STATISTICAL COMPUTER SIMULATIONS AND MONTE CARLO METHODS

Sanda Micula [1*]

ABSTRACT
In this paper we present Monte Carlo methods for estimating distribution characteristics of random variables. We describe algorithms for computer simulations of some common distributions and their implementation in MATLAB. The paper concludes with examples and applications.

AMS Subject Classification: 60E05, 60G25, 60G99, 60J10, 62P99, 65C05, 65C60.

KEYWORDS: *Monte Carlo methods, random variables, random number generators, computer simulations, MATLAB.*

1. Introduction

In this day and age, modelling of phenomena has become necessary in every field of research. Scientists in every area, try to describe what they see, to explain what is observed and to use this knowledge to make inferences and predictions about events occurring in the world we live in. Probabilistic and statistical methods are employed in every research area. These are methods by which decisions are made based on the analysis of data gathered through carefully designed experiments. This entails building of models that closely resemble real life situations, studying the models, developing computational algorithms and procedures and, based on those, making inferences about unknown characteristics. Experimental Statistics provides models that allow us to predict the behavior of a complex system, *before* actually building or trying it out in real life. It also allows us to assess the probability of error.

In real practice, however, one often deals with rather complicated random phenomena, where computation of probabilities and other characteristics of interest is far from being straightforward. When direct computation is too complicated, resource or time consuming, too approximate, or simply not feasible, we make use of *Monte Carlo methods*. These are based on computer simulations involving random numbers. If we can write a computer code that replicates a certain phenomenon, we can simulate it any number of times (thousands or millions) and draw conclusions about its real life behavior. The longer run is simulated, the more accurate our predictions are. Monte Carlo methods can be used for computation of probabilities, expected values and other distribution characteristics.

2. Random variables; Definitions and classifications

[1*] Corresponding author. Department of Mathematics and Computer Science, Babeş-Bolyai University, Cluj-Napoca, smicula@math.ubbcluj.ro

Definition 1.1 The set of all possible outcomes of an experiment is called the **sample space** of that experiment and is denoted by S. Its elements are called **elementary events.** An **event** is a collection of elementary events, i.e. a subset of S.

Definition 1.2 A collection of events $\mathcal{K} \subseteq$ S is called a **σ-field** (or **σ-algebra**) on the sample space S, if it satisfies the conditions

(i) $\mathcal{K} \neq \emptyset$;
(ii) $\mathrm{A} \in \mathcal{K} \Rightarrow \bar{\mathrm{A}} \in \mathcal{K}$;
(iii) $A_1, A_2, \ldots, A_n \in \mathcal{K} \Rightarrow A_1 \cup A_2 \cup \ldots \cup A_n \in \mathcal{K}$.
Definition 1.3 Let S be a sample space and $\mathcal{K} \subseteq$ S a σ-field on it. **Probability** is a function P: $\mathcal{K} \to \mathbb{R}$ satisfying the conditions
(i) P (S) = 1;
(ii) P (A) ≥ 0, ∀ A ∈ $\mathcal{K}$;
(iii) (iii) P $(A_1 \cup A_2 \cup \ldots)$ = P $(A_1) +$ P $(A_2) + \ldots$, for any finite or countable collection of *mutually exclusive (disjoint)* events in $\mathcal{K}$;
The triplet (S, $\mathcal{K}$, P) is called a **probability space.**

Definition 1.4 Let (S, $\mathcal{K}$, P) be a probability space.
A **random variable** is a function X: S → ℝ for which the inverse image $X^{-1}((-\infty, x]) = \{e \in \mathrm{S} \mid X(e) \leq x\} \in \mathcal{K}$. If the set of values that X takes is at most countable, then X is a **discrete random variable**, if it is an interval in ℝ, then X is a **continuous random variable**.
The function F: ℝ → ℝ defined by $F(x)$= P $(X \leq x)$ is called the **cumulative distribution function (cdf)** of X.

If X is a discrete random variable, then a better way of describing it is to give its **probability distribution function (pdf)**, an array that contains all its values x_i, and the corresponding probabilities with which each value is taken $p_i = P(X = x_i)$

$$X \begin{pmatrix} x_i \\ p_i \end{pmatrix}_{i \in I}$$

(2.1)
In this case, the cdf $F(x) = \sum_{i \in I} p_i$ (hence the name).
If X is a continuous random variable, then its cdf is absolutely continuous, which means that there exists a function $f : \mathbb{R} \to \mathbb{R}$ such that $F(x) = \int_{-\infty}^{x} f(t)\, dt$. That function f is called the **probability density function (pdf)** of the continuous random variable X.
So, either way (the discrete or the continuous case), the pdf is what describes a random variable.
Remark 1.5 Although in both cases we call it generically a pdf, the word *distribution* emphasizes a discrete behaviour, whereas *density* suggests a continuous set. However, some authors do not make the distinction between the two cases and call it a probability *density* function in both cases.

2.1. Common discrete distributions

For describing discrete random variables associated with experiments, an important notion is that of *Bernoulli* trials. There are three characteristics that define such trials:

- they are independent;
- each trial has two possible outcomes, which we refer to as "success" and "failure", respectively;
- the probability of success, p, is the same in every trial (and then we denote the probability of failure by $q = 1 - p$).

Note that the second condition above is not at all restrictive, we simply divide the sample space into "whatever is of interest to us" (and call that success) and everything else (which will constitute failure).

Bernoulli distribution with parameter $p \in (0, 1)$, *Bern*(p). This is the simplest of distributions, with pdf

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

(2.2)

It can be used to model a Bernoulli trial.

Binomial distribution with parameters $n \in \mathbb{N}, p \in (0, 1)$, $B(n, p)$. Consider a series of n Bernoulli trials with probability of success p in every trial. Let X be the number of successes that occur in the n trials. That is a Binomial distribution, with pdf

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

(2.3)

Remark 1.6 Note that a binomial $B(n, p)$ variable is the sum of n independent *Bern*(p) variables.

Geometric distribution with parameter $p \in (0, 1)$, $G(p)$. Consider an infinite sequence of Bernoulli trials with probability of success p in every trial. Let X denote the number of failures that occurred before the first success. That is a Geometric distribution, with pdf

$$X\begin{pmatrix} k \\ p\, q^k \end{pmatrix}_{k\in\mathbb{N}}$$

(2.4)

Pascal (Negative Binomial) distribution with parameters $n \in \mathbb{N}, p \in (0, 1)$, $NB(n, p)$. Just as before, consider an infinite sequence of Bernoulli trials with probability of success p in every trial. Let X denote the number of failures that occurred before the nth success. That is a Negative Binomial distribution, with pdf

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

(2.5)

Remark 1.7 Note that a negative binomial $NB(n, p)$ variable is the sum of n independent $G(p)$ variables.

Poisson distribution with parameter $\lambda > 0$, $\mathcal{P}(\lambda)$, with pdf

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

(2.6)

Such a variable is defined in the context of a Poisson *process*: a process in which discrete events are observed in a continuous medium (length, aria, volume or time). Examples would be: messages that arrive within an hour, white blood cells existing in a drop of blood, earthquakes that happen in an area during a year, etc. Such events are called *rare events*, because they are extremely unlikely to occur simultaneously or within a short interval (of time, length, etc.). The Poisson variable X denotes the number of such rare events that occur in a given interval of the continuous medium. As that number increases (tending to ∞), the probability of that many rare events occurring tends to 0 (since they are rare…). The parameter of the Poisson distribution, λ, represents the average number of the considered rare events per unit (of time, length, etc.).

2.2. Uniform distribution

The continuous Uniform distribution plays a unique role in statistical modelling, in the sense that *any* random variable (with any pdf), discrete or continuous, can be generated from a uniformly distributed variable, as we shall see in the next section.

The pdf of a **Uniform distribution** $\mathcal{U}(a, b)$, of parameters $-\infty < a < b < \infty$, is given by

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in [a, b] \\ 0 \quad , & x \notin [a, b] \end{cases}$$

and is used whenever a value is picked "at random" from an interval, in situations when *all* values from an interval are equally probable to be taken by a random variable.

By integration, its cdf is

$$F(x) = \begin{cases} 0 \quad , & x < a \\ \dfrac{x-a}{b-a}, & x \in [a, b] \\ 1 \quad , & x > b \end{cases}$$

An important special case is that of a **Standard Uniform distribution** $\mathcal{U}(0,1)$, with pdf

$$f_U(x) = \begin{cases} 1, & x \in [0, 1] \\ 0, & x \notin [0, 1] \end{cases}$$

(2.7)

and cdf

$$F_U(x) = \begin{cases} 0, & x < 0 \\ x, & x \in [0, 1] \\ 1, & x > 1 \end{cases}$$

(2.8)

3. Monte Carlo methods

As mentioned before, Monte Carlo methods are used in simulating random phenomena and effectively estimating distribution characteristics. For the distributions discussed in Section 2.1, we generate a large number of simulations and estimate probabilities by long-run relative frequencies.

3.1. Simulation of Bernoulli and Binomial random variables

Bernoulli distribution

As we mentioned in Section 2.2., a Standard Uniform $\mathcal{U}(0,1)$ variable can be used to model a binary process, that is, a Bernoulli trial. Let $p \in (0,1)$ be fixed and let $U \in \mathcal{U}(0,1)$. Define

$$X = \begin{cases} 1, & U < p \\ 0, & U \geq p \end{cases}$$

We consider a "success" the event $X = 1$ and a "failure" the event $X = 0$. Then, by (2.8), we have

$$P(\text{success}) = P(X = 1) = P(U < p) = F_U(p) = p.$$

Thus, $X \in Bern(p)$ and we have generated a Bernoulli random variable.

Example 3.1 Consider the experiment of rolling a die**.** The event of interest, i.e. "success", is getting a 6.

Below is a MATLAB[1] code generating $N = 10000$ simulations of a Bernoulli random variable with parameter $p = 1/6$.

```
% Simulate Bernoulli distribution Bern(p).
p = 1/6; % the parameter of the Bern distr.
N = 1e5; % the number of simulations
U = unifrnd(1,N); % sequence of N U(0,1)random numbers
X = (U < p); % simulation of N Bernoulli trials

% Compare it to the Bern(p) distribution
U_X = unique(X) % the values of X listed ONLY ONCE, no
% repetitions
n_X = hist(X,length(U_X)); % the frequency of each value in U_X % (how many times each occurs)
rel_freq = n_X/N % the relative freq. n_X/N approximates the    % probability
```

The run of this code will yield the results

```
U_X =    0     1
rel_freq =   0.8331    0.1669
```

The probability $p = 1/6$ with which X takes the value 1 is well approximated by the relative frequency of the occurrence of the value 1, namely, 0.1669.

Now, having generated a Bernoulli variable, we can use that to simulate other discrete distributions described in Section 2.1.

Binomial Distribution

We use Remark 1.6. We generate n independent *Bern*(p) and add them to obtain a Binomial $B(n, p)$ distribution.

Example 3.2 A lab network consists of 20 computers. Twenty five percent of the computers are in need of updated antivirus software. Let X denote the number of computers in the lab who have the latest antivirus software installed.

Then, "success" here means that a computer is up-to-date in antivirus protection and the probability of success is $p = 0.75$. The number of trials is $n = 20$. The

[1]MATLAB® is a trademark of MathWorks Inc.

variable X denotes the number of successes occurred in the n trials and, therefore, has a Binomial $B(20, 0.75)$ distribution. The code for the simulation of this problem is given below. This time, instead of looking at each individual value of the variable and its relative frequency, we compare it graphically to the actual Binomial distribution considered (see Figure 1).

```
% Simulate Binomial distribution B(n,p).
n = 20;
p = 0.75 % the parameters of the Binomial distr.
N = 1e5; % the number of simulations
for i = 1 : N
   U = unifrnd(1,n); % a sequence of n U(0,1)random numbers
   X(i) = sum(U < p); % generate a B(n,p) variable as the sum
   % of the n Bernoulli variables
end

% Compare it to the B(n,p) distribution.
k = 0 : n; % all the values of a Binomial distr.
p_k = binopdf(k,n,p); % the probabilities of a Binomial distr.
U_X = unique(X); % the values of X listed ONLY ONCE, no
% repetitions
n_X = hist(X,length(U_X)); % the frequency of each value in U_X (how many times each
occurs)
rel_freq = n_X/N; % the relative freq. n_X/N approximates the
% probability
```

3.2. Simulation of Geometric and Negative Binomial random variables
Geometric distribution

A Geometric random variable represents the number of failures occurred before the first success, in an infinite sequence of Bernoulli trials. Therefore, we simulate Bernoulli trials and count the number of failures before the occurrence of the first success.

Example 3.3 A computer virus can damage a file in the system it has entered with probability 0.6. The system manager checks the files for this virus in order to clean them. Let X denote the number of files that were found to be clean before the first infected file is found.

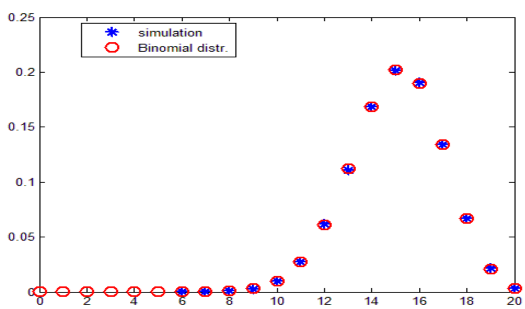

Figure 1. Binomial distribution simulation, N = 10000 simulations

Define "success" as a file being infected by this virus. Then variable X has a Geometric distribution with parameter $p = 0.6$. The code for simulating this variable and the graphical comparison with the $G(0.6)$ distribution (Figure 2) are given below.

```
% Simulate Geometric distribution G(p).
p = 0.6 % parameter of the Geometric distr.
N = 1e5; % the number of simulations
for i = 1 : N
   X(i) = 0; % initial number of failures
   U = unifrnd(1,1);
   while U >= p % "U < p" represents success, so "U >= p"
   % represents failure
      X(i) = X(i) + 1; % add the number of failures
   end  % stop at the first success
end

% Compare it to the G(p) distribution.
k = 0 : 20; % the relevant values of a Geometric distr. (those
% with probability > 0); otherwise, there is an infinite number % of values
p_k = geopdf(k,p); % the probabilities of a Geometric distr.
U_X = unique(X); % the values of X listed ONLY ONCE, no
% repetitions
n_X = hist(X,length(U_X)); % the frequency of each value in UX
rel_freq = n_X/N; % the relative freq. n_X/N approximates the
% probability
```

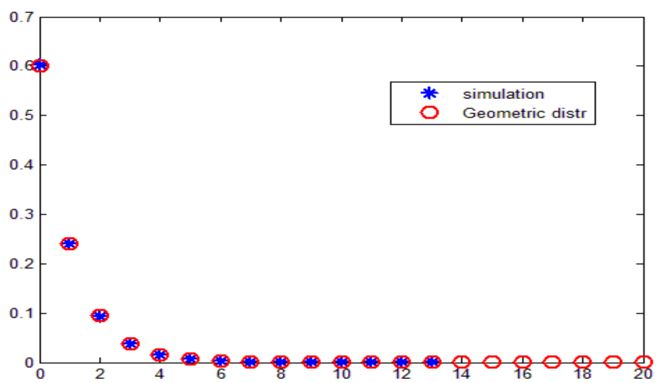

Figure 2. Geometric distribution simulation, N = 10000 simulations

Negative Binomial distribution

We make use of Remark 1.7. We generate n independent $G(p)$ variables and add them up.

Example 3.4 Consider, again, the experiment of rolling a die. Let X denote the number of even numbers rolled before the third odd number occurs.

Here, "success" means that an odd number shows on the die and its probability is $p = 0.5$. Then X has a Negative Binomial distribution with parameters $n = 3$ and $p = 0.5$. The code for simulating this variable and the graphical comparison with the $NB(3, 0.5)$ distribution (Figure 3) are given below.

```
% Simulate Negative Binomial distribution NB(n,p).
n = 0.5;
p = 0.6 % the parameters of the Negative Binomial distr.
N = 1e5; % the number of simulations
for i = 1 : N
   for j = 1 : n
      Y(j) = 0; % initial number of failures
     U = unifrnd(1,1);
      while U >= p % "U < p" represents success, so "U >= p"
          % represents failure
         Y(j) = Y(j) + 1; % add the number of failures
      end % stop at the first success
   end
   X(i) = sum(Y); % generate a NB(n,p) variable as the sum
   % of the n Geometric variables
end

% Compare it to the NB(n,p) distribution.
k = 0 : 25; % the relevant values of a Negative Binomial distr. % (those with probability > 0); otherwise, there is an infinite % number of values
p_k = nbinpdf(k,n,p); % the probabilities of a Geometric distr.
U_X = unique(X); % the values of X listed ONLY ONCE, no
% repetitions
n_X = hist(X,length(U_X)); % the frequency of each value in UX
rel_freq = n_X/N; % the relative freq. n_X/N approximates the
% probability
```

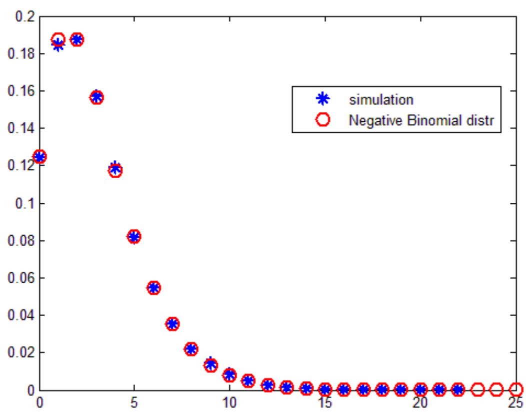

Figure 3. Negative Binomial distribution simulation, N = 10000 simulations

3.3. Simulation of arbitrary discrete distributions

There are several ways of generating a random variable with an arbitrary discrete pdf. We present here one of the simplest methods, which is based on our earlier simulation of a Bernoulli trial. Let us revisit how a Bernoulli variable was generated. We used a random number $U \in \mathcal{U}(0,1)$ to divide the interval $[0, 1]$ into two parts of lengths p and $1 - p$, respectively. Then the value of the Bernoulli variable was determined according to which part of the interval U fell in. Now we will do the same when the interval $[0, 1]$ is divided into an arbitrary number of subintervals. The procedure is described in the following algorithm.

Let X be any discrete random variable with pdf

$$X \begin{pmatrix} x_i \\ p_i \end{pmatrix}_{i \in I} .$$

Algorithm 3.5 Simulation of discrete random variables

1. Divide the interval $[0, 1]$ into the subintervals $\{A_i\}_{i \in I}$ as follows
   $A_0 \ = [0 \,, p_0 \,)$
   $A_1 \ = \ [p_0 \,, p_0 + p_1 \,)$
   $A_2 \ = \ [p_0 + p_1, p_0 + p_1 + p_2 \,)$
   $\vdots$
2. Let $U \in \mathcal{U}(0,1)$ be a Standard Uniform random variable.
3. When $U \in A_i$, let $X = x_i$.

Indeed, it follows that for each $i \in I$, subinterval A_i has length p_i and, by (2.8),
$P(p_0 + p_1 + \ldots + p_{i-1} \leq U < p_0 + p_1 + \ldots + p_i) = F_{U,i} - F_{U,i-1} = p_i,$
where we used the notation $F_{U,j} = F(p_0 + p_1 + \ldots + p_j)$.

Remark 3.6

1. Note that Algorithm 3.5 works both for a finite set of values I or for a countably infinite one.
2. The algorithm above can also be used to generate the common distributions discussed in Sections 3.1 and 3.2, but we preferred to use their special properties and the relationship to Bernoulli trials.

Let us use Algorithm 3.5 to generate a Poisson $\mathcal{P}(\lambda)$ variable with pdf given by (2.6). The initial value of the its cdf will be $F_{U,0} = p_0 = e^{-\lambda}$ and subsequent values of the cdf are obtained by the recurrence formula

$$F_{U,i} = F_{U,i-1} + p_i = F_{U,i-1} + \frac{\lambda^i}{i!} e^{-\lambda}.$$

Example 3.6 In a certain area, at a local firm, network breakdowns occur every 10 months, on the average. Let X denote the number of such network breakdowns that occur in the next 10 months.

Network breakdowns qualify as rare events, because no two such breakdowns can occur simultaneously. Thus, the number X of breakdowns in the next 10 months has a Poisson distribution with parameter $\lambda = 10$. We use Algorithm 3.5 to simulate it. The results of $N = 10000$ simulations are given below.

```
% Simulate Poisson distribution P(lambda).
lambda = 10; % the parameter of the Poisson distr
N = 1e5; % the number of simulations
for j = 1 : N
```

```
    U = unifrnd(1,1); % generated U(0,1) variable
    i = 0; % initial value
    F(j) = exp(-lambda); % initial value of the cdf F(0)
    while (U >= F(j)) % check that U is in A_i, i.e.
          % F(i-1) <= U <   F(i);
          % the while loop ends when U < F(i)
       F(j) = F(j) + exp(-lambda) * lambda^i/gamma(i+1); % new value
       % of F
       i = i + 1; % count the values in A_i
    end
    X(j) = i; % one Poisson variable
end
Y = sum(X); % N Poisson variables

% Compare it to the P(lambda) distribution.
k = 0 : 27; % the relevant values of a Poisson distr.(those with % probability > 0);
otherwise, there is an infinite number of
% values
p_k = poisspdf(k,lambda); % the probabilities of a Poisson
% distr.
U_X = unique(X); % the values of X listed ONLY ONCE, no
% repetitions
n_X = hist(X,length(U_X)); % the frequency of each value in UX
rel_freq = n_X/N; % the relative freq. n_X/N approximates the
% probability
```

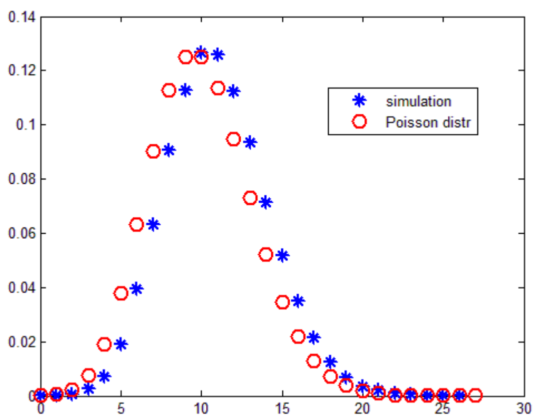

Figure 3. Poisson distribution simulation, N = 10000 simulations

4. Conclusions

We have described procedures and algorithms for computer simulations of some common discrete random variables. With the use of a Standard Uniform random variable, any distribution can be generated. Then we used these simulations to make inferences about each distribution, namely to estimate probabilities, i.e., we used Monte Carlo methods to approximate probabilities. As all the examples show, with a relatively moderate number of simulations, $N = 10000$ (moderate, in the sense that for some processes millions such simulations may be required), the results are very good, the approximations quite accurate.

References

[1] O. Agratini, P.Blaga, Gh. Coman, *Lectures on Wavelets, Numerical Methods and Statistics*, Casa Cărţii de Ştiinţă, Cluj-Napoca, 2005.
[2] C. Andrieu,, A. Doucet, R. Holenstein, *Particle Markov chain Monte Carlo methods,* J. Royal. Statist. Soc. B. Vol. 72, Part 3, 2010, 269–342.
[3] M. Baron, *Probability and Statistics for Computer Scientists, 2nd Edition*, CRC Press, Taylor & Francis Group, Boca Raton, FL, USA, 2014**.**
[4] Y. Chen, P. Diaconis, S. P. Holmes, J. S. Liu, *Sequential Monte Carlo Methods for Statistical Analysis of Tables*, J. American Statistical Assoc., Vol. 100(469) 2005, Theory and Methods, 109 – 120, doi:10.1198/016214504000001303.
[5] W. Feller, *An introduction to probability theory and its applications*, Vol. I-II, John Wiley, New York, 1957, 1966.
[6] H. Lisei, S. Micula, A. Soos, *Probability Theory trough Problems and Applications*, Cluj University Press, 2006.
[7] S. Micula, *Probability and Statistics for Computational Sciences*, Cluj University Press, 2009.
[8] J.S. Milton, J. C. Arnold, *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences, 3rd Edition*. McGraw-Hill, New York, 1995.
[9] R. T. Trîmbiţaş, *Metode statistice*, Cluj University Press, 2000.
[10] http://www.mathworks.com/help/matlab/.