



Komputery i los (II)

Jerzy Karczmarczuk

Zakład Informatyki, Uniwersytet w Caen, Francja

Twórczość jest zdolnością do uporządkowania losowości Natury. (Eric Hoffer)

Grajcie! 100 procent tych, co wygrali – grało!
(Reklama loterii we Francji)

1. Wstęp

Druga i zarazem ostatnia część artykułu porusza pobieżnie „losowanie” innych wielkości niż proste liczby, zajmuje się funkcjami losowymi (szumem) i niektórymi jego zastosowaniami, oraz metodami Monte-Carlo: technikami akumulacji informacji statystycznej otrzymanej dzięki symulacji przy użyciu liczb losowych. Te metody zostały po raz pierwszy rozpracowane przez Johna von Neumanna, Stanisława Ulama i Nicholasa Metropolisa podczas prac nad projektem Manhattan.

Z konieczności ten artykuł ograniczy się do dwóch części, mimo iż materiał zasługuje na grubą książkę. Sama teoria oraz praktyka „szumów”, czy procesów stochastycznych jest bardzo bogata i pełna wyników nie zawsze właściwie interpretowanych. W pierwszej części artykułu wspomnieliśmy – na przykładzie autobusów – o procesie Poissona, ciągu *niezależnych* zdarzeń, które mogą być również realizowane jako emisja i zliczanie indywidualnych fotonów przez osłabiony laser. Wyobraźmy sobie jednak słabe cieplne źródło światła, np. oddaloną żarówkę albo gwiazdę. Tutaj „statystyczność” przejawia się w dwóch kontekstach. Oczywiście dyskretna natura światła pozwoli nam – w sprzyjających okolicznościach – mierzyć pojedyncze fotony. Strumień fotonów jest dyskretnym procesem losowym. Ale wcześniej, klasycznie, natężenie fali elektromagnetycznej emitowanej przez źródło także jest obiektem losowym, fluktuującym w czasie i przestrzeni. Te dwa źródła losowości się nakładają, to, że jedno wynika z termiczności, a drugie z zasad fizyki kwantowej, może być źródłem nieporozumień.

W przedziale czasowym, w którym to natężenie jest większe, gęstość fotonów będzie większa, a gdy amplituda osłabnie, średnio fotonów będzie mniej. Pomiar zasugeruje, że fotony są „emitowane paczkami”, że są skorelowane ze sobą. Łatwo więc o złudzenie, że mechanizm emisji „lubi” produkować pakiety, a nie pojedyncze fotony. Nie ma to wiele wspólnego z rzeczywistością. Opisane zjawisko, charakteryzujące częściowo koherentne źródła światła, zwane efektem Hanbury Browna i Twissa znalazło zastosowania w astronomii [3].

Nauka wynikająca z tego przykładu jest następująca. Rozumowanie statystyczne, jeśli nie zna się mechanizmu jakiegoś zjawiska, może być prostą drogą na manowce. Kłopoty wynikną również w przypadku, gdy nie zna się próbki, z której szacuje się rozkład prawdopodobieństwa użyty następnie do przewidywań. W Stanach Zjednoczonych do tej pory pamięta się klapę prognozy wyborów prezydenckich 1936 r., dokonanej przez szacowne czasopismo *The Literary Digest*. Posłużyli się olbrzymią próbką, ponad dwumilionową (potworne koszty, nikt tego teraz nie robi...), i stwierdzili, że wygra Alf London, gubernator stanu Kansas, Republikanin, który następnie poniósł sromotną klęskę w starciu z Rooseveltem. Zostały popełnione dwa błędy, duży i mały.

Mały błąd polegał na ekstrapolacji *wyników* ankiety, a więc tej części ankietowanych, która odpowiedziała, na całą ankietowaną próbkę, ponad 10-milionową. Otóż w badaniach socjologicznych i innych dotyczących ludzi, trzeba się zastanowić, czy brak odpowiedzi nie oznacza już – statystycznie – pewnej orientacji, odmiennej niż ta, która charakteryzuje „posłusznych” ankietowanych.

Duży, a nawet olbrzymi błąd polegał na wzięciu próbki w oparciu o listy abonentów czasopisma, listy telefonów i danych z rejestracji samochodów. Jednym słowem, wybrano bezmyślnie pewną klasę społeczną, ludzi bogatszych, o określonym profilu (edukacja, kultura). Właściciele samochodów dzisiaj stanowią w USA próbkę dość zrównoważoną, ale nie w 1936 r., w okresie Wielkiej Depresji ekonomicznej! Ci bogatsi częściej głosowali na Republikanów, ale byli w mniejszości...

Czy fizykom mogą zdarzyć się podobne błędy? Owszem, i wiele innych. W fizyce wysokich energii w latach sześćdziesiątych obserwowano, że w zderzeniach produkujących dużo cząstek wtórnych (głównie mezonów) rozkład pędów podłużnych (wzdłuż osi zderzenia) sugeruje, że tworzone są dwa „bąble” zwane „fireballami” lecące jak cząstki pierwotne, i dopiero ich rozpad produkuje mezony. Widziano dużo lecących w przód, dużo w tył, a cząstek o małych pędach podłużnych było niewiele.

A zjawisko wzięło się z trudności z kalibracją rozkładu... Złe szacowanie położenia zera rozkładu pędu powodowało, że cząstki były „przesuwane” bardziej w przód, albo bardziej w tył, i próbka pozostawiała okolice zera niewypełnione. Powstał cały szereg modeli tłumaczących to fikcyjne zjawisko.

2. Co jeszcze możemy losować?

W pierwszej części artykułu omówiliśmy podstawowe metody generowania liczb losowych i wspomnieliśmy o wektorach losowych. Ale obiekty o charakterze losowym nie muszą sprowadzać się do prostych wartości numerycznych. Dość specyficznym, ale bardzo ważnym zagadnieniem, jest generowanie *konfiguracji*, np. losowy wybór m spośród n obiektów, z uwzględnieniem kolejności albo nie, z powtórkami, albo bez. Czasami musi się wprowadzać wagi, które

preferują niektóre rozkłady, a zakazują innych, jest to typowe np. w organizowaniu turniejów. Przy symulowaniu ruchu wielu animowanych postaci, ich kolejność winna być losowa, aby zachować pewien realizm, ale częściowo skorelowana: efekt tłumu bierze się częściowo z tego, że ludzie imitują swoich sąsiadów. Twórcy takich filmów jak „Titanic”, „Gladiator”, czy „Władca Pierścieni” korzystali z usług specjalistów od technik stochastycznych i jest dość prawdopodobne, że nie byli to „czyści informatycy”, lecz osoby mające niezłe pojęcie o własnościach systemów fizycznych...

2.1. Permutacje losowe

Jak wylosować kolejność elementów w ciągu o skończonej długości n , np. ciąg (3, 1, 8, 9, 6, 2, 10, 7, 5, 4) liczb od 1 do $n = 10$? Wspomnimy tutaj o dwóch algorytmach, aby pokazać, że w tej dziedzinie zbliżone wyniki można osiągnąć różnymi metodami.

Pierwsza metoda polega na wylosowaniu n liczb (standardowych, między 0 a 1) i użycie ich jako „kluczy” do wyboru liczb od 1 do n , albo do czegokolwiek innego. Praktycznie tworzymy tablicę par:

1	2	3	4	5	6	7	8	9	10
r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}

gdzie r_k jest standardową liczbą losową. Pierwszy rząd tablicy może zawierać inne, dowolne obiekty, których kolejność ma być przestawiona, lub – tak jak tutaj – liczby, które mogą indeksować te elementy.

Następnie sortujemy tę tablicę według wartości w drugim rzędzie, obiekty w pierwszym rzędzie zachowują tych samych partnerów z drugiego. Ponieważ kolejność liczb w drugim rzędzie jest „bylejaka”, po posortowaniu, kolejność w pierwszym rzędzie stanie się losowa. Jeśli wygeneruje się w ten sposób *bardzo dużo* permutacji, pewne rozumowanie pozwoli stwierdzić, że ta metoda *nie jest* idealnym próbkowaniem *wszystkich możliwych* permutacji, ale w praktyce nie jest to problemem. Technik sortowania nie będziemy omawiać, są one liczne, łatwe i dobrze udokumentowane.

Druga metoda polega na iterowaniu przestawień. Zaczynamy od permutacji trywialnej, tablica A zawiera elementy A_k w porządku „kanonicznym” (np. liczby 1, 2, ... n). Następnie iterujemy:

- Dla liczb i od $(n - 1)$ do 1, wykonać:
- j = całkowita liczba losowa między 0 a i ;
- następnie wymienić miejscami A_i i A_j .

2.2. Próbkowanie losowe

Wybór m spośród n elementów jest jedną z podstawowych technik statystycznych. Jest to użyteczne zarówno w *pollingu* społecznym jak i w testowaniu kombinacji składników jakiegoś lekarstwa w odniesieniu do konkretnego wroga bakteryjnego. W fizyce wybór kilku z możliwych zmian stanu, albo kilku możliwych stanów jest również jedną z podstawowych technik symulacji.

Pojęciowo algorytm jest prosty: wybieramy losowo jeden element spośród n , następnie drugi z próbki liczącej $n - 1$ itd., aż uzbiera się m . Ale to nie jest takie proste, jeśli obiekty są umieszczone w tablicy, po wybraniu jednego trzeba przesunąć inne, co zabiera sporo czasu. Okazuje się, że taki prosty problem wymaga różnych podejść w zależności od szczegółów.

Można po prostu wybierać losowo z n , ale potem trzeba sprawdzić, czy nie było powtórzeń, co także może drogo kosztować, zwłaszcza jeśli m jest porównywalne z n . Wtedy złożoność czasowa może się okazać większa niż n , bo pojawiają się wielokrotne powtórzenia.

Oto algorytm o złożoności m , dostarczający indeksów tablicy, w której mieszczą się wszystkie próbkowane elementy. Wymaga on dodatkowej pamięci, tablicy T o n elementach, której wartości początkowe są $0, 1, 2, \dots, n - 1$, tak więc $T_k = k$. Wyniki zostaną przechowane w tablicy S . Wartość $n - 1$ zostaje dodatkowo przechowana w zmiennej l .

Wykonać m przebiegów poniższej pętli, dla zmiennej k od 0 do $m - 1$:

- a. Wybrać całkowitą liczbę losową z od 0 do l .
- b. Zapisać w tablicy wynikowej zawartość odpowiedniego elementu T , czyli $S_k = T_z$.
- c. Przepisać $T_z = T_l$.
- d. Zmniejszyć l o 1 .

Dowód poprawności tego algorytmu jest bardzo dobrym ćwiczeniem, ale nie dla wszystkich.

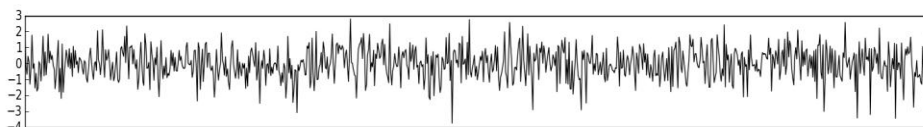
3. Funkcje losowe

Funkcje losowe, które będziemy nazywali także szumami, są to funkcje, których wartości mają charakter losowy: oglądając większą dyskretną próbkę takiej funkcji stwierdzamy, że przechodzi ona szereg testów statystycznych. Takie funkcje „wyglądają losowo”. Przebiegi losowe są wszędzie, napięcie elektryczne w funkcji czasu, prędkość płynu w przewodzie, ceny akcji na giełdzie itp.

Prosty wariant jednowymiarowej funkcji losowej przedstawia rys. 1. Taką „funkcję” (dyskretną), dla której kolejne wartości są dowolne i nieskorelowane ze sobą nazywa się często (choć niepoprawnie) „białym szumem”.

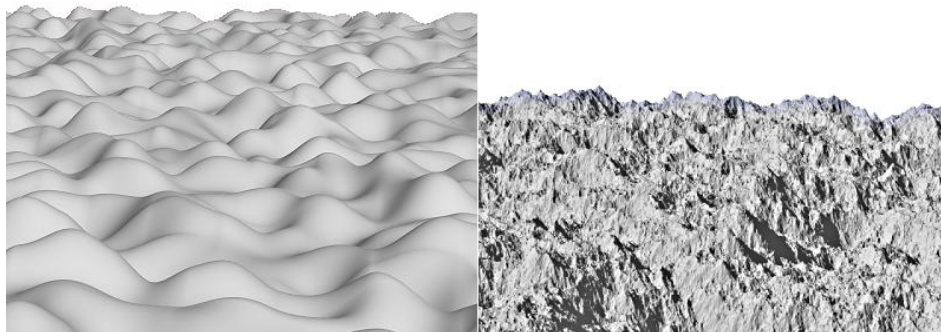
Wiemy już, że powyższa „definicja” jest niewiele warta, co to znaczy „wartości są dowolne”? Jaki jest ich rozkład? Zwykle – w fizyce – przyjmuje się rozkład Gaussowski, taki jest typowy rozkład prawdopodobieństwa szumów fizycznych. Dlaczego? Ponieważ szумы biorą się zwykle z fluktuacji jakiejś

wielkości wokół średniej; ten temat już poruszaliśmy. Rozkład normalny dominuje w przyrodzie i technice.



Rys. 1. Biały szum

„Prawdziwy” biały szum to funkcja tak dalece osobliwa, że nie da się jej narysować, jest nieskończenie gęsta. Jednak w informatyce wszystko jest dyskretne i skończone... Tak więc zwykle przyjmuje się, że jeśli charakterystyczny odstęp czasowy między próbkami wynosi Δt , wariancja rozkładu Gaussa charakteryzująca punktowo ten proces stochastyczny ma postać $\sigma^2 = c/\Delta t$, gdzie c jest pewną stałą. W granicy ciągłej ta wartość rozbiega się do nieskończoności, wykres białego szumu składałby się z linii nie tylko nieskończenie bliskich, ale nieskończenie wysokich. Technika informatyka to mało obchodzi, zażąda on podania Δt , i dobierze odpowiednio dyspersję w przybliżeniu dyskretnym. Fizyk, który pisze program symulacyjny winien wiedzieć, jakie są zależności między wielkościami, które mają osobliwe granice.



Rys. 2. Próbkę powierzchni losowych

Ale funkcje losowe w jednym lub większej liczbie wymiarów mogą być znacznie regularniejsze, gładkie, albo chropowate, jak widać na rys. 2.

Zastosowania funkcji losowych są bardzo szerokie. W grafice i syntezie obrazów mamy losowe wzory geometryczne, tekstury i rozkłady kolorów. Szумы w akustyce są niezbędne do generowania dźwięków naturalnych, w tym mowy i muzyki. Obiekty animowane udające rzeczywiste muszą się zachowywać trochę losowo, ich trajektorie, orientacje itp. są generowane przy użyciu funkcji losowych.

W jaki sposób je generować? Można mieć wrażenie, że jest to trudne i wymaga znacznej liczby parametrów. Metody dostosowane do specyficznych kategorii problemów są skomplikowane. Tutaj omówimy kilka metod prostych do zrozumienia i implementacji.

3.1. Całkowanie białego szumu

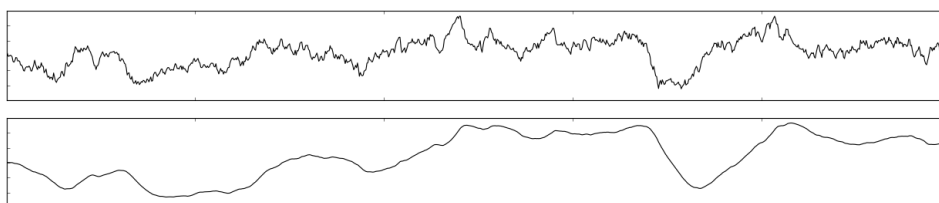
Wszyscy słyszeli o ruchach Browna. Pyłeczek zanurzony w cieczy jest bombardowany przez niezliczone molekule otaczające i porusza się chaotycznie. Mikroskopowo to zjawisko jest zbyt szybkie, aby je obserwować. W pewnym przybliżeniu można powiedzieć, że prędkość pyłku jest określona funkcją podobną do białego szumu. Obserwator mierzy jednak próbkowane położenie pyłku – sondy, co wymaga scałkowania tej osobliwej prędkości.

Oto solidnie uproszczony algorytm całkowania funkcji losowej, zadanej jako tablica ζ_n . Wynikiem jest ciąg w_n : $w_n = w_{n-1} + \zeta_n$. Otrzymaną w ten sposób funkcję nazywa się czasami procesem Wienera. W praktyce w pewnych okolicznościach ten algorytm ma pewną wadę. Nawet jeśli ζ_n jest funkcją stacjonarną, oscylującą wokół zera, więc wynik całkowania powinien też być stacjonarny, bez większych wychyleń dodatnich lub ujemnych, długofalowe duże fluktuacje występują. Wyobraźmy sobie jednak, że chcemy generować scałkowany szum akustyczny, niezbędny prawie wszędzie, np. w symulowaniu dźwięku fletni pana, która bardzo „szumi”.

Otóż bezwzględna wartość amplitudy dźwięku nie ma żadnego znaczenia, słyszy się jedynie fluktuacje wokół średniej, w określonym przedziale częstości (akustycznych). Programy generujące dźwięki korzystają więc czasami z algorytmu „całkowania ze stratami”:

$$w_n = a \cdot w_{n-1} + b \cdot \zeta_n,$$

gdzie a jest bliskie jedności, np. 0,98, a $b = (1 + a)/2$. Oto wynik scałkowania w ten sposób białego szumu gaussowskiego, oraz wynik powtórnego scałkowania poprzedniego wyniku.



Rys. 3. Całkowanie szumu

Jak widać, te funkcje są znacznie bardziej regularne i skorelowane niż biały szum. Istnieje wiele wariantów całkowania, np. w fizyce czasami korzysta się z równania Langevina, którego zapis funkcyjny ma postać:

$$\frac{d}{dt}v(t) = -\alpha v(t) + \beta \zeta(t).$$

Funkcja $\zeta(t)$ oznacza, jak poprzednio, biały szum. Pierwszy człon prawej strony „ściąga” wartość funkcji v do zera. Dyskretna postać tego równania dostarczy algorytmu „całkowania ze stratami”.

3.2. Metody spektralne

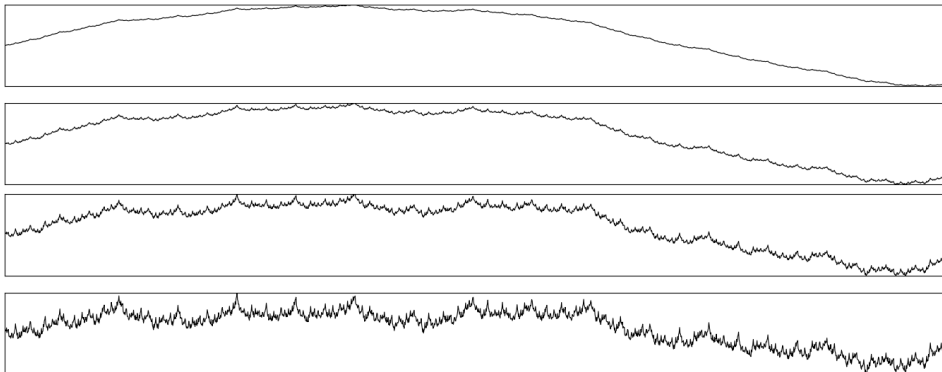
Nielokalny charakter powyższej metody jest niewygodny w implementacji. W praktyce, np. w syntezie obrazów, używa się często tzw. metody spektralnej, polegającej na dodaniu do siebie wielu wkładów funkcji regularnych, ale słabo skorelowanych, np. sinusoid z chaotycznie rozrzuconymi fazami i bardzo różnymi okresami. (Amplitudy także mogą być losowe, ale to nie jest istotne.)

Wzór na generowaną funkcję będzie miał postać

$$w(t) = \sum_{k=0}^N A_k \sin(\omega_k \cdot t + \phi_k),$$

gdzie losowe wartości ϕ_k , jednorodne między 0 a 2π , są podstawowymi parametrami losowymi algorytmu. Wartość N wcale nie musi być duża, zależy to od częstości ω_k w funkcji k . Często stosujemy zależność $\omega_k = \omega_0 2^k$, każdy nowy wkład podwaja częstość. Jeśli taki szum ma być użyty do grafiki komputerowej, nie ma sensu użyć tak wysokich częstości, że okres funkcji staje się mniejszy od jednego piksela. Więc, jeśli okres podstawowy jest porównywalny z wielkością ekranu, dla typowych ekranów $N = 12$ jest wystarczające.

Zależność amplitudy składnika od k jest dość zasadnicza. Jeśli maleje ona bardzo szybko, dominują niskie częstości, funkcja jest dość regularna. Jeśli wysokie częstości mają istotne wagi, szum staje się poszarpany. Bardzo często stosuje się zależność $A_k = k^{-\kappa}$, gdzie wykładnik κ mieści się typowo między 0 a, powiedzmy, 3. Wykresy poniżej przedstawiają próbki dla κ równych 2,5; 2,0; 1,5 i 1,0.



Rys. 4. Synteza spektralna szumów

3.3. Iteracyjne dzielenie przedziału

Ostatnią metodą poruszoną tutaj jest iteracyjne, albo rekurencyjne, podzielenie rozważanego przedziału na fragmenty i na losowym przesunięciu końców podprzedziałów. W żargonie używanym w grafice komputerowej, zwłaszcza w tworzeniu funkcji losowych dwuwymiarowych, technika ta nosi (niezbyt szczęśliwie) nazwę „plazmy”, co bierze się z faktu, że algorytm przypomina nieco iteracyjną technikę rozwiązywania równań pola dla przewodzącego ośrodka. W dziedzinie syntezy obrazów jest to ulubiona metoda generowania powierzchni terenów górskich.

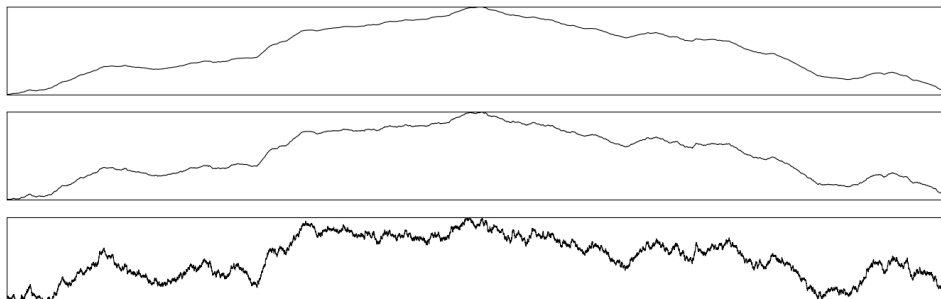
Algorytm w jednym wymiarze jest bardzo prosty. Bierze się dwa punkty – końce wykresu. Szum o „głębokości” zero jest po prostu linią łączącą te punkty. Aby otrzymać głębokość 1 dzielimy przedział na dwie połowy i środkowy punkt przesuwamy (np. dla prostoty tylko pionowo) o losową wartość, wybraną z jakiegoś rozkładu, jednorodnego, czy lepiej, Gaussowskiego. Wartość średnia losowanej zmiennej losowej wynosi zero, wtedy otrzymany szum winien oscylować wokół linii łączącej końce przedziału. Wariancja zależy od długości przedziału, powinna się zmniejszać wraz ze zmniejszaniem tej długości. Następnie stosujemy rekursję. Dwie połówki zostają znowu podzielone, a środki przesunięte. Wariancja losowych przesunięć powinna być mniejsza niż poprzednio. To jest szum o głębokości 2. Te podziały stosujemy tak długo, jak chcemy, w granicach rozsądku. Podobnie jak w syntezie spektralnej, jeśli celem szumu jest wizualizacja, nie warto schodzić poniżej długości jednego piksela. Głębokość rzędu kilkunastu jest wystarczająca.



Rys. 5. Chmury fraktalne, metoda „plazmy”

Aby kształt szumu, czy kształt jego funkcji autokorelacji, odpowiadał „typowym” funkcjom losowym opisującym obiekty naturalne, zwykle przyjmuje się dla głębokości p rozrzut $\sigma = \sigma_0 \cdot 2^{-\kappa p}$, gdzie κ jest stałą, tak jak poprzednio między 0 a 3 (powyżej wartości 2 widać właściwie tylko pierwszy podział, natomiast 0 generuje biały szum). Rysunek 6 obrazuje szum dla κ równych 1,0, 0,85 i 0,5.

Algorytm w dwóch wymiarach jest podobny, tylko bardziej skomplikowany. Przesuwa się środki kwadratów, tworzą one romby. Następnie dzieli się romby i przesuwa ich środki. Po tym etapie siatka zostaje zmniejszona dwa razy, mamy 4 razy więcej kwadratów i można procedurę posunąć dalej. W ten sposób możemy generować chmury lub szorstkie powierzchnie. Rysunek 5 przedstawia dwuwymiarowy szum wygenerowany tą metodą.



Rys. 6. Szum generowany przez podział odcinka

4. Metody Monte-Carlo

Przejdźmy do typowych – interesujących fizyków – zastosowań liczb losowych. Standardowe, podstawowe wykorzystanie liczb losowych, to próbkowanie statystyczne, zbieranie danych statystycznych, obliczanie średnich, dyspersji itp. W zasadzie można stwierdzić, że to wszystko sprowadza się do całkowania numerycznego (dyskretnego).

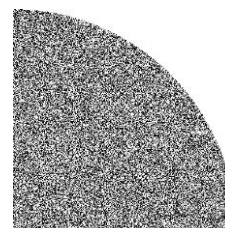
4.1. Całkowanie

Do obliczania pól figur geometrycznych można użyć metody „orzeł-reszka”. Poniższy przykład jest oczywiście akademicki. Losujemy w kwadracie jednostkowym pary liczb losowych – punkty (x,y) , i obliczamy liczbę punktów, dla których $\sqrt{x^2 + y^2} < 1$. Wynik dostarczy nam przybliżenia $\pi/4$.

Oczywiście wynik jest bardzo niedobrym przybliżeniem. Błąd względny przybliżenia zachowuje się jak $1/\sqrt{N}$, gdzie N jest liczbą losowań. Metody deterministyczne są tutaj znacznie precyzyjniejsze. Jednak obraz zmienia się zupełnie, jeśli całkowanie jest dokonywane w przestrzeni o dużej liczbie wymiarów. Zwróćmy uwagę, że nawet prosta molekula ma przynajmniej pięć stopni swobody: translacyjne i rotacyjne. Dla skomplikowanych cząsteczek wymiar przestrzeni parametrów jest znaczny. Okazuje się wtedy, że metody deterministyczne dzielenia przestrzeni na komórki dają zbieżność całkowania numerycznego już nie tak dobrą. W kilkunastu, lub więcej wymiarach metody losowe wygrywają bezapelacyjnie.

Metoda „orzeł-reszka” nie jest najbardziej klasyczną techniką całkowania. Zwykle używa się metody średniej ważonej:

$$\int_a^b f(x)dx \cong \frac{b-a}{N} \sum_{i=1}^N f(x_i),$$



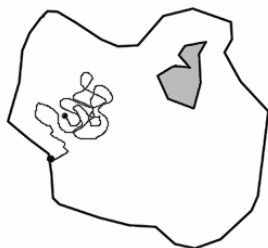
Rys. 7. Metoda „orzeł-reszka”

Gdzie x_i są liczbami losowymi z przedziału $[a, b]$. Dokładność tej metody jest nieco większa niż metody „orzeł-reszka” (np. dla $f(x) = \sqrt{1-x^2}$ między 0 a 1), ale zbieżność pozostaje proporcjonalna do $1/\sqrt{N}$. Im bardziej funkcja f jest fluktuująca, szybkozmienna, tym błąd jest większy. Lepsze (dokładniejsze) wyniki uzyska się, jeśli rozkład użytej zmiennej losowej nie jest jednorodny, ale przypomina zachowanie f (a właściwie jej wartości bezwzględnej).

4.2. Rozwiązywanie równań różniczkowych cząstkowych

To jest także całkowanie... Rozważmy bardzo ważne dla fizyków równanie różniczkowe Laplace'a w dwóch wymiarach:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) f(x, y) = 0.$$



Rys. 8. Równanie Laplace'a i ruch Browna

Równanie to opisuje np. rozkład wartości jakiegoś pola fizycznego w przestrzeni. (Może to być pole elektromagnetyczne, pole prędkości płynu, itp.) Obszar, w którym występuje to pole może mieć skomplikowaną geometrię. Znane są warunki brzegowe, tj. wartości pola na brzegach obszaru.

Metoda stochastyczna obliczania wartości pola w jakimś punkcie polega na losowaniu trajektorii, „ruchu Brownowskiego” teoretycznej cząstki, wychodząc od analizowanego punktu. Wybieramy losowo kierunek: góra, dół, lewo lub prawo i powtarzamy tę operację, aż „cząstka” dotrze do brzegu obszaru, gdzie wartość funkcji jest znana. Sumujemy wszystkie wartości. Ta średnia, po podzieleniu przez liczbę trajektorii, da nam oszacowanie wartości funkcji. Zauważmy, że szansa natrafienia na bliski brzeg jest większa niż na daleki. Wynik, jak zawsze, będzie obciążony błędem statystycznym, ale metoda jest użyteczna w praktyce.

5. Wybrane metody Monte-Carlo w fizyce

Jak wiemy, w stacjonarnej fizyce statystycznej prawdopodobieństwo jakiegoś stanu $\mathbf{S}(\dots)$ opisanego szeregiem parametrów mikroskopowych: położeń, orientacji itp. składników, zwykle wyraża się wzorem na rozkład kanoniczny Gibbsa

$$p(\mathbf{S}) = \mathcal{N} e^{-\beta E(\mathbf{S})},$$

gdzie \mathcal{N} jest stałą normującą, a E – energią stanu \mathbf{S} . Czynnikiem $\beta = 1/(kT)$, gdzie T jest temperaturą, a k – stałą Boltzmanna. Symulacja systemu powinna dostarczyć próbek zgodnych z tym prawdopodobieństwem. Mierzac zależności mię-

dzy symulowaną temperaturą a innymi wielkościami, np. średnią energią systemu.

Zależność energii od stanu może być bardzo skomplikowana i praktyczne obliczenia własności systemu za pomocą metod analitycznych jest rzadko możliwe. Dysponujemy wieloma metodami symulacyjnymi, z których ciągle dość popularną jest metoda Metropolis, którą pokrótce omówimy. Nie generujemy bezpośrednio prawdopodobieństwa jak we wzorze powyżej, ale wychodząc z jakiegoś stanu początkowego pozwalamy, aby układ się „uspokoił”, doszedł do swojego stanu równowagi, a raczej do zespołu takich stanów. Będą to stany minimalizujące energię.

Mierząc „doświadczalnie” energię systemu w stanie bliskim równowagi, możemy obliczyć takie własności termodynamiczne jak ciepło właściwe i otrzymać liczbowe wyniki dotyczące występowania przejść fazowych. Oto zarys metody:

1. Wybieramy jakiś stan początkowy \mathbf{S} i obliczamy energię systemu.
2. Iterujemy pewną liczbę razy, aż stwierdzimy, że układ się ustabilizował, następujące operacje:
 - a. Losujemy nowy *możliwy* stan, \mathbf{S}' . Zmieniamy wartości energii jednego lub kilku składników, położenia, orientacje cząsteczek itp., w zależności od szczegółów. Obliczamy nową energię $E' = E(\mathbf{S}')$. Zwykle, jeśli zmiany są lokalne, nie potrzeba przeliczyć całej energii, tylko oszacować jej zmianę, ΔE .
 - b. Jeśli ta zmiana jest ujemna, kierunek zmiany stanu jest korzystny, układ zbliża się do równowagi. Akceptujemy nowy stan. W przeciwnym wypadku energia się zwiększa, oddalamy się od równowagi, więc jest to niekorzystna zmiana stanu.
 - c. Jednak, aby uniknąć zablokowania systemu w stanie lokalnej pseudo-równowagi dalekiej od rzeczywistego minimum globalnego, pozwolimy systemowi się „ruszyć”:
 - i. Obliczamy $p = e^{-\Delta E/kT}$.
 - ii. Losujemy standardową liczbę z przedziału 0 – 1 i porównujemy z p . Jeśli $r < p$, nowy stan zostaje zaakceptowany. Jeśli nie, stan \mathbf{S}' zostaje odrzucony i wychodząc z \mathbf{S} generujemy nowy stan \mathbf{S}' .
 - d. Mierzmy wielkości, które nas interesują.
3. Zmieniamy temperaturę i kontynuujemy.

Zauważmy, że jeśli temperatura jest wysoka, wartość p pozostaje bliska jedności nawet dla dość sporych ΔE . System będzie fluktuował losowo, jego „energia termiczna” będzie dominowała nad energią „potencjalną” związaną ze stanem mikroskopowym. W niskich temperaturach niekorzystne zmiany stanu będą silnie hamowane. W praktyce, jeśli interesuje nas jakaś niska temperatura, zaczynamy od wysokiej i powoli ją obniżamy, bez reinicjalizacji parametrów

systemu Mamy w ten sposób nadzieję, że osiągniemy stany bliskie równowagi globalnej.

5.1. Model Isinga (2 wymiary)

Często używanym modelem, interesującym dydaktycznie, gdyż obliczalnym także analitycznie, jest model dwuwymiarowego „kryształu ferromagnetycznego” zawierającego spiny $S_{(k,l)}$ o dwóch możliwych orientacjach, powiedzmy „+1” i „-1”. Energia pary sąsiadów $a = (k, l)$ i np. $b = (k + 1, l)$ wynosi $E_{ab} = -cS_a \cdot S_b$, gdzie c jest dodatnią stałą. Jeśli dwa sąsiednie spiny są zorientowane identycznie, energia jest mniejsza, jeśli przeciwnie – większa. Całkowita energia systemu jest sumą

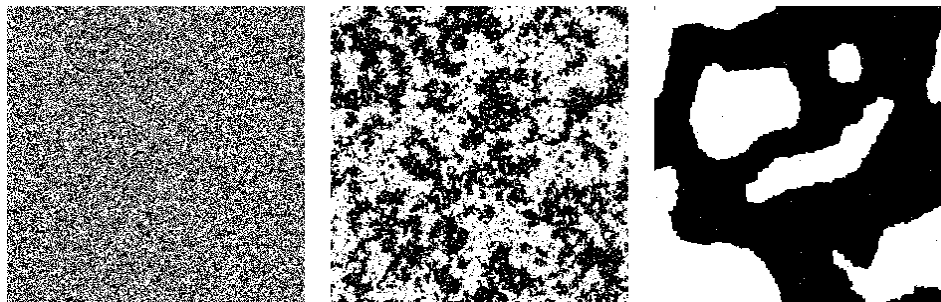
$$E(S) = -c \sum_{\text{sąsiedzi}(a,b)} S_a \cdot S_b.$$

Spin „do góry” może być wizualizowany jednym kolorem, np. czarnym, a „w dół” – białym. Oprócz energii możemy mierzyć również magnetyzację, średni spin systemu, a jeśli interesuje nas dynamika w obecności zewnętrznego pola magnetycznego, możemy dodać do energii człon $E_H(S) = H \sum_a S_a$. Zajmiemy się jednak doświadczeniem bez zewnętrznego pola. Takie stałe jak c , czy stała Boltzmann nie mają żadnego znaczenia, gdyż jednostki energii czy temperatury są umowne. Rysunek 9 obrazuje konfigurację systemu Isinga, dla temperatur wysokich, średnich: bliskich tzw. temperatury krytycznej, i niskich, po długim czasie symulacji.

Każdy spin ma czterech sąsiadów (zaniedbujemy sąsiadów diagonalnych; możemy ich uwzględnić, a także dalszych sąsiadów, ale tutaj podajemy najprostszy możliwy przykład). Lokalna zmiana stanu systemu polega na przekręceniu jednego spinu. W zależności od sąsiadów, zmiana energii wyniesie: +8, (z -4 do +4) jeśli wszystkie sąsiednie węzły miały tę samą orientację co centralny, +4, jeśli jeden miał inną, (z -2 do +2) i 0 jeśli dwa miały inną. Pozostałe przypadki dają ΔE_a równe -4 lub -8. Możemy stabilizować odpowiednie wartości funkcji wykładniczej, aby przyspieszyć obliczenia. Możemy także użyć wartości -2, -1, 0, 1, 2, sprowadzi się to do przedefiniowania wartości stałej c (albo temperatury).

To jest w zasadzie wszystko, pozostaje zaprogramować i puścić w ruch symulację. Jednak odpowiedź na pytanie: które, i w jakim porządku przekręcimy spiny, aby otrzymać nowy stan, nie jest prosta. Jeśli dokonamy tego sekwencyjnie, spin po spinie, linia po linii, kolumna po kolumnie, zmiana energii będzie zależeć od tego co stało się przed chwilą z sąsiadem, taki sekwencyjny algorytm jest nienaturalny, niefizyczny. Postąpimy inaczej, zajmiemy się równoległą (w symulacji) zmianą orientacji wszystkich spinów. Założymy okresowe (toroidalne) warunki brzegowe, tj. na prawo od ostatniej kolumny znajdzie się pierw-

sza, itd. Algorytm podany poniżej także jest nieco niefizyczny, ale realistyczny i szybki.



Rys. 9. Model Isinga, zachowanie w zależności od temperatury

5.2. Program i wyniki

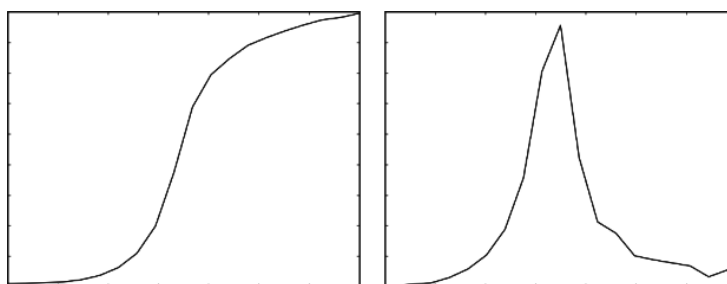
Naszym celem nie jest podawanie technik programowania, odstawimy jednak od tego, gdyż chcemy pokazać jak współczesny fizyk powinien programować zbliżone problemy, przy użyciu uniwersalnych języków i pakietów programowania i jak niewiarygodnie krótkie programy można tworzyć, jeśli ma się odrobinę kompetencji. Dość dawno temu proponowaliśmy na łamach *Fotonu* użycie pakietu Scilab, pozwalającego na tzw. programowanie wektorowe: całe tablice liczbowe, wielowymiarowe, mogą być w jednej instrukcji dodane do siebie, albo poddane innym operacjom matematycznym. Unika się pisania jawnych pętli iteracyjnych, co przyspiesza program (pętle są wykonywane przez wielokrotnie szybszy kod w skompilowanej bibliotece), a także unika się błędów.

Tym razem proponujemy użycie języka Python i pakietów **numpy** oraz **Matplotlib**, pozwalającymi razem na przetwarzanie tablic i ich wizualizację. Po załadowaniu tych pakietów, (ewentualnie innych jeśli trzeba) utworzenie tablicy liczb losowych $N \times N$ o wartościach ± 1 sprowadzi się do jednej instrukcji: **a = (2*rand(N,N)<1)-1**. Niech **beta** oznacza aktualną wartość $1/T$. W jaki sposób obliczyć energię systemu? Pakiet zawiera funkcję **roll**, pozwalającą przesunąć cyklicznie w lewo – prawo / górę – dół linie lub kolumny tablicy.

Poniższy program dokonuje modyfikacji tablicy **a**:

```
d=(roll(a,1,axis=0)+roll(a,-1,axis=0)+
    roll(a,1,axis=1)+roll(a,-1,axis=1))
en=a*d                                # Aktualna energia
z=(2*rand(N,N)<1)-1                    # Prop. nowego stanu; losowa.
p=exp(beta*(d*z-en))                   # Prawdopodobieństwo zmiany
rr=rand(N,N)<p
a=z*rr + a*(1-rr)
```

To wszystko, reszta to pętle zmieniające temperaturę, oraz iterujące kilkadziesiąt razy powyższe instrukcje dla każdej temperatury, aby układ się ustabilizował. Rysunek 10 przedstawia zależność sumarycznej energii systemu w funkcji temperatury oraz jej pochodną, tj. ciepło właściwe. Temperatura zmienia się od 0,5 do 4. Widać wyraźnie efekt progowy, w okolicy $T = 2$, nie zaznaczyliśmy jednostek, gdyż są one czysto umowne.



Rys. 10. Energia i ciepło właściwe w funkcji temperatury dla symulowanego modelu Isinga

Zakończenie

Metody Monte-Carlo zbliżone do metody Metropolis są także używane w technikach obliczeniowych dalekich od fizyki, gdyż stanowią one przykład technik *optymalizacji stochastycznej*, interesującej np. ekonomistów, inżynierów projektujących linie produkcyjne itp.

Literatura

- [1] R. Zieliński, (1970), *Metody Monte Carlo*, WNT, Warszawa.
- [2] N. Metropolis (1987). *The beginning of the Monte Carlo method*. Los Alamos Science (Specjalne wydanie zadedykowane Stanisławowi Ulamowi): 125–130.
- [3] R. Hanbury Brown, R.Q. Twiss (1958). *Interferometry of the intensity fluctuations in light. II. An experimental test of the theory for partially coherent light*. *Proc. of the Royal Society of London A* **243** (1234): 291–319.
- [4] N. Metropolis *et al* (1953). *Equations of state calculations by fast computational machine*. *Journal of Chemical Physics*, 21:1087–1091.