# Mathematica jako narzędzie badawcze
# Część piąta.
# Fraktale

*Czy koła są piękne?*
*Można udowodnić wiele teorii na ich temat,*
*wiele ich cech jest interesujących,*
*ale żeby koło miało być piękne?*
*Jest nudne, wszędzie takie samo;*
*podobnie linia prosta.*
*Coś, co jest takie samo w każdym miejscu,*
*nie wydaje się atrakcyjne.*
Benoît Mandelbrot
(za „Wiedza i życie", nr 7/2010 str. 60, „Pan od fraktali")

Celem zajęć będzie badanie struktur matematycznych generujących fraktale, a także analiza samych fraktali. Zajmiemy się także tworzeniem grafiki bazującej na fraktalach.

Na tych ćwiczeniach nie trzeba wykonywać wszystkich zadań. Zadania podzielone zostały na dwa bloki A i B - należy wybrać co najmniej 2 zadania kategorii A, oraz co najmniej 2 zadania z kategorii B. Po wykonaniu zadań z literkami proszę nie zapomnieć o wykonaniu dwóch zadań umieszczonych na samym końcu (obowiązkowo).

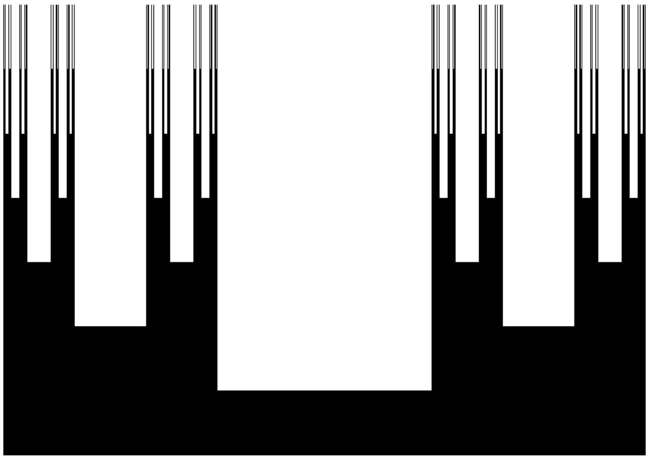
Sugerowane jest, aby przed zajęciami przypomnieć/przyswoić sobie następujące zagadnienia:

- Czym są i według jakich reguł powstają następujące struktury fraktalne: gwiazdka Kocha, trójkąt i dywan Sierpińskiego, zbiór Cantora, kostka Mengera. [5, 1]
- metoda Newtona-Raphsona (metoda stycznych) szukania pierwiastków funkcji - na czym polega? [4, 3].

- losowe metody generowania fraktali oraz afiniczne układy funkcji (IFS) [5, 6].

## Zadanie A1.

Napisz skrypt, który generuje kilka pierwszych kroków konstrukcji zbioru Cantora. Pomocne mogą okazać się funkcje `RealDigits[], Rectangle[]`.



Rysunek 1: Przykładowy wynik zadania A1.

## Zadanie A2.

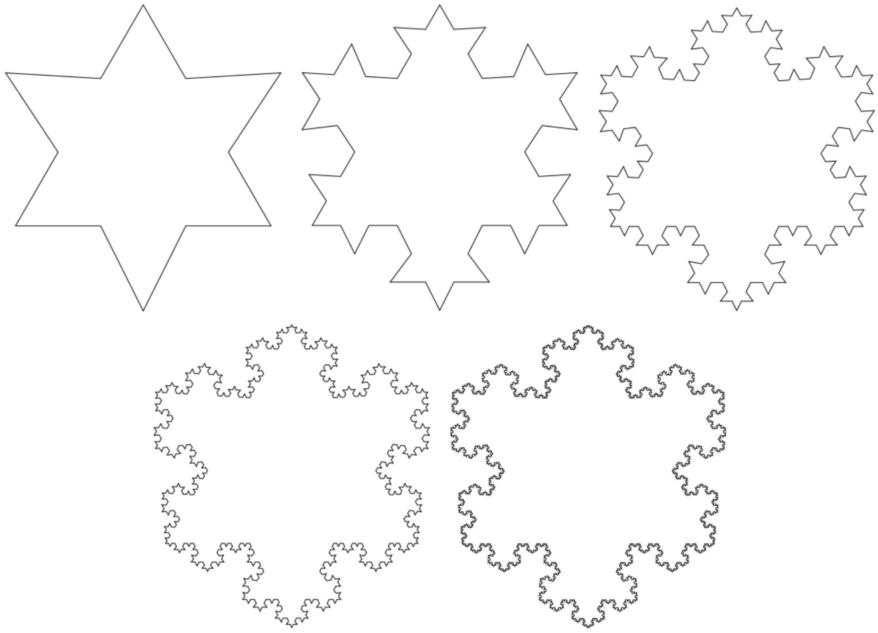
Napisz skrypt generujący kilka pierwszych kroków konstrukcji śnieżynki Kocha. Pomocna może okazać się funkcja z notatnika `http://if.pw.edu.pl/~siudem/MJNB/fraktale_pomoce.nb`.

## Zadanie A3.

Napisz skrypt generujący kilka pierwszych kroków konstrukcji trójkąta Sierpińskiego. Być może pomocne będzie tutaj umiejętne zastosowanie funkcji `Nest[]` albo `NestList[]` do operacji na listach.

## Zadanie A4.

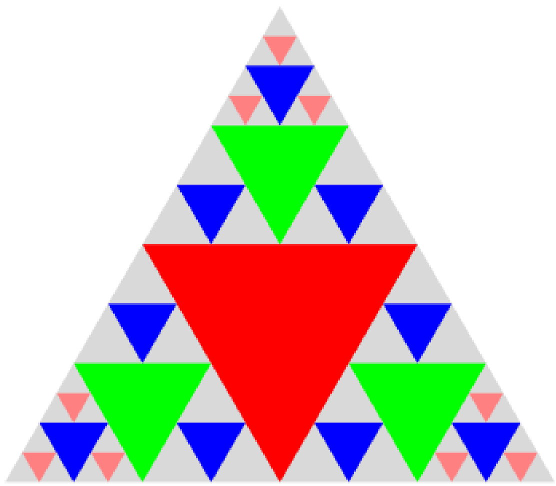
Napisz skrypt generujący kilka pierwszych kroków konstrukcji dywanu Sierpińskiego. Wystarczy tu niewielka modyfikacja skryptu z zadania A1.



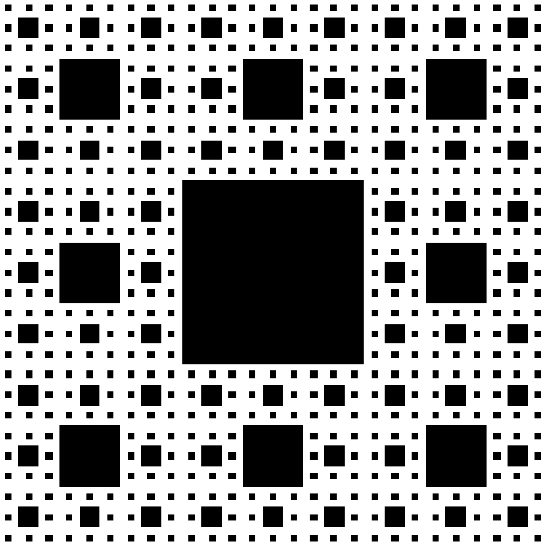
Rysunek 2: Przykładowy wynik zadania A2.

## Zadanie A5.

Napisz skrypt generujący kilka pierwszych kroków konstrukcji kostki Mengera. Pomocne może okazać się wcześniejsze wykonanie zadania A4. Niezbędna będzie funkcja `Graphicd3D[]`.



Rysunek 3: Przykładowy wynik zadania A3.

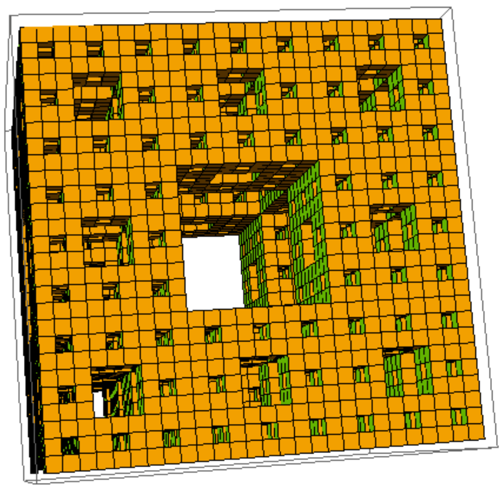


Rysunek 4: Przykładowy wynik zadania A4.

## Zadanie B1. (metoda Newtona-Raphsona)

Zaimplementuj, w postaci funkcji metodę Newtona-Raphsona. Zastosuj ją do wielomianu $p(z) = z^3 - 1$. Wygeneruj dla tego wielomianu zbiór Julii. Pomocne mogą okazać się funkcje `FixedPoints[]`, `Which[]`, `DensityPlot[]`, a w tej ostatniej użyteczna może okazać się opcja `ColorFunction[]`. Sprawdź jaki efekt daje zastosowanie tej metody do innych wielomianów.

W kolejnych zadaniach skupimy się na iterowaniu odwzorowań, które



Rysunek 5: Przykładowy wynik zadania A5.

dobierane będą losowo. przyjmijmy, że mamy pewien zbiór $X$. Na nasze potrzeby przyjmiemy, że $X = \mathbb{R}^2$. Na zbiorze tym zdefiniowane są funkcje $f_i \colon X \to X$, $i = 1, 2, ..., n$. Dysponujemy też wagami $w_i > 0$, $i = 1, 2, ..., n$. Jeden krok iteracji polega na wylosowaniu liczby $i$ ze zbioru $\{1, 2, ..., n\}$, a następnie wykonaniu funkcji $f_i$. Iteracja powtarzana wielokrotnie prowadzi do zaskakująco prostych i pięknych efektów.

## Zadanie B2. (Spirala, za [5])

Zaimplementuj poniższy układ dwóch odwzorowań dobieranych losowo

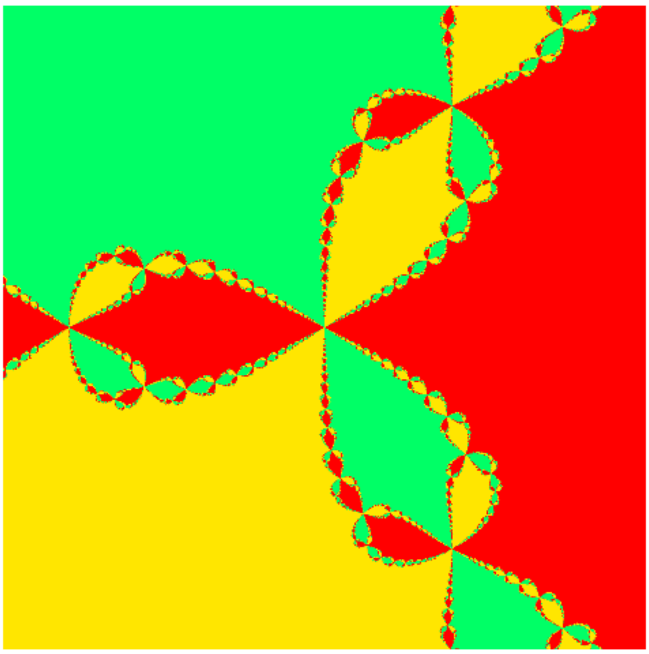
$$\begin{cases} x' = -0.4x - 1 \\ y' = -0.4y + 0.1 \end{cases}, \begin{cases} x' = 0.76x - 0.4y \\ y' = 0.4x + 0.76y \end{cases}.$$

Pomocne mogą okazać się funkcje `NestList[]`, `Point[]`, `RandomChoice[]`.

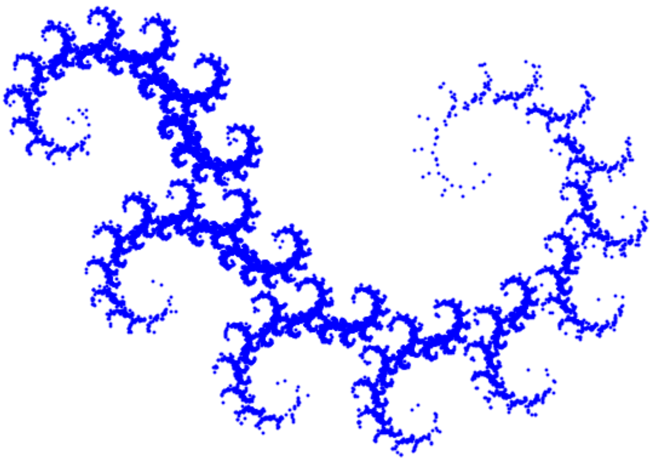
## Zadanie B3. (Choinka, za [5])

Zaimplementuj poniższy układ czterech odwzorowań dobieranych losowo

$$\begin{cases} x' = -0.67x - 0.02y \\ y' = -0.18x + 0.81y + 10 \end{cases}, \quad \begin{cases} x' = 0.4x + 0.4y \\ y' = -0.1x + 0.4y \end{cases},$$



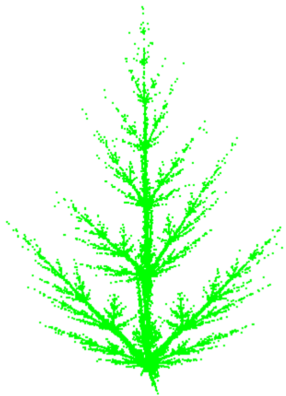
Rysunek 6: Przykładowy wynik zadania B1.



Rysunek 7: Przykładowy wynik zadania B2.

$$\begin{cases} x' = -0.4x - 0.4y \\ y' = -0.1x + 0.4y \end{cases}, \quad \begin{cases} x' = -0.1x \\ y' = 0.44x + 0.44y - 2 \end{cases}.$$

Pomocne mogą okazać się funkcje `NestList[]`, `Point[]`, `RandomChoice[]`.



Rysunek 8: Przykładowy wynik zadania B3.

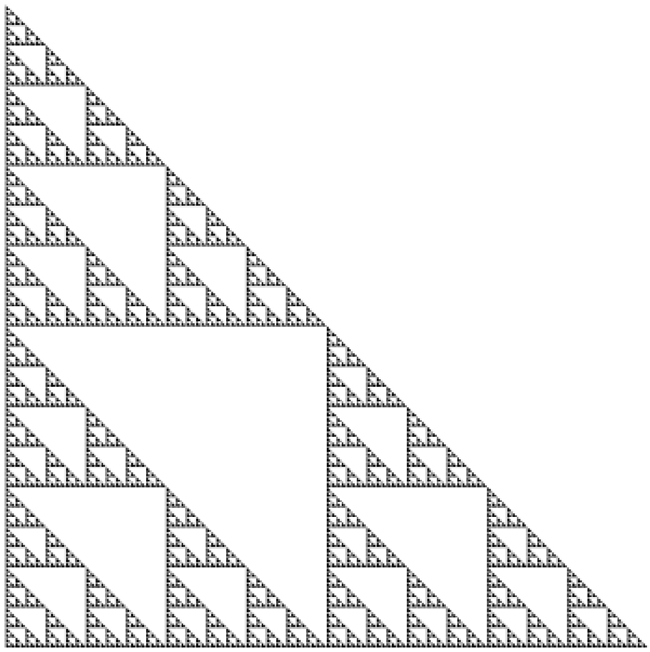
Oba powyższe przykłady wykorzystywały funkcje afiniczne, w kolejnych również będziemy korzystali z funkcji afinicznych, ale wygodniejsze okaże się wykorzystanie funkcji `afiniczne[]` z notatnika `http://if.pw.edu.pl/~siudem/MJNB/fraktale_pomoce.nb`. Wyraża on operacje afiniczne w postaci mnożenia przez macierz

$$A = \begin{bmatrix} r\cos(\theta) & -s\sin(\psi) \\ r\sin(\theta) & s\cos(\psi) \end{bmatrix},$$

i przesuwania o wektor $(a, b)^T$

## Zadanie B4. (Trójkąt Sierpińskiego inaczej, za [6])

Wygeneruj trójkąt Sierpińskiego, stosując trzy odwzorowania afiniczne o parametrach `afiniczne[0,0,0.5,0.5,0,0]`, `afiniczne[0,0,0.5,0.5,0,1]`, `afiniczne[0,0,0.5,0.5,1,0]` do trójkąta `Polygon[{{0,0},{0,2},{2,0}}]`. Pomocne okaże się wykorzystanie funkcji `Nest[]`, `GeometricTransformation[]`.



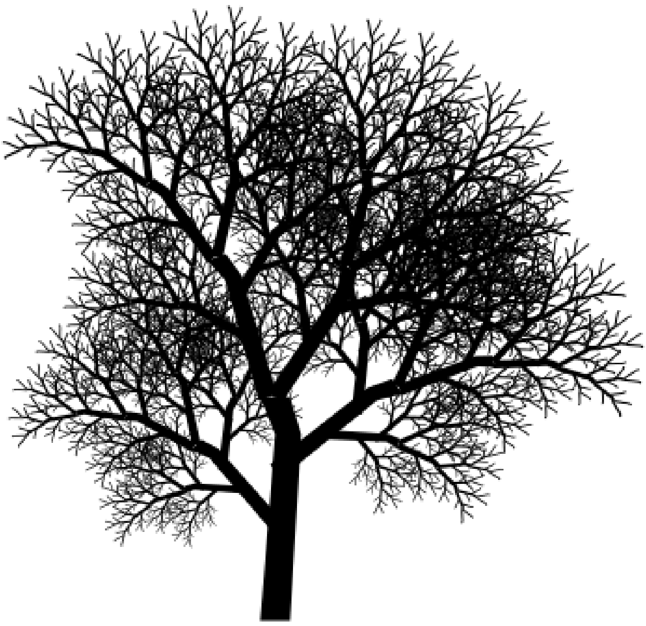
Rysunek 9: Przykładowy wynik zadania B4.

## Zadanie B5. (Drzewko, za [6])

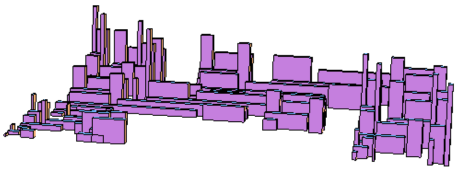
Wygeneruj drzewko, stosując cztery odwzorowania afiniczne o parametrach `afiniczne[20°,20°,0.65,2/3,-0.1,3]`, `afiniczne[45°,45°,0.5,0.5,0,1.3]`, `afiniczne[−30°,−30°,0.6,2/3,0,3]`,`afiniczne[−55°,−55°,-0.6,2/3,0,2]` do wielokąta `Polygon[{{-0.2,0},{0.2,0},{0.35,2.5},{0.2,3},{-0.2,3},{0,2.5}}]`. Pomocne okaże się wykorzystanie funkcji `NestList[]`, `GeometricTransformation[]`.

## Zadanie B6. (Miasto, za [6])

Wygeneruj fraktalny obraz miasta. Posłuż się w tym celu fragmentem kodu z pliku `http://if.pw.edu.pl/~siudem/MJNB/fraktale_pomoce.nb`. Pomocne okaże się wykorzystanie funkcji `Apply[]`, `Nest[]`, `Flatten[]`, `Cuboid[]`.



Rysunek 10: Przykładowy wynik zadania B5.



Rysunek 11: Przykładowy wynik zadania B6.

## Zadanie 1.

Należy wymyślić samemu albo znaleźć w literaturze lub internecie ciekawy, oryginalny i nietrywialny obraz generowany przez zastosowanie odwzorowań afinicznych i narysować taki obrazek.

## Zadanie 2.

Uzyskane wyniki z sekcji A i B oraz zadania 1. należy umieścić w prezentacji wykonanej w środowisku Mathematica.

## Literatura


[1] `http://mathworld.wolfram.com/`

[2] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, wydanie drugie, Westview Press, (2003).

[3] A. F. Beardon, *Iteration of Rational Functions*, Springer, New York, (2000).

[4] Z. Fortuna, B. Macukow, J. Wąsowski, *Metody Numeryczne*, Wydawnictwa Naukowo-Techniczne, (2006).

[5] J. Kudrewicz, *Fraktale i chaos*, wydanie trzecie zmienione, Wydawnictwa Naukowo-Techniczne, Warszawa, (1996).

[6] Ch. Getz, J. Helmstedt, *Graphics with Mathematica Fractals, Julia Sets, Patterns and Natural Forms.*, Elsevier, Amsterdam, (2004).