# How to generate random numbers with a power (or any other) density*

M. A. Furman†
*Center for Beam Physics*
*Lawrence Berkeley National Laboratory, MS 71B-287*
*1 Cyclotron Rd.*
*Berkeley, CA 94720*
(Dated: January 15, 2010)

We derive the "inversion algorithm" to generate random numbers in a specified interval $x_1 \le x \le x_2$ with 100% efficiency for a one-dimensional distribution with a power density, based on the uniform random number generator in $[0,1]$. We then review the method applicable to the case of the most general one-dimensional density, and apply it to obtain the algorithms for several other distributions of practical use. We derive the fundamental scaling relation obeyed by any distribution and apply it to obtain the algorithm for the generation of random numbers with a distribution with arbitrary RMS $\sigma$ and reference point $x_0$ out of the distribution with $\sigma = 1$ and $x_0 = 0$, in the given interval $x_1 \le x \le x_2$. We then review the 2D algorithms for the generation of gaussian random numbers and examine their shortcomings. Finally, we recapitulate the Monte Carlo algorithm, and describe some of the variants that improve its efficiency.

This note was written for practical convenience only; the methods and the results are well known.

## I. BASIC CASE: UNIFORM DENSITY.

At the base of essentially everything described in this note is the generation of random numbers uniformly distributed in the interval $[0,1]$. We define the basic distribution function $U(x)$ via

$$U(x) = \begin{cases} 1 & \text{if } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Suppose that we want to generate a set of $m$ random numbers $\{x\}$ such that the distribution of the $x$'s, in the limit $m \to \infty$, is uniform in $0 < x < 1$ and vanishes outside this interval. The distribution density of the $x$'s is therefore

$$\frac{dN}{dx} = kU(x) \tag{2}$$

where $k > 0$ is an unimportant normalization constant.[1] The $x$'s are therefore given by

$$x = \hat{u} \tag{3}$$

where $\hat{u}$ denotes here, and throughout this note, a random number uniformly distributed in $[0,1]$. We assume that the user has access to a computer function that performs such random number generation.

If the range of interest is not $[0,1]$ but rather an arbitrary interval $[x_1, x_2]$, with $x_1 < x_2$, then clearly

$$\frac{dN}{dx} = kU(\gamma x + \delta) \tag{4}$$

where the constants $\gamma$ and $\delta$ are determined by the requirements $\gamma x_1 + \delta = 0$ and $\gamma x_2 + \delta = 1$. In this case the algorithm is given by

$$\gamma x + \delta = \hat{u} \tag{5}$$

or, more explicitly, inserting the values for $\gamma$ and $\delta$,

$$x = (x_2 - x_1)\hat{u} + x_1 \tag{6}$$

Note that the normalization integral

$$K \equiv \int_{x_1}^{x_2} dx \, \frac{dN}{dx} = k(x_2 - x_1) \tag{7}$$

plays no direct role in the algorithm: only the ratio $K/k = x_2 - x_1$ matters, and the only requirement on it is that it must be finite.

## II. POWER DISTRIBUTION.

In this case we want to generate random numbers $x$ such that their distribution, in the $m \to \infty$ limit, is

$$\frac{dN}{dx} = kx^p \tag{8}$$

in the region $x_1 \le x \le x_2$, for a specified value of $p$.

While $p$ can, in general, be an arbitrary real number, we first impose the restrictions $p \neq -1$ and $0 < x_1 < x_2 < \infty$. Removal of these restrictions is possible in some cases, as discussed below. The power density clearly makes no sense for $x < 0$ unless $p$ is an even integer; we exclude this case from consideration because it is the mirror image of the $x > 0$ case.

The trick to generate the $x$'s is to make the change of variable defined by

$$dy = \lambda x^p dx \tag{9}$$

*Work supported by US DOE under contract DE-AC02-05CH11231.
†Electronic address: mafurman@lbl.gov; URL: http://mafurman.lbl.gov
[1] In fact, for the cases of uniform distribution, $k = m$.

where $\lambda$ is a positive constant to be determined. Eq. (8) then becomes

$$\frac{dN}{dy} = k/\lambda \tag{10}$$

which is similar to Eq. (4) except that $y$ replaces $x$, hence $y = \alpha\hat{u} + \beta$. Integrating Eq. (9) yields

$$y = ax^{p+1} - b \tag{11}$$

where $a = \lambda/(p+1)$, hence the algorithm is

$$x = \left(\frac{\alpha\hat{u} + \beta + b}{a}\right)^{1/(p+1)} \tag{12}$$

This equation shows that the constants $\alpha$, $\beta$, $a$ and $b$ have no independent meaning: only the combinations $\alpha/a$ and $(\beta + b)/a$ matter. We might as well set $\alpha = 1$ and $\beta = 0$ so that $y = \hat{u}$, which justifies the convenience of the introduction of the constant $\lambda$. To obtain $a$ (or, equivalently, $\lambda$) and $b$ we impose the requirements that $y = 0$ for $x = x_1$ and $y = 1$ for $x = x_2$, which yield

$$\frac{b}{a} = x_1^{p+1}\,, \qquad \frac{1}{a} = x_2^{p+1} - x_1^{p+1} \tag{13}$$

from which we get the explicit algorithm

$$x = \left[(x_2^{p+1} - x_1^{p+1})\hat{u} + x_1^{p+1}\right]^{1/(p+1)} \tag{14}$$

Note again that the normalization integral

$$K = k\int_{x_1}^{x_2} dx\, x^p = \frac{k(x_2^{p+1} - x_1^{p+1})}{p+1} \tag{15}$$

plays no direct role (only $K/k$ matters), and it must be finite.

### A. Special cases (1): $x_1 = 0$ or $x_2 = \infty$.

These cases are possible as long as $p$ is such that $K < \infty$, ie., as long as the distribution is normalizable. This condition implies the following restrictions on $p$:

$$x_1 = 0 \text{ and } x_2 < \infty : \; p > -1 \tag{16a}$$
$$x_1 > 0 \text{ and } x_2 = \infty : \; p < -1 \tag{16b}$$
$$x_1 = 0 \text{ and } x_2 = \infty : \; \text{impossible} \tag{16c}$$

### B. Special case (2): $p = -1$.

One can take the limit $p \to -1$ by using l'Hôpital's rule applied to either (12) or (14) or, more simply, by starting directly from Eqs. (8) and (9). In this case $dy/dx = \lambda/x$ hence

$$y = a\ln x - b \tag{17}$$

where $a = \lambda$. Imposing the conditions $y = 0$ when $x = x_1$ and $y = 1$ when $x = x_2$ yields

$$\frac{1}{a} = \ln\left(\frac{x_2}{x_1}\right)\,, \qquad \frac{b}{a} = \ln x_1 \tag{18}$$

therefore, setting $y = \hat{u}$, we obtain

$$x = \exp\left(\frac{\hat{u} + b}{a}\right) = x_1\left(\frac{x_2}{x_1}\right)^{\hat{u}} \tag{19}$$

Note that, in this case, both choices $x_1 = 0$ and $x_2 = \infty$ are forbidden because either one of these would make the distribution non-normalizable ($K = \infty$).

## III. THE INVERSION METHOD–GENERAL CASE.

The problem is stated as follows: generate a one-dimensional random number distribution with a given probability density $\rho(x)$,

$$\frac{dN}{dx} = \rho(x) \tag{20}$$

in $x_1 \le x \le x_2$. We require that $\rho(x)$ not vanish in any finite sub-interval within $[x_1, x_2]$. If it does, the problem reduces to generating random numbers from the linear superposition of two (or more), possibly disjoint, distributions, which is described in Sec. VI.

Even though we are interested in a given $x$-interval, in many cases $\rho(x)$ will be defined in a larger region of the $x$ axis. If $\rho(x)$ is given in analytic form, it is typically defined either in $-\infty < x < \infty$ or in $0 \le x < \infty$. Examples of distributions of the former kind are the Gaussian, $\rho(x) = k\exp(-(x-x_0)^2/2\sigma^2)$, the Lorentzian, $\rho(x) = k/((x-x_0)^2 + \gamma^2)$, etc. For these distributions $x_1$ may extend all the way to $-\infty$. Examples of the latter kind are the power density, $\rho(x) = kx^p$, the exponential, $\rho(x) = k\exp(-x)$, etc. For either kind, $x_2$ may extend to $+\infty$. The formulas below apply equally well to any such distribution. If $\rho(x)$ is only known in $x_1 \le x \le x_2$, see below.

The procedure to find the algorithm to generate the $x$'s according to (20) is to: (1) find a change of variables $x = f(y)$ such that the $y$'s are uniformly distributed, ie., $dN/dy$ =constant; (2) use the result of Sec. I, namely $y = \alpha\hat{u} + \beta$; (3) determine $\alpha$ and $\beta$ from the endpoints of the interval, $x_1 = f(\beta)$ and $x_2 = f(\alpha + \beta)$. The $x$'s are then given by $x = f(\alpha\hat{u} + \beta)$. In practice, it is usually easy to find $y$ as a function of $x$, but it's often not possible to invert this relation to find $x$ as a function of $y$ in analytic form.

The required change of variables is given by

$$dy = \lambda\rho(x)dx \tag{21}$$

where the scaling constant $\lambda$ is introduced, as in the power density case in Sec. II, to allow the choice $\alpha = 1$

and $\beta = 0$. Integrating (21) gives

$$y = \lambda P(x) - b \tag{22}$$

where the cumulative probability function $P(x)$ is defined to be the integral of $\rho(x)$ relative to $x = 0$,

$$P(x) \equiv \int_0^x dx' \rho(x') \tag{23}$$

Eq. (22) yields

$$x = P^{-1}\left(\frac{y+b}{\lambda}\right) \tag{24}$$

where $P^{-1}(z)$ is the *functional inverse* (not to be confused with the algebraic inverse) of $P(x)$ (that is to say, if $z = P(x)$, then[2] $x = P^{-1}(z)$).

The constants $\lambda$ and $b$ follow from the requirements that $x = x_1$ when $y = 0$ and $x = x_2$ when $y = 1$, ie.

$$x_1 = P^{-1}\left(\frac{b}{\lambda}\right), \qquad x_2 = P^{-1}\left(\frac{1+b}{\lambda}\right) \tag{25}$$

or, equivalently,

$$\frac{b}{\lambda} = P_1\,, \qquad \frac{1+b}{\lambda} = P_2 \tag{26}$$

where $P_i \equiv P(x_i)$, $i = 1, 2$. From here we easily obtain $b$ and $b/\lambda$, hence Eq. (24) yields the general algorithm

$$x = P^{-1}\left[(P_2 - P_1)\hat{u} + P_1\right] \tag{27}$$

Again, the normalization $K \equiv \int_{x_1}^{x_2} dx\, \rho(x) = P_2 - P_1$ must be finite.

If the density $\rho(x)$ is only defined, or only known, in the interval $[x_1, x_2]$, or is not given in analytic form (for example, it may be specified as a numerical table in $[x_1, x_2]$), then one may deal with an extended density defined by

$$\rho_{\rm ext}(x) = \begin{cases} \rho(x) & \text{if } x_1 \le x \le x_2 \\ 0 & \text{elsewhere} \end{cases} \tag{28}$$

and proceed as in the case above (a slight confusion may arise here because $P^{-1}(x)$ is not well defined outside $[x_1, x_2]$, although in practice this is not a problem). Equivalently, it is conceptually simpler to define $P(x)$ relative to $x = x_1$ rather than to $x = 0$, namely

$$P(x) \equiv \int_{x_1}^x dx' \rho(x')\,, \quad x_1 \le x \le x_2 \tag{29}$$

With this definition $P_1 = 0$ and $P_2 = K$, hence the algorithm is now written[3]

$$x = P^{-1}(K\hat{u}) \tag{30}$$

Depending on the complexity of the expressions for $P_1$ and $P_2$, one may be able to simplify somewhat expressions (27) or (30) by the replacement $\hat{u} \to 1 - \hat{u}$ because $1-\hat{u}$ is as good a uniform random number in $[0, 1]$ as $\hat{u}$ is. With this replacement, however, $\hat{u} = 0(1)$ gets mapped onto $x = x_2(x_1)$ rather than $x_1(x_2)$, an immaterial difference in the algorithm.

### A. Example: power density.

To show how the general formalism above applies to the power density we use Eq. (23) to obtain

$$z \equiv P(x) = k \int_0^x dx' x'^p = \frac{k x^{p+1}}{p+1} \tag{31}$$

therefore

$$x = P^{-1}(z) = \left[\frac{(p+1)z}{k}\right]^{1/(p+1)} \tag{32}$$

We now use

$$P_i = \int_0^{x_i} dx\, \rho(x) = \frac{k x_i^{p+1}}{p+1}\,, \quad i = 1, 2 \tag{33}$$

and, according to (27), insert $z = (P_2 - P_1)\hat{u} + P_1$ into (32), thus recovering the original expression (14).

Had we applied Eq. (29) instead of (23) we would have obtained

$$z \equiv P(x) = k \int_{x_1}^x dx' x'^p = \frac{k}{p+1}(x^{p+1} - x_1^{p+1}) \tag{34}$$

hence

$$x = P^{-1}(z) = \left[\frac{(p+1)z}{k} + x_1^{p+1}\right]^{1/(p+1)} \tag{35}$$

Following Eq. (30), we use $P_1 = 0$, $P_2 = K$ and $z = K\hat{u}$ in (35), yielding exactly the same result as above, namely Eq. (14).

---

[2] Note that there is a $1 \leftrightarrow 1$ correspondence between $x$ and $z$ owing to the monotonically increasing nature of $P(x)$.

[3] This variant of defining $P(x)$ relative to $x = x_1$ rather than to $x = 0$ is always valid, whether $\rho(x)$ is given in analytic form or not. Even though we use the same notation for $P(x)$ in (23) and (29), it should be clear that these are different functions, as are their functional inverses (27) and (30), respectively. Also, expression (29) for $P(x)$ contains an implicit dependence on $x_1$ which we suppress for notational compactness.

### B. Remark.

This algorithm presented here is most efficient when (a) $P(x)$ is obtainable in analytic form, and (b) its functional inverse has a simple analytic form. If these conditions are not satisfied, the next best option is to tabulate both $P(x)$ and $P^{-1}(x)$ in the interval $[x_1, x_2]$ and use table interpolation; this option works well if $P(x)$ is smooth enough and the interval is finite. If none of the above conditions are met, the conventional alternative to generating random numbers is the Monte Carlo ("accept-reject") method. This method has the advantages of simplicity and straightforward extension to higher dimensions, but it may have efficiency problems. See Sec. IX.

## IV. OTHER EXAMPLES.

Here we provide a list of examples that might be useful, without the detailed derivation. For each case we provide $\rho(x)$, the relevant interval, and any applicable condition on the parameters that might appear in $\rho(x)$.

1. $\rho(x) = ke^{-(x-x_0)^2/2\sigma^2}$, $x_1 \le x \le x_2$.
   Conditions: $\sigma > 0$, $x_0$ arbitrary.

$$x = x_0 + \sqrt{2}\sigma \,\mathrm{erf}^{-1}[(P_2 - P_1)\hat{u} + P_1] \tag{36a}$$

$$P_i \equiv \mathrm{erf}\left(\frac{x_i - x_0}{\sqrt{2}\sigma}\right), \quad i = 1, 2 \tag{36b}$$

   where $\mathrm{erf}(x)$ is the conventional error function,

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x dt\, e^{-t^2} \tag{37}$$

   so that $\mathrm{erf}(\pm\infty) = \pm 1$. Note that if $x_1 = -\infty$ and $x_2 = +\infty$ you get

$$x = x_0 + \sqrt{2}\sigma \,\mathrm{erf}^{-1}(2\hat{u} - 1) \tag{38}$$

2. $\rho(x) = k/[(x - x_0)^2 + \gamma^2]$, $x_1 \le x \le x_2$.
   Conditions: $\gamma > 0$, $x_0$ arbitrary.

$$x = x_0 + \gamma \tan[(T_2 - T_1)\hat{u} + T_1] \tag{39a}$$

$$T_i = \tan^{-1}\left(\frac{x_i - x_0}{\gamma}\right), \quad i = 1, 2 \tag{39b}$$

   where the $\tan^{-1}$ function is defined in $(-\pi/2, +\pi/2)$. Note that, if $x_1 = -\infty$ and $x_2 = +\infty$, you get

$$x = x_0 + \gamma \tan\left[\frac{(2\hat{u} - 1)\pi}{2}\right] \tag{40}$$

3. $\rho(x) = k \sin \pi x$, $0 \le x \le 1$

$$x = \frac{1}{\pi} \cos^{-1}(2\hat{u} - 1) \tag{41}$$

   where the function $\cos^{-1} x$ is defined in $0 \le x \le \pi$.

4. $\rho(x) = k \cos(\pi x/2)$, $-1 \le x \le 1$.

$$x = \frac{2}{\pi} \sin^{-1}(2\hat{u} - 1) \tag{42}$$

   where the function $\sin^{-1} x$ is defined in $-\pi/2 \le x \le \pi/2$.

5. $\rho(x) = k(x - a)(b - x)$, $a \le x \le b$.

$$x = x_0 + W \sin\left(\tfrac{1}{3} \sin^{-1}(2\hat{u} - 1)\right) \tag{43}$$

   where $x_0 \equiv (a + b)/2$, $W \equiv b - a$, and where the function $\sin^{-1} x$ is defined in $-\pi/2 \le x \le \pi/2$. Note that, if $b = -a = 1$, you get $\rho = k(1 - x^2)$, in which case

$$x = 2 \sin\left(\tfrac{1}{3} \sin^{-1}(2\hat{u} - 1)\right) \tag{44}$$

6. $\rho(x) = kx^{p-1}e^{-x^p}$, $0 \le x < \infty$.
   Conditions: $p \ne 0$.

$$x = (-\ln \hat{u})^{1/p} \tag{45}$$

7. $\rho(x) = kx^{p-1}e^{-x}$, $0 \le x < \infty$.
   Conditions: $p > 0$.

$$x = P^{-1}(p, \hat{u}) \tag{46}$$

   where $P^{-1}(p, x)$ is the functional inverse (in $x$) of the incomplete gamma function $P(p, x)$ defined by

$$P(p, x) = \frac{1}{\Gamma(p)} \int_0^x dt\, t^{p-1}e^{-t}, \quad p > 0,\ x \ge 0 \tag{47}$$

   If the desired interval in $x$ is $x_1 \le x \le x_2$ where $0 \le x_1 < x_2 < \infty$, then

$$x = P^{-1}\left[p, (P_2 - P_1)\hat{u} + P_1\right] \tag{48a}$$

$$P_i \equiv P(p, x_i), \quad i = 1, 2 \tag{48b}$$

8. $\rho(x) = ke^{-(x-x_0)/c}$, $x_0 \le x < \infty$.
   Conditions: $c > 0$. This is a special case of either of the above two examples, obtained by setting $x_0 = 0$ and $p = 1$, from which we have $(x - x_0)/c = -\ln \hat{u}$, hence

$$x = x_0 - c \ln \hat{u} \tag{49}$$

   If the desired interval in $x$ is $x_1 \le x \le x_2$, then

$$x = x_0 - c \ln\left[(E_2 - E_1)\hat{u} + E_1\right] \tag{50a}$$

$$E_i \equiv e^{-(x_i - x_0)/c}, \quad i = 1, 2 \tag{50b}$$

9. $\rho(x) = kx^{\mu-1}(1-x)^{\nu-1}$, $0 \le x \le 1$.

   Conditions: $\mu, \nu > 0$.

$$x = \beta^{-1}(\hat{u}, \mu, \nu) \tag{51}$$

   where $\beta^{-1}(x, \mu, \nu)$ is the functional inverse (in $x$) of the normalized incomplete beta function $\beta(x, \mu, \nu)$, defined by

$$\beta(x, \mu, \nu) = \frac{\Gamma(\mu+\nu)}{\Gamma(\mu)\Gamma(\nu)} \int_0^x dt\, t^{\mu-1}(1-t)^{\nu-1}\,, \qquad 0 \le x \le 1,\ \mu > 0,\ \nu > 0 \tag{52}$$

   If the desired interval in $x$ is $x_1 \le x \le x_2$ where $0 \le x_1 \le x_2 \le 1$, then

$$x = \beta^{-1}[(\beta_2 - \beta_1)\hat{u} + \beta_1, \mu, \nu] \tag{53a}$$
$$\beta_i \equiv \beta(x_i, \mu, \nu)\,, \quad i = 1, 2 \tag{53b}$$

## V. THE SCALING RELATION.

A fundamental scaling relation in all of physics expresses the fact that any physically acceptable function $F(x)$ of a physical (ie., dimensionful) variable $x$ must be of the form

$$F(x) = S^p \times G\left(\frac{x - x_0}{S}\right) \tag{54}$$

where $x_0$ is the origin (or reference point) for $x$, $S$ the units of $x$ (or the parameter that determines the units of $x$), $p$ is the dimensionality of $F(x)$, and $G(\cdot)$ is a dimensionless function of the dimensionless variable $(x-x_0)/S$. The above simply expresses the fact that the origin and scale of any physical variable can be chosen arbitrarily without changing the physics.

Probability distributions, in their general form, obey a similar scaling relation that can be used to generalize, albeit slightly, the algorithm to generate random numbers therefrom. Perhaps the simplest and most intuitive consequence of this generalization is the following rather obvious fact: if $x$ is a random number generated in $(-\infty, +\infty)$ out of a distribution with RMS $\sigma = 1$ and peak $x_0 = 0$, then the random number $y$ corresponding to the same distribution for arbitrary $\sigma$ and $x_0$ is simply given by

$$y = x_0 + \sigma x \tag{55}$$

where $\sigma$ plays the role of $S$. The simple transformation (55) is clearly not valid when $\sigma = \infty$. Nor is it valid when the desired interval for the random numbers is a finite domain $[x_1, x_2]$ because this interval changes to $[x_0 + \sigma x_1, x_0 + \sigma x_2]$, which is different from the original one. In this case, the generalized algorithm, which leaves the interval $[x_1, x_2]$ invariant, requires the derivation of the scaling relation. We thus focus attention on the generic form of a distribution density, namely $\rho = \rho(x; x_0, S)$. This form makes explicit the fact that, in addition to the variable $x$, $\rho$ depends on the two parameters $x_0$ and $S$, where $x_0$ is a reference value for $x$ (ie., a choice of origin), and $S$ sets the scale for $x$. For example, for the Gaussian distribution $\rho = k\exp[-(x-c)^2/2\sigma^2]$, $x_0$ might be chosen to be the location of the peak ($x_0 = c$) and $S$ might be the RMS ($S = \sigma$). For the Lorentzian distribution, $\rho = k/[(x-c)^2+\gamma^2]$, we might choose $x_0 = c$ and $S = \gamma$ (in this case $\sigma = \infty$ hence $\sigma$ is not available as a scale parameter). For the parabolic distribution, $\rho = k(x-c)(d-x)$, with $c \le x \le d$, we might choose $x_0 = c$ (the left edge) and $S = d - c$ (the full width). Any such distribution will be of the form

$$\rho(x; x_0, S) = \frac{dN}{dx} = \frac{k}{S} G\left(\frac{x - x_0}{S}\right) \tag{56}$$

where the prefactor $1/S$ accounts for the fact that $dN/dx$ has dimensions of $1/x$, and $k$ is an unimportant dimensionless normalization constant. On a computer, the random numbers $x$ would be generated in the interval $[x_1, x_2]$ via a call to a function $R$,

$$x = R(x_1, x_2, x_0, S) \tag{57}$$

For example, Eq. (36a) says that, for a gaussian distribution the $R$-function is given by

$$R(x_1, x_2, x_0, \sigma) = x_0 + \sqrt{2}\sigma\, \mathrm{erf}^{-1}[(P_2 - P_1)\hat{u} + P_1] \tag{58}$$

where the $P_i$'s are given by (36b).

Suppose now that we make the variable shift $x \to y = a + bx$ where $a$ and $b$ are arbitrary constants. Then Eq. (56) yields

$$\frac{dN}{dy} = \frac{k}{bS} G\left(\frac{y - y_0}{bS}\right) \tag{59}$$

where $y_0 \equiv a + bx_0$. This implies that $dN/dx$ is invariant under the simultaneous shifts

$$x \to y = a + bx \tag{60a}$$
$$S \to S' = bS \tag{60b}$$

To obtain the consequences on the $R$-function of the fundamental scale invariance of (56) under (60a-60b), we note that Eq. (59) implies that the $y$'s are generated via the operation

$$y = R(y_1, y_2, y_0, bS) \tag{61}$$

or, using $y = a + bx$,

$$R(y_1, y_2, y_0, bS) = a + bR(x_1, x_2, x_0, S) \tag{62}$$

ie.,

$$R(a + bx_1, a + bx_2, a + bx_0, bS) = a + bR(x_1, x_2, x_0, S) \tag{63}$$

Using the arbitrariness of $a$ and $b$ we define $a = -a'/b'$ and $b = 1/b'$ and plug this into both sides of (63). Upon relabeling $(a', b') \to (a, b)$, we obtain the fundamental scaling relation for the $R$-function

$$R(x_1, x_2, x_0, S) = a + bR[(x_1 - a)/b, (x_2 - a)/b, (x_0 - a)/b, S/b] \quad (64)$$

For the particular case of a distribution of finite RMS $\sigma$, (64) yields a particularly useful result by making the choices $a = x_0$ and $b = S = \sigma$. In this case Eqs. (57) and (64) yield

$$x = x_0 + \sigma R[(x_1 - x_0)/\sigma, (x_2 - x_0)/\sigma, 0, 1] \quad (65)$$

and

$$R(x_1, x_2, x_0, \sigma) = x_0 + \sigma R[(x_1 - x_0)/\sigma, (x_2 - x_0)/\sigma, 0, 1] \quad (66)$$

respectively. These equations provide the algorithm to generate random numbers with a distribution with peak at $x = x_0$ and RMS $\sigma$ out of the distribution with peak at $x = 0$ and $\sigma = 1$ *in the same interval* $[x_1, x_2]$: simply replace, in the $R$-function for the latter,

$$x \to (x - x_0)/\sigma \quad (67a)$$

$$x_i \to (x_i - x_0)/\sigma\,, \quad i = 1, 2 \quad (67b)$$

For example, gaussian random numbers in $x_1 \le x \le x_2$ are generated from a distribution with 0 mean and unit RMS according to Eqs. (36a-36b),

$$x = R(x_1, x_2, 0, 1) \equiv \sqrt{2}\,\mathrm{erf}^{-1}[(P_2 - P_1)\hat{u} + P_1] \quad (68)$$

where $P_i = \mathrm{erf}(x_i/\sqrt{2})$, $i = 1\,, 2$. According to the general formula (65) (or (67)), all we have to do to find the algorithm to generate the random numbers in case the gaussian has RMS $\sigma$ and is centered at $x_0$ is to replace $x \to (x - x_0)/\sigma$ and $x_i \to (x_i - x_0)/\sigma$ in (68). Indeed, these replacements yield precisely the general algorithm (36a)-(36b).

Some of the distributions in Sec. IV are not of the generic form (56) because either $x_0$ or $S$ or both have fixed numerical values. However, in all such cases it is straightforward to generalize $\rho$ to arbitrary $x_0$ and $S$ as follows: it is clear from the above arguments that $x_0$ need not have the meaning of being the peak of the distribution, nor an arbitrary origin for $x$. Indeed, $x_0$ may be any significant place in the distribution that serves as a reference, or origin, for $x$. For example, for a bimodal distribution, $x_0$ might be chosen to be, say, the right peak. Or, for a finite-extent distribution, $x_0$ might be the left edge, etc. For example, general form of the simple exponential distribution $\rho = ke^{-x}$ is $\rho = ke^{-(x-x_0)/c}$, where $x \ge x_0$ and $c > 0$; this expression is of the form (56) and its corresponding $R$-function, Eqs. (50a-50b), can be easily shown to obey the scaling relation (64).

Similarly, we may generalize the power distribution (8) to have an arbitrary left edge $x_0$, ie.

$$\rho = k(x - x_0)^p\,, \qquad x \ge x_0 \quad (69)$$

In this case (14) generalizes to

$$R(x_1, x_2, x_0, S) = x_0 + \left[\{(x_2 - x_0)^{p+1} - (x_1 - x_0)^{p+1}\}\hat{u} + (x_1 - x_0)^{p+1}\right]^{1/(p+1)} \quad (70)$$

where it is understood that $x_0 < x_1 < x_2$. Note that the power distribution is scale invariant, hence $S$ does not appear in the $R$-function. It is straightforward to verify that Eq. (70) obeys the scaling relation (64) for arbitrary $a$ and $b$.

## VI. LINEAR SUPERPOSITION OF TWO OR MORE DISTRIBUTIONS.

Suppose now that the problem is the following: generate a set of $m$ random numbers $\{x\}$ such that, in the limit $m \to \infty$, the $x$'s are distributed according to

$$\frac{dN}{dx} = \rho_1(x) + \rho_2(x)\,, \quad x_1 \le x \le x_2 \quad (71)$$

where $\rho_1$ and $\rho_2$ are given. In general, the $\rho_i$'s are defined over different intervals, which we call $[x_{1,1}, x_{2,1}]$ and $[x_{1,2}, x_{2,2}]$, respectively, and the interval $[x_1, x_2]$ is the union of the two, $[x_1, x_2] = [x_{1,1}, x_{2,1}] \cup [x_{1,2}, x_{2,2}]$. The four $x_{i,j}$'s may be different (they may be finite or infinite), and the two intervals may or may not intersect.

The technique described here amounts to generating the $x$'s from either $\rho_1$ or $\rho_2$, with specific weights to be determined. It is particularly useful,therefore, when one is able use the inversion algorithm described in Sec. III to generate random numbers individually from $\rho_1$ and $\rho_2$, but not from the sum $\rho_1 + \rho_2$. For example, $\rho_1(x)$ might be $k_1 x^p$ in $2 \le x \le 5$ and $\rho_2$ might be $k_2 e^{-cx}$ in $0 \le x \le \infty$: in this case it is straightfoward to find $y(x) = \int_0^x dx' \rho(x')$ in analytic form, but it is not possible to invert this function analytically.

Define the normalization integrals

$$K_i = \int_{x_{1,i}}^{x_{2,i}} dx\, \rho_i(x)\,, \quad i = 1, 2 \quad (72)$$

and the weights

$$w_i = \frac{K_i}{K_1 + K_2}\,, \quad i = 1, 2 \quad (73)$$

so that $w_i > 0$ and $w_1 + w_2 = 1$. Then the algorithm to generate $x$ is the following:

1. Generate a uniform random number $\hat{u}$ in $[0, 1]$.
2. If $0 \le \hat{u} < w_1$, generate $x$ with density $\rho_1(x)$ in $[x_{1,1}, x_{2,1}]$.

3. Otherwise, if $w_1 \le \hat{u} < w_1 + w_2 = 1$, generate $x$ with density $\rho_2(x)$ in $[x_{1,2}, x_{2,2}]$.

To prove the validity of the algorithm, one starts from the well-known theorem for joint probability distributions:

**Theorem 1** *Suppose that one generates a random number $x_1$ with probability distribution $\rho_1$ and a random number $x_2$ with probability distribution $\rho_2$. Suppose that one combines $x_1$ and $x_2$ via a given function, $x = f(x_1, x_2)$. Then the probability distribution of $x$ is given by*

$$\rho(x) = \int dx_1 dx_2 \, \rho_1(x_1)\rho_2(x_2)\delta[x - f(x_1, x_2)]$$

This theorem has the obvious generalization. If one combines $n$ random variables $x_1, \ldots, x_n$ via $x = f(x_1, \ldots, x_n)$, where $x_i$ is generated from the distribution $\rho_i(x_i)$ $(i = 1, \ldots, n)$, then the probability distribution of $x$ is given by

$$\begin{aligned}\rho(x) = \int dx_1 \cdots dx_n \, \rho_1(x_1) \cdots \rho_n(x_n) \\ \times \delta[x - f(x_1, \ldots, x_n)]\end{aligned} \tag{74}$$

To prove that the three-step algorithm above yields Eq. (71) for the distribution of the $x$'s, we first translate the algorithm into a joint distribution with a given combination function, then apply Theorem 1 to obtain $\rho(x)$. We first define the functions $\theta_i(x)$ in $0 \le x \le 1$ via

$$\theta_1(x) = \begin{cases} 1 & \text{if } 0 \le x < w_1 \\ 0 & \text{otherwise} \end{cases} \tag{75a}$$

$$\theta_2(x) = \begin{cases} 1 & \text{if } w_1 \le x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{75b}$$

and consider the joint distribution of *three* random variables, namely $x_1$ (generated from $\rho_1(x_1)$), $x_2$ (generated from $\rho_2(x_2)$), and $u$ (generated from $U(u)$) combined via the function

$$x = f(x_1, x_2, u) = \theta_1(u)x_1 + \theta_2(u)x_2 \tag{76}$$

Clearly, Eq. (76) is the mathematical expression of the algorithm. Substituting (76) into (74) for $n = 3$ yields

$$\begin{aligned}\rho(x) = &\int dx_1 dx_2 \int_0^1 du \, \rho_1(x_1)\rho_2(x_2) \\ &\times \delta(x - \theta_1(u)x_1 - \theta_2(u)x_2) &\text{(77a)}\\ = &\int dx_1 dx_2 \, \rho_1(x_1)\rho_2(x_2) \left[ \int_0^{w_1} du \, \delta(x - x_1) \right. \\ &\left. + \int_{w_1}^1 du \, \delta(x - x_2) \right] &\text{(77b)}\\ = &\int dx_1 dx_2 \, \rho_1(x_1)\rho_2(x_2) \, [w_1\delta(x - x_1) \\ &+ w_2\delta(x - x_2)] &\text{(77c)}\\ = &k(\rho_1(x) + \rho_2(x)) &\text{(77d)}\end{aligned}$$

which proves Eq. (71) (in (77d) $k = K_1K_2/(K_1 + K_2)$ is an unimportant normalization constant).

The algorithm generalizes in a straightforward way to the case of a sum of $n$ distributions: to generate a set of random numbers $\{x\}$ such that

$$\frac{dN}{dx} = \rho_1(x) + \cdots + \rho_n(x) \tag{78}$$

proceed as follows: define the normalization integrals

$$K_i = \int_{x_{1,i}}^{x_{2,i}} dx \, \rho_i(x) \,, \quad i = 1, \ldots, n \tag{79}$$

and the weights

$$w_i = \frac{K_i}{K_1 + \cdots + K_n} \,, \quad i = 1, \ldots, n \tag{80}$$

so that $w_i > 0$ and $w_1 + \cdots + w_n = 1$. Define the partial sums $s_i$ as follows:

$$\begin{aligned} s_0 &= 0 &\text{(81a)}\\ s_1 &= w_1 &\text{(81b)}\\ s_2 &= w_1 + w_2 &\text{(81c)}\\ &\vdots \\ s_n &= w_1 + w_2 + \cdots + w_n = 1 &\text{(81d)}\end{aligned}$$

so that $s_0 < s_1 < \cdots < s_n = 1$. Then the algorithm to generate $x$ is the following:

1. Generate a uniform random number $\hat{u}$ in $[0, 1]$.

2. Find the interval within the $s$'s where $\hat{u}$ belongs, i.e., find the value of $i$ for which $s_{i-1} \le \hat{u} < s_i$.

3. Generate $x$ with density $\rho_i(x)$.

## VII. OTHER APPLICATIONS OF THEOREM 1.

Theorem 1 and its generalization (74) allows a simple way to generate random numbers with (possibly) complicated distributions. For the simplest case in which $n = 1$, the algorithm yields

$$\rho(x) = \int dx_1\, \rho(x_1)\delta(x - f(x_1)) = \left.\frac{\rho(x_1)}{|f'(x_1)|}\right|_{x_1=f^{-1}(x)} \tag{82}$$

which is equivalent to the inversion method described in Sec. III.

### Examples.

For each case, we specify the value of $n$, the distribution(s) $\rho_i$, the combination function $f$, and any restriction on the function parameters.

1. $n = 1$, $\rho_1(x) = U(x)$, $f(x) = x^{1/p}$, $p > 0$:

$$\rho(x) = \int_0^1 dx_1\, \delta(x - x_1^{1/p}) = px^{p-1}U(x) \tag{83}$$

2. $n = 1$, $\rho_1(x) = U(x)$, $f(x) = x^{-1/p}$, $p > 0$:

$$\rho(x) = \int_0^1 dx_1\, \delta(x - x_1^{-1/p}) = \theta(x-1)|p|x^{-p-1} \tag{84}$$

3. $n = 1$, $\rho_1(x) = U(x)$, $f(x) = -\ln x$:

$$\rho(x) = \int_0^1 dx_1\, \delta(x + \ln x_1) = \theta(x)e^{-x} \tag{85}$$

4. $n = 1$, $\rho_1(x) = U(x)$, $f(x) = (-\ln x)^{1/p}$:

$$\rho(x) = \int_0^1 dx_1\, \delta(x - (-\ln x_1)^{1/p}) = \theta(x)|p|x^{p-1}e^{-x^p}\,, \quad p \neq 0 \tag{86}$$

5. $n = 1$, $\rho_1(x) = U(x)$, $f(x) = \ln[x/(1-x)]$:

$$\rho(x) = \int_0^1 dx_1\, \delta(x - \ln[x_1/(1-x_1)]) = \frac{1}{4\cosh^2(x/2)}\,, \quad -\infty < x < \infty \tag{87}$$

6. $n = 1$, $\rho_1(x) = U(x)$, $f(x) = -\ln[4x(1-x)]$:

$$\rho(x) = \int_0^1 dx_1\, \delta[x + \ln\{4x_1(1-x_1)\}] = \theta(x)\frac{e^{-x/2}}{4\sqrt{e^x - 1}} \tag{88}$$

(note that $\rho(x)$ has an integrable divergence as $x \to 0^+$, namely $\rho(x) \sim x^{-1/2}$).

7. $n = 2$, $\rho_1(x) = \rho_2(x) = U(x)$, $f(x_1, x_2) = x_1x_2$:

$$\rho(x) = \int_0^1 dx_1 dx_2\, \delta(x - x_1x_2) = -U(x)\ln x \tag{89}$$

8. $n = 2$, $\rho_1(x) = \rho_2(x) = U(x)$, $f(x_1, x_2) = x_1^{1/p}x_2^{1/q}$, $p, q > 0$:

$$\rho(x) = \int_0^1 dx_1 dx_2\, \delta(x - x_1^{1/p}x_2^{1/q}) = -U(x)\frac{pq(x^{p-1} - x^{q-1})}{p-q} \tag{90}$$

In the special case $p = q$ you get

$$\rho(x) = \int_0^1 dx_1 dx_2\, \delta(x - (x_1x_2)^{1/p}) = -U(x)p^2x^{p-1}\ln x \tag{91}$$

9. $n = 2$, $\rho_1(x) = \rho_2(x) = U(x)$, $f(x_1, x_2) = x_1 - x_2$:

$$\rho(x) = \int_0^1 dx_1 dx_2\, \delta(x - x_1 + x_2) = \begin{cases} 1 + x & \text{for } -1 < x < 0 \\ 1 - x & \text{for } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{92}$$

10. $n = 2$, $\rho_1(x) = \rho_2(x) = U(x)$, $f(x_1, x_2) = x_1 + x_2$:

$$\rho(x) = \int_0^1 dx_1 dx_2\, \delta(x - x_1 - x_2) = \begin{cases} x & \text{for } 0 < x < 1 \\ 2 - x & \text{for } 1 < x < 2 \\ 0 & \text{otherwise} \end{cases} \tag{93}$$

11. $n = 2$, $\rho_i(x) = \exp(-x^2/2\sigma_i^2)/\sqrt{2\pi}\sigma_i$ $(i = 1, 2)$,

$f(x_1,x_2)=x_1+x_2$:

$$\begin{aligned}\rho(x)=&\frac{1}{2\pi\sigma_1\sigma_2}\int_{-\infty}^{\infty}dx_1dx_2\,\delta(x-x_1-x_2)\\&\times e^{-x_1^2/2\sigma_1^2-x_2^2/2\sigma_2^2}\\=&\frac{e^{-x^2/2\Sigma^2}}{\sqrt{2\pi}\Sigma}\,,\quad \Sigma=(\sigma_1^2+\sigma_2^2)^{1/2}\end{aligned}\tag{94}$$

12. $n=2$, $\rho_i(x)=\exp(-x^2/2\sigma_i^2)/\sqrt{2\pi}\sigma_i$ $(i=1,2)$, $f(x_1,x_2)=|x_1x_2|^{1/p}$, $p\neq 0$:

$$\begin{aligned}\rho(x)=&\frac{1}{2\pi\sigma_1\sigma_2}\int_{-\infty}^{\infty}dx_1dx_2\,\delta(x-|x_1x_2|^{1/p})\\&\times e^{-x_1^2/2\sigma_1^2-x_2^2/2\sigma_2^2}\\=&\frac{|px|^{p-1}}{\pi\sigma_1\sigma_2}K_0\left(\frac{|x|^p}{\sigma_1\sigma_2}\right)\end{aligned}\tag{95}$$

where $K_0$ is the usual modified Bessel function.

13. $n=3$, $\rho_1(x)=\rho_2(x)=\rho_3(x)=U(x)$, $f(x_1,x_2,x_3)=x_1x_2x_3$:

$$\rho(x)=\int_0^1 dx_1dx_2dx_3\,\delta(x-x_1x_2x_3)=\tfrac{1}{2}U(x)\ln^2 x\tag{96}$$

14. $n=2$, $\rho_1(x)=\rho_2(x)=U(x)$, $f(x_1,x_2)=(x_1+x_2)/2$:

$$\begin{aligned}\rho(x)=&\int_0^1 dx_1dx_2\,\delta(x-(x_1+x_2)/2)\\=&\begin{cases}4x, & 0<x<1/2\\ 4(1-x), & 1/2<x<1\end{cases}\end{aligned}\tag{97}$$

15. $n=3$, $\rho_1(x)=\rho_2(x)=\rho_3(x)=U(x)$, $f(x_1,x_2,x_3)=(x_1+x_2+x_3)/3$:

$$\begin{aligned}\rho(x)=&\int_0^1 dx_1dx_2dx_3\,\delta(x-(x_1+x_2+x_3)/3)\\=&\begin{cases}27x^2/2, & 0<x<1/3\\ 9(6x-6x^2-1)/2, & 1/3<x<2/3\\ 27(1-x^2)/2, & 2/3<x<1\end{cases}\end{aligned}\tag{98}$$

16. $n=n$, $\rho_i(x)=\exp(-x^2/2\sigma_i^2)/\sqrt{2\pi}\sigma_i$ $(i=1,\ldots,n)$, $f(x_1,\ldots,x_n)=(x_1+\cdots+x_n)/n$:

$$\begin{aligned}\rho(x)=&\int_{-\infty}^{\infty}dx_1\cdots dx_n\,\rho_1(x_1)\cdots\rho_n(x_n)\\&\times\delta(x-(x_1+\cdots+x_n)/n)\\=&\frac{e^{-x^2/2\Sigma^2}}{\sqrt{2\pi}\Sigma}\,,\quad \Sigma=\frac{1}{n}(\sigma_1^2+\cdots+\sigma_n^2)^{1/2}\end{aligned}\tag{99}$$

Note that if all the $\sigma_i$'s are equal, you get $\Sigma=\sigma/\sqrt{n}$. This is perhaps the simplest example of the central limit theorem.

## VIII. TRICKS SPECIAL TO THE GAUSSIAN DISTRIBUTION.

The unique property of the exponential function $f(x+y)=f(x)f(y)$ allows a simple way to generate Gaussian random numbers in 1D by generating 2D normally distributed random numbers $(x,y)$, which are simpler to generate than in 1D, and then simply selecting either $x$ or $y$. This is clearly seen from the 2D distribution

$$\frac{dN}{dxdy}=ke^{-(x^2+y^2)/2}\tag{100}$$

and noting that its 1D projection, say onto the $x$-axis, is

$$\begin{aligned}\frac{dN}{dx}&=\int dy\frac{dN}{dxdy}\\&=k'e^{-x^2/2}\end{aligned}\tag{101}$$

where $k'$ is an unimportant constant.

### A. The basic 2D algorithm [1].

By going to cylindrical coordinates $(x,y)=r(\cos\phi,\sin\phi)$ with $0\le\phi<2\pi$ and $0\le r<\infty$, Eq. (100) becomes

$$\frac{dN}{dxdy}=\frac{dN}{rdrd\phi}=ke^{-r^2/2}\tag{102}$$

Defining $t\equiv r^2/2$, this equation is further simplified to

$$\frac{dN}{dtd\phi}=ke^{-t}\tag{103}$$

which means that $\phi$ and $t$ are decoupled: $\phi$ is generated uniformly in $[0,2\pi)$ and $t$ is generated in $[0,\infty)$ with distribution $e^{-t}$. These actions are implemented by the following algorithm:

1. Generate $\phi$ in $[0,2\pi)$ with uniform density, ie., $\phi=2\pi\hat{u}_1$, where $\hat{u}_1$ is a uniform random number in $[0,1]$.

2. Generate $t$ in $[0,\infty)$ with density $e^{-t}$, ie., $t=-\ln\hat{u}_2$ (cf. Eq. 49), where $\hat{u}_2$ is another uniform random number in $[0,1]$.

3. Compute $x=\sqrt{2t}\cos\phi$ and/or $y=\sqrt{2t}\sin\phi$.

Both $x$ and $y$ are normally distributed about 0 with unit standard deviation.

### B. The Box-Muller 2D algorithm [2].

The above basic algorithm has the disadvantage that it requires the evaluation of trigonometric functions, hence it might be too slow in applications dominated by the generation of random numbers. The trick to get around this deficiency is the following: make the change of variables $(x, y) \to (x', y')$ defined by

$$(x, y) = g(s) \times (x', y') \tag{104}$$

where $s \equiv x'^2 + y'^2$, and the function $g(s)$ is to be determined. The Jacobian of this transformation is given by

$$J = \frac{\partial(x, y)}{\partial(x', y')} = g^2 + 2sg\frac{dg}{ds} = \frac{d(sg^2)}{ds} \tag{105}$$

hence Eq. (100) becomes

$$\frac{dN}{dx'dy'} = ke^{-sg^2/2} \times \frac{d(sg^2)}{ds} \tag{106}$$

The simplest algorithm to generate $(x', y')$ is obviously given by $dN/dx'dy' =$ constant, which implies that $x'$ and $y'$ are to be generated uniformly in a 2D domain that is yet to be specified. The uniformity condition yields the elementary differential equation

$$\frac{d(sg^2)}{ds} = k'e^{sg^2/2} \tag{107}$$

whose solution is

$$g(s) = \sqrt{\frac{-2\ln(k_1 s + k_2)}{s}} \tag{108}$$

where $k_2$ is an integration constant and $k_1 = -k'/2$.

Defining $r^2 = x^2 + y^2$, we obtain

$$r^2 = sg^2 = -2\ln(k_1 s + k_2) \tag{109}$$

The constants $k_1$ and $k_2$ are determined by the choice of domain in the $x'-y'$ plane that gets mapped onto the entire $x-y$ plane according to Eq. (104). Now Eq. (104) also maps circles around the origin in the $x-y$ plane to circles around the origin in the $x'-y'$ plane. Since $x$ and $y$ must range in $(-\infty, +\infty)$, we conclude that the points $(x', y')$ must be allowed in all four quadrants. Since $0 \le r^2 < \infty$, we conclude from (109) that $0 < k_1 s + k_2 \le 1$ hence $s$ is allowed to range only in a finite interval. Furthermore, the function $r(s)$ is $1 \leftrightarrow 1$. Combining these arguments, we conclude that the most general acceptable domain for $(x', y')$ is an annulus defined by $s_1 < s < s_2$, where $0 < s_1 < s_2$. The constants $k_1$ and $k_2$ can be readily expressed in terms of the endpoints $s_1$ and $s_2$ and these, in turn, can be computed by the requirement that $s = s_1$ map onto $r = \infty$ in the $x-y$ plane, and that $s_2$ map onto $r = 0$, or viceversa. However, a bit of hindsight quickly shows that by far the simplest algorithm arises when the domain for $(x', y')$ includes the origin, ie., $s_1 = 0$, which implies $k_2 \ge 0$. The upper limit $s_2$ can be chosen to be any positive number, but the simplest choice is $s_2 = 1$, so that Eq. (104) maps the unit disc in the $x' - y'$ plane to the entire $x - y$ plane, and the function $r(s)$ is monotonically decreasing. Requiring that $r = 0$ for $s = 1$ and $r = \infty$ for $s = 0$ yields $k_1 = 1$ and $k_2 = 0$, hence

$$g(s) = \sqrt{\frac{-2\ln s}{s}} \tag{110}$$

The algorithm, therefore, is the following:

1. Generate two uniform random numbers $(x', y')$ in $[-1, +1]$, ie., $x' = 2\hat{u}_1 - 1$ and $y' = 2\hat{u}_2 - 1$, where $\hat{u}_1$ and $\hat{u}_2$ are uniform random numbers in $[0, 1]$.
2. Compute $s = x'^2 + y'^2$. If $s > 1$, reject $x'$ and $y'$ and go back to step 1. Repeat until you get $s \le 1$.
3. Compute $g(s)$ according to Eq. (110), and then $x = g(s)x'$ and $y = g(s)y'$.

The execution of this algorithm yields *two* random numbers ($x$ and $y$) normally distributed with zero mean and unit standard deviation. On the other hand, the algorithm is not 100% efficient owing to the possible rejection of $(x', y')$ in step 2: the acceptance rate (efficiency) is $\pi/4 \simeq 78.5\%$ (= the ratio of the area of the unit circle to the area of the square of side 2), but since you get two random numbers in one shot, the overall efficiency of the algorithm is $1 - (1 - \pi/4)/2 \simeq 89.3\%$ which, depending on the particular application, might be advantageous vis-à-vis the computation of two trigonometric function in the basic algorithm of Sec. VIII A.

### C. Caveat.

These 2D algorithms are simple to implement and work well when the desired range for the random numbers is $(-\infty, +\infty)$ (or half of this interval, say $(0, \infty)$). Unfortunately, they are not successfully modified when the desired range is a given, finite, interval $[x_1, x_2]$. One can always, of course, use the algorithms as described above and simply reject the $x$'s that fall outside $[x_1, x_2]$, but this introduces a possibly unacceptable inefficiency.

Consider the basic 2D algorithm of Sec. VIII A. Depending on the interval $[x_1, x_2]$, there are three sub-cases: (a) $0 \le x_1 < x_2$; (b) $x_1 < x_2 \le 0$; and (c) $x_1 < 0$, $x_2 > 0$. For case (a), a bit of algebra and a sharp pencil yield the following algorithm:

1. Generate $\phi$ uniformly[4] in $[0, \pi]$, ie. $\phi = \pi\hat{u}_1$ where

[4] If the reader attempts to reproduce this derivation, he/she should be aware that, in the $x-y$ plane, this $\phi$ is measured relative to the semiaxis $(x = 0, y \le 0)$ rather than to the conventional semiaxis $(y = 0, x \ge 0)$.

$\hat{u}_1$ is a uniform random number in $[0,1]$; compute $\sin\phi$.

2. Compute $r_i = x_i/\sin\phi$ for $i = 1, 2$.

3. Compute $E_i = e^{-r_i^2/2}$ for $i = 1, 2$.

4. Compute $x = (\sin\phi)\times[-2\ln\{(E_2-E_1)\hat{u}_2+E_1\}]^{1/2}$, where $\hat{u}_2$ is another uniform random number in $[0,1]$.

What's the deficiency of this algorithm? It is 100% efficient and, mathematically, yields the desired result. The problem is with numerical round-off: even if the $x_i$'s are relatively small numbers, ie. of order of a few, the $r_i$'s can easily be so large (due to the $\sin\phi$ denominator) that, even if using double precision, the $E_i$'s will easily underflow (or be set to 0), effectively yielding a distribution for the $x$'s that is substantially non-gaussian or that may have the incorrect limits $[x_1, x_2]$.

The algorithms for cases (b) and (c) are minor variants of the above,[5] but the underflow problem is equally severe. This underflow problem is probably the only disadvantage of this 2D algorithm over the erf/erf$^{-1}$ algorithm, Eq. (36).

We have not succeeded in finding the generalization of the Box-Muller algorithm applicable to the case of a given, finite, interval, but we have reasons to believe that, at least in the general case, it is not possible to express such an algorithm in terms of simple analytic expressions.

## IX. THE MONTE-CARLO METHOD.

Suppose you want to generate random numbers with a given, regular (i.e., nondivergent), distribution density function $\rho(x)$ in a given interval $x_1 \le x \le x_2$. Say that $M$ maximizes $\rho(x)$ in the interval $[x_1, x_2]$. The algorithm is:

1. Generate a random number $x$ uniformly distributed in $[x_1, x_2]$, i.e., $x = (x_2 - x_1)\hat{u}_1 + x_1$.

2. Generate a random number $y$ uniformly distributed in $[0, M]$, i.e., $y = M\hat{u}_2$.

3. Compute $\rho(x)$; if $y \le \rho(x)$, accept $x$, otherwise reject it and go back to Step 1.

Step 3 implies that $x$ is accepted with probability $\rho(x)/M$ which implies that the accepted $x$'s are distributed according to $dN/dx \propto \rho(x)$, justifiying the validity of the algorithm. Its efficiency $\epsilon$ is given by

$$\epsilon = \frac{\int_{x_1}^{x_2} dx\, \rho(x)}{(x_2 - x_1)M} \tag{111}$$

[5] I'll be happy to provide the fortran code for the three cases, but the user would be foolish to use them.

Note that $M$ doesn't have to be the exact maximum of $\rho(x)$: as long as $M \ge \rho(x)$ in $[x_1, x_2]$ the method works. However, the tighter the bound $M$ is the more efficient the algorithm. Figure 1 illustrates the method.

This technique works for any function $\rho(x)$ whose analytic form may be complicated or nonexistent. It has the advantages of simplicity, simple coding, and straightforward generalization to higher dimensions.

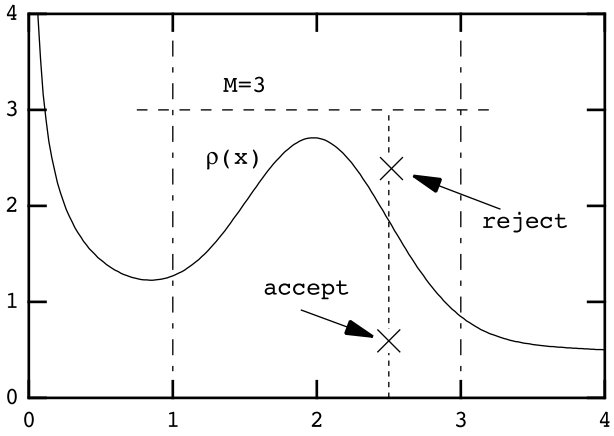


FIG. 1: The Monte Carlo method of generating random numbers $x$ in the range $1 \le x \le 3$ with a probability distribution $\rho(x) = x^{-1/2} + 2e^{-2(x-2)^2}$. The approximate maximum in this interval is $M = 3$.

### A. Refinements of the Monte Carlo algorithm.

The disadvantage of the Monte Carlo method is that it can be very inefficient when $\rho(x)$ is highly peaked, especially when the interval $[x_1, x_2]$ is large compared to the width of the peak (or peaks) of $\rho(x)$. For example, for the case shown in Fig. 1, if $x_1$ were 0.01 instead of 1, we would be forced to choose $M \simeq 10$ just to capture all possible values of $x$ near $x = 0$, resulting in a very low value for $\epsilon$. There are various techniques that ameliorate this inefficiency.

#### 1. *Splitting $\rho(x)$.*

In case $\rho(x)$ has a high, sharp, peak plus a low, smooth, background in $[x_1, x_2]$, it is far more efficient to split $\rho(x)$ into the sum of a peaked plus a background pieces, $\rho = \rho_p + \rho_b$, and then apply the technique described in Sec. VI. This splitting allows the application of the Monte Carlo algorithm to $\rho_p$ and $\rho_b$ separately, typically resulting in a much higher efficiency than for the sum $\rho_p + \rho_b$.

### 2. Partial integration.

Suppose that you want to generate random numbers $x$ in an interval $[0, x_2]$ and the function $\rho(x)$ has an integrable divergence at $x = 0$, i.e., it is of the form $\rho(x) = x^{-p} g(x)$ with $p < 1$ and $g(0) \neq 0$, and where $g(x)$ is regular and smooth in $[0, x_2]$. In this case $M = \infty$, hence the Monte Carlo algorithm is not applicable. However, the form $dN/dx \propto x^{-p} g(x)$ suggests a change of variables similar to that in Sec. II, given by $x^{-p} dx = \lambda dy$ where $\lambda > 0$ is an unimportant proportionality constant. This yields $dN/dy \propto g(x)$, which is amenable to the basic Monte Carlo method of Sec. IX.

Integrating $x^{-p} dx = \lambda dy$ yields, up to an irrelevant additive constant,

$$x = [(1-p)\lambda y]^{1/(1-p)} \tag{112}$$

Since $1-p > 0$, this transformation maps $x = 0$ to $y = 0$. Without any loss in generality we choose $(1-p)\lambda = 1$ so that the top value of $y$ is given by $y_2 = x_2^{1-p}$. The algorithm is therefore:

1. Generate a random number $y$ uniformly distributed in $[0, y_2]$, i.e. $y = y_2 \hat{u}_1$.
2. Compute $x = y^{1/(1-p)}$ and $g(x)$.
3. Generate a random number $z$ uniformly distributed in $[0, M]$ where $M$ is the maximum of $g(x)$ in $0 \leq x \leq x_2$, i.e. $z = M\hat{u}_2$.
4. If $z \leq g(x)$, accept $x$, otherwise reject it and go back to Step 1.

The efficiency of this algorithm is determined by the smoothness of $g(x)$. If the singularity is not at $x = 0$ but somewhere else in the interval $x_1 \leq x \leq x_2$, the method may require splitting the interval into two sections and then applying the technique described in Sec. VI.

### 3. Hybrid Monte Carlo-inversion method.

Suppose you can find a function $M(x)$ such that: (a) $M(x) \geq \rho(x)$ in $x_1 \leq x \leq x_2$; (b) $M(x)$ is amenable to the inversion method described in Sec. III (we are assuming that $\rho(x)$ is not amenable to this inversion method); and (c) $M(x)$ reasonably follows the shape of $\rho(x)$ in $[x_1, x_2]$. Then the efficiency of the algorithm in Sec. IX can be greatly improved.

The algorithm in this case is the following:

1. Generate a random number $x$ distributed with density $M(x)$ in $[x_1, x_2]$ by the inversion method (Sec. III).
2. Compute $M(x)$ and $\rho(x)$.
3. Generate a random number $y$ uniformly distributed in $[0, M(x)]$ i.e., $y = M(x)\hat{u}$.
4. If $y \leq \rho(x)$ accept $x$; otherwise reject it and go back to Step 1.

Figure 2 shows an example of $\rho(x)$ and $M(x)$ for which the technique is applicable. Step 1 implies that the tentative $x$'s are generated with distribution $dN/dx \propto M(x)$, while Steps 3-4 imply that the acceptance probability is $\rho(x)/M(x)$, hence the resultant distribution of the accepted $x$'s is given by

$$\frac{dN}{dx} \propto M(x) \times \frac{\rho(x)}{M(x)} = \rho(x) \tag{113}$$

which justifies the validity of the algorithm. Its efficiency is given by

$$\epsilon = \frac{\int_{x_1}^{x_2} dx\, \rho(x)}{\int_{x_1}^{x_2} dx\, M(x)} \tag{114}$$

which is closest to 100% the closer $M(x)$ follows $\rho(x)$ in $[x_1, x_2]$ subject to the constraint $M(x) \geq \rho(x)$.

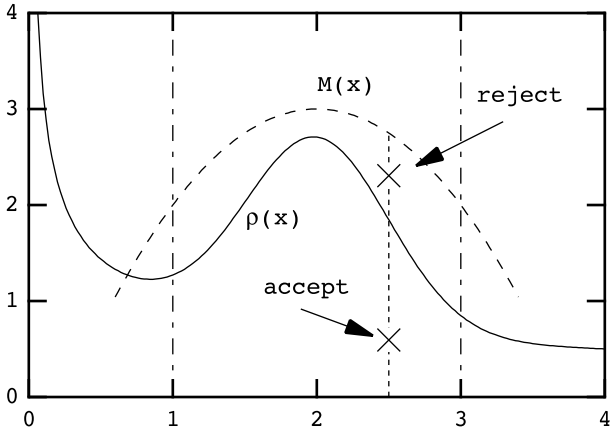


FIG. 2: The hybrid Monte Carlo-inversion method of generating random numbers $x$ in the range $1 \leq x \leq 3$. The parabolic function $M(x)$ satisfies $M(x) > \rho(x)$ in this interval.

**Acknowledgments**

I am grateful to J. Crittenden for bringing to my attention the 2D algortihms for generating gaussian random numbers, and for many discussions. I am grateful to M. Blaskiewicz for bringing to my attention the algorithm for efficiently generating random numbers out of a linear superposition of two or more distributions.

[1] See, eg., L. Devroye, *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, 1986. A discussion appears in the Wikipedia, `http://en.wikipedia.org/wiki/BoxMullertransform#cite_note-3`.

[2] G. E. P. Box and Mervin E. Muller, "A Note on the Generation of Random Normal Deviates," The Annals of Mathematical Statistics **29**(2), 610611 (1958).

## DISCLAIMER