

# Generatory liczb pseudolosowych i modelowanie Monte Carlo

Zadanie: Dokonać aproksymacji wielomianowej przykładowej funkcji. Jaki to ma związek z hordą zombie? Nie mam pojęcia. Jest strajk scenarzystów. Poproście ChatGPT by wam coś wymyślił.

Prawdopodobnie najprostszym rodzajem generatorów liczb pseudolosowych są generatory multiplikatywne. Ogólne wyrażenie na pozwalające obliczać ww. wartości wyraża się jako:

$$X_i = aX_{i-1} \bmod m$$

$$x_i = \frac{X_i}{m-1}$$

gdzie  $X_{i-1}$  to poprzednio zwracana wartość,  $X_0$  to ziarno metody,  $\bmod$  oznacza operator modulo - reszta z dzielenia, a  $a$  i  $m$  są stałymi liczbami całkowitymi. Drugi wzór odpowiada za normalizację liczby pseudolosowej do zakresu (0;1)

Generator ten jest daleki od ideału, np. w najlepszym przypadku wylosuje on w pseudolosowej kolejności wszystkie liczby z zakresu 1 do  $m-1$ , a następnie zaczynię się powtarzać. Również, po normalizacji dokładność naszych liczb losowych będzie ograniczona do  $1/(m-1)$ . A w przypadku źle dobranych parametrów  $a$  i  $m$  jego jakość będzie jeszcze mniejsza.

Problem z szybkim zapętlaniem generatorów można rozwiązać poprzez zastosowanie nieco bardziej skomplikowanego generatora np.:

$$X_i = (aX_{i-1} + bX_{i-2} + cX_{i-3}) \bmod (m)$$

## Zadania do wykonania:

1. Napisz trzy generatory liczb pseudolosowych:

U1 dla:  $a = 17$ ,  $m = 2^{13} - 1$ ,  $X_0 = 10$

U2 dla:  $a = 85$ ,  $m = 2^{13} - 1$ ,  $X_0 = 10$

U3 dla:  $a = 1176$ ,  $b = 1476$ ,  $c = 1176$ ,  $m = 2^{32} - 5$ ,  $X_0 = X_1 = X_2 = 10$

\* Wartości  $m$  i  $X$  powinny być zadeklarowane jako wartości całkowite. W większości przypadków powinna wystarczyć zmienna typu long int, ale w niektórych

przypadkach potrzebna jest wartość long long int. Wartości X z poprzednich iteracji można zachować definiując je jako: static long int X=10;

2. Dla każdego z tych generatorów wypisz pierwsze 2000 liczb pseudolosowych i przygotuj wykresy  $x_i = f(x_{i-1})$ ,  $x_i = f(x_{i-2})$ ,  $x_i = f(x_{i-3})$ .

3. Wykorzystaj generator U3 do wyznaczenia przybliżenia liczby Pi metodą Monte Carlo.

W tym celu, w pierwszej kolejności musimy policzyć powierzchnię koła  $S_k$ . Przyjmujemy koło o promieniu 1 z środkiem w początku układu współrzędnych. Oraz kwadrat o boku 2l i środkiem również w środku układu współrzędnych. A dokładniej pierwszą ćwiartkę z tego koła. W każdej iteracji skryptu losujemy dwie wartości z zakresu [0, 1] tworzące punkt i sumujemy wszystkie punkty, których odległość od początku układu współrzędnych jest mniejsza od r.

Jeżeli niczego nie schrzaniłmy, to stosunek punktów wewnątrz koła do wszystkich punktów powinien być równy stosunkowi powierzchni koła do powierzchni kwadratu:  $\frac{S_k}{(2l)^2} = \frac{suma}{n}$ . Teraz wystarczy tylko obliczyć wartość liczby Pi.

Całość wykonaj dla n=20 000 iteracji. Co 100 iteracji wypisz kolejne przybliżenia obliczonej liczby oraz błąd wyznaczonej liczby. Przygotuj wykres logarytmu błędu od iteracji.

No dobra... bardzo przyśpieszony sposób jak aproksymować funkcję, ale spójrzcie na wykład by wiedzieć co i dlaczego:

Jeżeli wektor  $y_i$  zawiera znane wartości aproksymowanej funkcji dla położenia zawartych w wektorze  $x_i$  i chcemy aproksymować go wielomianem rzędu  $m-1$ , naszą aproksymowaną funkcję (a) możemy zapisać w postaci macierzowej jako:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{m-1} \\ 1 & x_3 & x_3^2 & \dots & x_3^{m-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^{m-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_3 \\ \vdots \\ b_{m-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix},$$

$$\mathbf{Xb} = \mathbf{y}$$

Ponieważ jeszcze nie znamy wartości wektora  $\mathbf{b}$  a macierz  $\mathbf{X}$  nie jest kwadratowa nie możemy po prostu rozwiązać tego problemu. Ale możemy wszystko pomnożyć przez  $\mathbf{X}'$ :

$$\mathbf{X}'\mathbf{Xb} = \mathbf{X}'\mathbf{y}$$

$$\mathbf{D}\mathbf{b} = \mathbf{r}$$

$\mathbf{D} = \mathbf{X}'\mathbf{X}$  i jest macierzą  $m \times m$ , a  $\mathbf{r}$  jest  $m$  elementowym wektorem, tak więc nie ma już problemu by rozwiązać taki układ równań.

Teraz trzeba jedynie policzyć wartości aproksymowanej funkcji korzystając z pierwszego równania w tym akapicie.

### Zadania do wykonania:

1. Oblicz wartości funkcji  $f(x)$  dla  $n=11$ , równo rozłożonych węzłów w przedziale od -5 do 5.
2. Stwórz/oblicz macierze  $\mathbf{X}$ ,  $\mathbf{D}$  oraz wektor  $\mathbf{r}$ .
3. Rozwiąż układ równań liniowych by wyznaczyć parametry wielomianu aproksymacyjnego. Policz wartości funkcji aproksymacyjnej.
4. W sprawozdaniu zaprezentuj wykresy prezentujące wartości aproksymowanej funkcji na tle danych początkowych dla  $n=11$ , 51 i 101. Porównaj otrzymane parametry wielomianu z wartościami rzeczywistymi.