Computer Generation of Poisson Deviates from Modified Normal Distributions

J. H. AHRENS
University of Kiel, West Germany
and
U. DIETER
Technical University, Austria

Samples from Poisson distributions of mean $\mu \geq 10$ are generated by truncating suitable normal deviates and applying a correction with low probability. For $\mu < 10$, inversion is substituted. The method is accurate and it can cope with changing parameters μ. Using efficient subprograms for generating uniform, exponential, and normal deviates, the new algorithm is much faster than all competing methods.

Categories and Subject Descriptors: G.3 [**Mathematics of Computing**]: Probability and Statistics—*random number generation*; *statistical software*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Poisson distribution, acceptance-rejection method

1. INTRODUCTION

In 1969 D. E. Knuth proposed the following research problem [8, Sect. 3.4.1, Ex. 22]: "Can the exact Poisson distribution for large μ be obtained by generating an appropriate normal deviate, converting it to an integer in some convenient way, and applying a (possibly complicated) correction a small percent of the time?"

We are going to solve this exercise for all Poisson distributions with mean $\mu \geq 10$—in the case of smaller $\mu < 10$, a simple inversion method is substituted in our proposed new Algorithm PD.

Since the right-hand tail of a Poisson distribution does not fit under any normal density, an acceptance-rejection method would have to use a normal "hat" covering the bulk of the Poisson distribution, and a separate majorizing function for large arguments. A diploma thesis at Kiel University did not overcome the technical difficulties of this approach; the good fit of the normal envelope was upset by tedious initializations and case distinctions.

This research was supported by the Austrian Research Council (Fonds zur Förderung der wissenschaftlichen Forschung).

Authors' addresses: J. H. Ahrens, Department of Mathematics, University of Kiel, D 2300 Kiel, Olshausenstrasse 40-60, West Germany; U. Dieter, Institute of Statistics, Technical University, A 8010 Graz, Hamerlinggasse 6, Austria.

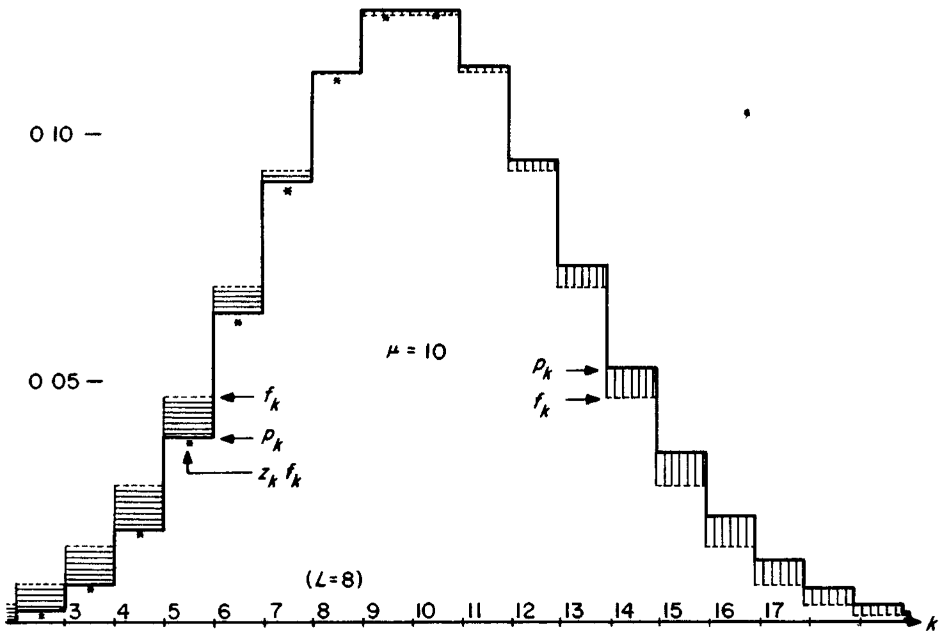

Figure 1

In [5] we dodged the problem by using double exponential hats covering all binomial and Poisson distributions. The Algorithm BP in [5] leads to constant computation times for large parameters. For Poisson distributions with potentially variable means μ, this seems to be the most efficient method to date.

However, quite recently we developed a very successful sampling procedure for gamma distributions [6] in which we modified J. von Neumann's acceptance-rejection method. This new approach is now applied to Poisson distributions.

In Figure 1 the case $\mu = 10$ is displayed. The probability function p_k of the Poisson distribution is compared with a suitable probability density f_k of a "discrete normal distribution" (dotted lines). The f_k are defined as integrals over equal intervals of the standard normal probability density function. Since $\sum p_k = \sum f_k = 1$, we must expect that $p_k < f_k$ for some k but $p_k \geq f_k$ for others. This situation will not be mended: no scaling factor $\alpha > 1$ is applied such that $p_k < \alpha f_k$ becomes true for more k. The f_k are not even the best overall discrete normal approximations to the p_k. Instead they are contrived such that $p_k < f_k$ for all $k < m$ and $p_k \geq f_k$ for all $k \geq m$, where $m \leq L = \lfloor \mu - 1.1484 \rfloor$ if $\mu \geq 10$.

The new method starts with a standard normal deviate T which is transformed quickly into a sample $K \leftarrow \lfloor \mu + \sqrt{\mu} T \rfloor$ from a discrete normal distribution. If $K \geq L$, we know that $p_K \geq f_K$ and accept K *immediately* as a Poisson (μ) variate (Case I). Otherwise we perform the usual acceptance-rejection test: a uniform deviate is compared with p_K/f_K. The calculations of p_K (Section 2) and f_K (Section 3) are involved, but in most cases they are avoided through a simple *squeeze* function (Section 4) $z_K \leq p_K/f_K$ (Case S). The asterisks (*) in Figure 1 depict the products $z_K f_K$, and illustrate the tightness of the squeeze. If the comparison with

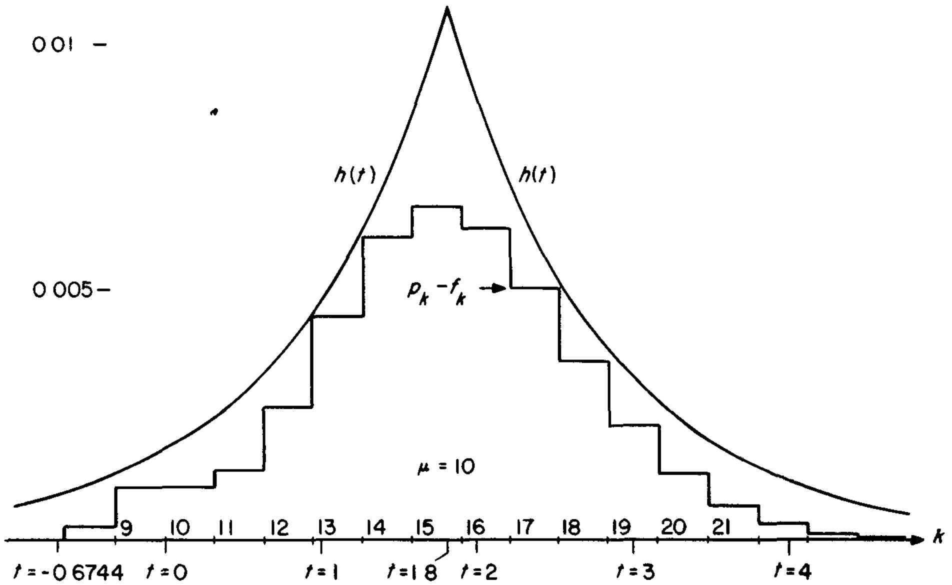

Figure 2

z_K does not lead to acceptance, the *quotient* p_K/f_K has to be worked out; the probability of still accepting K will be rather small (Case Q).

Whenever K is finally rejected, it must be replaced with a new sample, and this has to be from the difference distribution whose probability function is proportional to $p_K - f_K(K \geq m)$. Thus the rejected excess on the left (horizontal shades in Figure 1) is transformed to the defect on the right (vertical shades) which has the same area. Sampling from the difference distribution will be carried out by means of double exponential *hats* on $p_K - f_K$ (Case H); for $\mu = 10$ the hat is displayed in Figure 2. Fortunately, the resulting more laborious acceptance-rejection test (Section 5) occurs only rarely: see Table II for the probabilities P(I), P(S), P(Q), and P(H) of the four cases.

Finally, we state the Algorithm PD in the style of Knuth [8] (Section 6), report computational experience (Section 7), and include a sample computer program (Section 8). With assembler subprograms for uniform, exponential, and normal deviates this FORTRAN code returns Poisson variates in about twice the time required for a single precision logarithm (ALOG, 50 μs)—three ALOG times if the mean μ is continually changing between calls. But the new algorithm is really designed as part of a machine code sampling package, and our assembler version of Algorithm PD cuts the time down to 50–70 μs, so Poisson sampling becomes almost as fast as taking *one* logarithm.

2. POISSON DISTRIBUTIONS

The Poisson (μ) probability function is given by

$$p_k = \frac{e^{-\mu}\mu^k}{k!} \qquad k = 0, 1, 2, \ldots. \tag{1}$$

Table I

	$\lvert\varepsilon\rvert < 2 \times 10^{-8}$	$\lvert\varepsilon\rvert < 2 \times 10^{-9}$	$\lvert\varepsilon\rvert < 2.5 \times 10^{-10}$
a_0	−0.49999999	−0.500000000	−0.5000000002
a_1	0.33333328	0.333333278	0.3333333343
a_2	−0.25000678	−0.249999856	−0.2499998565
a_3	0.20001178	0.200011780	0.1999997049
a_4	−0.16612694	−0.166684875	−0.1666848753
a_5	0.14218783	0.142187833	0.1428833286
a_6	−0.13847944	−0.124196313	−0.1241963125
a_7	0.12500596	0.125005956	0.1101687109
a_8		−0.114265030	−0.1142650302
a_9			0.1055093006

Note: ε = truncation error.

The p_k are calculated directly from (1) only if k is small. For large k the Stirling approximation

$$\ln k! = \left(k + \frac{1}{2}\right)\ln k - k + \ln\sqrt{2\pi} + \frac{1}{12k} - \frac{1}{360k^3} + \frac{1}{1260k^5} + o(k^{-5}) \tag{2}$$

is used. The resulting expression

$$p_k = \frac{1}{\sqrt{2\pi k}}\exp(k\ln(1+v) - (\mu - k) - \delta), \tag{3}$$

where

$$v = \frac{\mu - k}{k} \quad \text{and} \quad \delta = \frac{1}{12k} - \frac{1}{360k^3} + \frac{1}{1260k^5}, \tag{4}$$

is not prone to floating-point overflow. However, if v is small, the rounding errors of (3) become severe. Therefore, whenever $|v| \le 0.25$ we expand

$$k\ln(1+v) - (\mu - k) = kv^2\left(-\frac{1}{2} + \frac{v}{3} - \frac{v^2}{4} + \frac{v^3}{5} - \cdots\right) = kv^2\phi(v), \tag{5}$$

and approximate $\phi(v)$ by an economized polynomial

$$\phi(v) = \frac{1}{2} + \frac{v}{3} - \frac{v^2}{4} + \frac{v^3}{5} - \cdots \approx \sum_{j=0}^{n} a_j v^j, \tag{6}$$

which conforms to the standard precision accuracy of the computer. Coefficients a_j for 7–10 decimal digits accuracy are listed in Table I.

On our Siemens 7760 computer, with its 24-bit mantissa, the first set of coefficients $a_j (n = 7)$ is sufficient, and (1) is used if $k < 10$ aided by a table of $k!$ for $0 \le k \le 9$. If $k \ge 10$, the last term $1/(1260k^5)$ of δ is smaller than 8×10^{-9}; so it can be ignored in (4). For more accurate floating-point arithmetics the third set of coefficients $a_j (n = 9)$ in Table I and the inclusion of the term $1/(1260k^5)$ in (4) results in truncation errors below 6×10^{-10} if $k \ge 10$.

3. DISCRETE NORMAL DISTRIBUTIONS

Since Poisson (μ) distributions tend to normal distributions with mean μ and standard deviation $s = \sqrt{\mu}$, one can approximate the Poisson probabilities p_k in (1) by the integrals

$$f_k = \frac{1}{\sqrt{2\pi}} \int_t^{t'} \exp\left(-\frac{x^2}{2}\right) dx, \text{ where } \begin{cases} t' = \dfrac{k-\mu+1}{s} \\ t = \dfrac{k-\mu}{s} \end{cases} \text{ and } s = \sqrt{\mu}. \tag{7}$$

These $f_k(-\infty < k < \infty)$ constitute the probability function of a *discrete normal distribution.* The Taylor expansions around the midpoints

$$x = \frac{t+t'}{2} = \frac{k-\mu+\frac{1}{2}}{s} \tag{8}$$

may be expressed in terms of Hermite polynomials $\mathrm{He}_n(x)$. Using

$$\mathrm{He}_n(x) = \frac{(-1)^n Z^{(n)}(x)}{Z(x)}$$

[1, 26.2.3], where $Z(x)$ is the standard normal probability density function, we obtain

$$f_k = \frac{1}{\sqrt{2\pi}} \int_{x-1/2s}^{x+1/2s} \exp\left(-\frac{\xi^2}{2}\right) d\xi = \frac{1}{s\sqrt{2\pi}} \sum_{n=0}^{\infty} \frac{\mathrm{He}_{2n}(x)}{2^{2n}(2n+1)!s^{2n}}. \tag{9}$$

The factors $\mathrm{He}_{2n}(x)$ may be worked out recursively from [1, 22.7.14]:

$$\mathrm{He}_0(x) = 1, \qquad \mathrm{He}_1(x) = x, \qquad \mathrm{He}_{m+1}(x) = x\mathrm{He}_m(x) - m\mathrm{He}_{m-1}(x). \tag{10}$$

Explicitly, (9) reads

$$\begin{aligned} f_k = \frac{1}{s\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)\Bigg(1 &+ \frac{x^2-1}{24s^2} + \frac{x^4-6x^2+3}{1920s^4} \\ &+ \frac{x^6-15x^4+45x^2-15}{322560s^6} \\ &+ \frac{x^8-28x^6+210x^4-420x^2+105}{92897280s^8} \\ &+ \frac{x^{10}-45x^8+630x^6-3150x^4+3725x^2-945}{40874803200s^{10}} + \cdots\Bigg). \end{aligned} \tag{11}$$

We use as many terms as required for the given precision of the computer. In the final method there are only two applications of (11). We shall need the quotients p_k/f_k in cases $\mu \geq 10$ and $K < \lfloor \mu - 1.1484 \rfloor$. Second, when $\mu \geq 10$ and $p_k - f_k > 0$, we consider expressions $(p_k - f_k)/h(t)$, where $h(t)$ is defined as a hat function majorizing the differences $p_k - f_k$. These two quantities are compared with (0, 1)-uniform deviates, and we have to make sure that their absolute errors are small enough.

We established numerically (by means of extensive computer-generated error tables) that the largest errors occur at the smallest mean $\mu = 10$. Let $f_k^{(2n)}$ be the approximation to f_k which is obtained by terminating (11) after the $1/s^{2n}$-term.

Then

$$\varepsilon'(2n) = \max\left\{ \left| \frac{p_k}{f_k} - \frac{p_k}{f_k^{(2n)}} \right| 0 \le k \le \mu \right\}$$

and

$$\varepsilon''(2n) = \max\left\{ \left| \frac{p_k - f_k}{h(t)} - \frac{p_k - f_k^{(2n)}}{h(t)} \right| \text{all } k \text{ for which } p_k - f_k > 0 \right\}$$

are bounded by their values at $\mu = 10$ (for $\mu > 10$ they decline steadily):

$$\varepsilon'(6) < 1.5 \times 10^{-10}, \qquad \varepsilon'(8) < 3.2 \times 10^{-13}, \qquad \varepsilon'(10) < 4.5 \times 10^{-16},$$
$$\varepsilon''(6) < 1.0 \times 10^{-8}, \qquad \varepsilon''(8) < 2.0 \times 10^{-11}, \qquad \varepsilon''(10) < 3.3 \times 10^{-14}.$$

Hence for up to 8 digits precision the first two lines of (11) are sufficient, and we work out $f_k^{(6)}$ in the following way. Whenever the mean μ changes, define

$$\omega = \frac{1}{s\sqrt{2\pi}} = \frac{0.3989422804}{s}, \qquad b_1 = \frac{\frac{1}{24}}{\mu}, \qquad b_2 = \frac{3}{10}\, b_1^2,$$

$$c_3 = \frac{1}{7}\, b_1 b_2, \qquad c_2 = b_2 - 15c_3, \qquad c_1 = b_1 - 6b_2 + 45c_3, \tag{12}$$

$$c_0 = 1 - b_1 + 3b_2 - 15c_3.$$

With these coefficients the approximation to $f_k(x)$ becomes

$$f_k^{(6)}(x) = \exp\left(-\frac{x^2}{2}\right)\omega(((c_3x^2 + c_2)x^2 + c_1)x^2 + c_0). \tag{13}$$

4. COMPARISONS

The Poisson and discrete normal probability functions p_k and f_k are now compared, and a squeeze function $z_k \le p_k/f_k$ is established. For the study of

$$q_k = \ln\frac{p_k}{f_k} = \ln p_k - \ln f_k \qquad \text{and} \qquad q'_k = \frac{dq_k}{dk},$$

k is treated as a continuous variable in accordance with (8):

$$x = \frac{k - \mu + \frac{1}{2}}{s} = -s + \frac{k + \frac{1}{2}}{s}, \qquad \sqrt{\mu} = s.$$

From (1) and (11) we have

$$q_k = -s^2 + k \ln s^2 - \ln k! + \ln(s\sqrt{2\pi}) + \frac{s^2}{2}\left(1 - \frac{k + \frac{1}{2}}{s^2}\right)^2$$
$$- \ln\left(1 + \frac{1}{24}\left(\left(1 - \frac{k + \frac{1}{2}}{s^2}\right)^2 - \frac{1}{s^2}\right)\right.$$
$$\left. + \frac{1}{1920}\left(\left(1 - \frac{k + \frac{1}{2}}{s^2}\right)^4 - \frac{6}{s^2}\left(1 - \frac{k + \frac{1}{2}}{s^2}\right)^2 + \frac{3}{s^4}\right) + \cdots\right).$$

It is easy to verify numerically that q_0 and q_1 are negative within our range $\mu \geq 10$; even $s^2 = \mu > 2$ suffices. For $k \geq 2$ the Stirling approximation (2) to $\ln k!$ yields

$$q_k = -s^2 + \frac{s^2}{2}\left(1 - \frac{k+\frac{1}{2}}{s^2}\right)^2 + k$$
$$+ \left(k + \frac{1}{2}\right)\ln\frac{s^2}{k} - \frac{1}{12k} + \frac{1}{360k^3} - \frac{1}{1260k^5} + \cdots$$
$$- \ln\left(1 + \frac{1}{24}\left(\left(1 - \frac{k+\frac{1}{2}}{s^2}\right)^2 - \frac{1}{s^2}\right)\right.$$
$$\left. + \frac{1}{1920}\left(\left(1 - \frac{k+\frac{1}{2}}{s^2}\right)^4 - \frac{6}{s^2}\left(1 - \frac{k+\frac{1}{2}}{s^2}\right)^2 + \frac{3}{s^2}\right) + \cdots\right).$$

Using $t = (k - \mu)/s = k/s - s$, that is, $k = s^2 + st = s^2(1 + t/s)$, we obtain

$$q_k = st + \frac{1}{2}\left(t + \frac{1}{2s}\right)^2 - \left(s^2 + st + \frac{1}{2}\right)\ln\left(1 + \frac{t}{s}\right) - \frac{1}{12(s^2+st)} + \cdots$$
$$- \ln\left(1 + \frac{1}{24s^2}\left(\left(t + \frac{1}{2s}\right)^2 - 1\right) + \cdots\right) \tag{14}$$

$$q'_k = \frac{1}{s}\frac{dq_k}{dt} = 1 + \frac{t}{s} + \frac{1}{2s^2} - 1 - \frac{1}{2(s^2+st)}$$
$$- \ln\left(1 + \frac{t}{s}\right) + \frac{1}{12(s^2+st)^2} - \cdots$$
$$- \frac{\frac{1}{12s^3}\left(t + \frac{1}{2s}\right)\left(1 + \frac{1}{40s^2}\left(\left(t + \frac{1}{2s}\right)^2 - 3\right) + \cdots\right)}{\left(1 + \frac{1}{24s^2}\left(\left(t + \frac{1}{2s}\right)^2 - 1\right) + \cdots\right)}$$

$$q'_k = \frac{t}{s} - \ln\left(1 + \frac{t}{s}\right) + \frac{t}{2s^2(s+t)} - \frac{t + \frac{1}{2s}}{12s^3} + \frac{1}{12s^4} + o(s^{-4})$$

$$q'_k = \left\{\frac{t}{s} - \ln\left(1 + \frac{t}{s}\right)\right\} + \frac{t(5s-t)}{12s^3(s+t)} + \frac{1}{24s^4} + o(s^{-4}). \tag{15}$$

The expression in curly braces is never negative. The second term in (15) is negative for $t < 0$ since $t > -s$, and it is positive for $0 < t < 5s$. The case $t \geq 5s$ is irrelevant since the second term can dominate the curly bracket only near

$t = 0$. Because of

$$q'_k = \frac{t^2}{2s^2} - \frac{t^3}{3s^3} + \frac{5t}{12s^3} + o(s^{-3})$$
$$= -\frac{t}{3s^3}\left(\left(t - \frac{3s}{4}\right)^2 - \left(\left(\frac{3s}{4}\right)^2 + \frac{5}{4}\right)\right) + o(s^{-3}),$$

q'_k changes sign near

$$t_1 = \frac{3s}{4} - \frac{3s}{4}\sqrt{1 + \frac{20}{9s^2}} \approx -\frac{5}{6s} \quad \text{and near} \quad t_2 = 0.$$

Hence q_k increases for $t < t_1$, it decreases if $t_1 < t < 0$ and it increases again for $t > 0$. Expanding the logarithm in (14) yields

$$q_k = st + \frac{t^2}{2} + \frac{t}{2s} + \frac{1}{8s^2}$$
$$-\left(s^2 + st + \frac{1}{2}\right)\left(\frac{t}{s} - \frac{t^2}{2s^2} + \frac{t^3}{3s^3} - \frac{t^4}{4s^4}\right) - \frac{1}{12s^2} - \frac{t^2 - 1}{24s^2} + o(s^{-2}) \quad (16)$$
$$q_k = \frac{t^3}{6s} + \frac{1}{24s^2}(2 + 5t^2 - 2t^4) + o(s^{-2}),$$

and this approximation of q_k is zero if

$$t_0 = -\frac{1}{(2s)^{1/3}} - \frac{5}{12s} + o(s^{-1}); \qquad k_0 = \mu + st_0 = \mu - \left(\frac{\mu}{2}\right)^{1/3} - \frac{5}{12} + o(1). \quad (17)$$

Furthermore, substituting $t = 0$ into (16), we obtain

$$q_k \approx \left(\frac{1}{12s^2} = \frac{1}{12\mu}\right) > 0 \quad \text{at} \quad t = 0 \text{ (corresponding to } k = \mu). \quad (18)$$

The overall behavior of q_k is now clear: we have $q_k \leq 0$ if $t \leq t_0 < 0$ and $q_k \geq 0$ for all $t \geq t_0$, especially if $t > 0$. Consequently, there is an integer $m(\mu)$ such that $p_k < f_k$ if $k < m$ but $p_k \geq f_k$ if $k \geq m$. For numerical bounds we need a few of the actual differences $p_k - f_k$.

$\mu = 10.0000$	$k = 7$	$t = -0.94868330$	$p_7 - f_7 = -0.00207455$
	$k = 8$	$t = -0.63245553$	$p_8 - f_8 = +0.00022884$
$\mu = 10.1484$	$k = 7$	$t = -0.98830528$	$p_7 - f_7 = -0.00243641$
	$k = 8$	$t = -0.67439813$	$p_8 - f_8 = +0.00000001$
$\mu = 10.1485$	$k = 7$	$t = -0.98833180$	$p_7 - f_7 = -0.00243666$
	$k = 8$	$t = -0.67442619$	$p_8 - f_8 = -0.00000015$

These data are reasonably close to the above approximations: at $\mu = 10.1484$, eq. (17), yields $t_0 \approx -0.6702$ and $k_0 \approx 8.0133$. Now consider $\mu = n + 0.1484$, where $n = 10, 11, 12, \ldots$. Then t_0 decreases and $\mu - k_0$ increases in (17), and therefore we can be certain that

$$p_k > f_k \quad \text{if} \quad \mu \geq 10 \quad \text{and} \quad k \geq L \geq m, \quad \text{where} \quad L = \lfloor \mu - 1.1484 \rfloor \tag{19}$$

$$p_k < f_k \quad \text{if} \quad \mu \geq 10 \quad \text{and} \quad t = \frac{k - \mu}{s} \leq -0.6744. \tag{20}$$

Finally a squeeze function $z_k \leq p_k/f_k$ is constructed. From $q_k = (t^3/6s) + o(s^{-1})$, eq. (16), we conjecture that $z_k \approx \exp q_k \approx \exp(t^3/6s) \approx 1 + (t^3/6s)$ will serve the purpose. Hence we set $z_k = 1 + t^3/Cs$, and from (15) we calculate

$$\begin{aligned} \frac{d}{dk}\left(\ln \frac{p_k}{f_k} - \ln z_k\right) &= \frac{dq_k}{dk} - \frac{1}{z_k}\frac{dz_k}{dk} = q'_k - \frac{1}{sz_k}\frac{dz_k}{dt} \\ &= \frac{t}{s} - \ln\left(1 + \frac{t}{s}\right) + \frac{t(5s - t)}{12s^3(s + t)} - \frac{3t^2}{s(Cs + t^3)} + o(s^{-3}) \\ &= \left\{\frac{t}{s} - \frac{t^2}{2s^2} - \ln\left(1 + \frac{t}{s}\right)\right\} \\ &\quad + \frac{t(5s - t)}{12s^3(s + t)} + \left\{\frac{t^2}{2s^2} - \frac{3t^2}{s(Cs + t^3)}\right\} + o(s^{-3}). \end{aligned} \tag{21}$$

Here the first part

$$\frac{t}{s} - \frac{t^2}{2s^2} - \ln\left(1 + \frac{t}{s}\right) = \frac{t^3}{3s^3}\left(1 - \frac{3}{4}\frac{t}{s} + \frac{3}{5}\frac{t^2}{s^2} - \cdots\right)$$

and the middle term are negative if $t < 0$ and positive if $t > 0$. The last part becomes

$$\frac{t^2}{2s^2} - \frac{3t^2}{s(Cs + t^3)} = \frac{t^2((C - 6)s + t^3)}{2s^2(Cs + t^3)},$$

and for $C = 6$ this is also an odd function of t.

Consequently we replace C with 6 as the best possible constant which guarantees that $\ln(p_k/f_k) - \ln z_k$, eq. (21), decreases when $t < 0$. Hence $z_k/(p_k/f_k)$ increases for negative t, but at $t = 0$ we have $z_k = 1$, $p_k/f_k = \exp q_k > 1$, eq. (18), and $z_k/(p_k/f_k)$ is still smaller than 1. Using $t = (k - \mu)/s$, eq. (7), the final squeeze inequality reads

$$z_k = 1 + \frac{t^3}{6s} = 1 + \frac{(k - \mu)^3}{6\mu^2} \leq \frac{p_k}{f_k}, \qquad \text{if} \quad t \leq 0 \quad (\text{corresponding to } k \leq \mu). \tag{22}$$

5. THE HAT FUNCTION

In order to achieve a good fit to the Poisson distribution we developed the discrete versions of normal distributions in Section 3. An alternative would have been to compare ordinary normal probability densities with "continuous Poisson distributions," that is with densities

$$\psi(x) = \frac{\partial \Gamma(x, \mu)}{\partial \mu}, \qquad \text{where} \quad \Gamma(x, \mu) = \int_0^{\mu} \frac{e^{-t} t^{x-1}}{\Gamma(t)}\, dt.$$

Table II

μ	P(I)	P(S)	P(Q)	P(H)	b	σ	$c\mu$	α	$\hat{\alpha}$
10	0.736455	0.211282	0.008939	0.043324	1.7958	0.9376	0.1103	1.5097	1.5606
15	0 697212	0.260763	0.006848	0.035177	1.7931	0.8535	0.1188	1.4886	1.5693
20	0.672640	0.291613	0.005416	0.030332	1.8023	0.7826	0.1279	1.4759	1.5761
30	0.642500	0.329034	0.003835	0.024631	1.7892	0.8216	0.1199	1.4598	1 5847
50	0.611351	0.367261	0.002379	0 019009	1.7906	0 7778	0.1243	1 4382	1 5906
100	0 579260	0 406141	0.001213	0 013387	1.7875	0 7500	0.1263	1 4152	1.5971
200	0.556231	0.433718	0.000607	0.009443	1.7848	0.7425	0.1258	1.3987	1 6010
500	0.535635	0.458162	0.000242	0.005961	1.7837	0.7234	0.1273	1.3817	1.6040
1000	0.525215	0.470453	0.000121	0.004211	1.7812	0.7273	0 1257	1.3735	1.6054

We decided against this possibility since $\psi(x)$ presented too many numerical problems. Faced with the task of designing an acceptance-rejection method for the differences $p_k - f_k > 0$ (Case H), we should have been consistent by selecting a discrete hat as well, for instance, a two-tailed geometric distribution. But this would have burdened the final algorithm with uncomfortable additional calculations. So we decided on a continuous double exponential (Laplace) hat $h(t)$ which has to majorize the entire *histogram* of $p_k - f_k > 0$, as illustrated in Figure 2 (Section 1) in the case $\mu = 10$.

$$h(t) = c \exp\left(\frac{-\mid t - b \mid}{\sigma}\right) \geq p_k - f_k, \qquad \text{if} \quad t \in \left[\frac{k-\mu}{s}, \frac{k-\mu+1}{s}\right). \tag{23}$$

The optimum parameters b, σ, and c in (23) are the ones that lead to the smallest areas $2c\sigma$ under the hat. They were determined for many μ by a complicated search program. The right-hand part of Table II contains some optimal values of b, σ, and $c\mu$, resulting in the best possible efficiencies $\alpha = 2c\sigma/(s\,\mathrm{P(H)})$; α is the expected number of trials before accepting a truncated Laplace deviate as a sample from the difference distribution proportional to $p_k - f_k$. (The probabilities P(I), P(S), P(Q), and P(H) in the left-hand part of Table II were obtained after tabulations and summations of the p_k, f_k, and z_k.)

For the final algorithm we need *simple* choices of the parameters in (23), and after some experimentation and timing we settled for $b = 1.8$ and $\sigma = 1$ ($\sigma = 1$ saves one multiplication). Therefore c had to be determined such that

$$h(t) = c \exp(- \mid t - 1.8 \mid) \geq p_k - f_k, \qquad \text{if} \quad t \in \left[\frac{k-\mu}{s}, \frac{k-\mu+1}{s}\right) \tag{24}$$

for all $\mu \geq 10$. Large tables of lower bounds

$$a\mu = \max_k \left\{\frac{\mu(p_k - f_k)}{\exp(- \mid t - 1.8 \mid)}\right\} \leq c\mu$$

displayed the same wobbly behavior that is visible in Table II in respect to the optimum parameters b, σ, and $c\mu$. Because, with changing mean μ the critical outer corners of the staircase in Figure 2 move and change their identities. So the tightest bound $a\mu = 0.1068446$ does not occur at $\mu = 10$, but it is the first local maximum near $\mu = 10.464$. Hence $c = 0.1069/\mu$ satisfies (24) safely for all k and for all $\mu \geq 10$.

The final choices $b = 1.8$, $\sigma = 1$, and $c = 0.1069/\mu$ lead to efficiencies $\hat{\alpha}$ (Table II), which are not so good as the optimum values α, particularly when μ is large. But then the probabilities P(H) of the hat case are small and declining. The expected number $\hat{\alpha}$ P(H) of hat calculations per sample is always below 6.8 percent.

6. THE ALGORITHM

With the expositions in Sections 1–5 the formal statement of the new Algorithm PD (below) should be comprehensible. In the main case (A) of medium or large $\mu \geq 10$ step N creates the discrete normal deviate $K = \lfloor \mu + sT \rfloor$ (7), which is accepted immediately in step I if $K \geq L$ (19). The squeeze function is employed in step S: using the (0, 1)-uniform deviate $1 - U$ (instead of U), $1 - U \leq z_K = 1 + (K - \mu)^3/6\mu^2$, eq. (22) translates into $U \geq (\mu - K)^3/d$, where $d = 6\mu^2$. If this fails, K may still be accepted after comparing $1 - U$ with the quotient $p_K/f_K = p_y \exp p_x/f_y \exp f_x$ in step Q. The Poisson parts p_x and p_y are worked out in the procedure F using (1), (3), (4), where $\delta \leftarrow 1/(12K)$, $\delta \leftarrow \delta - 4.8\delta^3$ yields $\delta = 1/(12K) - 1/(360K^3)$, (5) and (6). The discrete normal parts f_x and f_y in F comply with (13) using the coefficients (12) as precalculated in step P.

If K is finally rejected in step Q, the hat case is entered. The double exponential deviate T in step E will rarely be below -0.6744, in which case $p_K - f_K < 0$, eq. (20), allows us to reject immediately and try again. When $T > -0.6744$ holds, the new sample $K = \lfloor \mu + sT \rfloor$ requires another application of the procedure F for the test in step H where rejection is indicated whenever the (0, 1)-uniform deviate $|U|$ is larger than $(p_K - f_k)/h(T)$, eq. (23), or $c|U| > (p_y \exp p_x - f_y \exp f_x)\exp|T - 1.8|$, eq. (24), but $|T - 1.8|$ is the original exponential sample E from step E and $c = 0.1069/\mu$ is precalculated in step P.

In the case (B) of small means $\mu < 10$, table-aided inversion is substituted: the (0, 1)-uniform deviate U in step U is compared with cumulative Poisson probabilities $P_K = p_0 + p_1 + \cdots + p_K$. These P_K are stored (step C) so that they may be used again (step T) provided that μ has not changed in the meantime. The P_K-table is useless if the mean μ shifts after every sample, but in most simulations with variable μ the changes will occur only from time to time. If $U > 0.458 > 0.4579297 = P_9$ (at $\mu = 10$), we know that $K \geq \lfloor \mu \rfloor$ will result. Hence, if $U > 0.458$, the search starts at $\lfloor \mu \rfloor = M$ (at 1 if $\mu < 1$) or at L (current length of the P_K-table), whichever is smaller.

Algorithm PD

Case A. Input: mean $\mu \geq 10$. Output: Poisson deviate K.

Case A requires Table I (coefficients a_i) and a table of $k!$ ($k = 0, 1, \ldots, 9$).
If the mean μ is not the same as before, the following three quantities are recalculated: $s \leftarrow \sqrt{\mu}$, $d \leftarrow 6\mu^2$, and $L \leftarrow \lfloor \mu - 1.1484 \rfloor$ ($\lfloor \cdot \rfloor$ = floor function).

N (*Normal sample*). Generate T (standard normal deviate) and set $G \leftarrow \mu + sT$.
If $G \geq 0$ set $K \leftarrow \lfloor G \rfloor$. In the rare case $G < 0$ immediate rejection is indicated: if $G < 0$ go to P.

I (*Immediate acceptance*). If $K \geq L$ return K.

S (*Squeeze acceptance*). Generate U ((0, 1)-uniform deviate).
If $dU \geq (\mu - K)^3$ return K.

P (*Preparations for Q and H*). If the mean μ has changed since this step P was carried out the last time, the following eight quantities are recalculated: $\omega \leftarrow 1/\sqrt{2\pi}/s$, $b_1 \leftarrow (1/24)/\mu$, $b_2 \leftarrow (3/10)b_1^2$, $c_3 \leftarrow (1/7)b_1b_2$, $c_2 \leftarrow b_2 - 15c_3$, $c_1 \leftarrow b_1 - 6b_2 + 45c_3$, $c_0 \leftarrow 1 - b_1 + 3b_2 - 15c_3$, and $c \leftarrow 0.1069/\mu$.
If $G \geq 0$ apply the procedure F (below) to evaluate p_x, p_y and f_x, f_y.
If $G < 0$ go to E (skip Q).

Q (*Quotient acceptance*). If $f_y(1 - U) \leq p_y \exp(p_x - f_x)$ return K.

E (*Double exponential sample*). Generate E (standard exponential deviate) and U ((0, 1)-uniform deviate). Set $U \leftarrow U + U - 1$ and $T \leftarrow 1.8 + E \operatorname{sgn} U$.
If $T \leq -0.6744$ this step E has to be restarted. Otherwise set $K \leftarrow \lfloor \mu + sT \rfloor$ and apply procedure F to evaluate p_x, p_y and f_x, f_y.

H (*Hat acceptance*). If $c|U| > p_y \exp(p_x + E) - f_y \exp(f_x + E)$ go back to E (reject). Otherwise return K.

F *Procedure F.*

1. Poisson probabilities p_k expressed by p_x and p_y ($p_K = p_y \exp p_x$).
Case $K < 10$: Set $p_x \leftarrow -\mu$ and $p_y \leftarrow \mu^K/K!$ using the table of $K!$.
Case $K \geq 10$: Prepare $\delta \leftarrow 1/(12K)$, $\delta \leftarrow \delta - 4.8\delta^3$ and $V \leftarrow (\mu - K)/K$. Then $p_x \leftarrow K \ln(1 + V) - (\mu - K) - \delta$. However, if $|V| \leq 0.25$ substitute $p_x \leftarrow KV^2 \sum a_j V^j - \delta$ for improved accuracy using coefficients a_j from Table I. Finally, set $p_y \leftarrow 1/\sqrt{2\pi}/\sqrt{K}$.

2. Discrete normal probabilities f_K expressed by f_x and f_y ($f_K = f_y \exp f_x$).
Set $X \leftarrow (K - \mu + 0.5)/s$, $f_x \leftarrow 0.5X^2$ and $f_y = \omega(((c_3X^2 + c_2)X^2 + c_1)X^2 + c_0)$.

Case B. Input: mean $\mu < 10$. Output: Poisson deviate K.

Case B is treated by table-aided inversion, and space must be provided for the cumulative probabilities P_k($k = 1, 2, \ldots, 35$ for up to 9 digits accuracy).
If the mean μ is not the same as before, initialize the following five quantities: $M \leftarrow \max(1, \lfloor\mu\rfloor)$, $L \leftarrow 0$, $p \leftarrow \exp(-\mu)$, $q \leftarrow p$, and $p_0 \leftarrow p$.

U (*Uniform sample*). Generate U ((0, 1)-uniform deviate). Set $K \leftarrow 0$.
If $U \leq p_0$ return K.

T (*Comparison of U with existing table*). If $L = 0$ (empty table of P_k) go to C. Otherwise set $J \leftarrow 1$, but if $U > 0.458$ set $J \leftarrow \min(L, M)$ (because, if $U > 0.458 > P_9$ (at $\mu = 10$) then K will never be below $\lfloor\mu\rfloor$).
For $K \leftarrow J, J + 1, \ldots, L$ do: as soon as $U \leq P_K$ return K.
If this search is unsuccessful and $L = 35$ go back to U. (This is a safety measure: $1 - P_{35} < 0.2 \times 10^{-9}$ for all $\mu < 10$, but rounding errors could still cause an infinite loop.)

C (*Creation of new P_k and comparison with U*). For $K \leftarrow L + 1, L + 2, \ldots, 35$ do: set $p \leftarrow p\mu/K$, $q \leftarrow q + p$, $P_K \leftarrow q$, and as soon as $U \leq q$ set $L \leftarrow K$ and return K.
If this research is unsuccessful then $L \leftarrow 35$ and go back to U (safety).

7. COMPUTATIONAL EXPERIENCE

All inequalities and the accuracy of the calculations were checked out on a Siemens 7760 computer. Several batches of 10,000 Poisson deviates each passed various statistical tests. The FORTRAN and assembler versions of Algorithm PD returned identical sets of samples for each choice of μ, since the same assembler subprograms were used:

T = SNORM(IR) (standard normal deviates),	24 μs, Algorithm FL_5 [3].
E = SEXPO(IR) (standard exponential deviates),	20 μs, Algorithm SA [2].
U = SUNIF(IR) ((0, 1)-uniform deviates),	10 μs.

Table III

μ	ε	1	2	5	$10-\varepsilon$	10	15	20	30	50	100	10^3	10^4	10^5	10^6
FOR FIX	62	77	81	90	108	114	111	111	109	105	101	97	96	95	95
FOR VAR	131	165	190	256	369	174	168	164	160	156	155	149	148	147	148
ASS FIX	36	49	50	56	66	70	67	67	65	61	57	54	52	52	51
ASS VAR	67	109	123	158	215	111	109	106	103	98	97	90	90	89	88

(The parameter IR transmits the current state of our basic generator: IR = IR $\times$ 663608941 (mod 2^{32}), $U = \mathrm{IR}/2^{32}$. IR is initialized to some integer $4m + 1$.) FORTRAN (FOR) and assembler (ASS) times [μs] in Table III were based on 10,000 samples each for fixed (FIX) and variable (VAR) μ; in the latter case, the means were subject to small random oscillations around the table entries μ. In order to predict the performance of PD on other computers, Table III should be compared with Siemens 7760 times for

Y = SQRT(X): 31–32 μs; Y = EXP(X): 48–51 μs; Y = ALOG(X): 47–51 μs.

The claims at the end of the introduction are based on these comparisons. The new method is also much faster than the Algorithm BP in [5]: there the Poisson sampling times stabilized at 390 μs (FOR) and 330 μs (ASS) for large parameters μ. We have no reason to compare PD with the older methods in [7] whose computation times grow with increasing μ.

Naturally, the new algorithm is harder to code, and we think that the inclusion of the FORTRAN FUNCTION KPOISS(IR, A) in Section 8 may be helpful. Moreover, K. D. Kohrt has designed a package of commented assembler routines for SUNIF, SEXPO, SNORM, KPOISS, and SGAMMA (Algorithm GD in [6] for gamma deviates, which is about as fast as PD). These programs will run on all large IBM and IBM-like machines. Listings are available on request from the first author at Kiel University.

8. A FORTRAN PROGRAM

```
      FUNCTION KPOISS(IR,MU)
C
C     INPUT:  IR=CURRENT STATE OF BASIC RANDOM NUMBER GENERATOR
C             MU=MEAN MU OF THE POISSON DISTRIBUTION
C     OUTPUT: KPOISS=SAMPLE FROM THE POISSON-(MU)-DISTRIBUTION
C
      REAL MU, MUPREV, MUOLD
C
C     MUPREV=PREVIOUS MU, MUOLD=MU AT LAST EXECUTION OF STEP P OR B.
C     TABLES: COEFFICIENTS A0-A7 FOR STEP F. FACTORIALS FACT
C
      DIMENSION FACT(10), PP(35)
      DATA MUPREV,MUOLD /0.,0./
      DATA A0,A1,A2,A3,A4,A5,A6,A7 /-.5,.3333333,-.2500068,
     ,.2000118,-.1661269,.1421878,-.1384794,.1250060/
      DATA FACT /1.,1.,2.,6.,2.,120.,720.,5040.,40320.,362880./
C
```

```
C     SEPARATION OF CASES A AND B
C
      IF (MU .EQ. MUPREV) GO TO 1
      IF (MU .LT. 1Ø.Ø) GO TO 12
C
C     C A S E  A. (RECALCULATION OF S,D,L IF MU HAS CHANGED)
C
      MUPREV=MU
      S=SQRT(MU)
      D=6.Ø*MU*MU
      L=IFIX(MU-1.1484)
C
C     STEP N. NORMAL SAMPLE - SNORM(IR) FOR STANDARD NORMAL DEVIATE
C
    1 G=MU+S*SNORM(IR)
      IF (G .LT. Ø.Ø) GO TO 2
      KPOISS=IFIX(G)
C
C     STEP I. IMMEDIATE ACCEPTANCE IF KPOISS IS LARGE ENOUGH
C
      IF (KPOISS .GE. L) RETURN
C
C     STEP S. SQUEEZE ACCEPTANCE - SUNIF(IR) FOR (Ø,1)-SAMPLE U
C
      FK=FLOAT(KPOISS)
      DIFMUK=MU-FK
      U=SUNIF(IR)
      IF (D*U .GE. DIFMUK*DIFMUK*DIFMUK) RETURN
C
C     STEP P. PREPARATIONS FOR STEPS Q AND H. (RECALCULATIONS OF
C             PARAMETERS IF NECESSARY)  .3989423=(2*PI)**(-.5)
C
    2 IF (MU .EQ. MUOLD) GO TO 3
      MUOLD=MU
      OMEGA=.3989423/S
      B1=.4166667E-1/MU
      B2=.3*B1*B1
      C3=.1428571*B1*B2
      C2=B2-15.*C3
      C1=B1-6.*B2+45.*C3
      CØ=1.-B1+3.*B2-15.*C3
      C=.1Ø69/MU
    3 IF (G .LT. Ø.Ø) GO TO 5
C
C             "SUBROUTINE" F IS CALLED (KFLAG=Ø FOR CORRECT RETURN)
C
      KFLAG=Ø
      GO TO 7
C
C     STEP Q. QUOTIENT ACCEPTANCE (RARE CASE)
C
    4 IF (FY-U*FY .LE. PY*EXP(PX-FX)) RETURN
C
```

```
C     STEP E. EXPONENTIAL SAMPLE - SEXPO(IR) FOR STANDARD EXPONENTIAL
C             DEVIATE E AND SAMPLING FROM THE LAPLACE HAT
C
    5 E=SEXPO(IR)
      U=SUNIF(IR)
      U=U+U-1.Ø
      T=1.8+SIGN(E,U)
      IF (T .LE. -.6744) GO TO 5
      KPOISS=IFIX(MU+S*T)
      FK=FLOAT(KPOISS)
      DIFMUK=MU-FK
C
C             "SUBROUTINE" F IS CALLED (KFLAG=1 FOR CORRECT RETURN)
C
      KFLAG=1
      GO TO 7
C
C     STEP H. HAT ACCEPTANCE (E IS REPEATED ON REJECTION)
C
    6 IF (C*ABS(U) .GT. PY*EXP(PX+E)-FY*EXP(FX+E)) GO TO 5
      RETURN
C
C     STEP F. "SUBROUTINE" F. CALCULATION OF PX,PY,FX,FY.
C             CASE KPOISS .LT. 1Ø USES FACTORIALS FROM TABLE FACT
C
    7 IF (KPOISS .GE. 1Ø) GO TO 8
      PX=-MU
      PY=MU**KPOISS/FACT(KPOISS+1)
      GO TO 11
C
C             CASE KPOISS .GE. 1Ø USES POLYNOMIAL APPROXIMATION
C             AØ-A7 FOR ACCURACY WHEN ADVISABLE
C
    8 DEL=.8333333E-1/FK
      DEL=DEL-4.8*DEL*DEL*DEL
      V=DIFMUK/FK
      IF (ABS(V) .LE. Ø.25) GO TO 9
      PX=FK*ALOG(1.Ø+V)-DIFMUK-DEL
      GO TO 1Ø
    9 PX=FK*V*V*(((((((A7*V+A6)*V+A5)*V+A4)*V+A3)*V+A2)*V+A1)*V+AØ)-DEL
   1Ø PY=.3989423/SQRT(FK)
   11 X=(Ø.5-DIFMUK)/S
      XX=X*X
      FX=-.5*XX
      FY=OMEGA*(((C3*XX+C2)*XX+C1)*XX+CØ)
      IF (KFLAG) 4,4,6
C
C     C A S E  B. (START NEW TABLE AND CALCULATE PØ IF NECESSARY)
C
   12 MUPREV=Ø.Ø
      IF (MU .EQ. MUOLD) GO TO 13
      MUOLD=MU
      M=MAXØ(1,IFIX(MU))
      L=Ø
```

```
      P=EXP(-MU)
      Q=P
      PØ=P
C
C     STEP U. UNIFORM SAMPLE FOR INVERSION METHOD
C
   13 U=SUNIF(IR)
      KPOISS=Ø
      IF (U .LE. PØ) RETURN
C
C     STEP T. TABLE COMPARISON UNTIL THE END PP(L) OF THE
C             PP-TABLE OF CUMULATIVE POISSON PROBABILITIES
C
      IF (L .EQ. Ø) GO TO 15
      J=1
      IF (U .GT. Ø.458) J=MINØ(L,M)
      DO 14 KPOISS=J,L
   14 IF (U .LE. PP(KPOISS)) RETURN
      IF (L .EQ. 35) GO TO 13
C
C     STEP C. CREATION OF NEW POISSON PROBABILITIES P
C             AND THEIR CUMULATIVES Q=PP(K)
C
   15 L=L+1
      DO 16 KPOISS=L,35
      P=P*MU/FLOAT(KPOISS)
      Q=Q+P
      PP(KPOISS)=Q
   16 IF (U .LE. Q) GO TO 17
      L=35
      GO TO 13
   17 L=KPOISS
      RETURN
      END
```

Remarks. The FUNCTION KPOISS(IR, MU) is presented with a conversion to *machine code* in mind; therefore low-level FORTRAN was chosen. For the sampling subfunctions SNORM(IR), SEXPO(IR), and SUNIF(IR), compare Section 7.

The constant 35 in the last part (Case B) corresponds to the dimension PP(35) of the table P_k (steps T, C); it is sufficient for up to 9-digit accuracy. If the standard precision of the computer is more than 7-8 decimals, the DATA A0, A1, ... may be modified according to the second block in Table I, and some other constants should be adjusted: $1/\sqrt{2\pi} = 0.398942280$, $1/24 = 0.416666667\text{E}-1$, $1/7 = 0.142857143$, and $1/12 = 0.833333333\text{E}-1$.

ACKNOWLEDGMENT

The authors wish to express their appreciation and thanks to the reviewers of this article for their constructive and valuable comments. Furthermore, they acknowledge stimulating conversations with E. Mayr and K. D. Kohrt.

REFERENCES

(Note. Reference [4] is not cited in the text.)

1. ABRAMOWITZ, M., AND STEGUN, I.A. *Handbook of Mathematical Functions.* Dover, New York, 1972.
2. AHRENS, J.H., AND DIETER, U. Computer methods for sampling from the exponential and normal distributions. *Commun. ACM 15,* 10(Oct. 1972), 873-882.
3. AHRENS, J.H., AND DIETER, U. Extension of Forsythe's method for random sampling from the normal distribution *Math Comput. 27* (1973), 927-937.
4. AHRENS, J.H., AND DIETER, U. Computer methods for sampling from gamma, beta, Poisson and binomial distributions *Computing 12* (1974), 223-246.
5. AHRENS, J.H., AND DIETER, U. Sampling from binomial and Poisson distributions: A method with bounded computation times. *Computing 25* (1980), 193-208.
6. AHRENS, J.H., AND DIETER, U. Generating gamma variates by a modified rejection technique. *Commun. ACM 25,* 1 (Jan. 1982), 47-54.
7. FISHMAN, G.S. *Principles of Discrete Event Simulation.* Wiley, New York, 1978.
8. KNUTH, D.E. *The Art of Computer Programming,* Vol. II: *Seminumerical Algorithms.* Addison-Wesley, Reading, Mass., 1969 and 1981

Received March 1981; revised January 1982; accepted February 1982