Homework4 - Gaussian Distribution,Uniform Distribution,Box-Müller transformation

Professor : Ren Jung Chang 張仁宗
Student : N16066176 Chen Wei Yu 余振瑋

Oct 14 2017

Abstract

The homework 4 is designed for disgussing the uniform distribution of an random sequences with the mean and standard deviation. And how to use the transformation technique to generate the exponential deviate or the normal deviate from the uniform distribution sequences.

List of Figures

1 Homework Problem

1.1 Prblem 1

1.1.1 Problem

Use Matlab to generate Gaussian x_i with specified mean μ_x and the standard deviation σ_x, and plot time series, distribution diagram, and histogram.

1.1.2 Answer

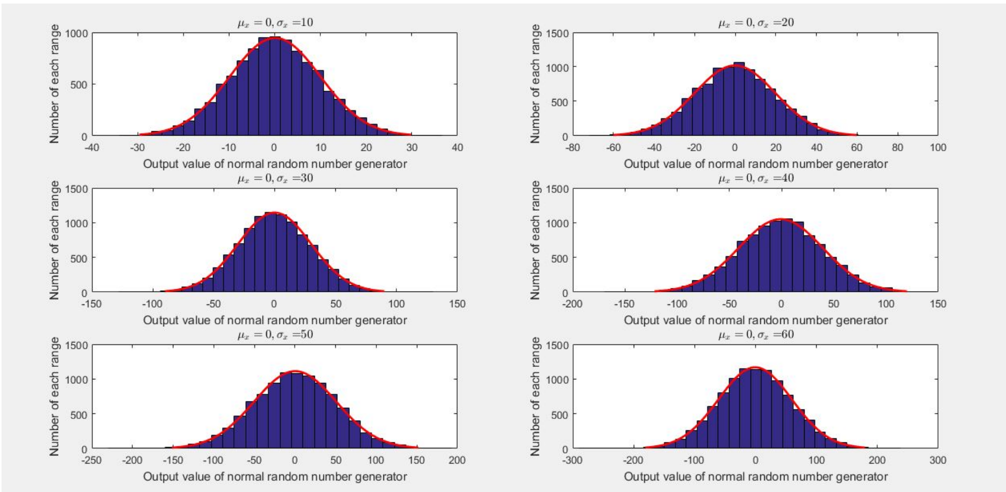

Figure 1: Histogram of normrnd

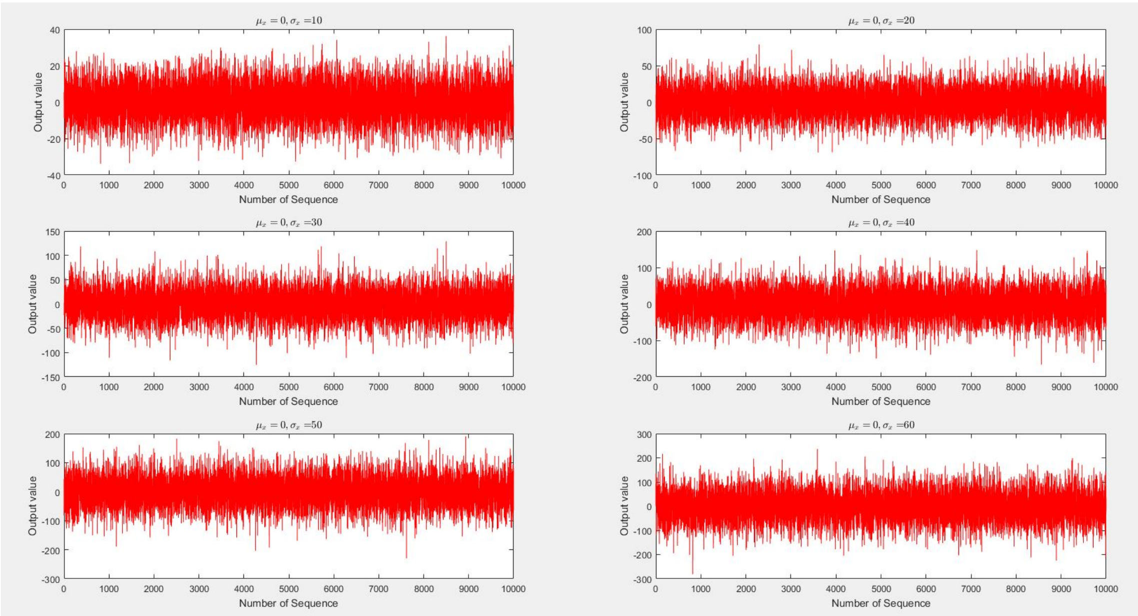

Figure 2: Sequence with normrnd

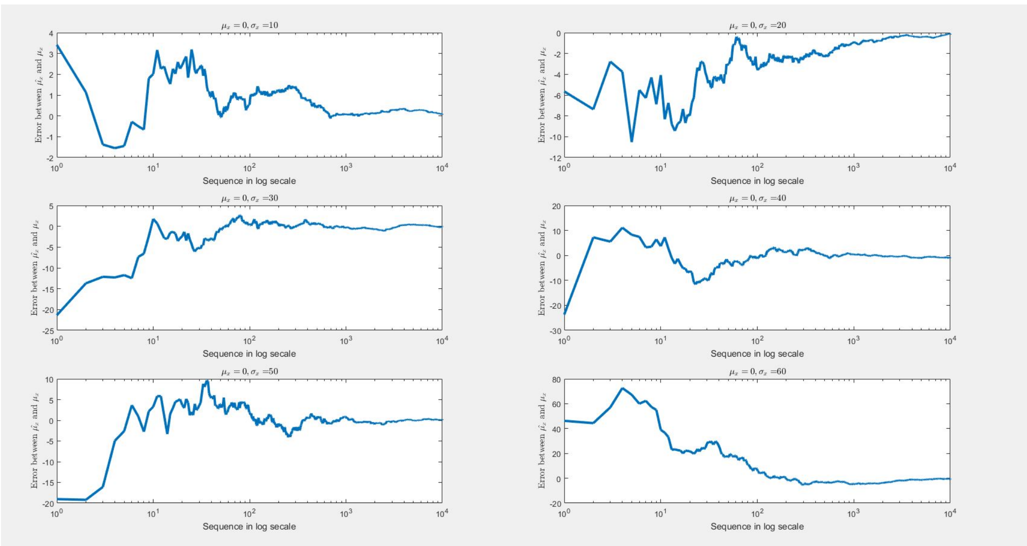

Figure 3: Error between estimated mean and true mean with different sequence in semilog diagram

1.2 Problem 2

1.2.1 Problem

Use Box-Müller transformation to generate two independent Gaussian sequences through input two independent uniform distributions generated by yourself.

1.2.2 Answer

There are two classical ways to implement the Box-Müller method[1][2]. One way is using two independent variables $x_1 \sim U[0,1] \, and \, x_2 \sim U[0,1]$ with the trigonometric operations in the Cartesian cordinate, the other is based on the previous method but changes the cordinate to the polar cordinate.

To understand the tranformation method, we discuss the "exponential deviate" at first.Consider $x \sim U[0,1]$ and y is an exponential distribution in $[0,1]$. The probability density function of y is defined as

$$g(y) = \{ \begin{array}{c} 0 \text{ , y<0} \\ \lambda e^{-\lambda y}, y \geq 0. \end{array} \tag{1}$$

By the fundamental transformation law of probabilities, we can get

$$x = G(y) = \int_{\infty}^{y} g(u)du = 1 - e^{-\lambda y} \tag{2}$$

so we can get the exponential deviates of y by inserting $x \sim U[0,1]$ to

$$y = -ln|1-x|. \tag{3}$$

The transformation can be applied on more than one dimension. If $x_1, x_2 \cdots$ are random deviates with a joint probability distribution $p(x_1, x_2, \cdots)dx_1 dx_2 \cdots$, and if $y_1, y_2, \cdots$ are each functions of all the x's(same number of as x's),then the joint probability distribution of the y's is

$$p(y_1, y_2, \cdots)dy_1 dy_2 \cdots = p(x_1, x_2, \cdots)|\frac{\partial(x_1, x_2, \cdots)}{\partial(y_1, y_2, \cdots)}|dy_1 dy_2 \cdots \tag{4}$$

where $\frac{\partial()}{\partial()}$ is the Jacobian determinant of the x's with respect to the y's.

The transformation is the spirit behind the Box-Müller method. The other descirptions are same as the hint of homework.

Figure 6 shows that the two random variables will be form an cirle region which is same as the assumption of the modified Box-Müller method.

Figure 17∼26 show that the estimated mean will converge to the ture mean no metter the different mean values or the different the standard deviation values.

Input: μ, σ
Result: Return two normal deviate random numbers
RSQ=0;(RSQ : square root of square sum)
while *RSQ=0 or RSQ≥1* **do**
  let $z_1\, and\, z_2 \sim U[-1, 1]$ which is in the unit circle
    $z_1 = rand() * 2 - 1.0$
    $z_2 = rand() * 2 - 1.0$
    RSQ=$z_1^2 + z_2^2$
    FAC=$\sqrt{\|(-2 \times ln(RSQ)/RSQ)\|}$
  if *(RSQ* $\neq 0$*) and (RSQ<1)* **then**
    break
  end
end
$x_1 = z_1 \times FAC * \sigma + \mu$
$x_2 = z_2 \times FAC * \sigma + \mu$
Output: x_1, x_2

Algorithm 1: BoxMüller method in the polar form

Input: μ, σ
Result: Return two normal deviate random numbers
R_1=0;(R_1 : first uniform random nomber)
while *R_1=0* **do**
  let $R_1 \sim U[0, 1]\, but \neq 0$
    $R_1 = rand()$
  if $R_1 \neq 0$ **then**
    break
  end
end
$R_2 = rand$
$Left = \sqrt{-2 * ln(R1)}$
$x_1 = Left \times \cos(2 \times \pi \times R_2) \times \sigma + \mu$
$x_2 = Left \times \sin(2 \times \pi \times R_2) \times \sigma + \mu$
Output: x_1, x_2

Algorithm 2: BoxMüller method with trigonometric operation

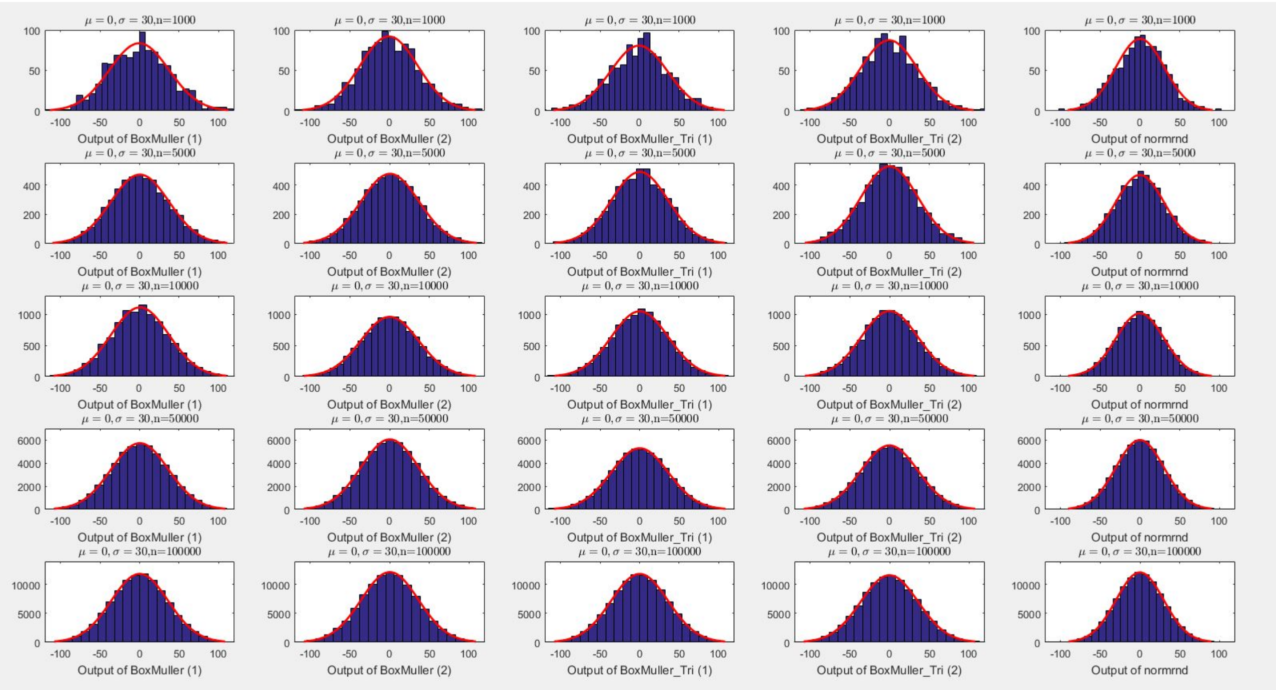

Figure 4: The first two columns are the random numbers generated by the BoxMüller method in polar form. The third and fourth column are the random numbers generated by the BoxMüller method with the trigonometric operation. Final one is generated by the naive Matlab function normrnd.

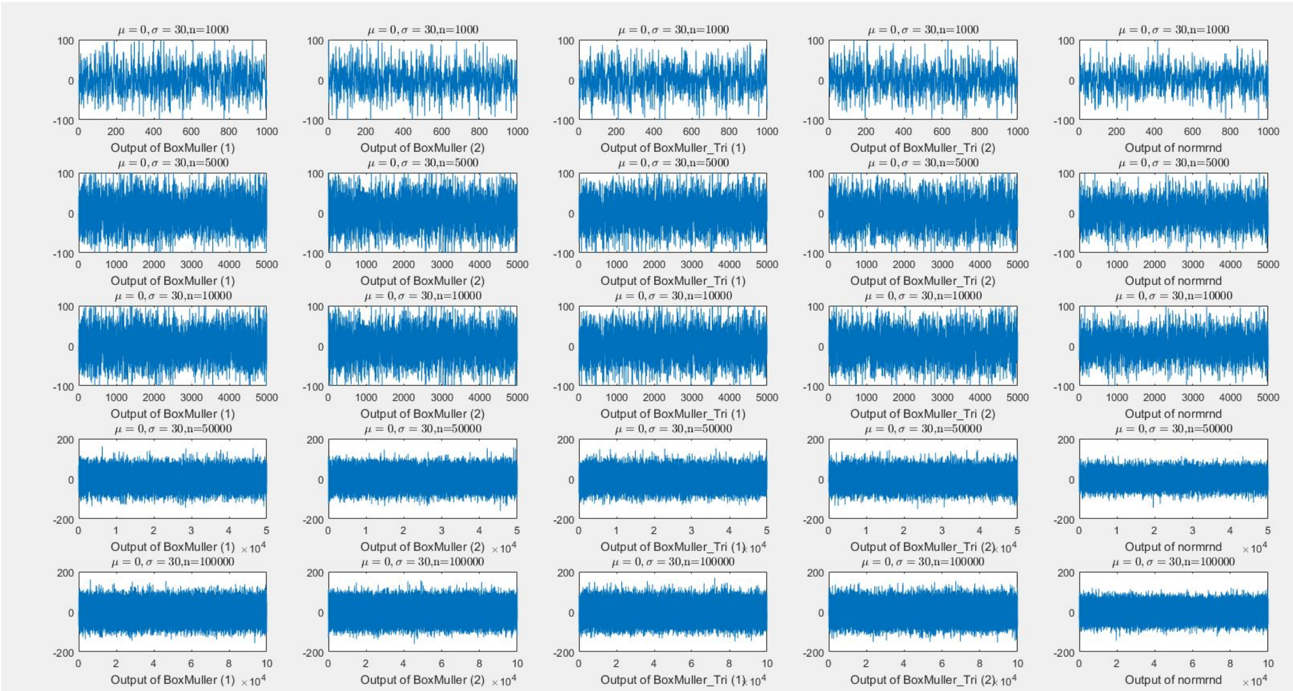

Figure 5: The first two columns are the random numbers generated by the BoxMüller method in polar form with the sequence. The third and fourth column are sequence plot of the random numbers generated by the BoxMüller method with the trigonometric operation. Final one is generated by the naive Matlab function normrnd.

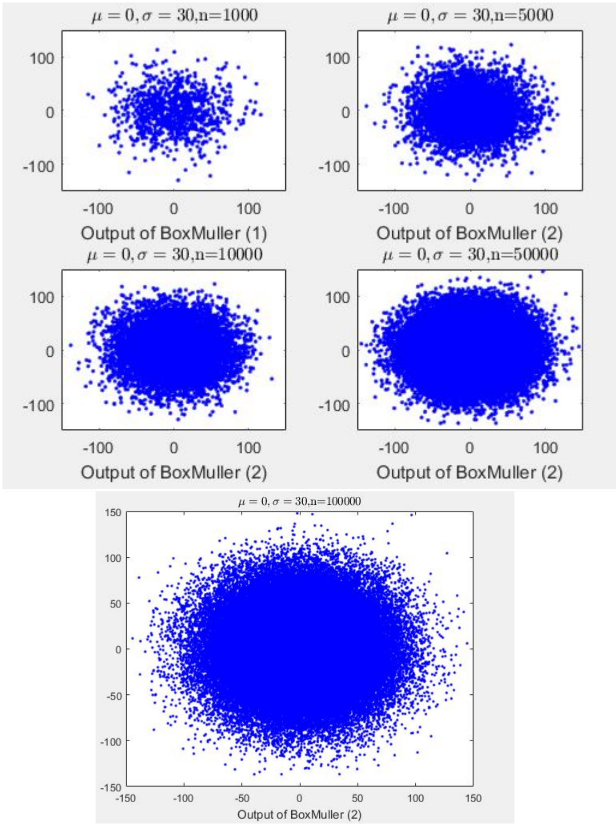

Figure 6: Using the first BoxMüller method shows that the two normal distribution random numbers can be generated from an circle region.

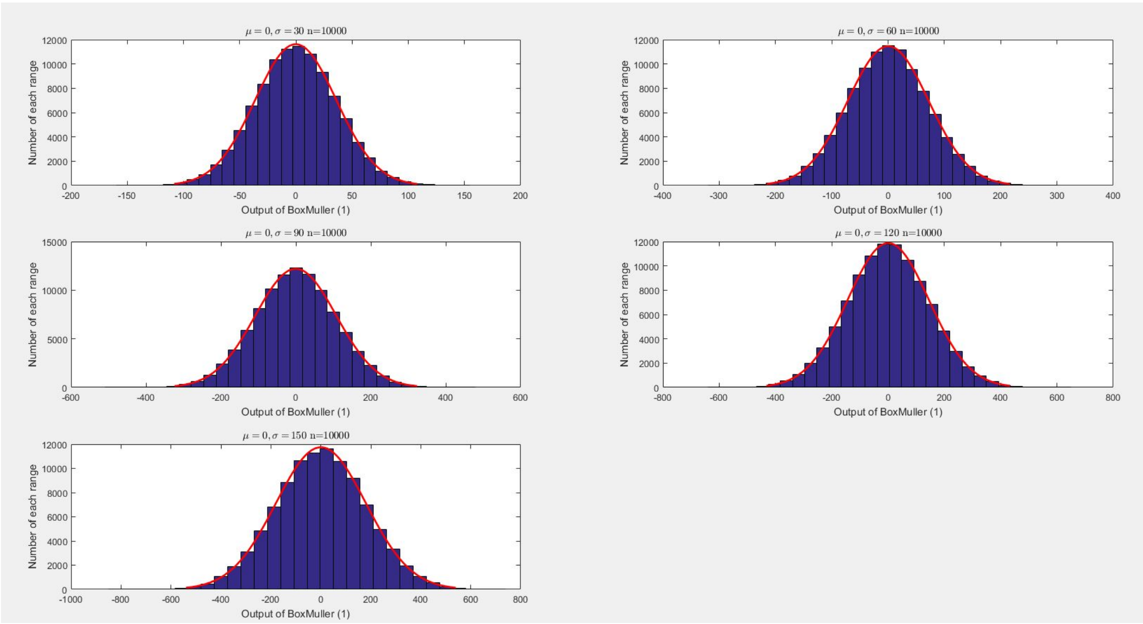

Figure 7: Histogram of first random number by BoxMüller method in polar form with different standard deviations

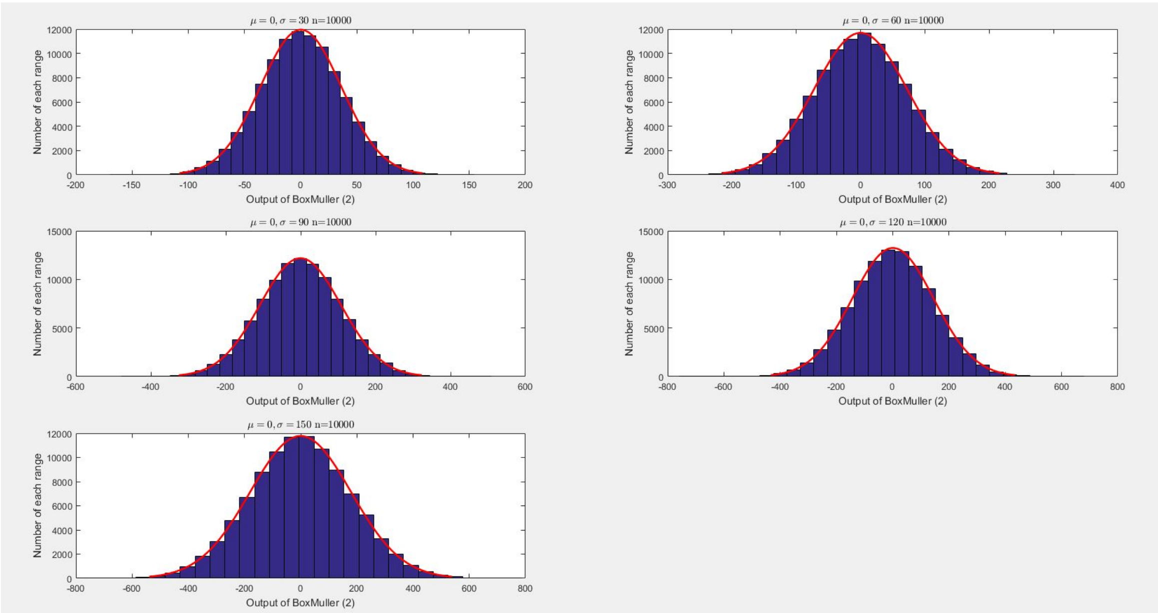

Figure 8: Histogram of second random number by BoxMüller method in polar form with different standard deviations

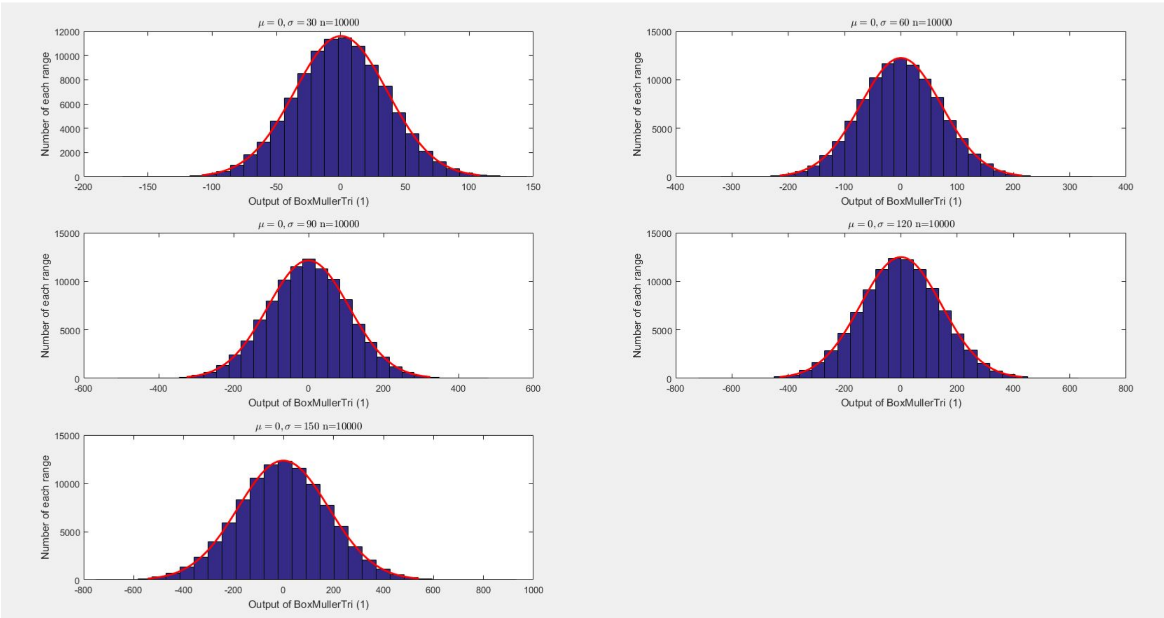

Figure 9: Histogram of first random number by BoxMüller method in trigonometric operation with different standard deviations

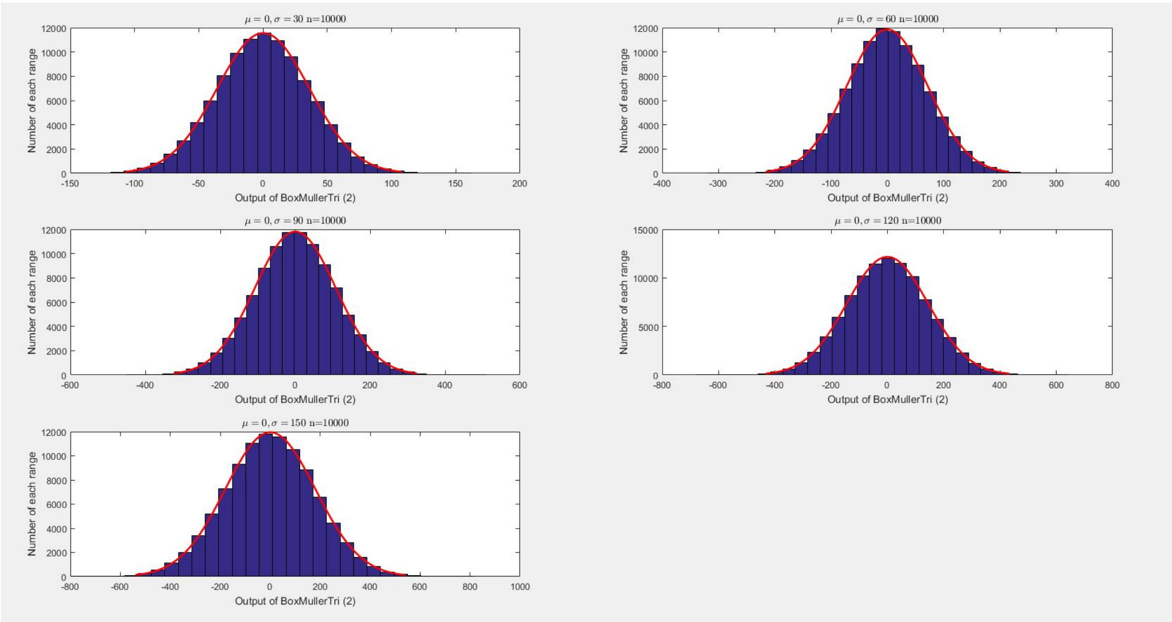

Figure 10: Histogram of second random number by BoxMüller method in trigonometric operation with different standard deviations

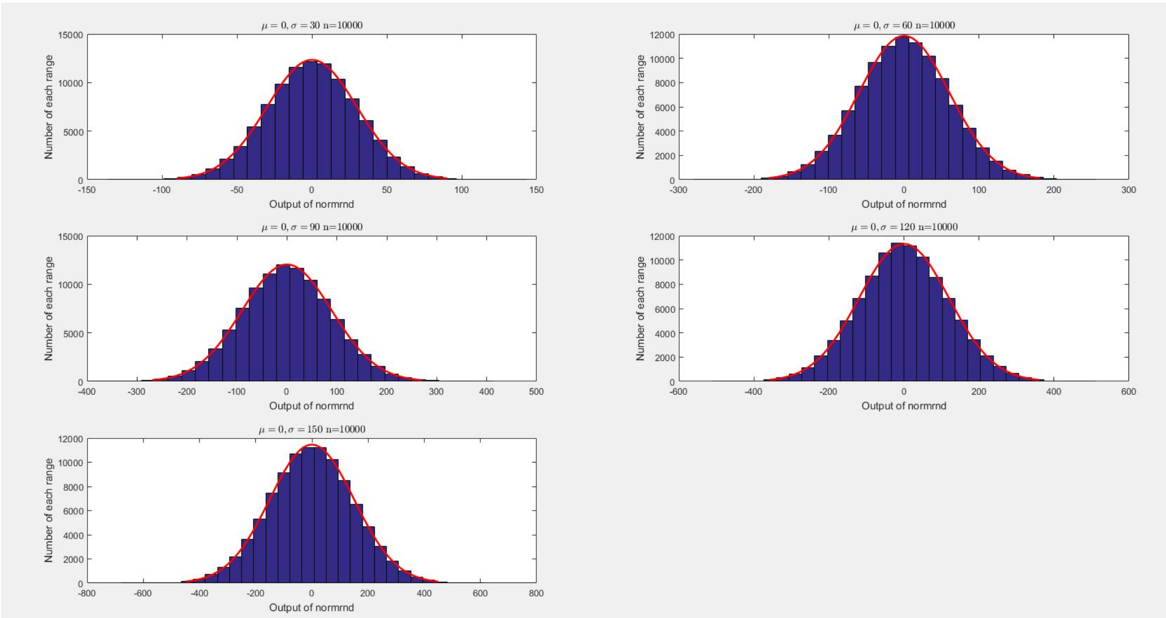

Figure 11: Histogram of random number by naive function normrnd with different standard deviations

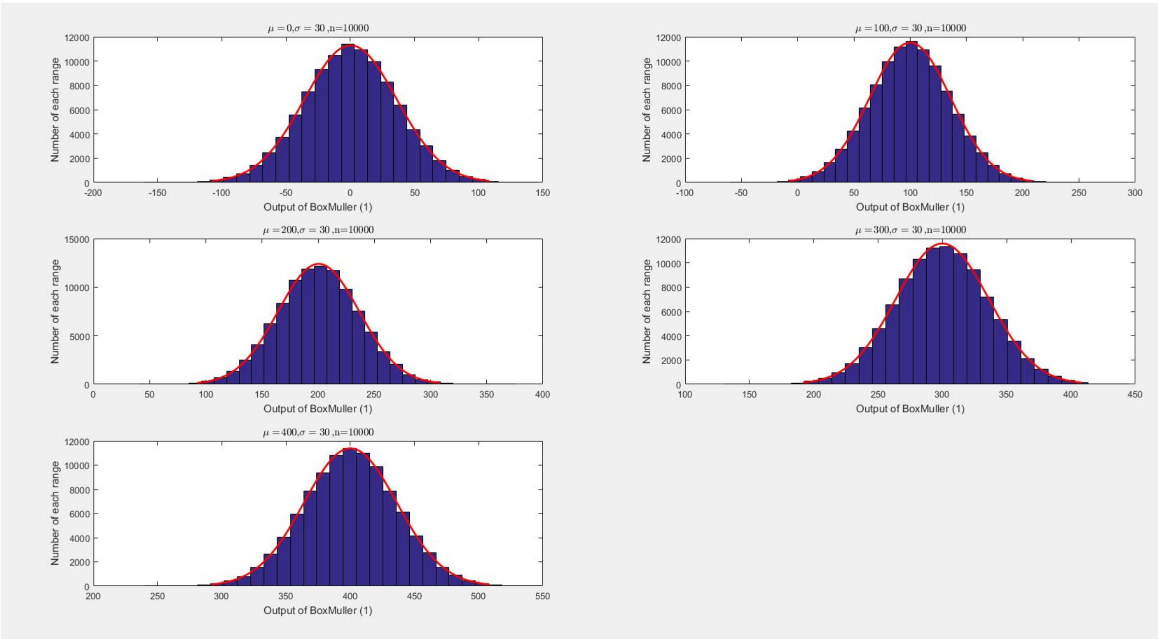

Figure 12: Histogram of first random number by BoxMüller method in polar form with different means

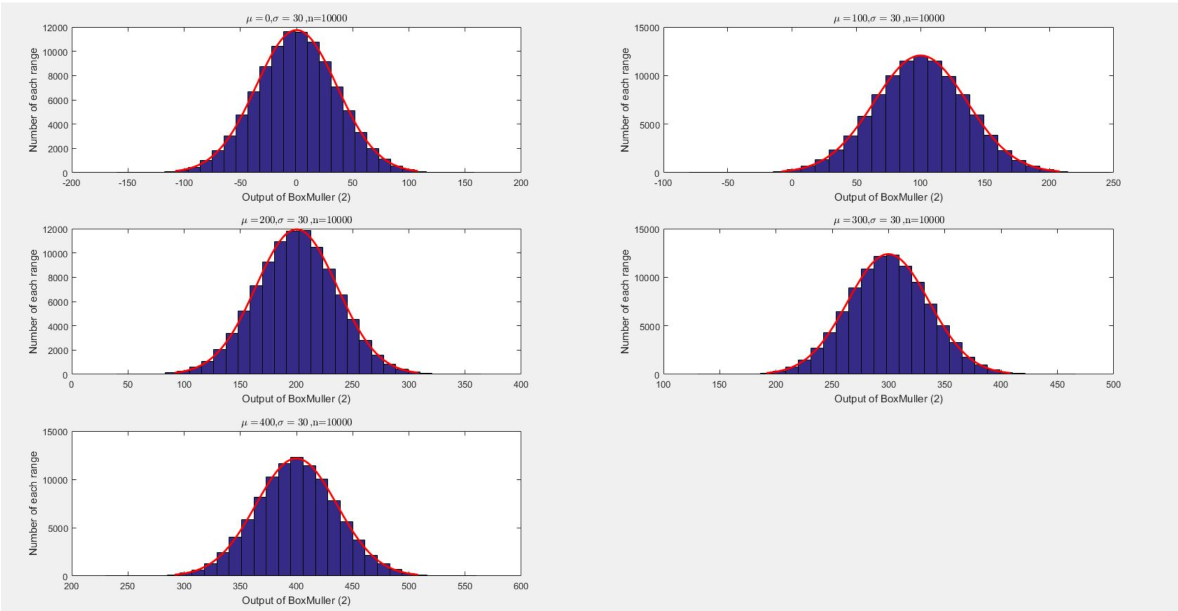

Figure 13: Histogram of second random number by BoxMüller method in polar form with different means

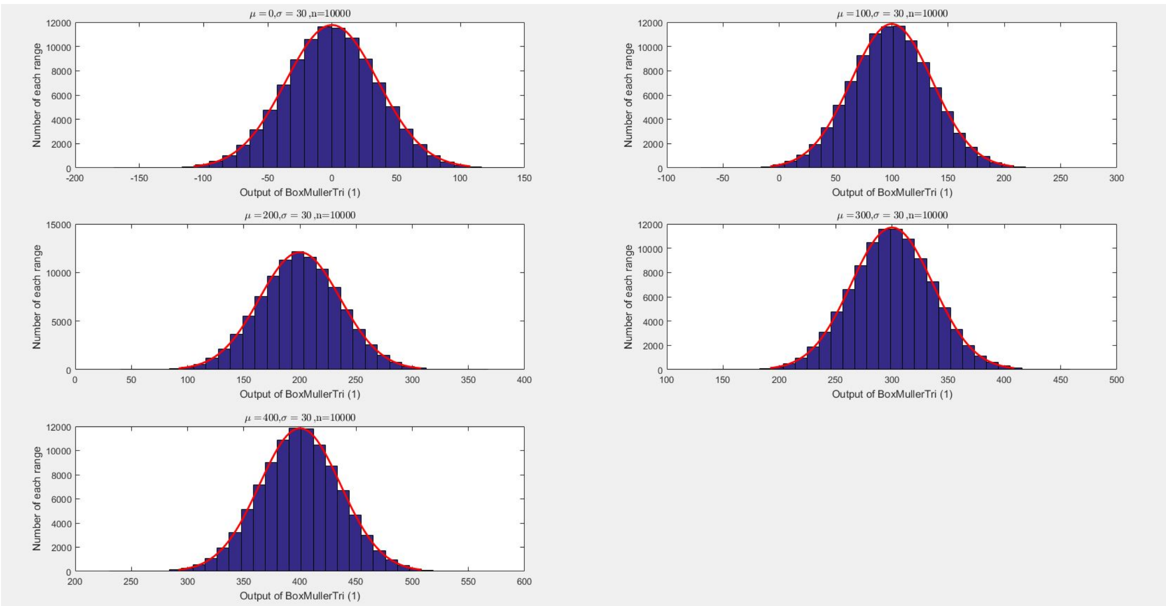

Figure 14: Histogram of first random number by BoxMüller method in trigonometric operation with different means

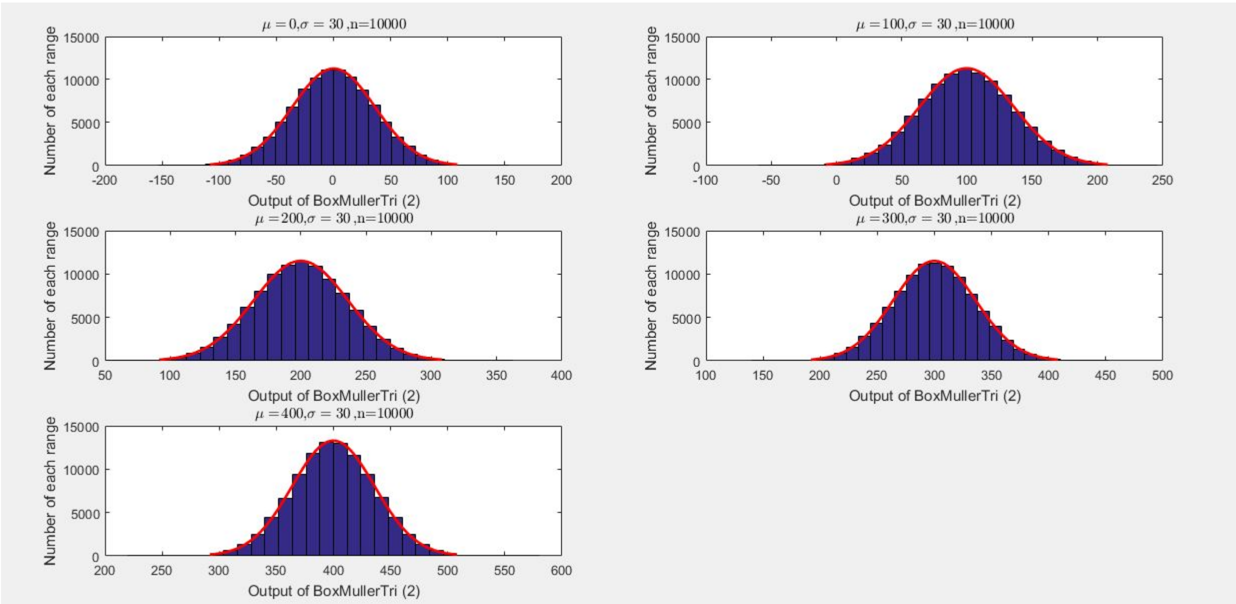

Figure 15: Histogram of second random number by BoxMüller method in trigonometric operation with different means

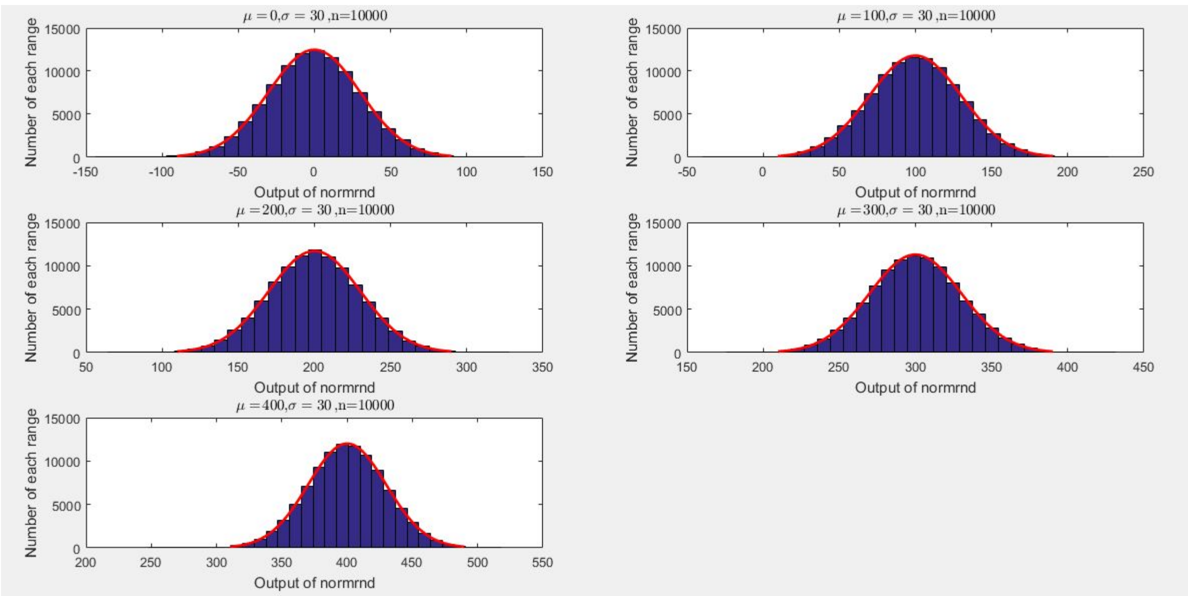

Figure 16: Histogram of random number by naive function normrnd with different means

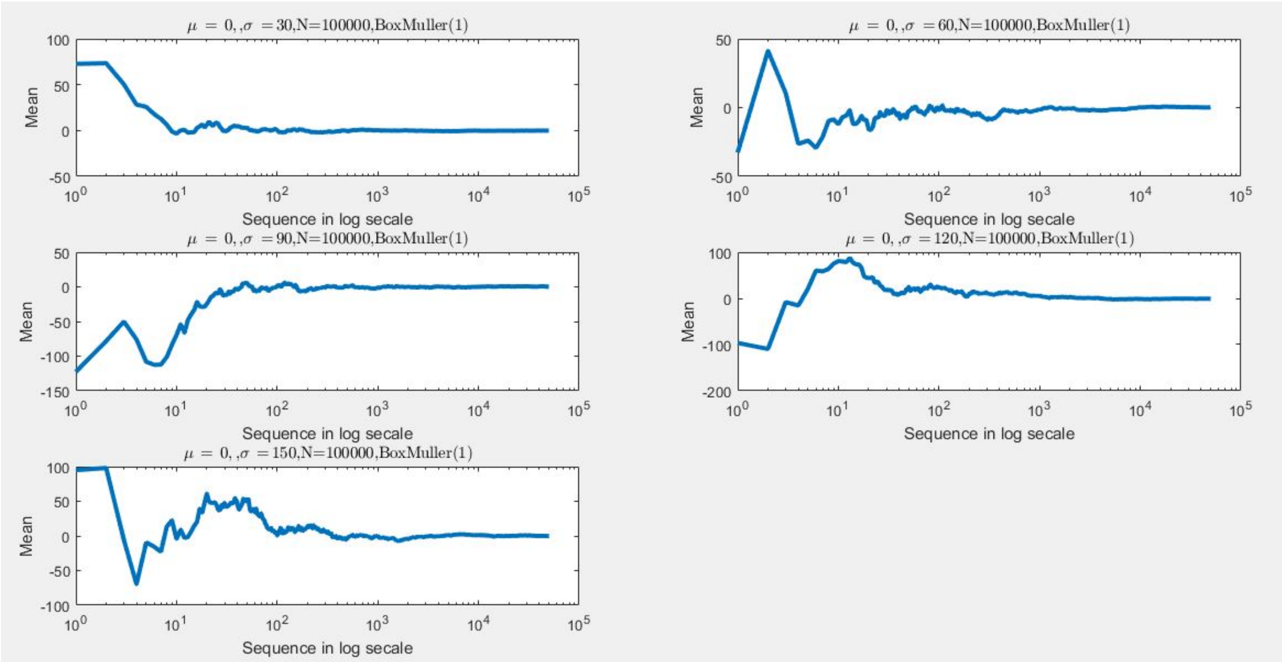

Figure 17: Mean of first random number by BoxMüller method in polar form with different deviations in semilog scale

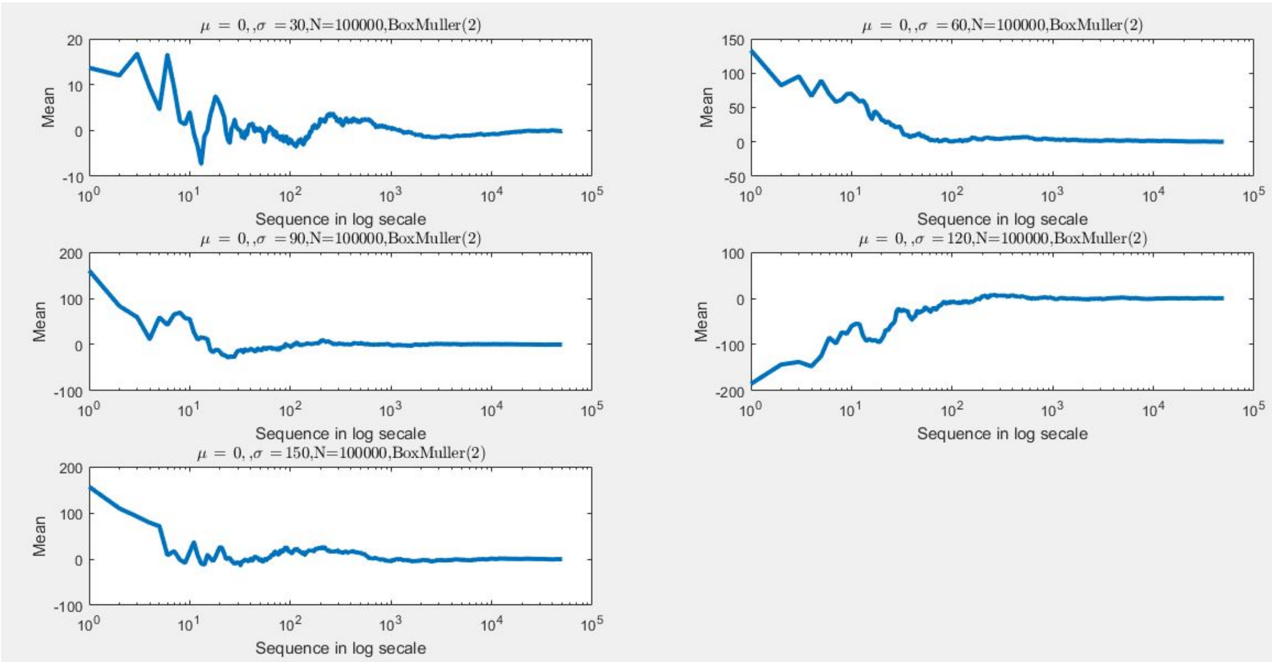

Figure 18: Mean of second random number by BoxMüller method in polar form with different deviations in semilog scale

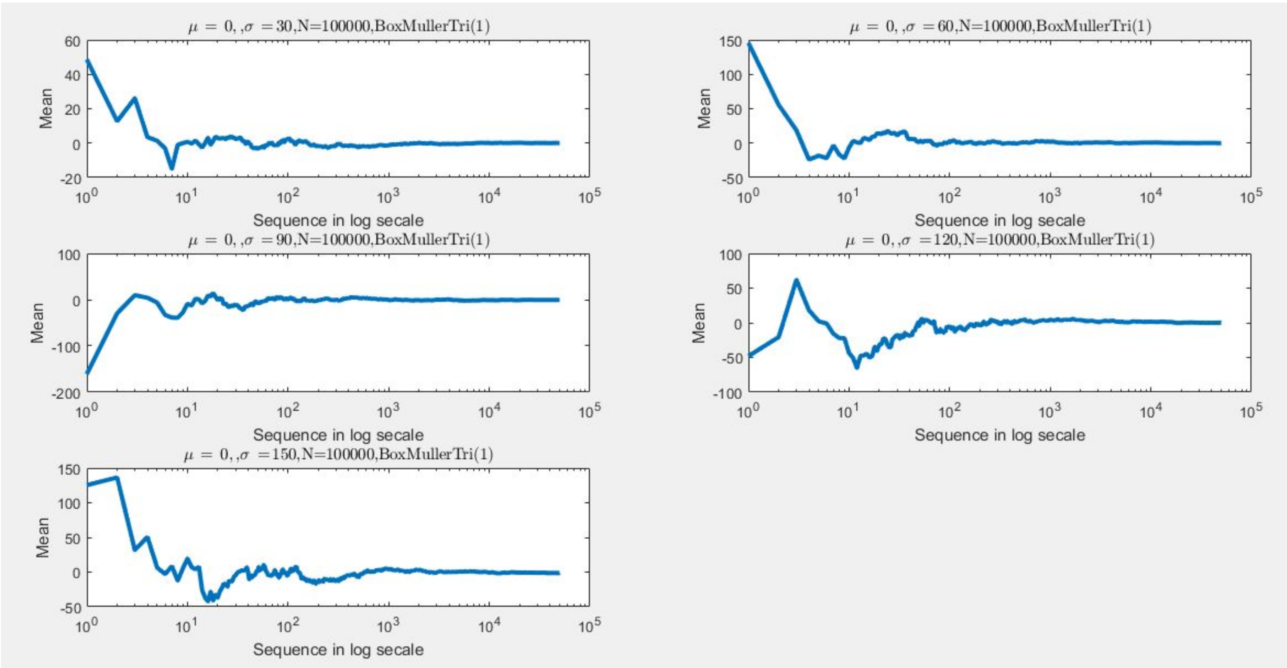

Figure 19: Mean of first random number by BoxMüller method in trigonometric operation with different standard deviations in semilog scale

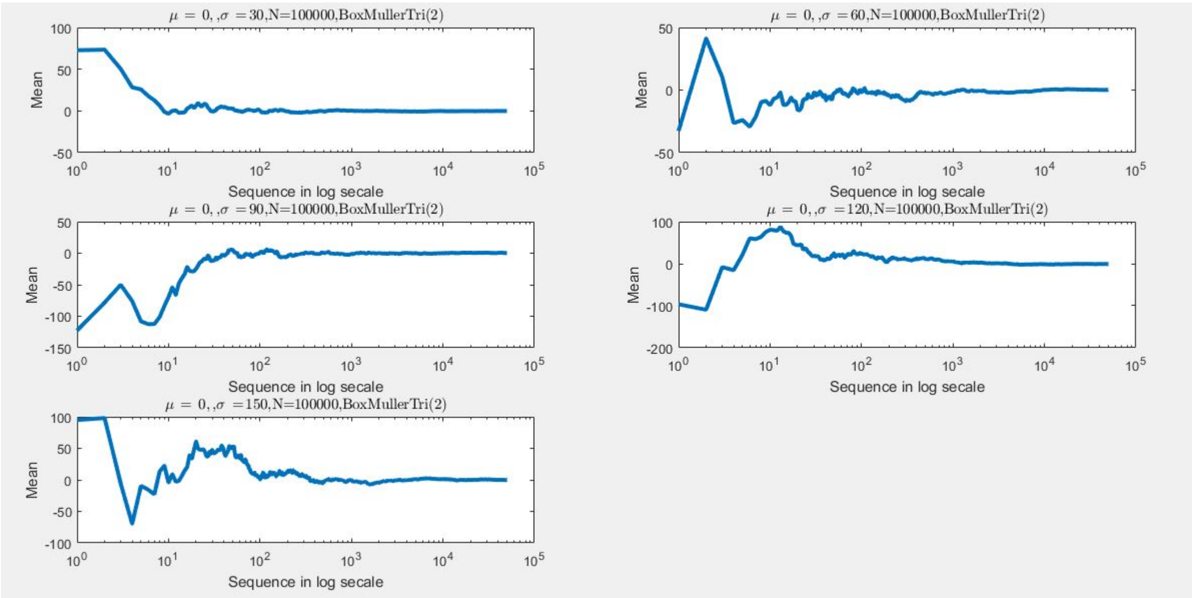

Figure 20: Mean of second random number by BoxMüller method in trigonometric operation with different standard deviations in semilog scale

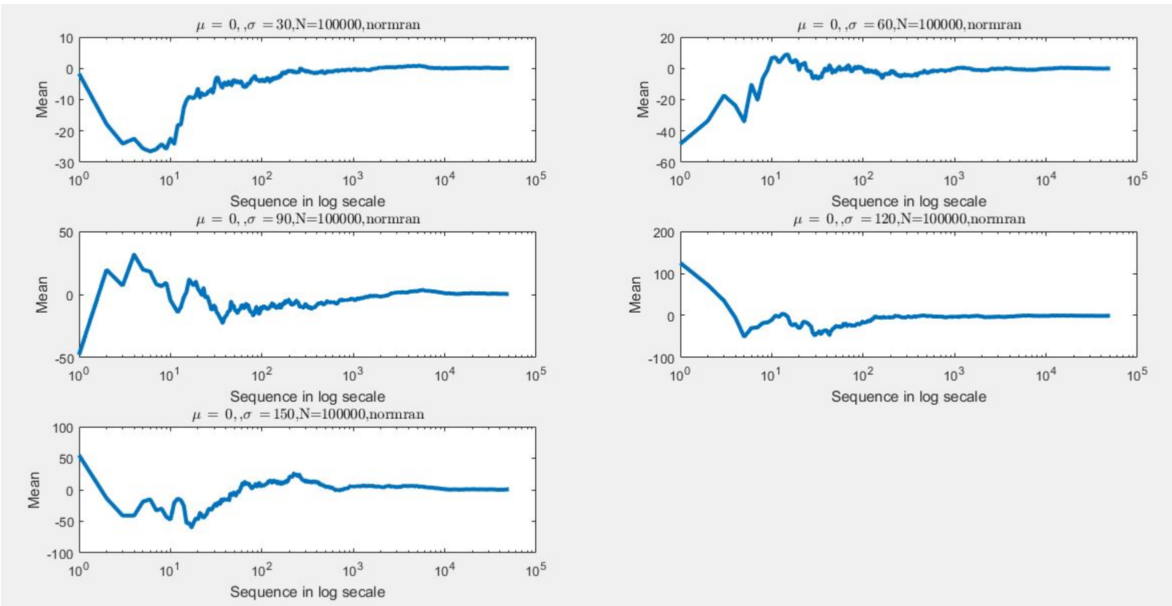

Figure 21: Mean of random number by naive function normrnd with different standard deviations

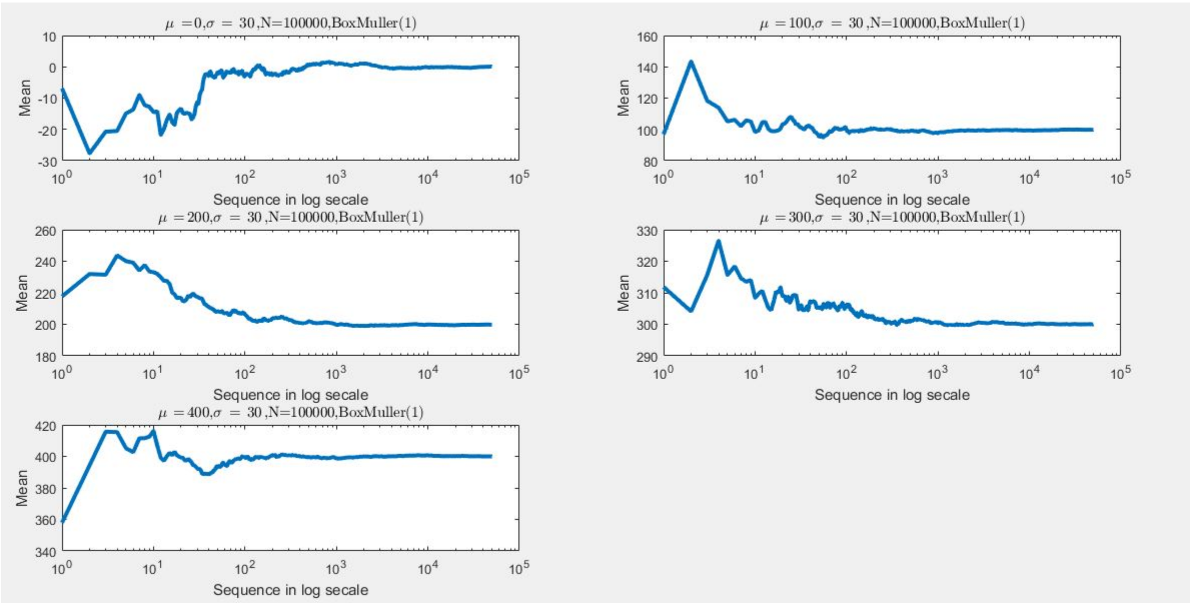

Figure 22: Mean of first random number by BoxMüller method in polar form with different means

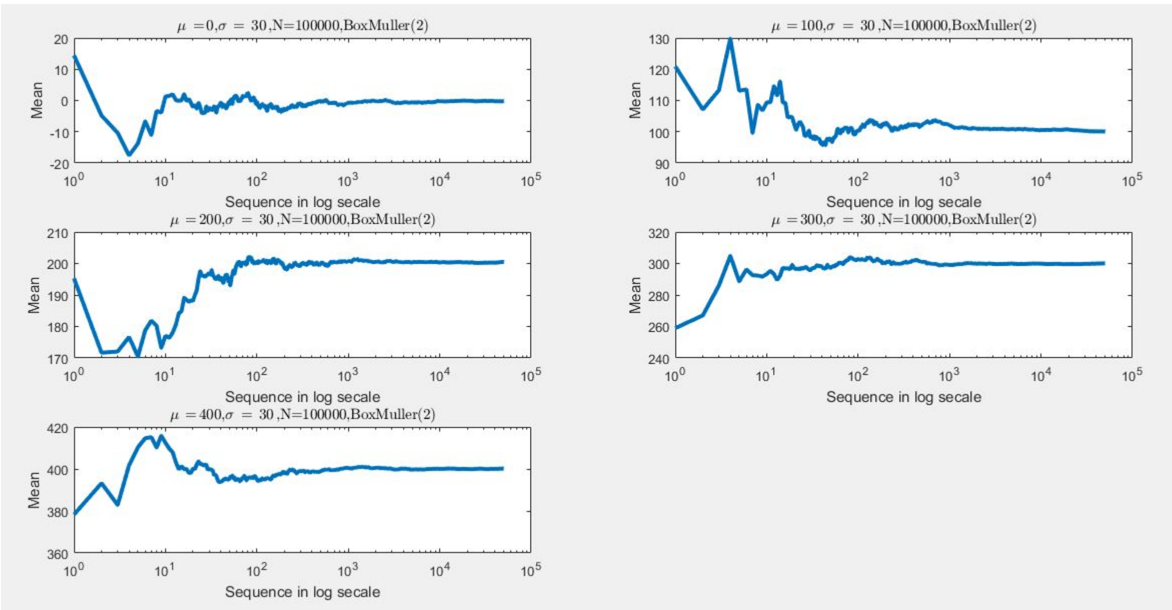

Figure 23: Mean of second random number by BoxMüller method in polar form with different means

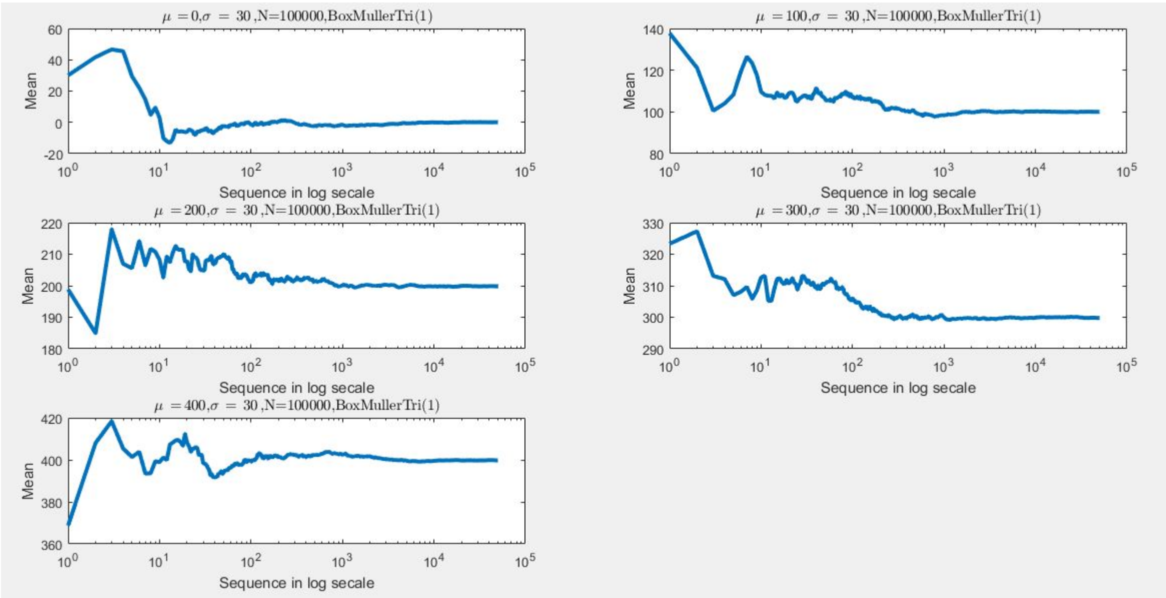

Figure 24: Mean of first random number by BoxMüller method using trigonometric operation with different means

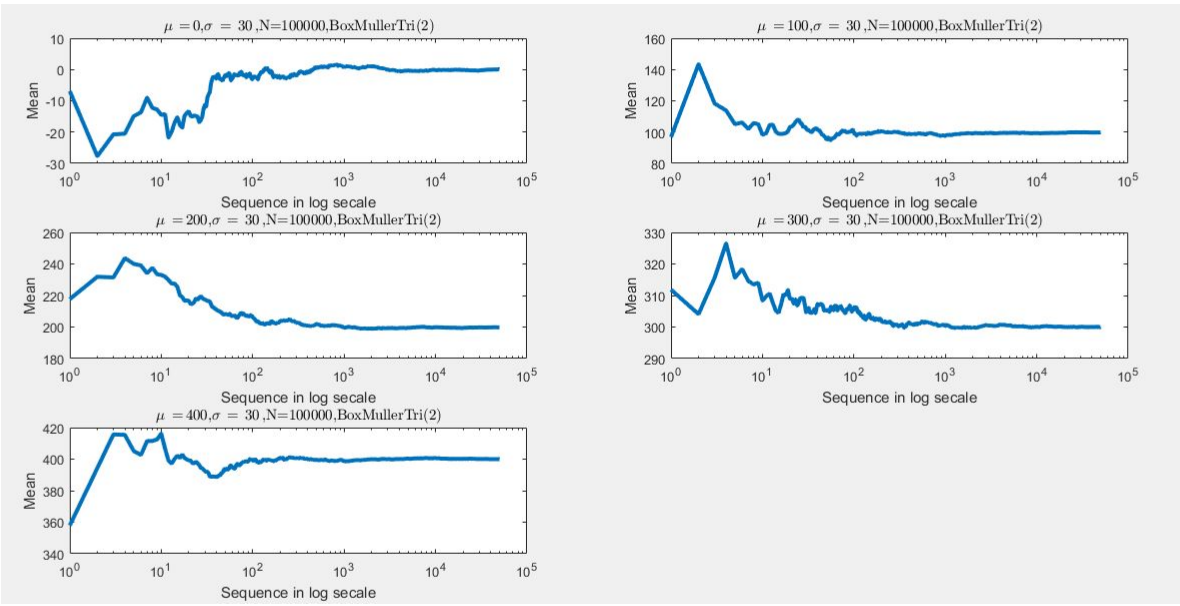

Figure 25: Mean of second random number by BoxMüller method using trigonometric operation with different means

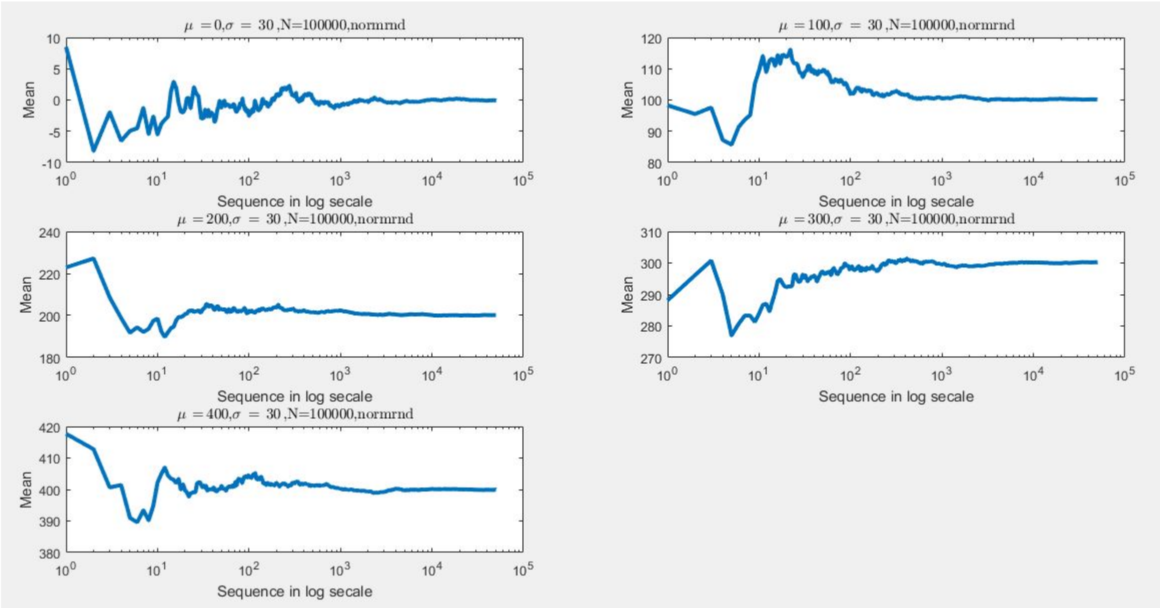

Figure 26: Mean of random number by naive function normrnd with different means

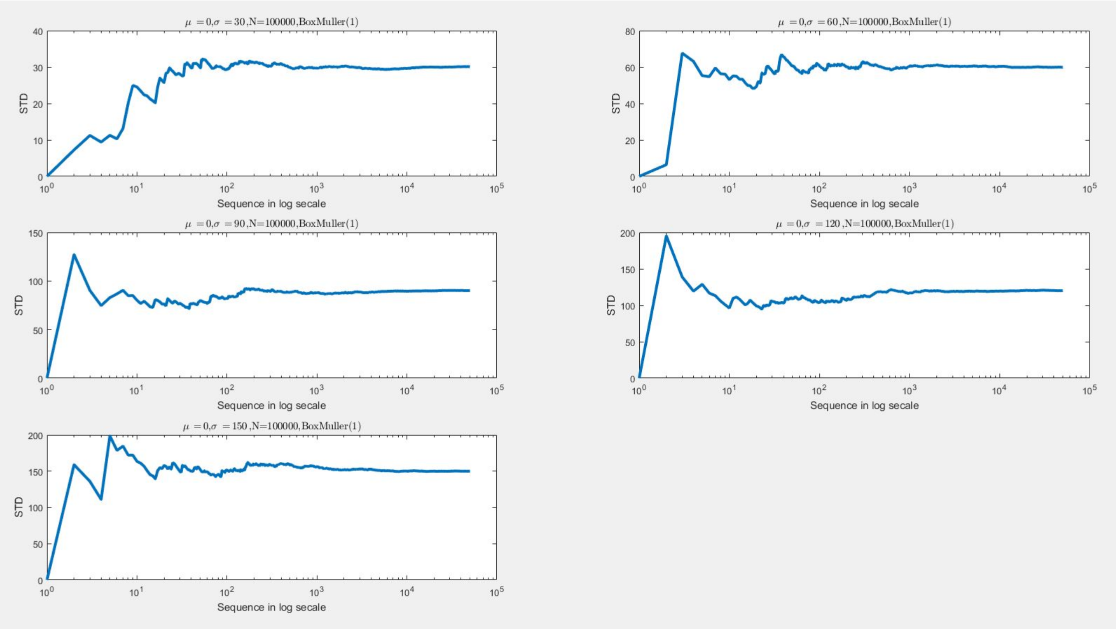

Figure 27: STD of first random number by BoxMüller method in polar form with different STDs

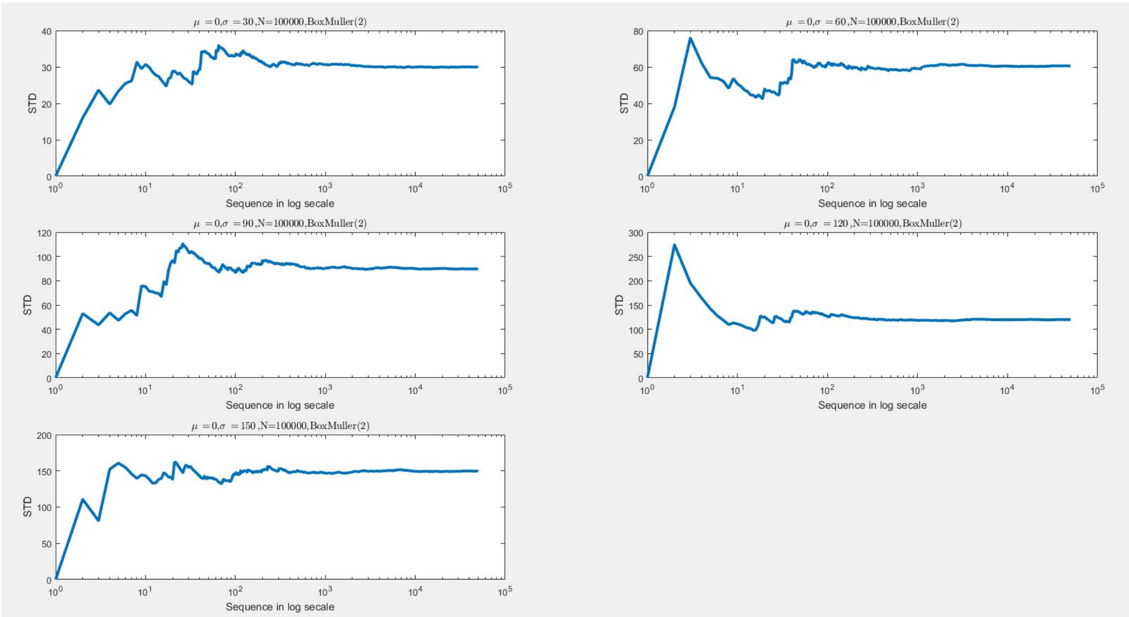

Figure 28: STD of second random number by BoxMüller method in polar form with different STDs

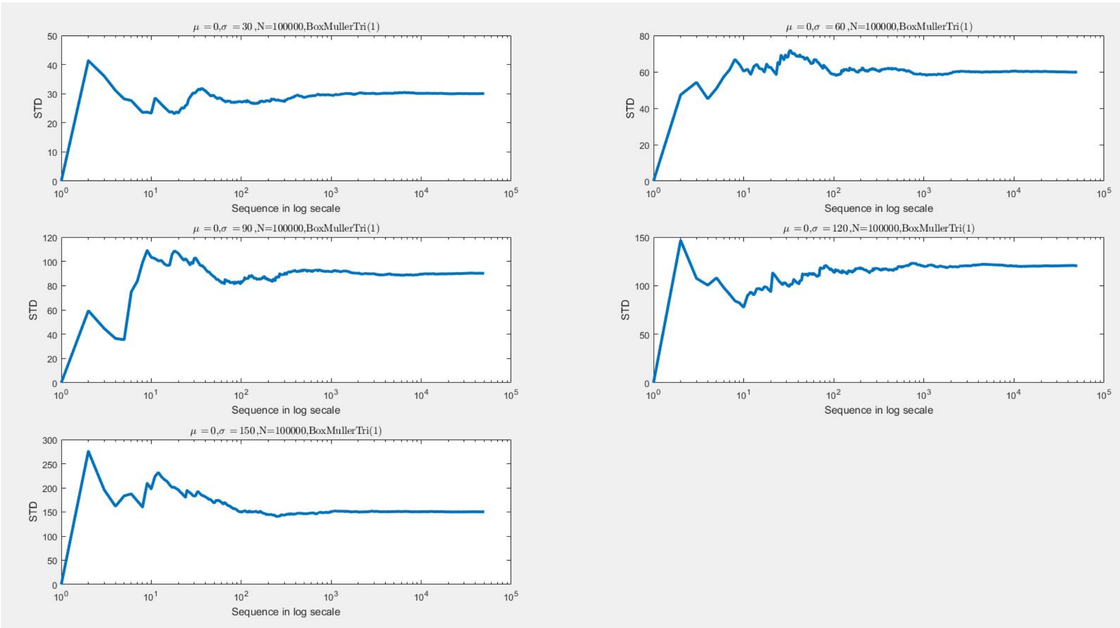

Figure 29: Mean of first random number by BoxMüller method using trigonometric operation with different STDs

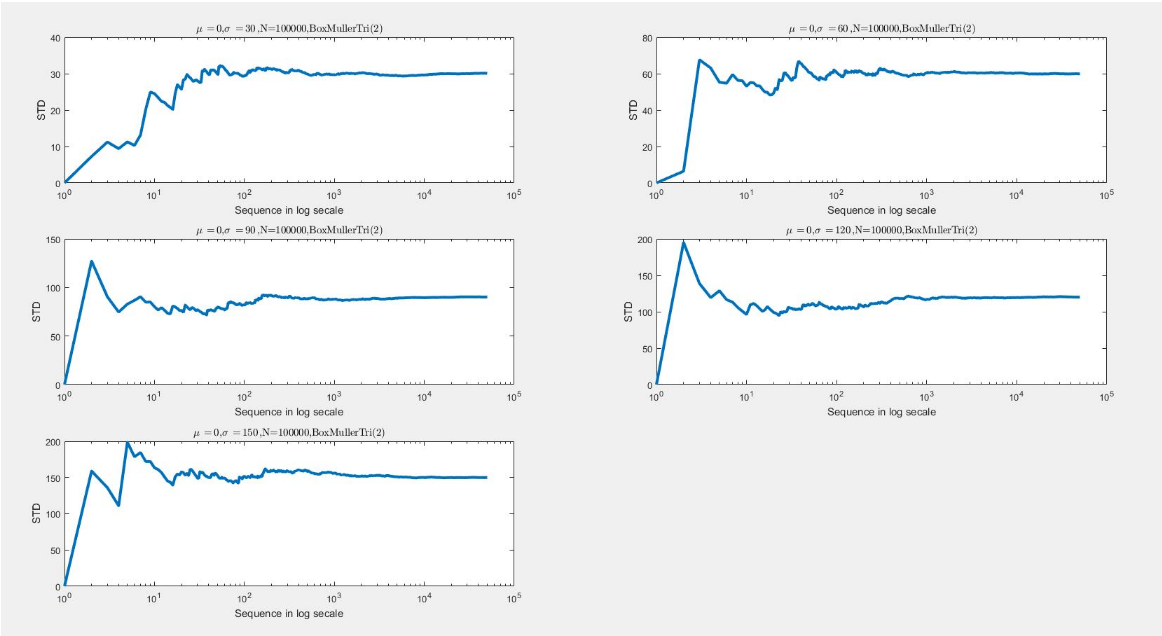

Figure 30: Mean of second random number by BoxMüller method using trigonometric operation with different STDs

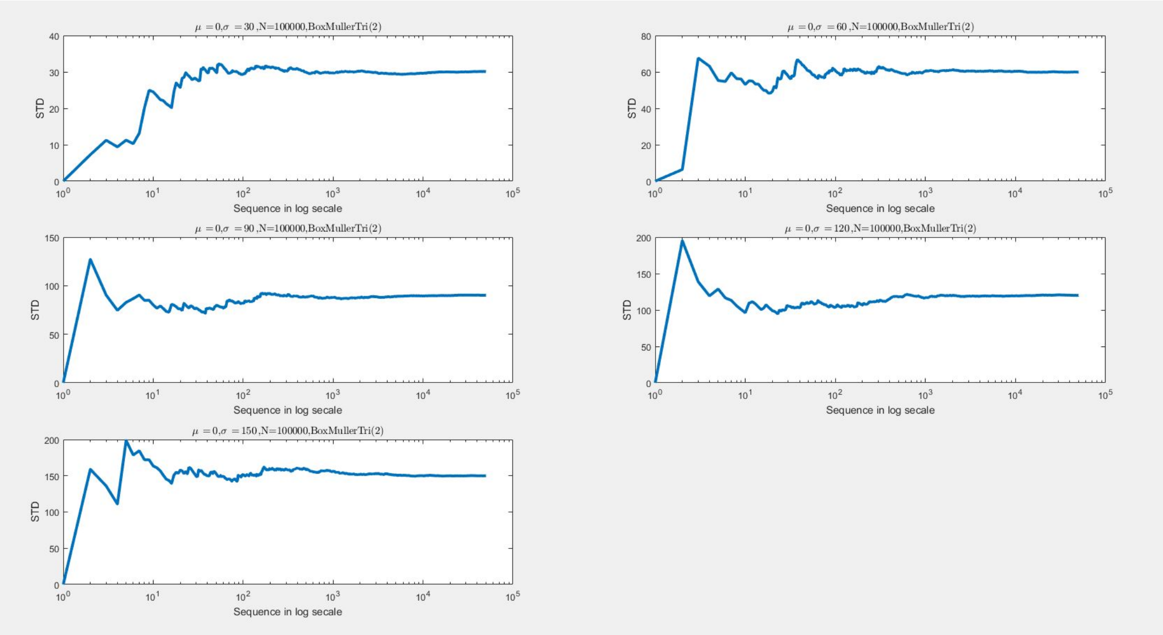

Figure 31: Mean of second random number by naive function normrnd with different STDs

1.3 Problem 3

1.3.1 Problem

Repeat problem 1 and 3, by generating a time series of uniform sequence with specified range.

1.3.2 Answer

The probability density of an continous uniform distribution sequence is defined as

$$f(x) = \{ \begin{array}{c} \frac{1}{b-a} \text{ , for } a \leq x \leq b \\ 0 \text{ , for } x < a \text{ or } x > b. \end{array} \tag{5}$$

[3]

$$\int_a^b f(x)dx = 1 \tag{6}$$

.

With the mean and the standard deviation, we can reform the equation 5 to this form

$$f(x) = \{ \begin{array}{c} \frac{1}{2\sigma\sqrt{3}} \quad , \text{ for } -\sigma\sqrt{3}+\mu \leq x \leq \sigma\sqrt{3}+\mu \\ 0 \ , \text{ for others} \end{array} \tag{7}$$

With the equation 7, we can get equation

$$\sigma = \frac{b-a}{\sqrt{12}}, \mu = \frac{a+b}{2} \tag{8}$$

Input: min,max
Result: Output of the uniform deviate random number in the range of $[min, max]$
DIS=(max-min)/2;
MEAN=(max+min)/2; Caculating the mean of the range.
STD=$(DIS)/\sqrt{3}$; Caculating the standard deviation of the range.
Initialized the variable TMP which $TMP > 1.0$
while $TMP > 1.0$ *or* $TMP < -1.0$ **do**
  TMP=(rand-0.5)/0.5
  OUT=TMP*DIS+MEAN
  if $TMP \leq (DIS + MEAN)$ *and* $TMP \geq (DIS - MEAN)$ **then**
    break
  end
end
Output: OUT

Algorithm 3: Uniform deviate random number generator in the specfied range $U[min, max]$

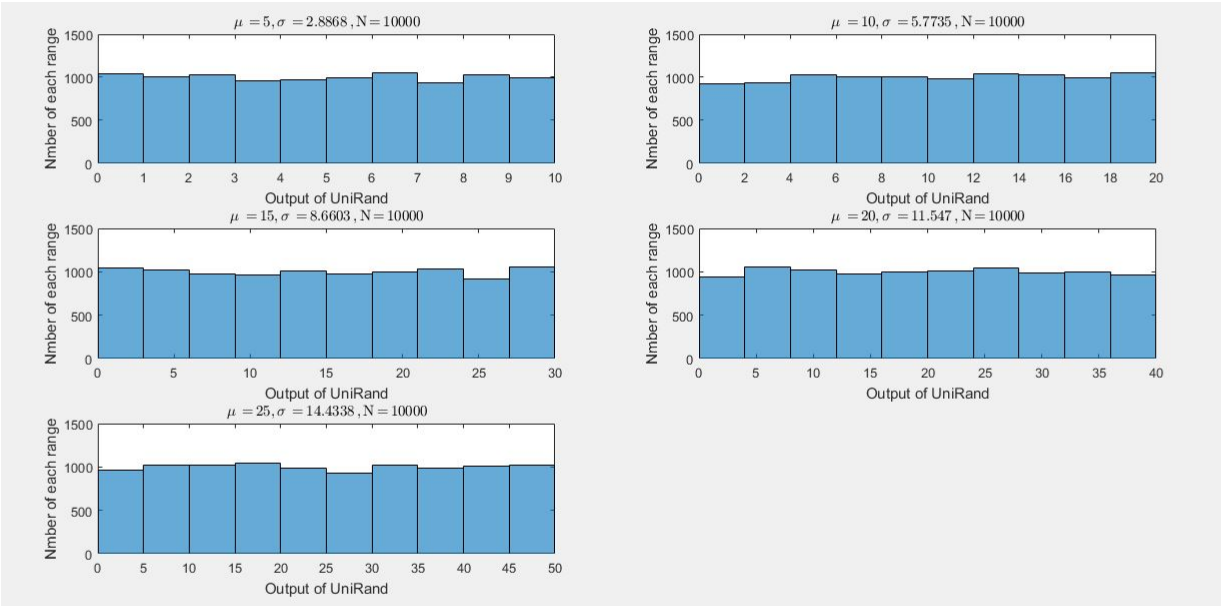

Figure 32: Uniform distribution in specified range

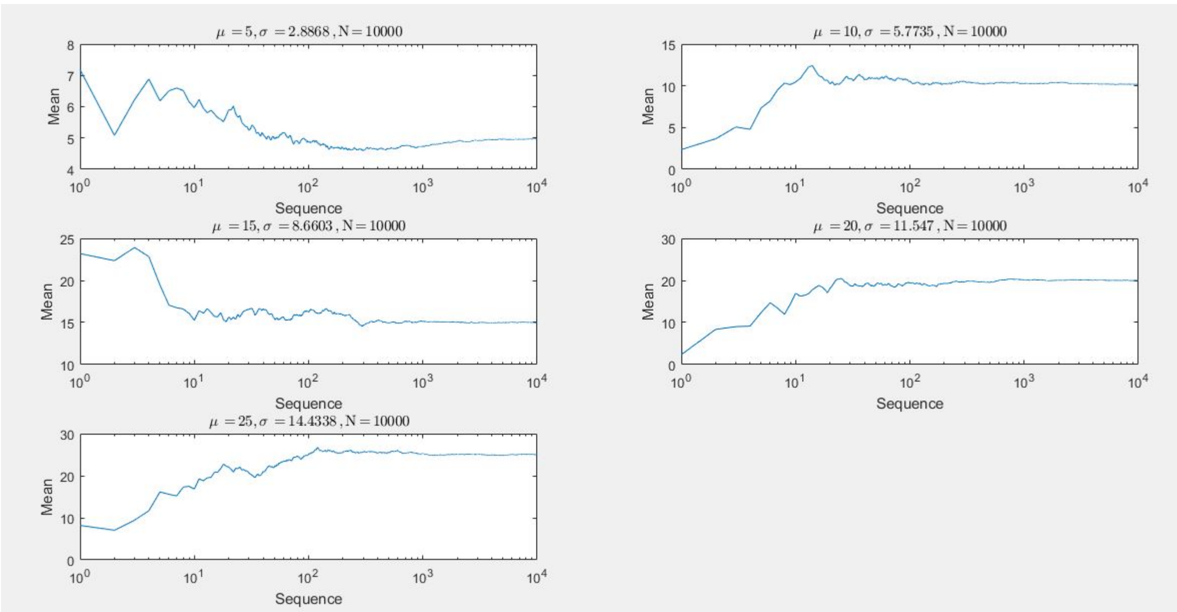

Figure 33: Mean of uniform distribution in spcfied range with different sequence length in semilog scale

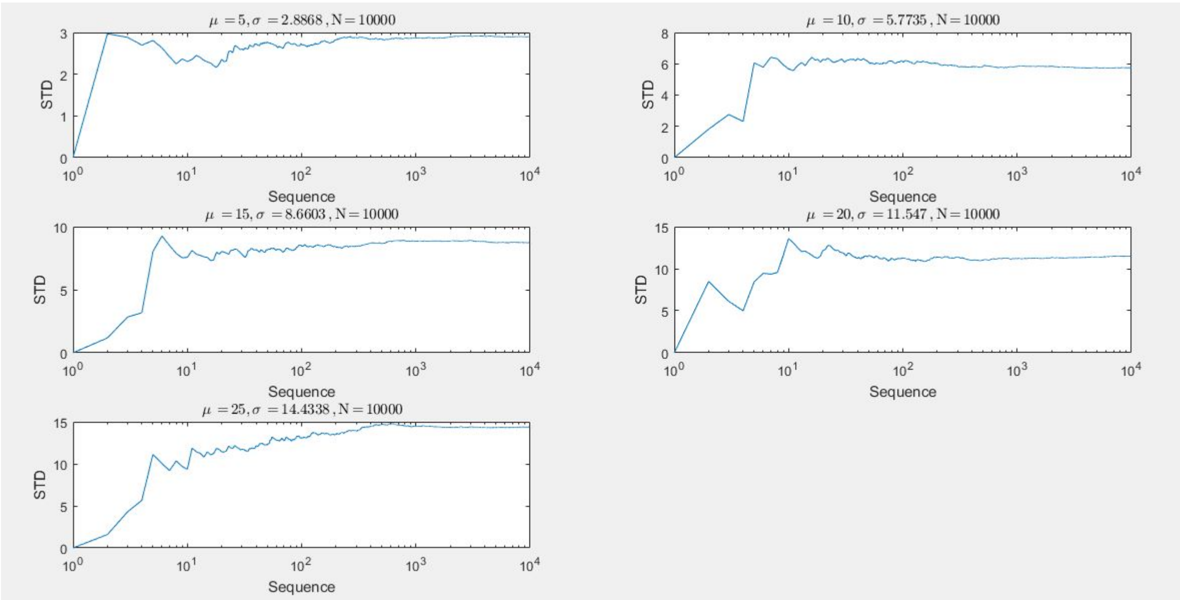

Figure 34: Standard deviation of uniform distribution in spcfied range with different sequence length in semilog scale

From the Figure 17 $\sim$ 31, we can know the Box-Müller can generated the normal deviate random numbers at well. We may use the third moment(skewness) or the fourth moment(kurtosis) to analyze the performance of the output distribution.

2 Code

All of my codes can be found at the Gihub StochasticMatlabCodeHomework4.

- BoxMüller.m : The file is my own function based on the Box-Müller method in polar form.
- BoxMüller_Tri.m : The file is my own function based on the Box-Müller method with trigonometric operations.
- UniRand.m : The file is my own function of uniform distribution random in specified range $U[min, max]$.
- Problem1.m : Figure 1 $\sim$ 3
- Problem2a.m : Figure 4 $\sim$ 6

- Problem2b.m : Figure 7 $\sim$ 11
- Problem2c.m : Figure 12 $\sim$ 16
- Problem2d.m : Figure 17 $\sim$ 21
- Problem2e.m : Figure 22 $\sim$ 26
- Problem2f.m : Figure 27 $\sim$ 31
- Problem3.m : Figure 32 $\sim$ 34

References

[1] Numerical Recipes in C : The Art of Scientific Computing

[2] Box—Muller transform from Wikipedia

[3] Uniform distribution from Wikipedia