Mathematical Problems of Computer Science 42, 73–80, 2014.

An Improved Algorithm for Generation of Truncated Normal Distributed Random Numbers

Vahe V. Sahakyan

Institute for Informatics and Automation Problems of NAS RA
e-mail: vahe@spir.me

Abstract

In this paper we discuss the computational problems of random numbers generation distributed by truncated normal distribution. It is shown that the standard methods and libraries have a limit for truncation point caused by the limit on the smallest number representable by double precision format. Theoretically the problems arise starting from the truncation point ≈ 40, but in practical calculations the limit is lower, starting from ≈ 8.5. An improved method is represented, based on the combination of two approximation algorithms, which with the represented coefficients has 4.5 times more coverage interval than the standard methods.

Keywords: Random numbers generation, Truncated normal distribution, Error function, Inverse error function.

1. Introduction

The need of generation of numbers distributed by truncated normal distribution arises across many statistical calculations and modelling problems. The family of normal and truncated normal distributions is defined using the error function, and their calculation is strictly related to the calculation of the error function and the inverse error function. Unfortunately, this functions are hard to compute numerically. Particularly, in case of the error function difficulties are visible in tail regions: it convergences to 0 or 1 values very fast and the floating point representation of the resulting values lose their accuracy. Obviously, the inverse case will convergence to $+\infty$ or $-\infty$ values very fast for arguments near 1 or 0. Due to these issues the generation of numbers distributed by truncated normal distribution requires a different approach. In this paper we are going to address possible ways to overcome those difficulties and generate numbers distributed by such distribution.

2. Definitions

The probability density function (PDF) of standard normal distribution is defined over the interval $(-\infty, +\infty)$. That function is usually denoted as $\Phi(0, 1, x)$ and is represented by the following formula:

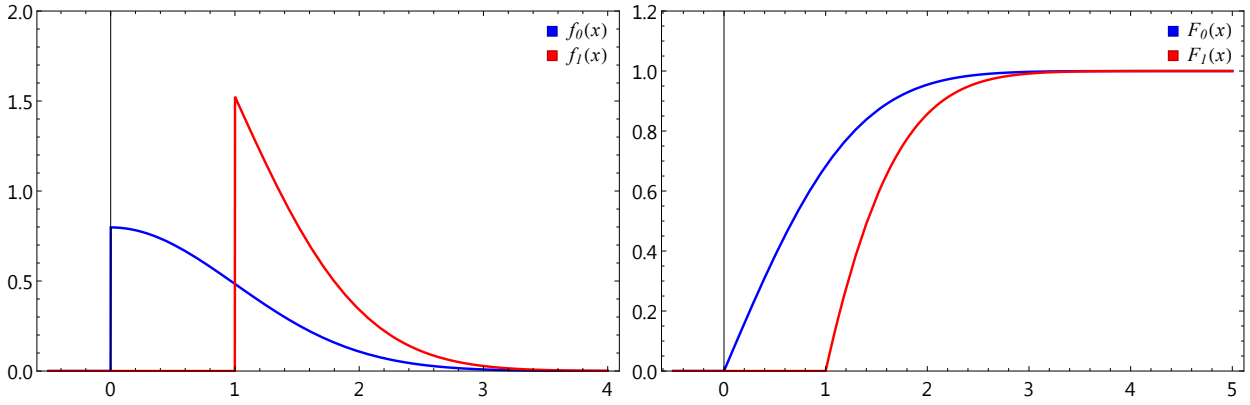

Fig. 1. PDF and CDF of half-normal standard distribution and truncated by 1 half-normal standard distributions.

$$\Phi(0,1,x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

In this article we will observe a distribution defined only on the positive range $[0, \infty)$, often called a half-normal standard distribution ($\mathcal{N}^+$). The normal distribution is symmetric about the origin, hence, it is enough to examine the half-normal case and reconstruct the normal distribution from it. The same is true for the truncated case.

PDF of half-normal standard distribution on interval $[0, +\infty)$ is defined as follows:

$$f(x) = \begin{cases} 2\,\Phi(0,1,x) & x \geq 0 \\ 0 & x < 0 \end{cases} = \begin{cases} \sqrt{\frac{2}{\pi}} e^{-\frac{x^2}{2}} & x \geq 0 \\ 0 & x < 0 \end{cases}.$$

We will say, that a is distributed by *half-normal standard distribution truncated by* z ($\mathcal{N}_z^+$), if a follows $\mathcal{N}^+$ and satisfies $a \geq z$, where $z \geq 0$. The PDF of this distribution will be:

$$f_z(x) = \begin{cases} \frac{\sqrt{\frac{2}{\pi}} e^{-\frac{x^2}{2}}}{\operatorname{erfc}\left(\frac{z}{\sqrt{2}}\right)} & x > z \\ 0 & x \leq z \end{cases}, \tag{1}$$

where erfc is a complementary error function, which is defined as

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2}\, dt.$$

The cumulative density function (CDF) of $\mathcal{N}_z^+$ can be easily found by integrating (1) on the interval $(-\infty, x]$, and it will be

$$F_z(x) = \int_{-\infty}^{x} f_z(t)\, dt = \begin{cases} 1 - \frac{\operatorname{erfc}\left(\frac{x}{\sqrt{2}}\right)}{\operatorname{erfc}\left(\frac{z}{\sqrt{2}}\right)} & x > z \\ 0 & x \leq z \end{cases}. \tag{2}$$

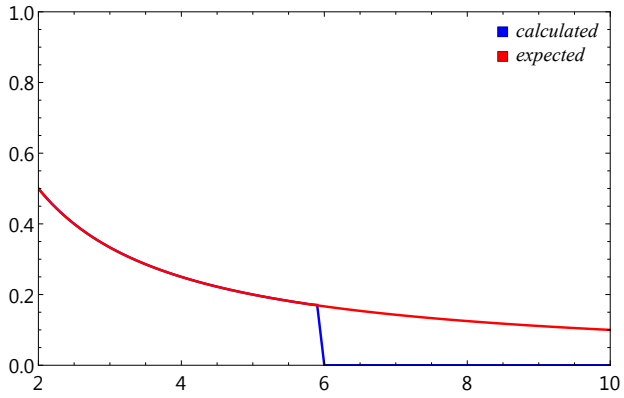

Fig. 2. This graphic shows calculated and expected values of $\frac{1}{\text{erfc}^{-1}(\text{erfc}(x))}$ using [5] for erfc^{-1} and [6] for erfc.

3. Numbers Generation

There are two common approaches to generate numbers distributed by an arbitrary distribution. The first one is the Von Neumann's rejection technique [10]. On truncated normal distribution this technique has very a low efficiency, because the probability that the generated number satisfies $x > z$, where $x \sim \mathcal{N}^+$ is equal to $f(z)$. In case, if $z = 5$ the $f(5) \approx 3 * 10^{-6}$, so approximately 99.9997% of probes will be rejected. Obviously, in applications this method is not acceptable.

The second approach is the inverse transform sampling method [2]. It claims that if u is uniformly distributed on the interval $(0, 1)$, then $F_z^{-1}(u)$ follows the distribution in which CDF is F_z. We will take $u = 1 - u$ to make the equation simple, and in this case the inverse of (2) will be

$$F_z^{-1}(u) = \sqrt{2}\ \text{erfc}^{-1}\left(\text{erfc}\left(\frac{z}{\sqrt{2}}\right) u\right). \tag{3}$$

Note, that during the inverse function calculation we are not considering the case when $x < z$, because in that case $F_z(x) = 0$, therefore the probability of that is 0, so it goes beyond our interests.

As we can see, computation of F_z^{-1} requires computation of erfc and erfc^{-1}. The argument range of erfc is $[0, \infty)$, therefore, the values are ranged in $[0, 1]$. The u is a random number from the range $(0, 1)$, hence, the argument range for erfc^{-1} is $[0, 1]$.

From a theoretical point of view the expression F_z^{-1} is correct, however, from computational perspective it could imply numerical errors. For example, if $z = 40$, then $\text{erfc}\left(\frac{40}{\sqrt{2}}\right) \approx 7.3 \times 10^{-350}$, which is not representable in double precision format [3] and resulting in 0, therefore also $\text{erfc}^{-1}(0) = \infty$. Figure 2 shows the actual numerical calculation results. It is clearly visible, that starting from values near 6 the result is ∞, even though theoretically it is becoming not computable starting from values higher than 28.28. The break point corresponds to the value $z = 6\sqrt{2} \approx 8.49$.

Conclusion from here is that independently from precision and accuracy of calculation, when erfc and erfc^{-1} are computed separately, starting from $z \geq 40$ the final result will always be ∞ in double precision representation. Obviously, their calculation should be done combined.

4. Numerical Approximations

The usual approach to the computation of erfc, is to divide the axis into segments and use different polynomials and asymptotic approximations for those segments ([1], [6], [7], [8] and [9]). Let us have a look at approach discussed in [1]. At first erfc can be approximated in the following form:

$$\operatorname{erfc}(x) = t \exp\left(-x^2 + P(t)\right) \quad x > 0, \tag{4}$$

where

$$t = \frac{2}{2+x},$$

and $P(t)$ is polynomial ($0 \le t \le 1$) which is found by Chebyshev polynomial approximation. The sample coefficients for 28 degree polynomial are shown in Table 1.

Now let us have a look at the calculation of erfc^{-1}. As suggested in [1], we apply the Halley's method on $g(x) = \operatorname{erfc}(x) - y$ and solve it for $g(x) = 0$. In that case the iteration variable will be

$$x_{n+1} = x_n - \frac{2g(x_n)g'(x_n)}{2[g'(x_n)]^2 - g(x_n)g''(x_n)} = x_n + \frac{\operatorname{erfc}(x_n) - y}{\frac{2}{\sqrt{\pi}}\exp(-{x_n}^2) - x_n(\operatorname{erfc}(x_n) - y)}.$$

Here, the following form of approximation is used for the initial value of x_n:

$$\operatorname{erfc}^{-1}(x) \approx R(v), \quad 0 < x < 1, \tag{5}$$

where $R(v)$ is a rational polynomial and

$$v = \sqrt{-2\log\left(\frac{x}{2}\right)}. \tag{6}$$

R has the following form and the coefficients are found by rational polynomial approximation [1]

$$R(v) = \frac{0.70711v^3 + 15.6585v^2 + 11.5099v - 36.4132}{v^2 + 22.1444v + 22.3164}.$$

Table 1. Coefficients of polynomial $P(t)$. Degree column corresponds to degree of variable t in polynomial P.

The represented form of both of these approximations are not the best ones from accuracy and performance perspective, but they have very simple and convenient form for further transformations. Accuracy can also be improved by using higher degree polynomials.

5. Improved Algorithm

As was shown in Section 3, with standard approach and using standard libraries the computation of (3) is impossible for values more than $6\sqrt{2}$. Here we will describe a combined method from two algorithms described in the previous section. Taking into consideration the argument of erfc^{-1} in (3) and (4), (6) will get

degree	value	degree	value
0	-1.265512123484646	14	$8.81583262585187 * 10^{1}$
1	$9.99999999999995 * 10^{-1}$	15	$-2.449786109546412 * 10^{2}$
2	$3.750000000013982 * 10^{-1}$	16	$5.592802242564082 * 10^{2}$
3	$8.33333331763132 * 10^{-2}$	17	$-1.037357191852551 * 10^{3}$
4	$-8.59374904138571 * 10^{-2}$	18	$1.541879449019999 * 10^{3}$
5	$-1.437503619675783 * 10^{-1}$	19	$-1.824712058124052 * 10^{3}$
6	$-9.17876834732462 * 10^{-2}$	20	$1.714187964988546 * 10^{3}$
7	$2.940960653857621 * 10^{-2}$	21	$-1.272415091307403 * 10^{3}$
8	$1.351072101958271 * 10^{-1}$	22	$7.388485218111491 * 10^{2}$
9	$9.90519562132564 * 10^{-2}$	23	$-3.293364188005826 * 10^{2}$
10	$1.566341639700399 * 10^{-1}$	24	$1.090336950910116 * 10^{2}$
11	-1.389503257698621	25	$-2.530009680666059 * 10^{1}$
12	5.910981818651237	26	3.677189095748008
13	$-2.552672826806604 * 10^{1}$	27	$-2.522015791327478 * 10^{-1}$

Table 1: Coefficients of polynomial $P(t)$. Degree column corresponds to degree of variable t in polynomial P.

$$v = \sqrt{-2\,log\left(\frac{t\,u\exp\left(-\frac{z^2}{2}+P(t)\right)}{2}\right)} = \sqrt{z^2 - 2P(t) - 2\log(t\,u) + \log(4)},$$

where

$$t = \frac{4}{4+\sqrt{2}z}.$$

By this simple insertion we combine both computation algorithms and as a result exempt from exponential part in (4). Based on this computation the algorithm will be defined as follows:

function InverseTruncatedCDF(z, u)
  $y \leftarrow \mathrm{erfc}(z) * u$
  $t \leftarrow 4/(4 + \mathrm{sqrt}(2) * z)$
  $v \leftarrow \mathrm{sqrt}(z * z - 2 * \mathrm{P}(t) - 2 * \log(t * u) + \log(4))$
  $x \leftarrow \mathrm{R}(v)$
  for $i \leftarrow 1, 4$ **do**
    $a \leftarrow \mathrm{erfc}(x) - y$
    $b \leftarrow 2/\mathrm{sqrt}(\pi) * \exp(-x * x) - x * a$
    if $b = 0$ **then**
      break
    end if
    $x \leftarrow x + a/b$
  end for
  return x
end function

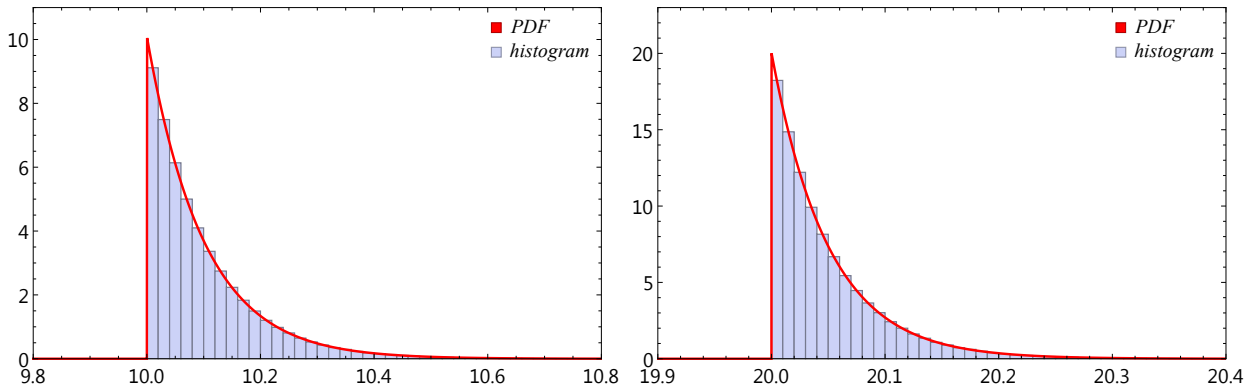

Fig. 3. Probability density function (PDF) and histogram of 10^6 samples calculated by RandTruncatedNormal function for $\mathcal{N}_{10}^{+}$ and $\mathcal{N}_{20}^{+}$, respectively.

The algorithm consists of maximum 4 loops of Halley's method. This amount was taken by applying a series of numerical experiments and the results show that the increase of loops doesn't improve the result.

Drawback of this algorithm is that with the increase of z the initial x is also increasing. With coefficients represented in paper for $z = 39$ we will have $x \approx 28.2833$ and the value of $\exp(-x * x) \approx 3.87 * 10^{-348}$, which is already not representable in double precision format. Because of that the Halley's method improvement steps are not possible, so accuracy is lost. Nevertheless, with current coefficients it is covering range $[0, 38]$ with maximum error smaller than 10^{-14}. The result can be improved by taking better approximation for initial x, which means using a higher degree polynomial for R.

InverseTruncatedCDF can be used for random number generation following distribution $\mathcal{N}_z^{+}$. The following algorithm can be used for that purpose:

function RandTruncatedNormal(z)
  $u \leftarrow$ rand() ▷ uniformly distributed random number from interval $(0, 1)$
  $r \leftarrow$ InverseTruncatedCDF(z, u)
  return $\sqrt{2} * r$
end function

In Figure 3 the result of numbers generation are shown using the above algorithm.

6. Conclusion and Remarks

The problems related to limits of double precision number representation are making impossible the calculation of such formulas in tail regions. The suggested way is one of the approaches that is improving the coverage range. Another way would be the use of Chebyshev multi-variate approximation, but as a rule a better coverage means higher degree of polynomial and consequently a slow computational time. The choice of algorithm is strictly dependent on the initial problem and here we represented a relatively light-weight method, also with a possibility to be improved by increasing degrees of approximation polynomials P and R.

References

[1] W. H. Press, S. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Third Edition, New York, 2007.

[2] L. Devroye, *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, 1986.

[3] American National Standards Institute and Institute of Electrical and Electronic Engineers, *Standard for Binary Floating-Point Arithmetic*, ANSI/IEEE Standard 754-1985, 1985.

[4] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standards Applied Mathematics Series - 55, New York, 1964.

[5] F. W. J. Olver, D. W. Lozier, R. F. Boisvert and C. W. Clark, "Error Functions, Dawsons and Fresnel Integrals" *NIST Handbook of Mathematical Functions*, Cambridge University Press, p. 166, New York, 2010

[6] S. L. Moshier, Cephes Math Library. [Online]. Available: http://www.netlib.org/cephes

[7] J. F. Hart, et al., *Computer Approximations*, The Siam Series in Applied Mathematics, New York, 1968

[8] W. J. Cody, "Rational Chebyshev approximations for the error function" *Math. Comp.*, vol. 23, pp. 631-637, 1969

[9] GNU Project, GNU Scientific Library. [Online]. Available: https://www.gnu.org/software/gsl/

[10] J. von Neumann, "Various techniques used in connection with random digits. Monte Carlo methods", *Nat. Bureau Standards*, vol. 12, pp. 36-38, 1951.

Submitted 05.09.2014, accepted 28.11.2014.

Կտրված ստանդարտ նորմալ բաշխված թվերի գեներացման բարելավված ալգորիթմ

Վ. Սահակյան

Ամփոփում

Հոդվածում քննարկված են կտրված ստանդարտ նորմալ բաշխված թվերի գեներացման գործընթացում առաջացող հաշվողական խնդիրները: Ապացույցվել է, որ ստանդարտ եղանակները և գրադարանները ունեն կտրման կետի ընտրության սահմանափակում, որը պայմանավորված է կրկնակի ճշտությամբ թվերի ներկայացման սահմանափակումներով: Տեսականորեն խնդիրներն առաջանում են ¼ 40-ից մեծ կտրման կետի համար, սակայն գործնականում այդ սահմանը շատ ավելի փոքր է՝ ¼ 8:5-ից սկսած: Առաջարկված է նոր եղանակ, որի հիմքում երկու մոտարկման ալգորիթմների միավորումն է: Կիրառելով հոդվածում ներկայացված գործակիցները՝ կտրման կետի համար ստացվում է 4.5 անգամ ավելի մեծ հաշվարկելի տիրույթ, քան ստանդարտ եղանակների դեպքում:

Улучшенный алгоритм генерации случайных чисел с усеченным нормальным распределением

В. Саакян

Аннотация

В этой работе рассмотрены вычислительные задачи генерации случайных чисел с усеченным нормальным распределением. В работе показано, что стандартные методы и библиотеки имеют ограничения точки усечения. Эти ограничения обусловлены лимитом на наименьшее число, представимое с двойной точностью.Теоретически, сложности возникают начиная с усечения в точке ¼ 40, но в практических расчетах предел намного ниже, начиная с ¼ 8:5. Представлен усовершенствованный метод, созданный на основе комбинации двух аппроксимационных алгоритмов. В сравнении со стандартными методами, при показанных коэффициентах, данный метод обеспечивает в 4.5 раз больший интервал покрытия.