



Maszynki do fraktali

Studenci: Marcin Figiel, Tomasz Sabala
Pod opieką prof. dr. hab. Macieja A. Nowaka
Instytut Fizyki UJ

1. Wstęp

W poprzednim numerze (*Foton 111*) znalazł się artykuł pt. *Fraktale*, w którym na prostych przykładach przedstawiliśmy zagadnienie samopodobieństwa, ułamkowego wymiaru oraz zbiorów Julii. Czytelnikom niezaznajomionym z tymi tematami gorąco polecamy lekturę wspomnianego artykułu, którego ten tekst jest kontynuacją. Z jego treścią można zapoznać się także w Internecie (link podany na końcu artykułu).

Wszystko, co mówimy, jest zakodowane językiem, którego używamy. Składa się on z abstrakcyjnych pojęć określających przedmioty, czynności czy cechy. Pojedyncze pojęcia składają się w większe grupy tworzące zdania, które niosą ze sobą więcej treści niż pojedyncze słowo. Okazuje się, że fraktale także można zapisać za pomocą „słów” – prostych instrukcji, których zestawienie daje zaskakujący efekt w postaci skomplikowanej matematycznej struktury. Do takiej konstrukcji może służyć abstrakcyjne pojęcie wirtualnej maszyny zwanej MRCM (*multiple reduction copy machine* – maszyna wielokrotnie redukująca i kopiująca), czyli tytułowa maszynka do fraktali.

2. Czym jest MRCM?

Wyobraźmy sobie jakąś maszynę. Aby mogła ona pracować, trzeba jej dostarczyć materiału, nasza maszyna będzie działać na obrazkach. Nad tym materiałem ma ona wykonać konkretne czynności, a na koniec zwraca „obrobiony” materiał. Jeżeli dołożymy sprzężenie zwrotne, to otrzymamy maszynę, która działa w nieskończoność „obrabiając” to, co już „obrobiła”. Mamy teraz maszynę działającą w cyklu. Teraz przyjrzyjmy się jej działaniu. Załóżmy, że nasza maszyna pomniejsza to, co do niej wrzucamy (choćbyż dozwolone są wszystkie transformacje afiniczne – skalowanie, obrót, odbicie, jednokładne odwzorowania itp.).

Jeżeli dodatkowo założymy, że wynik składa się z wielu pomniejszonych kopii obrazka początkowego, otrzymamy właśnie MRCM. MRCM można odnieść w matematyce do deterministycznych iterowanych układów funkcji (*deterministic iterated function systems* – IFS). Każda taka maszyna charakteryzuje się tym, ile tworzy kopii obrazka wejściowego, jak bardzo pomniejszona lub przekształcona jest każda z kopii oraz jak poukładane są kopie tworząc ostateczny obrazek.

2.1. Transformacje afiniczne

W tym miejscu warto wspomnieć, czym dokładnie są transformacje afiniczne. Wiemy, że nasza maszyna może dokonywać skalowania, obrotów, odbić oraz odwzorowań jednokładnych. Nasza MRCM pracuje na płaskich obrazkach, podzbiorach płaszczyzny. Taka płaszczyzna może być opisana za pomocą kartezjańskiego układu współrzędnych: dobrze znanych osi x oraz y . Położenie każdego punktu P płaszczyzny możemy określić względem początku układu podając jego współrzędne: $P = (x_p, y_p)$. Takie punkty można dodawać lub mnożyć przez skalar (liczbę). Przy dodaniu punktów $P_1 = (x_1, y_1)$ i $P_2 = (x_2, y_2)$ powstanie punkt $P_3 = (x_1 + x_2, y_1 + y_2)$. A jeżeli pomnożymy punkt P_1 przez liczbę b to otrzymamy punkt $P_4 = (bx_1, by_1)$. Co jednak najbardziej interesuje nas w tej dziedzinie, to odwzorowania afiniczne. Odwzorowania te przekształcają jedne punkty płaszczyzny w inne. Są one liniowe, czyli dla każdego punktu P_1 oraz P_2 , na które działa odwzorowanie F oraz liczb p i q prawdziwe jest równanie: $F(pP_1 + qP_2) = pF(P_1) + qF(P_2)$. Transformacje, które wykonuje nasza maszyna, można zapisać w języku takich właśnie odwzorowań. Załóżmy, że mamy punkt $P = (x, y)$, który przekształcany jest w punkt $F(P) = (u, v)$. Wtedy możemy zapisać:

$$u = ax + by + c$$

$$v = dx + ey + f$$

Takie odwzorowanie jest zdefiniowane przez 6 liczb a, b, c, d, e, f . Za obroty, skalowania i tego typu przekształcenia odpowiedzialne są współczynniki a, b, c, d , natomiast pozostałe dwa za translację (przesunięcie).

2.2. Cechy MRCM

Wyobraźmy sobie maszynę, która pobiera obrazek początkowy, pomniejsza go 2 razy i umieszcza 3 kopie w wierzchołkach trójkąta równobocznego. Weźmy za obrazek początkowy:



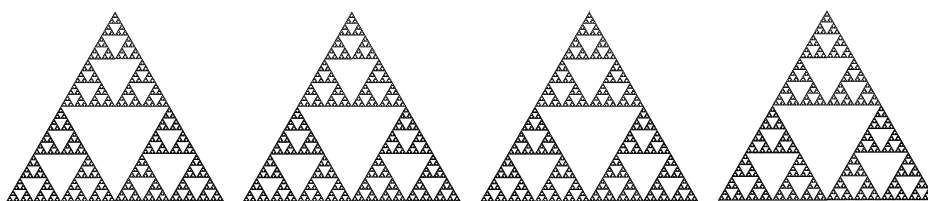
Rys. 1. Przykładowy obrazek początkowy

W wyniku działania maszyny otrzymamy:



Rys. 2. Pierwsze kroki działania maszyny na naszej róży

Po trzech cyklach działania maszyny widzimy coś zbliżonego do trójkąta Sierpińskiego. Zobaczmy, co się stanie, gdy użyjemy właśnie jego jako obrazka początkowego:

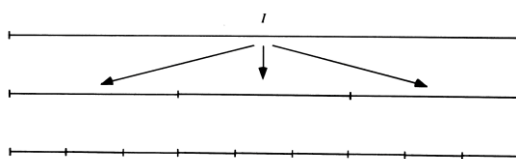


Rys. 3. Działanie maszyny nad trójkątem Sierpińskiego

Dana maszyna działająca na trójkąt Sierpińskiego nic nie zmieniła. Taki obrazek, do którego dąży maszyna niezależnie od obrazka na wejściu, nazywamy atraktorem. Do jednego atraktora może jednak dążyć wiele maszyn. Dodatkowo dana maszyna daje taki sam wynik niezależnie od wejściowego obrazka. To fascynujące, że nieważne, co wrzucimy do naszej maszyny, i tak otrzymamy taki sam wynik. Wyobraźmy sobie, że chcielibyśmy poznać sposób działania jakiejś konkretnej maszyny MRCM. W takim wypadku wystarczy uruchomić maszynę tylko raz: jeden cykl zdradzi nam wszystkie przekształcenia, które dokonywane są nad wejściowym obrazkiem. Oczywiście musimy wybrać kształt na tyle dowolny, by dało się zauważyć wszystkie przekształcenia afiniczne, bowiem wybierając przykładowo kwadrat nie dałoby się zauważyć np. odbicia lustrzanego czy obrotu o 90° . Sposób przekształcenia wejściowego obrazka przez MRCM nazwiemy schematem. Okazuje się, że przy użyciu MRCM można tworzyć nie tylko fraktale, ale różne inne struktury. Jednak ich opis przy użyciu tej metody wcale nie jest prostszy od innych, więc nie poświęca się temu dużo uwagi.

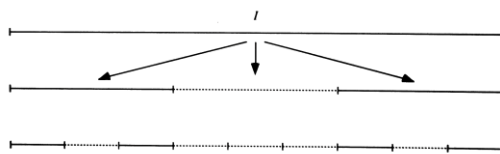
2.3. Konstrukcje fraktali

Teraz pokażemy, jak za pomocą deterministycznych iterowanych funkcji (które, jak już powiedzieliśmy, są matematycznym odpowiednikiem MRCM) konstruować znany nam już zbiór Cantora oraz dywan Sierpińskiego. Weźmy odcinek o długości 1. Niech nasza funkcja transformuje go zgodnie ze schematem:



Rys. 4. Przekształcenie odcinka w identyczny odcinek

Uzyskana transformacja nie jest zbyt ciekawa, zmienia odcinek w odcinek o tej samej długości. Jednak wystarczy „wyrzucić” środkową część, aby po kilku iteracjach otrzymać charakterystyczne punkty zbioru Cantora, według schematu:



Rys. 5. Przekształcenie odcinka w zbiór Cantora

Wynik działania możemy obejrzeć na poniższym obrazku:

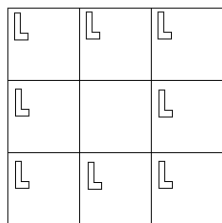


Rys. 6. Kolejne etapy powstawania zbioru Cantora

Chcąc zapisać te transformacje w języku MRCM musielibyśmy tylko ustalić, że obrazek początkowy skalujemy do $1/3$ wielkości i umieszczamy na środku przeciwległych końców obrazka wynikowego.

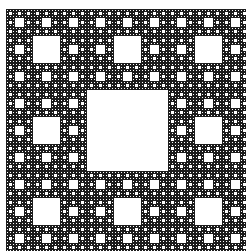
Skonstruujmy teraz dywan Sierpińskiego. Zróbmy schemat działania takiej funkcji. Niech bierze ona obrazek wejściowy, skaluje go do $1/9$ początkowej

wielkości (powierzchni) i umieszcza 9 kopii tak, aby na siebie nie zachodziły. Schemat wygląda tak:



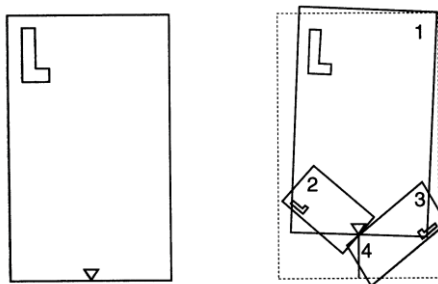
Rys. 7. Schemat działania maszyny. Pola z literką L to kopie rysunku wejściowego

Teraz usuńmy tę część, która znajduje się w środku. Jeżeli przeprowadzimy odpowiednio wiele iteracji, to otrzymamy taki kształt:



Rys. 8. Dywan Sierpińskiego

Co ciekawe, bardzo prosto można stworzyć na pierwszy rzut oka skomplikowane twory, takie jak paproć Barnsleya. Jest ona zakodowana za pomocą tylko czterech przekształceń, którym odpowiadają na poniższym schemacie kopie obrazka po lewej, oznaczone cyframi 1, 2, 3, 4. Transformacja 4 to przekształcenie wejściowego obrazka w kształt bliski odcinkowi. Transformacje zaś 1, 2 oraz 3 to proste translacje, obroty i przeskalowania, a w przypadku 3 obrazek został również rozciągnięty.



Rys. 9. Schemat przekształceń, dzięki którym otrzymamy paproć Barnsleya

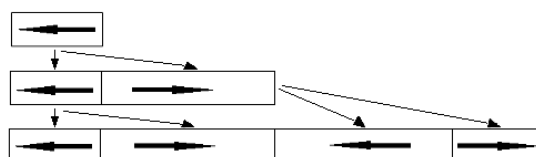
Kompletna paproć przedstawia się następująco:



Rys. 10. Gotowa paproć. Podobieństwo do naturalnej jest uderzające

3. L-układy

Załóżmy, że mamy prosty wielokomórkowy organizm, algę. Składa się ona tylko z jednego łańcucha komórek. Są dwa rodzaje komórek: wyspecjalizowane, które nie dzielą się, oraz niewyspecjalizowane, które ulegają podziałowi. W wyniku podziału powstają dwie komórki, z których jedna jest nieznacznie większa od drugiej. Załóżmy, że alga ułożona jest poziomo. Rozmnażaniem się komórek tego organizmu rządzą proste zasady: lewa komórka potomna lewej komórki potomnej jest mniejsza oraz prawa komórka potomna prawej komórki potomnej również jest mniejsza. Zaczynając od mniejszej, lewej komórki potomnej schematycznie wygląda to tak:



Rys. 11. Schemat mnożenia się komórki algi

Załóżmy, że chcielibyśmy zapisać zasady wzrostu algi w systematyczny sposób. Oznaczmy sobie strzałką, odpowiednio lewą i prawą, z której strony powstała komórka potomna. Oznaczmy też małą komórkę literą A , a dużą komórkę literą B . Wtedy nasz schemat wyglądał będzie tak:

$$\begin{array}{c} \overline{A} \\ \overline{AB} \\ \overline{ABBA} \end{array}$$

Zasady wzrostu komórki można więc zapisać jako:

Stan początkowy: \bar{A}

Zasady:

$$\bar{A} \rightarrow \bar{A}\bar{B}$$

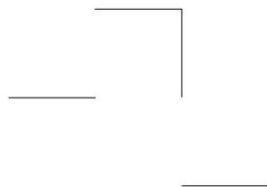
$$\bar{A} \rightarrow \bar{B}\bar{A}$$

$$\bar{B} \rightarrow \bar{A}\bar{B}$$

$$\bar{B} \rightarrow \bar{B}\bar{A}$$

Zbiór reguł oraz stan początkowy (zwany axiomem) stanowią istotę L-układów.

Chcielibyśmy teraz zająć się tworzeniem klasycznych fraktali za pomocą L-układów. Jako pierwszy weźmiemy krzywą Kocha, ale najpierw wprowadzimy pewne oznaczenia. Oznaczmy F jako odcinek o ustalonej długości, f jako odcinek o długości takiej, jak F , którego jednak nie rysujemy na naszym obrazku, \oplus jako obrót przeciwnie do ruchu wskazówek zegara o ustalony kąt α , a \oslash jako obrót zgodnie z ruchem wskazówek zegara o taki sam kąt α . Za pomocą tych znaków będziemy mogli zakodować konstrukcję wyspy Kocha jak i innych fraktali. Zobaczmy, jak to wygląda w praktyce. Rysować zaczynamy od dowolnego punktu w prawo. Załóżmy, że długość F to 1, a kąt, o który chcemy obracać, to $\alpha = 90^\circ$. Przykładowa instrukcja $F, \oplus, f, \oslash, F, \oslash, F, f, \oplus, F$ wygląda następująco (rysujemy kolejne graficzne interpretacje symboli):



Rys. 12. Rysunek pokazujący, jak graficznie zinterpretować przykładową instrukcję

Takie oznaczenia są bardzo wygodne w języku L-układów. Zapiszemy w nim klasyczną konstrukcję znanej nam już krzywej Kocha. Aby utworzyć naszą krzywą musimy wziąć parametr $\alpha = 60^\circ$ i zacząć od prostej linii (która jest teraz naszym axiomem).

Axiom: F

Zasady:

$$F \rightarrow F, \oplus, F, \oslash, \oslash, F, \oplus, F$$

$$\oplus \rightarrow \oplus$$

$$\oslash \rightarrow \oslash$$

Parametr: $\alpha = 60^\circ$

W rzeczywistości powstający obraz jest coraz większy, ale wystarczy przeskalować go i otrzymamy to, co chcieliśmy. Zapręgnijmy teraz nasze nowo odkryte narzędzie do konstrukcji prostej trawy. Potrzebne nam będzie jeszcze jedno oznaczenie na rozgałęzienia: będą to nawiasy kwadratowe. To, co będzie w nie ujęte, należeć będzie do jakiegoś odgałęzienia. Dodatkowo wprowadzimy symbol B , który będzie po prostu pomijany przy rysowaniu. Przykładowe zasady wyglądają tak:

Axiom: B

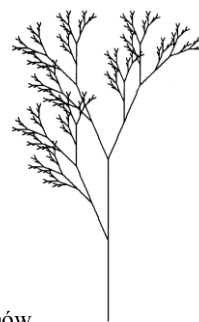
Zasady:

$$F \rightarrow FF$$

$$B \rightarrow F[\oplus B] F[\oslash B] \oplus B$$

Parametr: $\alpha = 20^\circ$

Stosując je otrzymamy:



Rys. 13. Takie i wiele podobnych roślin można łatwo stworzyć za pomocą L-systemów

Opisane przykłady to tylko bardzo proste przypadki omawianych zagadnień. Bardziej zaawansowane L-układy mogą uwzględniać w zasadach np. sąsiedztwo danego symbolu i reagować na nie. Takie L-układy, zwane parametrycznymi, służą do tworzenia zadziwiająco realistycznych modeli biologicznych oraz przepięknych krajobrazów, często wykorzystywanych w filmach *science fiction*.

4. Podsumowanie

Maszynki do fraktali to bardzo proste konstrukcje mające zadziwiające właściwości. Zestaw prostych poleceń zarówno w MRCM jak i L-systemach może pomóc nam w uzyskaniu różnorodnych fraktali. Te z kolei możemy wykorzystać w sposób różnorodny. Od zastosowań artystycznych do naukowych.

Literatura

- [1] Peitgen Heinz-Otto, Jürgens Hartmut, Saupe Dietmar, *Granice chaosu. Fraktale, cz. I*, PWN, Warszawa 2002
- [2] Kudrewicz Jacek, *Fraktale i chaos*, WNT, Warszawa 2007.
- [3] Pierański Piotr, *Fraktale: od geometrii do sztuki*, Ośrodek Wydawnictw Naukowych, Poznań 1992.
- [4] <http://www.if.uj.edu.pl/Foton/111/index.html>