Josai Mathematical Monographs
vol. 8 (2015), pp. 85-95

Pseudo-Normal Random Number Generation via the Eulerian Numbers

Nagatomo NAKAMURA

Abstract. The discrete probability distribution that appears in the sorting process of the modified bucket sorting is the Eulerian distribution. The Eulerian distribution gives a good approximation to the normal distribution. We propose an algorithm to generate pseudo-normal random numbers by adopting the properties of the Eulerian distribution. The capability of approximation to the normal distribution and the computational efficiency of the proposed method are demonstrated by numerical experiments.

1. Introduction

The distribution of the number of buckets (or bins) appearing in the process of the modified bucket sorting is the Eulerian distribution (Tsuchiya and Nakamura, 2009; Tsuchiya, 2015). The Eulerian distribution is a good approximation to the normal distribution. It is just similar to a binomial distribution. The objective of this paper is to propose a method for a fast generation of a normal random number using the property of the approximation compared to the Box-Muller method (Box and Muller, 1958).

The proof that a series of a random numbers follows to any probability distribution relies on a theoretical background. All the random numbers following to any probability distribution can be considered a locally uniform distribution. Our proposal can be confirmed that they are the normal random numbers, because the shape of the distribution of the generated random numbers is normally distributed and the result of the hypothesis testing for the data is accepted. A theoretical basis of the pseudo-normal random number is proposed in this paper, that is the data have a global normal distribution and a locally uniform distribution.

In the next section, the bucket sorting and the modified bucket sorting are described. The method of generation of the pseudo-normal random number is proposed in section 3. The effectiveness of the proposed method is verified through numerical experiments in section 4.

2. The Bucket Sorting and the Modified bucket Sorting

2.1. The bucket sorting (the bin sorting)

The principle of the bucket sorting (or bin sorting) is described.

Sorting Procedure (Bucket Sorting or Bin Sorting):

Suppose that there are n cards with $m\,(m \leq n)$ distinct numbers and m empty labeled buckets.

STEP 1 Distribute the cards into the buckets according to their numbers.

STEP 2 Sort the non-empty buckets.

The performance of an average case for computational time of the bucket sorting is $O(n+k)$, where n is the size of data and k is the number of different buckets. The sorting method is relatively faster than any other sorting algorithm.

2.2. The Modified Bucket Sorting and Eulerian Numbers

We prepare a deck of well shuffled n cards from 1 to n. The following procedure is carried out in order to sort a deck of cards in ascending order. At this time, Tsuchiya and Nakamura (2009) and Tsuchiya(2015) propose the modified algorithm of the bucket sorting as follows:

Sorting Procedure (Modified Bucket Sorting):

STEP 1 If the top of a deck of cards at hand is a card k and a card $k+1$ is on the table, then we put the card k on top of the card $k+1$.

STEP 2 If the card $k+1$ is not on the table, then the card k is not on any card on the table but is put directly on the surface of the table.

STEP 3 STEPs 1 and 2 are repeated until the cards at hand run out.

STEP 4 We bundle all the sets of cards in ascending order.

The differences between the bucket sorting and the modified ones are as follows: in the modified bucket sort, (i) it is assumed that none of the cards have the same numbers, (ii) we do not need to know how many buckets we should set up in advance, and (iii) if the card $k+1$ is on the table, then the card k is put on top of it. The number of bunches of cards in STEP 4 is the Eulerian numbers (Graham, Knuth and Patashnik, 1994; Kimber, 1989; Kunuth, 1997; Sloane, online reference) and dividing by $n!$ leads to the Eulerian distribution (Tsuchiya and Nakamura, 2009; Tsuchiya, 2015). Hereafter, the "bunches" will be called the "bins".

All the sequences of cards in $n = 2, 3$ and 4 related to the number of bins are as follows:

$$
\begin{aligned}
n=2: &\quad (1,2)\ \cdots\ 2, \quad (2,1)\ \cdots\ 1,\\
n=3: &\begin{cases} (1,2,3)\ \cdots\ 3,\ (1,3,2)\ \cdots\ 2,\ (2,1,3)\ \cdots\ 2,\\ (2,3,1)\ \cdots\ 2,\ (3,1,2)\ \cdots\ 2,\ (3,2,1)\ \cdots\ 1, \end{cases}
\end{aligned}
$$

$$n=4: \begin{cases} (1,2,3,4)\cdots 4,\ (2,1,3,4)\cdots 3,\ (3,1,2,4)\cdots 3,\ (4,1,2,3)\cdots 3, \\ (1,2,4,3)\cdots 3,\ (2,1,4,3)\cdots 2,\ (3,1,4,2)\cdots 3,\ (4,1,3,2)\cdots 2, \\ (1,3,2,4)\cdots 3,\ (2,3,1,4)\cdots 3,\ (3,2,1,4)\cdots 2,\ (4,2,1,3)\cdots 2, \\ (1,3,4,2)\cdots 3,\ (2,3,4,1)\cdots 3,\ (3,2,4,1)\cdots 2,\ (4,2,3,1)\cdots 2, \\ (1,4,2,3)\cdots 3,\ (2,4,1,3)\cdots 2,\ (3,4,1,2)\cdots 3,\ (4,3,1,2)\cdots 2, \\ (1,4,3,2)\cdots 2,\ (2,4,3,1)\cdots 2,\ (3,4,2,1)\cdots 2,\ (4,3,2,1)\cdots 1. \end{cases}$$

The total number of bins i in any number of cards n is obtained by the following recurrence relation (Tsuchiya and Nakamura, 2009):

$$\begin{cases} M_n(1)=M_n(n)=1 \quad (n\geq 1), \\ M_n(i)=iM_{n-1}(i)+\{n-(i-1)\}M_{n-1}(i-1) \quad (n\geq 3,\ i=2,...,n-1), \end{cases} \tag{1}$$

where, let X be the random variable which denotes the number of bins and let $M_n(i)$ be the total number of case that $X=i$ in n cards.

The frequency distribution of Eulerian numbers until $n=7$ is computed by the formula, shown in Table 1.

Table 1. Eulerian Numbers

$n\backslash i$	1	2	3	4	5	6	7	sum $(=n!)$
1	1							1
2	1	1						2
3	1	4	1					6
4	1	11	11	1				24
5	1	26	66	26	1			120
6	1	57	302	302	57	1		720
7	1	120	1191	2416	1191	120	1	5040

The discrete probability distribution is obtained by dividing $n!$. We call this distribution "Eulerian distribution." This distribution has statistically good properties, for example, its approximation to a normal distribution is excellent compared with a binomial distribution (Tsuchiya and Nakamura, 2009).

3. The principle of the pseudo normal random number generation

3.1. An Outline of Random Number Generation

The idea of generation of a pseudo normal random number is as follows: Given distinct n numbers over the range 1 to n, the total number of combinations is $n!$. We choose an integer random number q from $[1,n!]$. The number of bins k (or the Eulerian numbers k) is given by the sequence of the numbers t corresponding q. The $r=q/n!$ is normalized by the interval $(0,1]$, so the Eulerian numbers are dispersed by $k+r$ in $[k,k+1]$. Moreover, the Eulerian distribution approximately

follows the normal distribution $N((n+1)/2, (n+1)/12)$ (Tsuchiya and Nakamura, 2009), then

$$\frac{(k+r)-(n+1)/2}{\sqrt{(n+1)/12}}$$

follows a standard normal distribution. It is assumed that q and k are statistically independent. The random numbers generated by this method will be called pseudo-normal random numbers or Eulerian pseudo-normal random numbers.

As an intuitive understanding, it can be said that (i) the Eulerian distribution is an asymptotically normal distribution, and (ii) any normal random number generated by any methods, i.e. the Box-Muller method, is assumed to locally uniform distribution in a narrow interval. The proposed method gives us a normal distribution globally and a uniform distribution locally. If the number of bins is small and the data size is large (Figure 1 (a)–(c)), the distribution shape is stepped. But, if the number of bins is more than one hundred (Figure 1(e)–(h)), the shape is indistinguishable from a normal distribution (Figure 1(i)).

3.2. The relationship to the Eulerian distribution

More detailed description of the proposed method is as follows.

The number of combination of n cards is $n!$, and t is assumed to be a sequence of any cards. It can be written as:

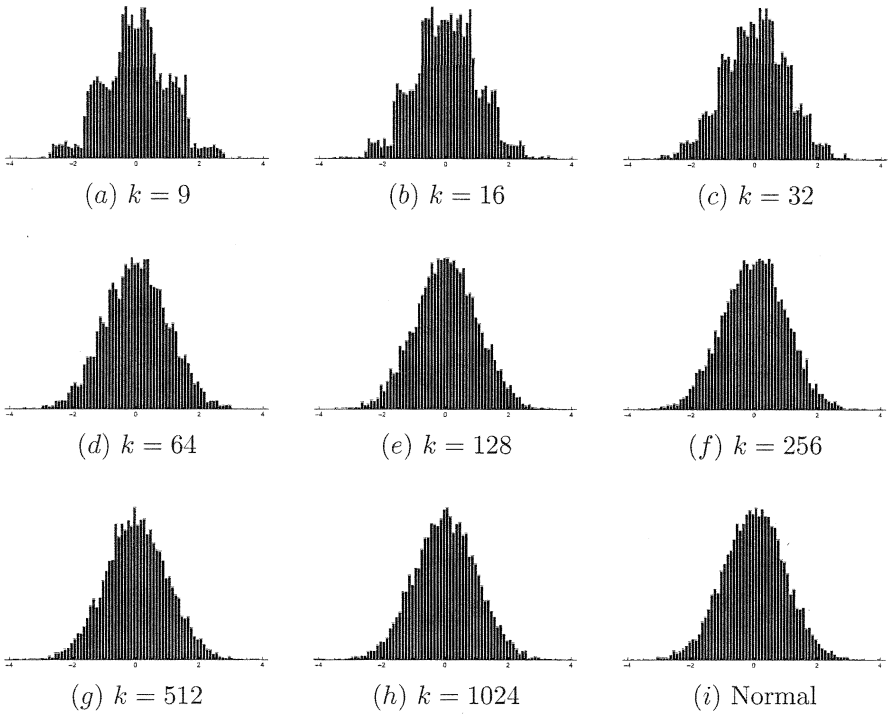
$$t = a_1 a_2 a_3 \cdots a_{n-1} a_n,$$

where $a_j \in [1, n]$, $a_j \neq a_{j'}$, $1 \leq j, j' \leq n$ and $j \neq j'$. The number of bins k is uniquely determined when t is given, and k is distributed in an approximately normal distribution, which in this case is an Eulerian distribution. Using this relation, the integer value q is randomly sampled from $[1, n!]$, the number of bins k is given by any card sequence t. Moreover, the q is transformed by $r = q/n!$, and will be a uniform distribution on $(0, 1]$. Next, the s computed by $s = k + r$ follows approximately a normal distribution. Subsequently, a random number following to $N(0, 1)$ is obtained by an appropriate standardization. In this connection, the Eulerian distribution has an average $(n+1)/2$ $(n \geq 1)$, and the variance $(n+1)/12$ $(n \geq 2)$, follows approximately a normal distribution $N((n+1)/2, (n+1)/12)$.

Q is the set of combinations T from 1 to n consisting of $n!$ array, and each combination has the number of bins K. These situations can be illustrated in figure 2.

The function $h(\cdot)$ transforms the sequence of card t into a q-th array of $n!$, and the function $g(\cdot)$ transforms q to the number of bins k. Then the following relation holds:

$$q = h(t), \quad g(q) = g(h(t)) = k$$



(a) $k = 9$ (b) $k = 16$ (c) $k = 32$

(d) $k = 64$ (e) $k = 128$ (f) $k = 256$

(g) $k = 512$ (h) $k = 1024$ (i) Normal

Figure 1. Histograms for several Pseudo-Normal Random Numbers and Normal Random Numbers. (a) to (h) are the distribution of the Eulerian normal random numbers, where the number of bins k is from 9 to 1024, respectively. (i) is the distribution of the normal random numbers. The number of data for each figure is 10,000.

As a result, we could obtain k from q without t.

3.3. The algorithm

Based on the previous section, we can generate a pseudo-normal random number with the following procedures.

STEP 1 Transformation Table: Preparing the transformation function $k = g(p)$.

STEP 2 Uniform Random Data: Generating an integer random number p from the uniform distribution $U(1, n!)$.

STEP 3 Number of Bins: Obtaining the number of bins k from $k = g(p)$.

STEP 4 Continuation: Computing $s = k + r$ in order to get a continuation via computing $r = p/n!$.

			T	(t)				Q		K	
a_1	a_2	a_3	$\cdots$	a_{n-2}	a_{n-1}	a_n		(q)		(k)	
1	2	3	$\cdots$	$(n-2)$	$(n-1)$	n	$\Rightarrow$	1	$\Rightarrow$	n	
1	2	3	$\cdots$	$(n-2)$	n	$(n-1)$	$\Rightarrow$	2	$\Rightarrow$	$n-1$	
1	2	3	$\cdots$	$(n-1)$	$(n-2)$	n	$\Rightarrow$	3	$\Rightarrow$	$n-1$	
1	2	3	$\cdots$	$(n-1)$	n	$(n-2)$	$\Rightarrow$	4	$\Rightarrow$	$n-1$	
1	2	3	$\cdots$	n	$(n-2)$	$(n-1)$	$\Rightarrow$	5	$\Rightarrow$	$n-1$	$n!$ array
1	2	3	$\cdots$	n	$(n-1)$	$(n-2)$	$\Rightarrow$	6	$\Rightarrow$	$n-2$	
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	
n	$(n-1)$	$(n-2)$	$\cdots$	3	1	2	$\Rightarrow$	$n!-1$	$\Rightarrow$	2	
n	$(n-1)$	$(n-2)$	$\cdots$	3	2	1	$\Rightarrow$	$n!$	$\Rightarrow$	1	

n card numbers

Figure 2. Transformation Table of the Sequence of the Numbers (T), Serial Number (Q), and the Number of Bins (K).

STEP 5 Standardization: Standardizing s by $(s-(n+1)/2)/\sqrt{(n+1)/12}$.

STEP 6 Repeating as necessary from STEPs 2 to 5.

3.4. The Computational Costs

In this subsection, the operational costs among (i) the proposed method, (ii) the Box-Muller method and (iii) the central limit theorem based generation method, are compared.

The Box-Muller method generates two normal random numbers, X and Y, by the independent two uniformly distributed random numbers, U and V, via the following transformation:

$$X = \sqrt{-2\log U}\cos 2\pi V, \tag{2}$$

$$Y = \sqrt{-2\log U}\sin 2\pi V, \tag{3}$$

where, $U, V \in (0, 1]$.

The central limit theorem based method is as follows: take the sum of 12 independent uniform random numbers and subtract 6 then the generated data follows a normal distribution $N(0, 1)$ (Shimizu, 1976).

We now compare the computational costs for generating a normal random number. However, in general, if the clock multiplier (frequency) in many modern microcomputers of the four fundamental operations (addition, subtraction, multiplication and division) is assumed to be one, the trigonometric cost is more than five times, the square root cost is at least four times or more. Although these costs are dependent on the performance of the compiler optimization.

First, the computational costs of generating the number of bins from the array in STEP 3 of the algorithm are zero. Arithmetic operation costs in STEP 4 are three, and in STEP 5 are two. The total costs are $5 + 0$ (arithmetic operations + mathematical functions). Second, in the Box-Muller method, the number of arithmetic operations for generating the two normal random numbers are $3 \times 2 = 6$, and the number of operations of trigonometric functions is $1 \times 2 = 2$. For common square root, four arithmetic operations is 1, the logarithm is 1, the square root is 1. Computational cost per one normal random number is $((6-1)+(6-2))/2 = 2.5+2$. Finally, in the central limit theorem, the arithmetic operations are 12. From the above illustrations, when the cost of the mathematical function is assumed to be at least four times more than the four arithmetic operations, the total cost of the proposed method is 4, the Box=Muller method is $2.5 + 2 \times 4 = 10.5$, and the method by central limit theorem is 12. Clearly our proposal has the smallest cost.

3.5. Simpilfication of the Algorithm

In the previous section, the strict algorithm for all of the combination for the card number has been shown. However, it must be considered a combination of $n! = 3628800$ when $n = 10$. Moreover, it becomes necessary an astronomical number for $n = 64$, i.e. $64! = 1.27 \times 10^{89}$, then it is not practical to obtain the number of bins for all combinations. Thus, we adopt the subset of the true Eulerian distribution to generate the random numbers, that is, we sample the subset randomly from the transformation table in STEP 1 of subsection 3.3. Then we generate the numbers with the subset in the remained STEPs in the algorithm. Then, the sampling for obtaining the pseudo-normal random number from the pseudo-probability distribution can help us to simplify the procedures and the computation.

4. Numerical Experiments

4.1. The purpose of the experiments

We verify the performance of the proposed method through various numerical experiments. In the numerical experiments, four tests of normality are applied to the pseudo-normal random numbers generated by the proposed method. Furthermore, the computational time of the proposed method and the Box-Muller methods are compared.

The conditions for the numerical experiments are as follows:

- The number of bins : $9, 16, 32, 64, 128, 256, 512$.
- The number of generated data: $20, 50, 100, 200, 500, 1000, 10000, 100000$.
- The number of simulations: 100,000.

Table 2. Performance Comparison

Methods	Eulerian-Normal (Pseudo-Normal)			Normal (Box-Muller)
Number of Bins	16	32	64	—
Average Time	43.1	44.2	43.0	78.0
Standard Deviation	0.946	0.628	0.812	0.907

The table shows the average time spent (millisecconds) and its standard deviations in generating a random number of 10^6 via 10,000 simulations. The source codes are written in Visual Basic 2010, and the computer processor is Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz with 8.00GByte memory.

- The methods for testing: (*a*) Anderson-Darling test, (*b*) Kolmogorov-Smirnov test, (*c*) Jarque-Bera test, and (*d*) Kuiper test.

On the other hand, the comparison experiments of the computational time by generating a 10^6 of the normal random number. The experiments were performed 10,000 times, the the average times were compared.

4.2. The results

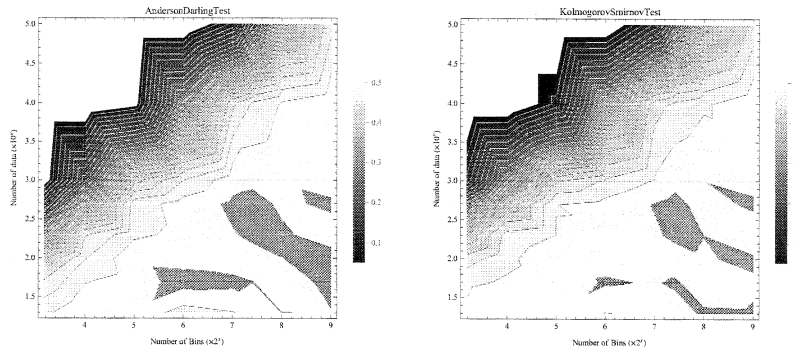
Figure 3 illustrates the contour maps of p-values with the number of bins and the number of data. The figures show the results of Anderson-Darling test, Kolmogorov-Smirnov test, Jarque-Bera test, and Kuiper test, respectively. The x-axis shows the number of bins and y-axis shows the number of generated data. The white area of the upper left of each figure is a rejection region of a p-value of less than five percent.

The four simulation results show that (*i*) when the number of data is small ($n < 1000$), the hypothesis of normality is not rejected by five percent regardless of the number of bins, (*ii*) when the number of bins is small and the number of data is large, the hypothesis is rejected, and (*iii*) when the number of bins is larger, the hypothesis can be regarded as a normal random number. For conditions different to those of rejection, it was verified that the pseudo-normal random numbers can be used as a normal random number.

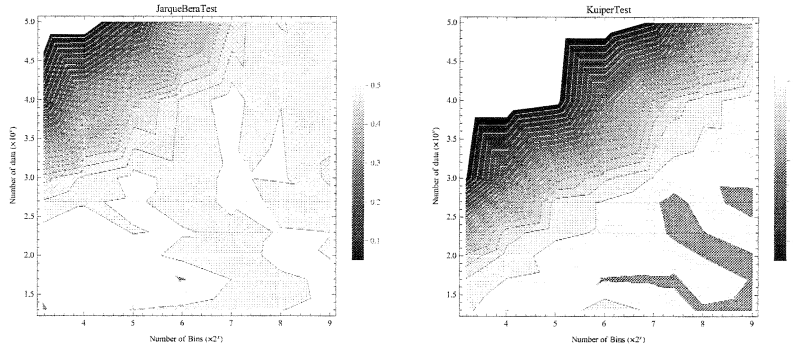
On the other hand, the results of computational time for generating 10^6 pseudo-normal and normal random numbers on different conditions are shown in table 2. Regardless of the number of bins, the results obtained by the proposed method are almost a certain period of time. It is about 1.8 times faster than the Box-Muller method.

5. Concluding Remarks

The proposed method can be viewed as a kind of the inverse function method for generating a random number of arbitrary probability distribution. The key issue of the proposed method is that a uniform random number behaves in two



(a) Anderson-Darling test (b) Kolmogorov-Smirnov test



(c) Jarque-Bera test (d) Kuiper test

Figure 3. The Results of Four Normality Tests. The x-axis indicates the number of bins, the y-axis indicates the number of samples. The white area of the upper left of each figure is a rejection region of a p-value of less than five percent.

ways. One is that it serves to select the number of bins, and the other is that helps to scatter the data in the bin.

We also now compare the tails truncation of the normal distribution for each generator. Since the most popular personal computer is equipped with 32-bit or 64-bit, maximum values by the Box-Muller transform are, $\sqrt{-2\ln(2^{-32})}\cos(2\pi 2^{-32}) \approx 6.66$ for 32-bit and $\sqrt{-2\ln(2^{-64})}\cos(2\pi 2^{-64}) \approx 9.42$ for 64-bit. In the standard normal distribution, twofold probability of greater than these values are shown in Table 3. On the other hand, the probabilities of the standard normal distribution obtained by changing the number of bins in the proposed method are also shown in Table 3. From these facts, we could understand where the distribution is truncated. From this table, the counterpart of the proposed method with 32 bins is 64-bit version of the Box-Muller method. When we increase the number of bins, the truncation location of the distribution is away

Table 3. Probability of Tails Truncation

	Number of Bins in Eulerian Distribution				Box-Muller Method	
	16	24	32	64	32bit	64bit
z-value	6.30	7.97	9.35	13.53	6.66	9.42
p-value	2.95×10^{-10}	1.62×10^{-15}	9.03×10^{-21}	9.78×10^{-42}	2.73×10^{-11}	4.54×10^{-21}

This table illustrates the twofold probability in a standard normal distribution given by a maximum value of each normal random generator. The z-value means the value of standard normal deviates, and the p-value means the probability of outside $\pm z$-values in a standard normal distribution. For 32 bit or 64 bit computers, the smallest number that can be generated is 2^{-32} or 10^{-64}, respectively. When U and V of uniform random number on $(0, 1]$ are equal to these values, the Box-Muller transformation in equation (2) produces a normal random variable equal to 6.66 or 9.42, respectively.

from the average.

When n is sufficiently large, the approximation accuracy for the normal distribution of the Eulerian distribution is good enough. Since the proposed method generates uniform distributions in the bins, it cannot be approximated to the slope of the normal distribution. This effect appears when the number of bins is small, and a large size of random number generates. However, the effect of the uniform distribution has been reduced when the number of bins is large, and its width is narrow. From the viewpoint of capability for approximation to a normal distribution and computational costs of generating numbers, the effectiveness of the proposed method was verified. The proposed method would be more effective when large-scale experiments are performed and the required number of normal random numbers per one dataset is not so large (about less than 10^4).

The remaining problem is to prove the local uniformity of random numbers following to any probability distribution mathematically and/or statistically.

Acknowledgements

The author would like to thank Professor Takahiro Tsuchiya from Josai University, and the two anonymous referees for their valuable comments and discussions. This research was supported by the ISM Cooperative Research Program (2013-ISM, CRP-2070).

References

[1] Box, G.E.P. and Muller, M.E. (1958). A note on the generation of random normal deviates, Annals Mathematical Statatistics, **29**, 610-611, doi:10.1214/aoms/1177706645.

[2] Graham, R. L., Knuth, D. E. and Patashnik, O. (1994). "Eulerian Numbers." §6.2 in Concrete Mathematics: A Foundation for Computer Science, 2nd ed. Reading, MA: Addison-Wesley, 267-272.

[3] Kimber, A. C. (1989). "Eulerian Numbers," *Supplement to Encyclopedia of Statistical Sciences*, (Ed. S. Kotz, N. L. Johnson, and C. B. Read), New York: Wiley, 59-60.

[4] Knuth, D. (1997). *Art of Computer Programming*, Volume 2: Seminumerical Algorithms, 3rd ed., Addison-Wesley.

[5] Shimizu, R. (1976). *Central Limit Theorem*, (in Japanese), Kyouiku-Shuppan, Tokyo, Japan.
[6] Sloane, N. J. A., Sequences A000295/M3416, A000460/M4795, A000498/M5188, and A008292 in "The On-Line Encyclopedia of Integer Sequences," URL: http://oeis.org/.
[7] Tsuchiya, T. (2015). Eulerian distribution with a missing number, Josai Mathematical Monographs, **8**, 73-83.
[8] Tsuchiya, T. and Nakamura, N. (2009). Eulerian numbers in modified bucket sorting and its related distribution theories, Proceedings of the Institute of Statistical Mathematics, **57**-1, 159-178.

Nagatomo NAKAMURA

Department of Economics, Sapporo Gakuin University
Bunkyodai 11, Ebetsu, Hokkaido, 0698555, Japan
nagatomo@sgu.ac.jp