Laborator 8

I. Generarea de numere pseudo-aleatoare, ce urmează o distribuţie discretă dată:
Se dau $(x_1, \ldots, x_n)$ (valorile) şi $(p_1, \ldots, p_n)$ (probabilităţile lor). Să se realizeze un program în Matlab care generează N numere pseudo-aleatoare, care urmează *legea discretă*

$$X \sim \begin{pmatrix} x_1 & x_2 & \ldots & x_n \\ p_1 & p_2 & \ldots & p_n \end{pmatrix},$$

folosind numere aleatoare uniform distribuite pe [0,1].

Procedeul de generare al numerelor aleatoare $Y(i)$, $i = \overline{1, N}$, *este*:

• Se citesc valorile $x_1, x_2, \ldots, x_n$ şi probabilităţile corespunzătoare $p_1, p_2, \ldots, p_n$, precum şi numărul N.
• Pentru optimizarea procedeului, se ordonează descrescător $p_1, p_2, \ldots, p_n$. Fie $p_0 := 0$.
• Se generează N numere aleatoare uniform distribuite pe [0,1]: $U(i)$, $i = \overline{1, N}$ (în Matlab U=*unifrnd*(0,1,1,N)).
• Pentru fiecare $i = \overline{1, N}$: $Y(i) := x_k$ dacă şi numai dacă

$$p_0 + p_1 + \cdots + p_{k-1} < U(i) \leq p_0 + p_1 + \cdots + p_k, \ k \in \{1, \ldots, n\}.$$

• Se afişează: $Y(i), i = \overline{1, N}$.

Verificarea procedeului: Deoarece U urmează legea uniformă, avem pe baza procedeului de mai sus

$$P(\text{se generează } x_k) = P(p_0 + p_1 + \cdots + p_{k-1} < U \leq p_0 + p_1 + \cdots + p_k) = p_k, \ k = 1, \ldots, n,$$

deci numerele generate urmează legea de distribuţie discretă dată.

Aplicaţie: Într-o urnă sunt 23 de bile numerotate cu cifra 1, 20 de bile numerotate cu cifra 2, 5 bile numerotate cu cifra 3 şi 2 bile numerotate cu cifra 4. Simulaţi de $N(= 100, 1000, \ldots)$ ori extragerea unei bile şi afişaţi frecvenţa de apariţie a fiecărei cifre. Comparaţi rezultatele obţinute cu cele teoretice.

II. Generarea de numere pseudo-aleatoare, ce urmează o distribuţie continuă dată:

Fie X o variabilă aleatoare ce are funcţia de repartiţie F. Din teorie se ştie ca F este continuă şi monoton crescătoare. Presupunem că F *este inversabilă*, adică există F^{-1}: pentru orice $y \in (0,1)$ există un unic $x \in \mathbb{R}$ astfel încât $F(x) = y$, ceea ce este echivalent cu $F^{-1}(y) = x$.

Procedeul de generare al numerelor aleatoare $Y(i)$, $i = \overline{1,N}$, *este*:

• Se citeşte numărul N, se defineşte funcţia F^{-1}.
• Se generează N numere aleatoare uniform distribuite pe [0,1]: $U(i)$, $i = \overline{1,N}$ (în Matlab U=*unifrnd*(0,1,1,N)).
• Pentru fiecare $i = \overline{1,N}$: $Y(i) := F^{-1}(U(i))$.
• Se afişează: $Y(i), i = \overline{1,N}$.

Verificarea procedeului: Fie variabila aleatoare $U \sim Unif[0,1]$ şi definim variabila aleatoare $Y := F^{-1}(U)$. Arătăm că Y are aceeaşi funcţie de repartiţie ca X: pentru orice $y \in \mathbb{R}$ are loc

$$F_Y(y) = P(Y \leq y) = P(F^{-1}(U) \leq y) = P(U \leq F(y)) = F(y),$$

deci Y are aceeaşi distribuţie ca X, pentru că $F_Y = F$.

Realizaţi un program care generează N numere pseudo-aleatoare conform procedeului de mai sus, care urmează legea Exponenţială cu valoarea medie λ.

Aplicaţie: Timpul necesar ca o imprimantă să printeze un afiş urmează legea Exponenţială cu media de 12 secunde. Simulaţi de $N(= 100, 1000, \ldots)$ ori printarea unui afiş si afişaţi frecvenţa de apariţie a unui timp mai mic decât k secunde, unde $k = 5, 10, 15$. Comparaţi rezultatele obţinute cu cele teoretice.

Funcţii Matlab: ecdf, expcdf, sort, cumsum.