Laboraratorul 4. Crearea de algoritmi proprii pentru generarea variabilelor aleatoare discrete

Bibliografie:

1. I. Văduva. *Modele de simulare*, Editura Universitatii din Bucureşti, 2004.
2. R. Trandafir, **I. Iatan**, Modelare- Simulare. Noţiuni teoretice şi Aplicaţii", Ed. Conspress, Bucureşti, 2013.
3. I. Vladimirescu, *Probabilitati si statistica*, Note de curs, Facultatea de Matematica si Informatica, Universitatea din Craiova, an III, 1995-1996.
4. R. Trandafir, *Modele de simulare*, Note de curs, Facultatea de Hidrotehnica, an III, AIA, 2011-2012.
5. I. Armeanu, V. Petrehus, *Probabilitati si statistica aplicate in biologie*, MatrixRom, Bucuresti, 2006.
6. I. Iatan, *Îndrumător de laborator în Matlab 7.0*, Ed. Conspress, Bucureşti, 2009.

Scopuri:

1) Implementarea in Matlab a metodei inverse de simulare a unei variabile aleatoare discrete.
2) Construirea unor functii in Matlab pentru simularea unei variabile discrete, cu o repartitie particulara (Bernoulli, binomiala, Poisson, geometrica, hipergeometrica), utilizata in fiabilitate.

METODA INVERSA DE SIMULARE A UNEI VARIABILE ALEATOARE DISCRETE

Vom descrie o metoda de simulare a unei variabile aleatoare discrete, ce ia un numar finit de valori, numita *metoda inversa*. Conform acestei metode putem simula orice variabila aleatoare X daca cunoastem functia sa de repartitie F si putem calcula functia inversa F^{-1}.

Folosind aceasta metoda vom construi un program Matlab pentru simularea variabilei discrete X, ce are repartitia

$$X:\begin{pmatrix} a_1 & \dots & a_k & \dots & a_m \\ p_1 & \dots & p_k & \dots & p_m \end{pmatrix},\ \sum_{k=1}^{m} p_k = 1.$$

Functia de repartitie a acesteia este:

$$F_X(x)=\begin{cases} 0, & x \le a_1 \\ p_1, & x \in (a_1, a_2] \\ p_1 + p_2, & x \in (a_2, a_3] \\ \vdots & \\ p_1 + p_2 + \ldots + p_k, & x \in (a_k, a_{k+1}] \\ \vdots & \\ 1, & x > a_m \end{cases}$$

iar functia inversa va fi:

$$F_X^{-1}(u) = a_k,\ u \in (F_X(a_{k-1}), F_X(a_k)], (\forall) k = \overline{1, m},$$

unde:

$$a_0 = -\infty, \text{ iar } F_X(a_0) = 0.$$

Algoritmul pentru simularea variabilei aleatoare X consta in:

- generararea unei valori u uniform repartizate in $[0,1]$,
- determinarea indicelui k pentru care

$$F_X(a_{k-1}) < u \le F_X(a_k) \tag{1}$$

Relatia (1) rezulta din faptul ca:

$$a_{k-1} < x \le a_k \Rightarrow F_X(a_{k-1}) < F_X(x) \le F_X(a_k)$$

si $F_X(x) = u$.

Vom construi programul Matlab corespunzator:

```
function x=simdiscrv(F,a,m)
u=rand;
k=1;
while (u>F(k)) & (k<=m)
   k=k+1;
end
x=a(k);
end
```

Vom aplica functia Matlab precedenta pentru a genera o variabila aleatoare discreta care da numarul punctelor obtinute in experienta aruncarii unui zar o data.

Astfel

$$X:\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix},\ \sum_{k=1}^{6} p_k = 1$$

iar

$$F_X(x)=\begin{cases} 0, & x \le 1 \\ 1/6, & x \in (1, 2] \\ 2/6, & x \in (2, 3] \\ 3/6 & x \in (3, 4] \\ 4/6, & x \in (4, 5] \\ 5/6, & x \in (5, 6] \\ 1, & x > 6 \end{cases}$$

```
>> a=1:7;
>> F=0:1/6:1;
>>x=simdiscrv(F,a,7)
        u =
           0.6557
        x =
            5
```

SIMULAREA UNEI VARIABILE DISCRETE CU REPARTITIE BERNOULLI

Fie X o variabila aleatoare binara astfel incat: $P(X=1)=p,\ p\in[0,1]$ si $P(X=0)=q=1-p$, adica: $X=1$ daca intr-o experienta intamplatoare, un eveniment aleator observabil A se produce cu probabilitatea p (avem de-a face cu un *succes*) si $X=0$ daca se produce evenimentul contrar $\overline{A}$ cu probabilitatea $q=1-p$ (se realizeaza un esec).

Asadar X are repartitia:

$$X:\begin{pmatrix} 0 & 1 \\ q & p \end{pmatrix},\ M(X)=p\,,\ \mathrm{Var}(X)=pq=p(1-p).$$

Functia de repartitie a lui X este:

$$F(x)=P(X<x)=\begin{cases} 0, & x \le 0 \\ q, & x \in (0,1] \\ 1, & x > 1. \end{cases}$$

Simularea unei probe Bernoulli revine la a simula un eveniment aleator de probabilitate constanta p. Un astfel de eveniment este de exemplu $\{U \le p\}$, unde U este o variabila aleatoare uniforma pe $[0,1]$. Evenimentul contrar $\{U > p\}$ este un esec.

Vom construi programul Matlab pentru simularea lui X:

```
function x=bern(p)
u=rand;
if u>p
   x=0;
else
   x=1;
end
end
```

Observatie. Daca aplicam de n ori aceasta functie putem spune ca am simulat o selectie de volum n asupra lui X.

```
>> for i=1:100
        x(i)=bern(0.1);
end
```

SIMULAREA UNEI VARIABILE DISCRETE CU REPARTITIE BINOMIALA

Repartitia binomiala se foloseste la probleme de tipul urmator: Intr-o urna se afla bile albe si negre in proportii date. Fie p probabilitatea ca la o extragere de bila din aceasta urna sa obtinem culoarea alba si $q = 1 - p$. Extragem din aceasta urna n bile, succesiv cate una, cu revenirea bilei extrasa, de fiecare data.

Atunci probabilitatea ca din n bile extrase k sa fie albe este

$$P_n(k) = C_n^k p^k q^{n-k},$$

adica este termenul general al dezvoltarii binomului $(p+q)^n$.

Fie variabila discreta $X \sim B(n, p)$, ce are repartitie binomiala

$$X : \begin{pmatrix} 0 & 1 & \dots & k & \dots & n \\ q^n & C_n^1 pq^{n-1} & & C_n^k p^k q^{n-k} & & C_n^n p^n \end{pmatrix}, \; p, q \in (0,1), \; p + q = 1.$$

Functia de repartitie a acesteia este:

$$F_X(x) = \begin{cases} 0, & x \le 0 \\ q^n, & x \in (0,1] \\ q^n + C_n^1 pq^{n-1}, & x \in (1,2] \\ \vdots & \\ q^n + C_n^1 pq^{n-1} + \ldots + C_n^k p^k q^{n-k}, & x \in (k, k+1] \\ \vdots & \\ 1, & x > n. \end{cases}$$

Media si dispersia variabilei aleatoare X vor fi:

$$\begin{cases} M(X) = np \\ D^2(X) = npq. \end{cases} \qquad (2)$$

Pentru n suficient de mare, conform teoremei limita centrala rezulta ca variabila

$$W = \frac{X - np}{\sqrt{npq}} \sim N(0,1).$$

Pe baza acestei relatii vom construi programul Matlab de generare a lui $X \sim B(n, p)$.

```
function x=binom(n,p)
w=rand;
q=1-p;
x=round(n*p+w*sqrt(n*p*q));
end
>> x=binom(5000,1/4)
x =
      1271
```

Observatie. Variabila aleatoare discreta $X \in \mathbb{N}$ este o variabila binomiala $B(n, p), n \in \mathbb{N}^*$, $0 < p < 1$ daca X reprezinta numarul de succese in n probe Bernoulli independente, adica

$$X = \sum_{i=1}^{n} Z_i ,$$

unde Z_i sunt n variabile aleatoare Bernoulli, independente.

Pornind de la aceasta observatie putem simula variabila discreta X cu repartitie binomiala numarand succesele in n probe Bernoulli independente.

```
function x =bern2(n,p)
for h=1:10
```

```
for k=1:n
xx(k)=bern(p);
end
x(h)=sum(xx);
end
end
>> x=bern2(100,0.2)
x =
  Columns 1 through 9
    15    22    16    18    20    22    26    17    16
  Column 10
    17
```

Vom folosi formulele (2) pentru validarea algoritmului precedent.

```
>> mean(x)
ans =
   18.9000
>> 100*0.2
ans =
    20
>> std(x)
ans =
    3.5103
>> sqrt(100*0.2*0.8)
ans =
     4
```

SIMULAREA UNEI VARIABILE DISCRETE CU REPARTITIE POISSON

Repartitia Poisson este o repartitie asemanatoare cu cea binomiala, deosebindu-se de aceasta prin faptul ca numarul n de extrageri din urna este foarte mare iar probabilitatea p de extragere a unei bile albe este foarte mica. Cu alte cuvinte, repartitia Poisson este un caz limita al repartitiei binomiale, pentru $n \to \infty$ si $p \to 0$, unde produsul $np = \lambda$ este constant.

Deoarece repartitia Poisson se foloseste in cazul aparitiei foarte rare a unui eveniment se mai numeste si repartitia evenimentelor rare.

Aceasta repartitie este frecvent intalnita in studiul unor fenomene din biologie, telecomunicatii, controlul statistic al calitatii (atunci cand probabilitatea obtinerii unui defect este foarte mica).

Probabilitatea ca din cele n bile extrase, k sa fie albe este

$$P(k) = \lim_{n\to\infty} P_n(k) = \lim_{n\to\infty} C_n^k p^k q^{n-k} = \lim_{n\to\infty} C_n^k \left(\frac{\lambda}{n}\right)^k \left(1-\frac{\lambda}{n}\right)^{n-k} =$$
$$= \frac{\lambda^k}{k!} \cdot \underbrace{\lim_{n\to\infty} \frac{n!}{n^k (n-k)!}}_{=1} \cdot \underbrace{\lim_{n\to\infty} \left(1-\frac{\lambda}{n}\right)^{n-k}}_{e^{-\lambda}} = \frac{\lambda^k}{k!} \cdot e^{-\lambda}, \quad \lambda > 0.$$

Fie variabila discreta $X \sim P(\lambda)$, ce are repartitie Poisson

$$X : \begin{pmatrix} 0 & 1 & 2 & \dots & k & \dots \\ e^{-\lambda} & e^{-\lambda} \cdot \frac{\lambda}{1!} & e^{-\lambda} \cdot \frac{\lambda^2}{2!} & & e^{-\lambda} \cdot \frac{\lambda^k}{k!} & \end{pmatrix},$$

Media si dispersia variabilei aleatoare X vor fi:

$$\begin{cases} M(X) = \lambda \\ D^2(X) = \lambda. \end{cases}$$

Functia Matlab urmatoare simuleaza $X \sim P(\lambda)$ pornind de la o repartitie binomiala $B(n, p)$, pentru care se noteaza $\lambda = np$ si se face presupunerea ca $n \to \infty$ si $p \to 0$, λ ramanand constant.

```
function x=poisson(la,p)
n=round(la/p)
x=binom(n,p);
end
>> x=poisson(1,0.001)
        n =
              1000
        x =
            2
```

SIMULAREA UNEI VARIABILE DISCRETE CU REPARTITIE GEOMETRICA

Fie X o variabila aleatoare care semnifica numarul de esecuri pana la aparitia unui success intr-un sir oarecare de probe Bernoulli independente.

Deci X are repartitia:

$$X:\begin{pmatrix} 0 & 1 & 2 & \dots & k & \dots & n \\ p & pq & pq^2 & & pq^k & & pq^n \end{pmatrix},\ M(X)=\frac{q}{p},\ \mathrm{Var}(X)==\frac{q}{p^2}.$$

Functia de repartitie a lui X este:

$$F(x)=P(X<x)=\sum_{k=0}^{x} pq^k=1-q^{x+1},\ x=0,1,2,\dots n$$

adica este o functie de repartitie discreta.

Numele de repartitie geometrica provine din faptul ca $P(X=x)=pq^x$ reprezinta termenul unei progresii geometrice.

Programul Matlab urmator simuleaza X, pe baza unui algoritm care numara esecurile produse pana la realizarea unui succes intr-un sir de probe Bernoulli independente:

```
function x=rgeom(p)
x=0;
k=0;
while(k~=1)
   u=rand;
   if u<=p
      k=k+1;
   else
      x=x+1;
   end
end
end
>>x=rgeom(0.2)
x=
  7
```

Observatie. Simularea variabilei aleatoare X, ce are repartitie geometrica se poate realiza si prin metoda inversa cu formula:

$$X = \left[\frac{\log(U)}{\log(q)}\right],$$

unde :

- ✓ $[a]$ este partea intreaga a lui a,
- ✓ U este o variabila aleatoare uniform repartizata in $[0,1]$.

Vom construi functia Matlab ce simuleaza $X \sim \mathrm{Geom}(p)$ cu metoda inversa.

```
function x =rgeom2(p)
q=1-p;
u=rand;
x=round(log(u)/log(q));
end
>> x=rgeom2(0.3)
x =
   2
```

SIMULAREA UNEI VARIABILE DISCRETE CU REPARTITIE HIPERGEOMETRICA

Vom considera experimentul cu urna descris in legatura cu repartitia binomiala, cu deosebirea ca acum cele n bile se extrag din urna, ce contine a bile albe si b bile negre, fara revenirea bilei extrasa inaintea extragerii urmatoare.

Probabilitatea ca din cele n bile extrase, k sa fie albe este

$$P_n(k) = \frac{C_a^k C_b^{n-k}}{C_{a+b}^n}.$$

Daca $X \sim H(n,a,b)$, adica X are repartitie hipergeometrica de parametri n,a,b atunci

$$X : \begin{pmatrix} 0 & 1 & \dots & k \\ \frac{C_a^0 C_b^n}{C_{a+b}^n} & \frac{C_a^1 C_b^{n-1}}{C_{a+b}^n} & & \frac{C_a^k C_b^{n-k}}{C_{a+b}^n} \end{pmatrix},$$

pentru orice $\max(0, n-b) \le k \le \min(n,a)$.

Functia de repartitie a lui X este:

$$F(x) = \sum_{k=\max(0,n-b)}^{\min(n,a)} \frac{C_a^k C_b^{n-k}}{C_{a+b}^n}.$$

Media si dispersia variabilei aleatoare X vor fi:

$$\begin{cases} M(X) = np \\ D^2(X) = np(1-p)\dfrac{a+b-n}{a+b-1}. \end{cases} \quad (3)$$

Se introduc urmatoarele notatii:

- $N_i = N_{i-1} - 1,\ (\forall) i = \overline{1, n}$ semnifica numarul bilelor din urna dupa extractia de rang i (unde $N_0 = a + b$),
- $p_i = \dfrac{N_{i-1} p_{i-1} - S}{N_{i-1} - 1},\ (\forall) i = \overline{1, n}$ reprezinta probabilitatea de a extrage o bila alba la extragerea i, unde $S = 1$ daca la extractia $i-1$ a fost gasita o bila alba si $S = 0$ in caz contrar.

Functia Matlab urmatoare permite simularea variabilei hipergeometrice X.

```
function x=hipergeom(n,a,b,p)
i=0;
x=0;
N=a+b;
while (i<n)
u=rand;
i=i+1;
if u<=p
   S=1;
else
   S=0;
end
x=x+S;
p=(N*p-S)/(N-1);
N=N-1;
end
end
```

Vom simula o selectie de volum 20 asupra lui X si vom folosi formulele (3) pentru validarea algoritmului precedent.

```
>>for k=1:20
      x(k)=hipergeom(10,15,13,0.9);
end
>> mean(x)
ans =
        8.9000
>> 10*0.9
      ans =
         9
>> std(x)
 ans =
        0.7182
>> 10*0.9*0.1*18/27
ans =
        0.6000
```