m <= 150, m++,
 kold = kk;
 For[i = 1, i <= n, i++, For[j = 1, j <= n, j++, zywe = 0;
   If[i > 1 && j > 1 && kold[[i - 1, j - 1]] == 1, zywe++];
   If[i > 1 && kold[[i - 1, j]] == 1, zywe++];
   If[i > 1 && j < n && kold[[i - 1, j + 1]] == 1, zywe++];
   If[ j > 1 && kold[[i, j - 1]] == 1, zywe++];
   If[ j < n && kold[[i, j + 1]] == 1, zywe++];
   If[i < n && j > 1 && kold[[i + 1, j - 1]] == 1, zywe++];
   If[i < n && kold[[i + 1, j]] == 1, zywe++];
   If[i < n && j < n && kold[[i + 1, j + 1]] == 1, zywe++];
   If[kold[[i, j]] == 1,
    If[zywe == 2 || zywe == 3, kold[[i, j]] = 1, kk[[i, j]] = 0],
    If[zywe == 3, kk[[i, j]] = 1, kk[[i, j]] = 0]]
   ]];
 AppendTo[lista, kk];
 ]
Manipulate[ArrayPlot[lista[[m]], Mesh -> True], {m, 1, 150, 1}]
```

Zadanie 4

```
n = 100;
kk = Table[RandomInteger[], {n}, {n}];
lista = {kk};
For[m = 2, m <= 30, m++,
 kold = kk;
 For[i = 1, i <= n, i++, For[j = 1, j <= n, j++, zywe = 0;
   If[kold[[i, j]] == 1, zywe++];
   If[i > 1 && j > 1 && kold[[i - 1, j - 1]] == 1, zywe++];
   If[i > 1 && kold[[i - 1, j]] == 1, zywe++];
   If[i > 1 && j < n && kold[[i - 1, j + 1]] == 1, zywe++];
   If[ j > 1 && kold[[i, j - 1]] == 1, zywe++];
   If[ j < n && kold[[i, j + 1]] == 1, zywe++];
   If[i < n && j > 1 && kold[[i + 1, j - 1]] == 1, zywe++];
   If[i < n && kold[[i + 1, j]] == 1, zywe++];
   If[i < n && j < n && kold[[i + 1, j + 1]] == 1, zywe++];
   If[zywe >= 5 , kk[[i, j]] = 1, kk[[i, j]] = 0]
   ]];
 AppendTo[lista, kk];
 ]
Manipulate[ArrayPlot[lista[[m]], Mesh -> True], {m, 1, 30, 1}]
```

Pracownia 19

Żółwik

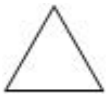
Wyobraźmy sobie żółwia znajdującego się na kartce papieru. Jest to inteligentny żółw i rozumie kilka prostych poleceń:

'F' idź do przodu o jedną jednostkę;

'+' obróć się w lewo o ustalony kąt;

'-' obróć się w prawo o ustalony kąt.

Żółwik ma troche przybrudzony ogon, więc podczas poruszania zostawia za sobą ślad. Droga żółwia zależy zarówno od polecenia jak i ustawionego kąta. Dla przykładu jeżeli ustalony kąt wynosi $\frac{2\pi}{3}$, to żółw na polecenie 'F+F+F' narysuje trójkąt równoboczny:



Natomiast jeżeli ustalony kąt wynosi $\frac{\pi}{6}$, to widząc polecenie

'FF+++FF+++F+++FF+++FF+++FFF+FFF++++FFF++++FFF++++++FFF+++FFF'

żółw narysuje domek

(w stanie początkowym żółw znajdował się w lewym dolnym rogu i był skierowany w prawo).

Przykład 1. *Narysuj rysunki, które wykona żółw widząc polecenia: 'F+F+F-F-F' oraz 'FF+FF+F+F-F-F+F+FF+F' jeżeli ustawiony kąt wynosi: a)* $\frac{\pi}{3}$*, b)* $\frac{\pi}{2}$*, c)* $\frac{2\pi}{3}$*.*

Przykład 2. *Napisz funkcję, która przyjmuje 2 parametry: polecenie, kąt i rysuje obrazek utworzony przez żółwia.*

```
rysuj[wzorzec_, kat_] := Module[{s = wzorzec, k = kat},
  znaki = Characters[s];
  n = Length[znaki];
  x = 0; y = 0; alfa = 0;
   lista = {{0, 0}};
  For[i = 1, i <= n, i++,
```

```
   Switch[znaki[[i]],
     "+", alfa = alfa + k,
     "-", alfa = alfa - k,
     "F", x = x + Cos[alfa]; y = y + Sin[alfa];
    AppendTo[lista, {x, y}]]];
  Show[Graphics[Line[lista]]]
  ]
```

Przykład 3. *Wykorzystując funkcje rysuj rozwiąż ponownie przykład 1.*

Okazuje się, że mimo tak niewielkiego zbioru poleceń żółwik potrafi rysować bardzo skomplikowane obiekty np. fraktale. Zaczniemy od narysowania śnieżynki Kocha. W pierwszym kroku wywołujemy funkcję

```
rysuj["F+F--F+F", Pi/3]
```

W wyniku otrzymamy

Następnie zmniejszmy powyższy rysunek trzykrotnie i wklejmy go powyżej w miejsce każdego odcinka. Otrzymamy wówczas

W terminach komend dla żółwia powyższa operacja polega na zastąpieniu w ciągu 'F+F–F+F' każdego polecenia 'F' tym samym ciągiem. Tak więc powyższy obrazek został narysowany przez żółwia poleceniem

```
rysuj["F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F", Pi/3]
```

(Zauważmy, że tak naprawdę rysunek powinien być trzykrotnie większy. Jednak Mathematica skaluje rysunek do standardowego okna, dzięki czemu nie musimy się troszczyć o zmniejszanie rysunku).

Powtórzmy ponownie tą samą czynność, a więc w miejsce każdego małego odcinka wklejmy krzywą od której zaczęliśmy. Tak więc w powyższym poleceniu musimy zastąpić każde 'F' przez 'F+F–F+F'. W wyniku otrzymamy komendę

```
rysuj["F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+
F--F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F+F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F", Pi/3]
```

i odpowiadający jej obrazek

Wykonywanie kolejnych kroków wydaje się być kłopotliwe. Chcielibyśmy zautomatyzować powyższą procedurę. I tu z pomocą przychodzą nam dwa polecenia: **Nest** i **StringReplace**.

Komendy

```
Nest[f, x, 3]
Nest[(1 + #)^2 &, x, 3]
```

zwracają odpowiednio $f[f[f[x]]]$ i $(1+(1+(1+x)^2)^2)^2$. Polecenie

```
 StringReplace["string",{s1->sp1,s2->sp2,...}]
```

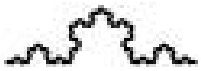
zastępuje w napisie 'string' wszystkie pojawienia się 's1' przez 'sp1', 's2' przez 'sp2' itd. Dla przykładu wynikiem działania

```
StringReplace["abbaabbaa", "ab" -> "X"]
```

jest 'XbaXbaa'. Możemy zatem automatycznie wygenerować kolejne kroki opisujące śnieżynkę Kocha:

```
rysuj[Nest[StringReplace[#, {"F" -> "F+F--F+F"}] &, "F", 5], Pi/3]
```

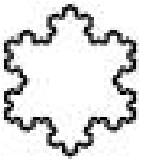
gdzie 5 oznacza stopień zagnieżdżenia funkcji (dokładność). W wyniku otrzymamy



Otrzymany powyżej rysunek stanowi jedynie fragment śnieżynki Kocha. Aby narysować całą śnieżynkę należy wystartować z trójkąta równobocznego i zastąpić każdy z boków powyższą krzywą. Polecenie

```
rysuj[Nest[StringReplace[#, {"F" -> "F-F++F-F"}] &, "F++F++F", 5],Pi/3]
```

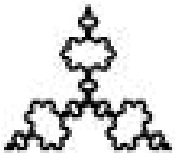
rysuje pełną śnieżynkę



Zauważmy, że użyta powyżej formuła różni się nieco od powyższych reguł. Mianowicie żółwik obraca się w odwrotną stronę. Ta zmiana spowodowana jest tym, że podczas rysowania wyjściowego trójkąta równobocznego żółw idzie w przeciwnym kierunku. Używając poprzedniego wzorca

```
rysuj[Nest[StringReplace[#, {"F" -> "F+F--F+F"}] &, "F++F++F", 5],Pi/3]
```

otrzymamy



Powyższa konstrukcja jest przykładem L-systemu. W dużym skrócie jest to zestaw reguł podstawiania, które pozwalają tworzyć słowa (ciągi znaków) składające się z 'F', '+' i '-', a więc mogą generować polecenia dla żółwia.

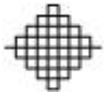
L-system, dzięki któremu utworzyliśmy śnieżynkę Kocha składa się z:

- słowa początkowego: 'F++F++F';
- reguł podstawiania: F -> F-F++F-F;
- parametru: kąta $\pi/3$.

Używając tej metody możemy wygenerować kolejne przykłady ciekawych krzywych. Polecenie

```
rysuj[Nest[StringReplace[#, {"F" -> "FF+F+F+FF+F+F-F"}] &, "F", 2], Pi/2]
```

narysuje krzywą Kocha



a dzięki poleceniu

```
rysuj[Nest[StringReplace[#, {"F" -> "F+F-F-FF+F+F-F"}] &, "F+F+F+F", 2], Pi/2]
```

otrzymamy tzw. wyspę Kocha

L-systemy umożliwiają używanie wielu reguł, zamiast jak dotychczas jednej. Dla kąta $\pi/3$, poleceniem '+F-F-F-' narysujmy

Następnie odcinek środkowy zamieńmy tą samą regułą F->+F-F-F+, ale pozostałe odcinki zastąpmy odbitą regułą F->-F+F+F-. Otrzymamy wówczas

Chcemy kontynuować tą czynność. Mamy więc następujący L-system

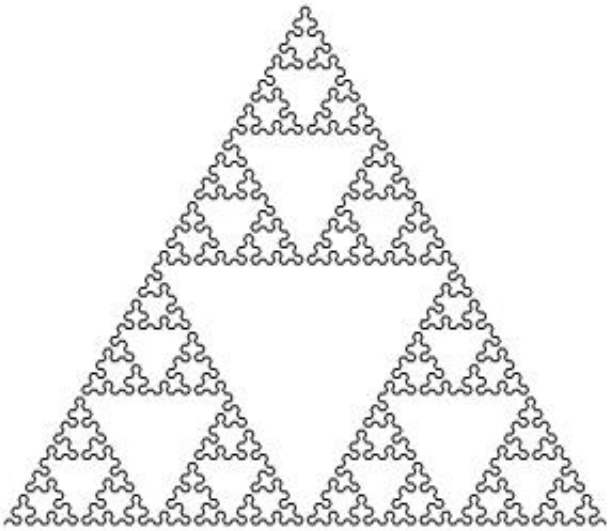
- słów początkowe: 'L';
- reguły podstawiania: L->+R-L-R+, R->-L+R+L-, L->F, L->R;
- kąt $\pi/3$.

Aby narysować więc krzywą ustalamy stopień zagnieżdżenia k. Startujemy z 'L' i k-krotnie używamy dwóch pierwszych reguł, a następnie zamieniamy wszystkie pojawienia się 'L' i 'R' przez 'F' (a więc korzystamy z dwóch ostatnich reguł). W ten sposób, ciąg poleceń

```
s = Nest[StringReplace[#, {"L" -> "+R-L-R+","R" -> "-L+R+L-"}] &, "L", 4]
s1 = StringReplace[s, {"L" -> "F", "R" -> "F"}]
rysuj[s1, Pi/3]
```

wygeneruje krzywą

Po zwiększeniu stopnia zagnieżdżenia do 7 otrzymamy krzywą przybliżającą trójkąt Sierpińskiego.

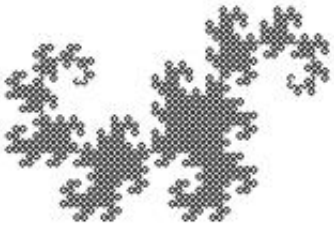


Kolejnym naturalnym krokiem jest więc napisanie funkcji, która jako parametry przyjmuje opis L-systemu (symbol startowy, wzorce, stopień zagnieżdżenia, kąt) i rysuje otrzymaną krzywą

```
rysuj1[start_, wzorce_, zaglebienie_, kat_] :=
 rysuj[
  StringReplace[
   Nest[
    StringReplace[#, wzorce] &, start, zaglebienie],
   LetterCharacter -> "F"],
  kat]
```

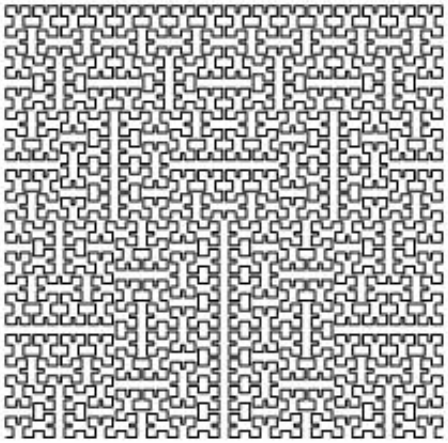
Kolejne przykłady: Smok

```
rysuj1[ "D", {"D" -> "-D++E", "E" -> "D--E+"}, 11, Pi/4]
```



Krzywa Hilberta (krzywa bez samoprzecieć wypełniająca cały kwadrat)

```
rysuj1[ "L", {"L" -> "+RF-LFL-FR+", "R" -> "-LF+RFR+FL-"}, 6, Pi/2]
```



Powyższy schemat nie pozwala na utworzenie rozgałęzionych struktur, np. krzewów i drzew. Problem jest następujący. Różne części drzewa powinny być narysowane przy pomocy różnych wzorców, inaczej rysujemy pień a inaczej koronę. Aby rozwiązać ten problem wprowadza się dodatkowe symbole '[' i ']'. Symbol '[' oznacza punkt rozgałęzienia, a więc początek rysowania gałęzi, a symbol ']' oznacza zakończenie definicji gałęzi. Dla przykładu weźmy ciąg

ABA[BBAA][CCBB] ABA[AABB] ABA

po opuszczeniu nawiasów otrzymamy ABAABAABA co odpowiada pniu a wyrażenia w nawiasach określają gałęzie. Żółw po napotkaniu na znak '[' powinien zapamiętać swoje położenie, narysować gałąź, a następnie gdy napotka ']' powrócić do poprzedniego stanu. Nawiasy mogą się zagnieżdżać.

Poniższy fragment kodu uwzględnia rozszerzenie L-systemów. Po napotkaniu na '[' aktualna pozycja żółwia, a więc pozycja i kąt nachylenia odkładane są na stos (z uwagi na możliwość zagnieżdżania nawiasów struktura stosu jest niezbędna)

```
lsystem[wzorzec_, kat_] := Module[{s = wzorzec, k = kat},
  znaki = Characters[s];
  n = Length[znaki];
  x = 0; y = 0; alfa = Pi/2;
   lista = {{0, 0}};
  stos = {};
  For[i = 1, i <= n, i++,
   Switch[znaki[[i]],
     "+", alfa = alfa + k,
     "-", alfa = alfa - k,
     "F", x = x + Cos[alfa]; y = y + Sin[alfa];
    AppendTo[lista, {x, y}],
     "[", AppendTo[stos, {x, y, alfa}],
     "]", {x, y, alfa} = Last[stos]; stos = Drop[stos, -1]
    ]];
  Show[Graphics[Line[lista]]]]

lsystem1[start_, wzorce_, zaglebienie_, kat_] :=
 lsystem[
  StringReplace[
   Nest[
    StringReplace[#, wzorce] &, start, zaglebienie],
   LetterCharacter -> "F"],
  kat]
```

Przykłady

```
lsystem1["F", "F" -> "F[+F]F[-F]F", 5, 25.7/90 * Pi/2]
```

```
lsystem1["B", {"F" -> "FF", "B" -> "F[+B]F[-B]+B"}, 6, 20/90 * Pi/2]
```

```
lsystem1["VZFFF", {"V" -> "[+++W][---W]YV", "W" -> "+X[-W]Z",
  "X" -> "-W[+X]Z", "Y" -> "YZ", "Z" -> "[-FFF][+FFF]F"}, 8,20/180*Pi]
```

```
lsystem1["x", {"x" -> "fy[+fy][--fy]fy", "y" -> "fx[++fx][-fx]fx"}, 6, Pi/3]
```