Seminar Nr. 2

Computer Simulations of Discrete Random Variables; Discrete Methods

```
clear all;
format longg;
```

1. Function **rnd** in Statistics Toolbox; special functions **rand** and **randn**.

```
rand
```

```
ans =
       0.882402795931734
```

```
rand(1, 20)
```

```
ans = 1×20
      0.0198754228247094        0.34176488323731        0.766027445694415 ···
```

```
randn
```

```
ans =
      -0.354291622942114
```

```
randn(1, 20)
```

```
ans = 1×20
       0.434227622152402      -0.101478020404053        2.02876798537404 ···
```

```
generator = input('Generator (1-6): ');
```

2. Using a Standard Uniform $U(0,1)$ random number generator, write Matlab codes that simulate the following common discrete probability distributions:

a. Bernoulli Distribution $Bern(p)$, with parameter $p \in (0,1)$:

$$X \begin{pmatrix} 0 & 1 \\ 1-p & p \end{pmatrix}$$

```
if generator==1
    p = input('p in (0,1): ');

    N = input('Nr of simulations: ');
    X = rand(1, N) < p

    values = unique(X);
    nX = histcounts(X, 2);
    relfreq = nX/N
end
```

b. Binomial Distribution $B(n, p)$, with parameters $n \in \mathbb{N}, p \in (0, 1)$:

$$X \begin{pmatrix} k \\ C_n^k p^k q^{n-k} \end{pmatrix}_{k=\overline{0,n}}$$

```
if generator==2
    p = input('p in (0,1): ');
    n = input('n (positive number): ');

    N = input('Nr of simulations: ');
    X = zeros(1, N);
    for i = 1:N
        X(i) = sum(rand(1, n) < p);
    end
    X

    fprintf('simulated probab. P(X = 2) = %1.5f\n', mean(X==2));
    fprintf('true probab. P(X = 2) = %1.5f\n', binopdf(2, n, p));
    fprintf('error= %e\n\n', abs(binopdf(2, n, p)) - mean(X == 2));

    fprintf('simulated probab. P(X <= 2) = %1.5f\n', mean(X<=2));
    fprintf('true probab. P(X <= 2) = %1.5f\n', binocdf(2, n, p));
    fprintf('error= %e\n\n', abs(binocdf(2, n, p)) - mean(X <= 2));

    fprintf('simulated probab. P(X < 2) = %1.5f\n', mean(X < 2));
    fprintf('true probab. P(X < 2) = %1.5f\n', binocdf(1, n, p));
    fprintf('error= %e\n\n', abs(binocdf(1, n, p)) - mean(X <= 2));

    fprintf('simulated mean E(X) = %5.5f\n', mean(X));
    fprintf('simulated mean E(X) = %5.5f\n', n*p);
    fprintf('error= %e\n\n', abs(n*p - mean(X)));
end
```

c. Geometric Distribution $Geo(p)$, with parameter $p \in (0, 1)$:

$$X \begin{pmatrix} k \\ pq^k \end{pmatrix}_{k\in\mathbb{N}}$$

```
if generator==3
    p = input('p in (0,1): ');

    N = input('Nr of simulations: ');
    X = zeros(1, N);
    for i=1:N
        while rand >= p
            X(i) = X(i) + 1;
        end
    end
    X

    fprintf('simulated probab. P(X = 2) = %1.5f\n', mean(X==2));
    fprintf('true probab. P(X = 2) = %1.5f\n', geopdf(2, p));
    fprintf('error= %e\n\n', abs(geopdf(2, p)) - mean(X == 2));

    fprintf('simulated probab. P(X <= 2) = %1.5f\n', mean(X<=2));
    fprintf('true probab. P(X <= 2) = %1.5f\n', geocdf(2, p));
    fprintf('error= %e\n\n', abs(geocdf(2, p)) - mean(X <= 2));

    fprintf('simulated probab. P(X < 2) = %1.5f\n', mean(X < 2));
```

```
    fprintf('true probab. P(X < 2) = %1.5f\n', geocdf(1, p));
    fprintf('error= %e\n\n', abs(geocdf(1, p)) - mean(X <= 2));

    fprintf('simulated mean E(X) = %5.5f\n', mean(X));
    fprintf('simulated mean E(X) = %5.5f\n', (1 - p) / p);
    fprintf('error= %e\n\n', abs((1 - p) / p - mean(X)));
end
```

d. Negative Binomial Distribution $NB(n,p)$ with parameters $n \in \mathbb{N}, p \in (0,1)$:

$$X\begin{pmatrix} k \\ C_{n+k-1}^{k} p^n q^k \end{pmatrix}_{k\in\mathbb{N}}$$

```
if generator==4
    p = input('p in (0,1): ');
    n = input('n (positive number): ');

    N = input('Nr of simulations: ');
    X = zeros(1, N);
    for i=1:N
        cnt = 0;
        while cnt < n
            if rand >= p
                X(i) = X(i) + 1;
            else
                cnt = cnt + 1;
            end
        end
    end
    X

    fprintf('simulated probab. P(X = 2) = %1.5f\n', mean(X==2));
    fprintf('true probab. P(X = 2) = %1.5f\n', nbinpdf(2, n, p));
    fprintf('error= %e\n\n', abs(nbinpdf(2, n, p)) - mean(X == 2));

    fprintf('simulated probab. P(X <= 2) = %1.5f\n', mean(X<=2));
    fprintf('true probab. P(X <= 2) = %1.5f\n', nbincdf(2, n, p));
    fprintf('error= %e\n\n', abs(nbincdf(2, n, p)) - mean(X <= 2));

    fprintf('simulated probab. P(X < 2) = %1.5f\n', mean(X < 2));
    fprintf('true probab. P(X < 2) = %1.5f\n', nbincdf(1, n, p));
    fprintf('error= %e\n\n', abs(nbincdf(1, n, p)) - mean(X <= 2));

    fprintf('simulated mean E(X) = %5.5f\n', mean(X));
    fprintf('simulated mean E(X) = %5.5f\n', n * (1 - p) / p);
    fprintf('error= %e\n\n', abs(n * (1 - p) / p - mean(X)));
end
```

e. Poisson Distribution $\mathcal{P}(\lambda)$ with parameter $\lambda > 0$:

$$X\begin{pmatrix} k \\ \frac{\lambda^k}{k!} e^{-\lambda} \end{pmatrix}_{k\in\mathbb{N}}$$

```
if generator==5
    L = input('lambda (positive number): ');

    N = input('Nr of simulations: ');
    X = zeros(1, N);
```

```
    for i=1:N
        cnt = exp(-L);
        U = rand;
        while U >= cnt
            X(i) = X(i) + 1;
            cnt = cnt + exp(-L) * L^X(i) / factorial(X(i));
        end
    end
    X

    fprintf('simulated probab. P(X = 2) = %1.5f\n', mean(X==2));
    fprintf('true probab. P(X = 2) = %1.5f\n', poisspdf(2, L));
    fprintf('error= %e\n\n', abs(poisspdf(2, L)) - mean(X == 2));

    fprintf('simulated probab. P(X <= 2) = %1.5f\n', mean(X<=2));
    fprintf('true probab. P(X <= 2) = %1.5f\n', poisscdf(2, L));
    fprintf('error= %e\n\n', abs(poisscdf(2, L)) - mean(X <= 2));

    fprintf('simulated probab. P(X < 2) = %1.5f\n', mean(X < 2));
    fprintf('true probab. P(X < 2) = %1.5f\n', poisscdf(1, L));
    fprintf('error= %e\n\n', abs(poisscdf(1, L)) - mean(X <= 2));

    fprintf('simulated mean E(X) = %5.5f\n', mean(X));
    fprintf('simulated mean E(X) = %5.5f\n', L);
    fprintf('error= %e\n\n', abs(L - mean(X)));
end
```

Optional

f. Discrete Uniform Distribution $U(m)$ with parameter $m \in \mathbb{N}$:

$$X \begin{pmatrix} k \\ \frac{1}{m} \end{pmatrix}_{k=\overline{1,m}}$$

```
if generator==6
    m = input('m (positive integer): ');

    N = input('Nr of simulations: ');
    X = ceil(rand(1, N) * m)

    fprintf('simulated probab. P(X = 2) = %1.5f\n', mean(X==2));
    fprintf('true probab. P(X = 2) = %1.5f\n', unidpdf(2, m));
    fprintf('error= %e\n\n', abs(unidpdf(2, m)) - mean(X == 2));

    fprintf('simulated probab. P(X <= 2) = %1.5f\n', mean(X<=2));
    fprintf('true probab. P(X <= 2) = %1.5f\n', unidcdf(2, m));
    fprintf('error= %e\n\n', abs(unidcdf(2, m)) - mean(X <= 2));

    fprintf('simulated probab. P(X < 2) = %1.5f\n', mean(X < 2));
    fprintf('true probab. P(X < 2) = %1.5f\n', unidcdf(1, m));
    fprintf('error= %e\n\n', abs(unidcdf(1, m)) - mean(X <= 2));

    fprintf('simulated mean E(X) = %5.5f\n', mean(X));
    fprintf('simulated mean E(X) = %5.5f\n', (m + 1) / 2);
    fprintf('error= %e\n\n', abs((m + 1) / 2 - mean(X)));
end
```

```
X = 1×1000
     8     1     6     1     5     5     1     1    10     6     4     4     2···
simulated probab. P(X = 2) = 0.09800
true probab. P(X = 2) = 0.10000
error= 2.000000e-03
simulated probab. P(X <= 2) = 0.20500
```

```
true probab. P(X <= 2) = 0.20000
error= -5.000000e-03
simulated probab. P(X < 2) = 0.10700
true probab. P(X < 2) = 0.10000
error= -1.050000e-01
simulated mean E(X) = 5.51100
simulated mean E(X) = 5.50000
error= 1.100000e-02
```