Generating Beta Variates Via Patchwork Rejection*

H. Zechner and E. Stadlober, Graz

Dedicated to Professor Ulrich Dieter on the occasion of his 60^{th} *birthday*

October 19, 1992

Abstract — Zusammenfassung

Generating Beta Variates Via Patchwork Rejection. A new algorithm for sampling from beta(p, q) distributions with parameters $p > 1$, $q > 1$ is developed. It is based on a method by Minh [9] which improves acceptance–rejection sampling in the main part of the distributions. Additionally, transformed uniform deviates can often be accepted immediately, so that much fewer than two uniforms are needed for one beta variate, on the average. The remaining tests for acceptance are enhanced by 'squeezes'. Experiments covering a wide range of pairs (p, q) showed improvements in speed over competing algorithms in most cases.

AMS Subject Classification: 65C10, 65C05.

Keywords and Phrases: Random number generation, beta distribution, patchwork rejection.

Erzeugung von beta–verteilten Zufallszahlen mittels Patchwork–Verwerfung. Ein neuer Algorithmus zur Erzeugung von Stichproben aus Beta(p, q)–Verteilungen mit Parametern $p > 1$, $q > 1$ wird entwickelt. Er basiert auf einer Methode von Minh [9], welche die Verwerfungstechnik im Hauptteil der Verteilungen verbessert. Zusätzlich können transformierte gleichverteilte Zufallszahlen oft sofort akzeptiert werden, so daß im Durchschnitt viel weniger als zwei gleichverteilte Zufallszahlen pro Beta–Stichprobe benötigt werden. Die verbleibenden Annahmetests werden durch "Squeezes" verbessert. Versuche über einen weiten (p, q)–Bereich ergaben in den meisten Fällen Geschwindigkeitssteigerungen im Vergleich zu konkurrierenden Methoden.

1 Introduction

The beta distribution is an important tool in stochastic modelling and simulation; it can have a number of different shapes. When parameters p and q are both less than 1, the density is U–shaped, if exactly one parameter is less than 1, the density is J–shaped, and

* This research was supported by the Austrian Research Council (Fonds zur Förderung der wissenschaftlichen Forschung Nr. P8257–PHY).

when both parameters are greater than 1, the density is unimodal. This variety makes the development of sampling methods covering all possible shapes difficult. A survey of existing beta generators is given in the book of Devroye [6]. The algorithms discussed in this article deal with unimodal densities only. For the other cases $p < 1$ and/or $q < 1$ we refer to the papers of Atkinson [2], Atkinson & Whittaker [3], [4], Cheng [5], Jöhnk [8] and Sakasegawa [10].

Competitive methods for $p > 1$ and $q > 1$ are based on acceptance–rejection. The first method of this type is due to Ahrens & Dieter [1], [7]. They constructed a normal envelope which is inefficient in case of skew beta distributions. Uniformly fast algorithms (i. e. algorithms with bounded computation times for all $p > 1$, $q > 1$) have been established in [5], [10], [11]. Cheng [5] applied a log–logistic envelope to the beta distribution of the second kind. Samples from this distribution can be transformed easily to the desired beta variates. The resulting algorithm BB is rather simple, has low set–up costs and reasonable speed. Schmeiser & Babu [11] and Sakasegawa [10] utilized piecewise linear envelopes in the center, and exponentials in the tails. Their sampling routines B4PE and B11 are involved, but they are fast at the expense of a more time–consuming set–up.

Our new sampling method improves acceptance–rejection in the center by using the patchwork–rejection idea of Minh [9]. A certain modification of the beta density (reflection at one point $(x_2,\, f(x_2))$ to the left of the mode and at one point $(x_4,\, f(x_4))$ to the right of the mode) leads to a more efficient sampling process which uses rectangular majorizing and minorizing functions (see Figure 1). The expected number of uniforms needed per variate is bounded by $2\,(2 - e^{-1}) = 3.2642\ldots$ $(p \searrow 2,\, q \to 1$ or $p \to 1,\, q \searrow 2)$, it decreases to $6/\sqrt{2\pi e} = 1.4518\ldots$ as $p,\, q \to \infty$. The experimental results in Section 6 indicate that in terms of execution time the resulting algorithm BPRS compares well with the algorithms mentioned above.

The next section contains a general description of the patchwork–rejection method and its application to the beta distribution. The basic sampling method is described in Section 3. Formal statements of the two new algorithms BPRS and BPRB are given in Section 4. Algorithms B4PE and BPRS are compared with respect to some performance measures in Section 5, and computational experience is reported in the final section. In the appendix the proof of the crucial Lemma 1 of Section 2 is established.

2 The Patchwork–Rejection Method

2.1 General Technique

Let $f(x)$ be proportional to a probability density function which is bounded and unimodal with the mode at m. The main idea of the method is to transform the left part T_l (in Figure 1) of the top region under $f(x)$ into the congruent area $T_l{}'$ by using the point P_l as the center of a reflection σ_l. Similarly the right part T_r of the top region under $f(x)$ is reflected by σ_r at P_r into $T_r{}'$. The points P_l and P_r must be determined carefully, such that the transformed regions $T_l{}'$ and $T_r{}'$ contain no points below $f(x)$. Formally the reflections σ_l and σ_r read

$$\sigma_l:\ (x,\, y)\ \mapsto\ (v,\, w) = (2x_2 - x,\, 2f(x_2) - y), \qquad x_2 < x < m \tag{1}$$

$$\sigma_r:\ (x,\, y)\ \mapsto\ (v,\, w) = (2x_4 - x,\, 2f(x_4) - y), \qquad m < x < x_4\,. \tag{2}$$

$x_1 = 2x_2 - m$ x_2 m x_4 $2x_4 - m = x_5$

Figure 1: Patchwork (Example beta(5, 3.5)–distribution)

There is no overlapping of the transformed region to the left of x_2 (to the right of x_4) with the original area below $f(x)$, if and only if condition (3) (condition (4)) holds:

$$f(x_2 + z) - f(x_2)\ \leq\ f(x_2) - f(x_2 - z)\,, \qquad 0 \leq z \leq m - x_2 \tag{3}$$

$$f(x_4 - z) - f(x_4)\ \leq\ f(x_4) - f(x_4 + z)\,, \qquad 0 \leq z \leq x_4 - m\,. \tag{4}$$

Differentiating (3) and (4) with respect to z and using that the equal sign holds at $z = 0$ yields the sufficient conditions

$$\begin{aligned} f'(x_2 + z) &\leq\ \ f'(x_2 - z)\,, \qquad & 0 \leq z \leq m - x_2 \\ -f'(x_4 - z) &\leq -f'(x_4 + z)\,, \qquad & 0 \leq z \leq x_4 - m\,. \end{aligned}$$

Since $f'(x)$ is positive for $x < m$ and negative for $x > m$, both inequalities contain positive terms only. Taking logarithms, differentiating and using $z = 0$ again leads to the simple conditions

$$\frac{d}{dx}\ln|f'(x)|\,\Big|_{x_2+z} + \frac{d}{dx}\ln|f'(x)|\,\Big|_{x_2-z} \le 0\,, \qquad 0 \le z \le m - x_2 \tag{5}$$

$$\frac{d}{dx}\ln|f'(x)|\,\Big|_{x_4-z} + \frac{d}{dx}\ln|f'(x)|\,\Big|_{x_4+z} \ge 0\,, \qquad 0 \le z \le x_4 - m \tag{6}$$

which will be the basis for the determination of valid 'reflection points' $P_l = (x_2,\, f(x_2))$ and $P_r = (x_4,\, f(x_4))$ in the next subsection.

2.2 Application to the Beta Distribution

Let

$$f_b(x) = x^{p-1}(1-x)^{q-1}/B(p,\, q)\,, \qquad 0 < x < 1, \quad p > 0,\ q > 0, \tag{7}$$

be the density of the beta distribution, where the normalizing constant

$$B(p,\, q) = \int_0^1 x^{p-1}(1-x)^{q-1}\,dx \;=\; \frac{\Gamma(p)\,\Gamma(q)}{\Gamma(p+q)}$$

is the complete beta function. We confine the shape parameters p, q to $p > 1$ and $q > 1$. Then $f_b(x)$ is bounded and it has one mode at $m = (p-1)/(p+q-2)$, $0 < m < 1$. In order to simplify the following calculations we set $a = p-1$, $b = q-1$ and define

$$f_B(x) = x^a(1-x)^b\,, \qquad 0 < x < 1, \quad a > 0,\ b > 0\,. \tag{8}$$

Differentiating $\ln|f_B{}'(x)|$ with respect to x leads to

$$\frac{d}{dx}\ln|f_B{}'(x)| \;=\; \frac{(a+b)(a+b-1)x^2 - 2a(a+b-1)x + a(a-1)}{x(1-x)(a-(a+b)x)}\,, \qquad x \ne \frac{a}{a+b}$$

which may be written as

$$\frac{d}{dx}\ln|f_B{}'(x)| \;=\; \frac{(s-1)x^2 - 2m(s-1)x + m(ms-1)}{x(1-x)(m-x)}\,, \qquad x \ne m = \frac{a}{a+b}, \tag{9}$$

where $s = a+b = a/m = b/(1-m)$.

The reflection point P_l. Inserting (9) into (5) yields, after some straightforward calculations, the following sufficient condition for x_2,

$$A(x_2)z^4 - B(x_2)z^2 + C(x_2) \le 0\,, \qquad 0 \le z \le m - x_2\,, \tag{10}$$

where

$$A(x) = (s-1)(x-(1-m))\,, \tag{11}$$

$$\begin{aligned} B(x) \;&=\; 2(s-1)x^3 - 2(m+1)(s-1)x^2 + \\ &\quad + m((3-m)s-2m)x - m(1-m)(ms+1)\,, \end{aligned} \tag{12}$$

and

$$C(x) = x(1-x)(m-x)((s-1)x^2 - 2m(s-1)x + m(ms-1))\,. \tag{13}$$

After an analytical and numerical examination of the inequality (10) we arrived at the following result which will be proved in the appendix.

Lemma 1 *Let x_{wl} be the left point of inflection of $f_B(x) = x^a(1-x)^b$:*

$$x_{wl} = \frac{1}{a+b}\left(a - \sqrt{\frac{ab}{a+b-1}}\right) = m - \sqrt{\frac{m(1-m)}{s-1}}\,, \quad a = ms \geq 1, \tag{14}$$

where $s = a+b$ and $m = a/(a+b)$. Furthermore let

$$z_2 = x_{wl} - \frac{x_{wl}(1-x_{wl})}{s\,(m-x_{wl})} \tag{15}$$

be the point at which the tangent of $f_B(x)$ to x_{wl} crosses the x–axis. Then candidates for abscissas $x_2 < m$ of reflection points P_l which fulfil the inequality (10) are

(i) $x_2 \geq \frac{m}{2}$, *if* $x_{wl} \leq \frac{m}{2}$ *and* $a \leq 2$ *or, equivalently,*

if $a \leq \min\left(\frac{1-b+\sqrt{b^2+14b+1}}{2}\,,\ 2\right)$.

(ii) $x_2 \geq x_{wl}$, *if* $x_{wl} \geq \frac{m}{2}$ *and* $(s-6)x_{wl} - ms + 3 \leq 0$ *or, equivalently,*

if $a \geq \max\left(\frac{1-b+\sqrt{b^2+14b+1}}{2}\,,\ \frac{-(b+3)^2+(b-3)\sqrt{b^2+54b+9}}{2\,(b-9)}\right)$.

(iii) $x_2 \geq \frac{z_2+m}{2}$, *if* $a > 2$, $x_{wl} < \frac{m}{2}$ *or* $(s-6)x_{wl} - ms + 3 > 0$ *or, equivalently,*

if $2 < a < \frac{-(b+3)^2+(b-3)\sqrt{b^2+54b+9}}{2\,(b-9)}$.

The two sketches in Figure 2 display the three regions of Lemma 1. For $a \geq 9$ the condition in (ii) is always satisfied. Therefore $x_2 = x_{wl}$ (and $x_4 = x_{wr}$) is feasible if $a \geq 9$ (and $b \geq 9$).

Remark. Let ${x_2}^{(min)}$ be the smallest x_2–value which is feasible in respect to the inequality (10). It is plausible that ${x_2}^{(min)}$ is a good choice for the abscissa of P_l. In fact, it was established numerically that the sum $A(x_2) = A_T(x_2) + A_R(x_2)$ of the tail area $A_T(x_2) = \int_0^{2x_2-m} f(x)\,dx$ and the (shaded) rejection area $A_R(x_2) = \int_{2x_2-m}^{x_2}(2f(x_2) - f(2x_2-x) - f(x))\,dx$ increases monotonically between ${x_2}^{(min)}$ and m except for small a below 2.5.

The proof of Lemma 1 in the appendix contains the additional information that ${x_2}^{(min)} = \frac{m}{2}$ in the case (i), and ${x_2}^{(min)} = x_{wl}$ in the case (ii). However, inside the region (iii) the

Figure 2: The three cases of Lemma 1 in the $(a,\, b)$ plane.

safe choice $x_2 = (z_2 + m)/2$ suggested in Lemma 1 is only an upper bound on ${x_2}^{(min)}$. We have no tighter bounds which would be simple enough for the use in the sampling routine.

The reflection point P_r. Interchanging the shape parameters p and q in (7) causes a reflection of the beta density $f_b(x)$ at $x = \frac{1}{2}$, i. e. the random variate X from beta$(p,\, q)$ is changed to $Y = 1 - X$ from beta$(q,\, p)$. Consequently the bounds in Lemma 1 for $x_2 < m$ can be easily modified to equivalent expressions for abscissas $x_4 > m$ of feasible reflection points P_r: interchange a and b (s remains unchanged), replace m by $1 - m$, x_{wl} by $1 - x_{wr}$, z_2 by $1 - z_4$, and x_2 by $1 - x_4$. The transformed bounds yield analogous suggestions for safe, but not always optimal, choices of x_4 which are spelt out under *Initialization* of algorithm BPRS in Section 4.

3 Sampling Method

From now on the rescaled function

$$f(x) = \frac{f_b(x)}{f_b(m)} = \Big(\frac{x}{m}\Big)^{p-1} \Big(\frac{1-x}{1-m}\Big)^{q-1}, \qquad 0 < x < 1, \quad p > 1,\ q > 1, \tag{16}$$

is employed instead of the original density $f_b(x)$. $f(x)$ needs no evaluation of the normalizing constant $B(p,\, q)$, and $f(x) \le f(m) = 1$ holds for all x.

From Section 2 we know that the mode m and the abscissas x_2 and x_4 are important ingredients of the sampling routine as well as $x_1 = 2x_2 - m$ and $x_5 = 2x_4 - m$. x_1 and x_5 split up the area under $f(x)$ into three parts: the *left tail*, the *center*, and the *right tail*. This suggests a sampling routine based on acceptance–rejection with a three part envelope $h(x)$. The graph of $h(x)$ in the center is the upper boundary of T_l', R_2, R_3 and T_r' in Figure 1. The left tail of $f(x)$ over $(0,\, x_1]$ and the right tail of $f(x)$ over $[x_5,\, 1)$ are majorized by exponential functions. The details are given in subsections 3.1 and 3.2.

3.1 Sampling from the Center

Let $f_i = f(x_i),\ i = 1,\,2,\,4,\,5$. The part $h^{(c)}(x)$ of the hat function $h(x)$ which covers the center region — after the reflections at P_l and P_r (Figure 1) — is defined as

$$h^{(c)}(x) = \begin{cases} f_2 = f(x_2) & x_1 < x < m \\ f_4 = f(x_4) & m \le x < x_5\,. \end{cases}$$

Immediate acceptance. Under the graph of $f(x)$ in Figure 1 we find four rectangles which are the 'immediate acceptance regions':

$$R_1 = \{(x,\,y)\,|\,x_1 < x < x_2,\ 0 \le y \le f_1\}, \quad R_2 = \{(x,\,y)\,|\,x_2 \le x < m,\ 0 \le y \le f_2\},$$
$$R_3 = \{(x,\,y)\,|\,m \le x \le x_4,\ 0 \le y \le f_4\}, \quad R_4 = \{(x,\,y)\,|\,x_4 < x < x_5,\ 0 \le y \le f_5\}.$$

For sampling from $\bigcup R_i$ only one uniform deviate U is needed (see algorithm BPRS, Step 2 and Step 4).

Acceptance–Rejection. Let $(X,\,Y)$ be a point under $h^{(c)}(x)$ outside $\bigcup R_i$. Then $X \in (x_1,\,x_2)$ or $X \in (x_4,\,x_5)$. If $Y \le f(X)$ then the point $(X,\,Y)$ is in either Q_l or Q_r in Figure 1, and X can be accepted as a beta variate. Otherwise $(X,\,Y)$ may belong to T_l' or T_r' in which case the test applies to the corresponding point $(V,\,W)$ in T_l or T_r. Since ${\sigma_l}^{-1} = \sigma_l$ and ${\sigma_r}^{-1} = \sigma_r$, the back–transformation can be copied from (1) and (2):

$$(X,\,Y) \mapsto (V,\,W) = (2x_2 - X,\ 2f_2 - Y), \qquad x_1 < X < x_2 \tag{17}$$

$$(X,\,Y) \mapsto (V,\,W) = (2x_4 - X,\ 2f_4 - Y), \qquad x_4 < X < x_5\,. \tag{18}$$

Now V is accepted if $W \le f(V)$. Otherwise the point $(X,\,Y)$ is in the shaded area of Figure 1, rejection is indicated, and a new sample from $h(x)$ is required.

3.2 Sampling from the Tails

The hat function $h(x)$ for the tails is to touch $f(x)$ at x_1 and x_5. This means

$$h(x_1) = f_1\,,\ h'(x_1) = f'(x_1) \quad \text{and} \quad h(x_5) = f_5\,,\ h'(x_5) = f'(x_5)\,. \tag{19}$$

Both tails can be covered by exponential functions which meet the requirements (19):

$$\begin{aligned} h^{(l)}(x) &= f_1\,e^{(x - x_1)/\lambda_l}\,, & -\infty < x \le x_1\,, && \lambda_l &= x_1(1 - x_1)\,/\,(s\,(m - x_1)) \\ h^{(r)}(x) &= f_5\,e^{(x_5 - x)/\lambda_r}\,, & x_5 \le x < \infty\,, && \lambda_r &= x_5(1 - x_5)\,/\,(s\,(x_5 - m))\,. \end{aligned}$$

Their areas are

$$H^{(l)} = \int_{-\infty}^{x_1} h^{(l)}(x)\,dx = f_1\,\lambda_l\,, \qquad H^{(r)} = \int_{x_5}^{\infty} h^{(r)}(x)\,dx = f_5\,\lambda_r\,.$$

Samples X from $h^{(l)}(x)$ and $h^{(r)}(x)$ are obtained by inversion as

$$X = x_1 + \lambda_l\,\log(U) \qquad \text{and} \qquad X = x_5 - \lambda_r\,\log(1 - U)\,.$$

3.3 Squeeze Functions

Since the acceptance tests $W \le f(V)$ and $Y \le f(X)$ use the complicated function $f(x)$, fast pre–tests based on simple lower bounds for $f(x)$ (called squeezes) will be added to our basic sampling routine. We employ linear squeezes which are obtained as follows.

Let $x_1 \le x_{wl}$, $x_5 \ge x_{wr}$. Furthermore, let $z = t(\xi) = \xi - \xi(1-\xi)/(s\,(m-\xi))$ be the point at which the tangent of $f(x)$ to ξ crosses the x–axis, and define

$$z_2 = \begin{cases} 0, & \text{if } a \le 1 \\ t(x_{wl}), & \text{if } a > 1\,, \end{cases} \qquad z_1 = t(x_1) = \begin{cases} 0, & \text{if } a \le 1 \\ x_1 - \lambda_l, & \text{if } a > 1\,, \end{cases}$$

$$z_4 = \begin{cases} 1, & \text{if } b \le 1 \\ t(x_{wr}), & \text{if } b > 1\,, \end{cases} \qquad z_5 = t(x_5) = \begin{cases} 1, & \text{if } b \le 1 \\ x_5 + \lambda_r, & \text{if } b > 1\,. \end{cases}$$

Then the line segments with end points $(z_i,\,0)$, $(x_i,\,f_i)$, $i = 1, 2, 4, 5$, are below $f(x)$. This is obviously valid for $i = 1$ and $i = 5$. For $i = 2$ and $i = 4$ the statement follows from $x_{wl} \le x_2$, $x_4 \le x_{wr}$ (compare the inequality (23) in the appendix) and from the concavity of $f(x)$ in $[x_{wl},\,x_{wr}]$. Consequently the segments with end points $(x_2,\,f_2)$, $(m,\,1)$ and $(m,\,1)$, $(x_4,\,f_4)$ are also below $f(x)$, and the complete piecewise linear squeeze function $L(x) \le f(x)$ reads

$$L(x) = \begin{cases} f_1(x-z_1)/(x_1-z_1), & 0 < x \le x_1 \\ f_2(x-z_2)/(x_2-z_2), & x_1 < x < x_2 \\ f_2 + (1-f_2)(x-x_2)/(m-x_2), & x_2 \le x < m \\ f_4 + (1-f_4)(x_4-x)/(x_4-m), & m \le x \le x_4 \\ f_4(z_4-x)/(z_4-x_4), & x_4 < x < x_5 \\ f_5(z_5-x)/(z_5-x_5), & x_5 \le x < 1\,. \end{cases}$$

Acceptance–rejection of samples $X \in (x_1,\,x_2)$ and $X \in (x_4,\,x_5)$ is modified by $L(x)$ as follows. First of all, the fast pre–test $Y \le L(X)$ is executed. Whenever it fails, $(X,\,Y)$ is back–transformed to $(V,\,W)$. If $W \ge 1$, then V is rejected, since $f(x) \le 1$. Otherwise the fast pre–test $W \le L(V)$ and, if necessary, the slow acceptance test $W \le f(V)$ is carried out. If both tests fail, the sample $(X,\,Y)$ is subjected to the test $Y \le f(X)$ which finally decides on acceptance or rejection of X.

Acceptance–rejection of samples $X \in (0,\,x_1]$ and $X \in [x_5,\,1)$ is also complemented by the fast pre–test $Y \le L(X)$ (Step 6 and Step 7 of the algorithm BPRS).

4 Algorithms BPRS and BPRB

The algorithm BPRS (**B**eta **P**atchwork **R**ejection method with **S**queezes) results directly from the theory in Section 2 and Section 3. The shorter version BPRB without squeezes is obtained from BPRS as follows. Remove all lines that are marked with a '**--**' sign, and change the lines which carry combinations of '∘' and '**•**':

```
○○  Set  D_l = m/2,  x_2 = D_l,  x_1 = 0;
○●  Set  x_1 = x_2 * (1 - (1 - x_2)/(s * (m - x_2)));
●○  Set  D_r = (1 - m)/2,  x_4 = 1 - D_r,  x_5 = 1;
●●  Set  x_5 = x_4 * (1 + (1 - x_4)/(s * (x_4 - m)));
```

Algorithm BPRS $(p > 1 \text{ and } q > 1)$

$f(x, a, b, m) = \exp(a * \log(x/m) + b * \log((1-x)/(1-m)))$

Initialization:

```
        Set  a = p - 1,  b = q - 1,  s = a + b,  m = a/s;
        If  (a > 1  or  b > 1)  then Set  D = sqrt(m * (1 - m) / (s - 1));
        If  (a ≤ 1)  then
○○         Set  D_l = m/2,  x_2 = D_l,  x_1 = z_2 = 0;              Lemma 1, (i)
           Set  f_1 = λ_l = 0;
        else
           Set  D_l = D,  x_2 = m - D_l,  x_1 = x_2 - D_l;          Lemma 1, (ii)
--         Set  z_2 = x_2 * (1 - (1 - x_2)/(s * (m - x_2)));
           If  (x_1 ≤ 0  or  (s - 6) * x_2 - a + 3 > 0)  then
○●            Set  x_1 = z_2;
              Set  x_2 = (x_1 + m)/2,  D_l = m - x_2;              Lemma 1, (i), (iii)
           Endif
           Set  f_1 = f(x_1, a, b, m);
           Set  λ_l = x_1 * (1 - x_1) / (s * (m - x_1));
        Endif                                                      z_1 = x_1 - λ_l
        Set  f_2 = f(x_2, a, b, m);
        If  (b ≤ 1)  then
●○         Set  D_r = (1 - m)/2,  x_4 = 1 - D_r,  x_5 = z_4 = 1;    Lemma 1, (i)
           Set  f_5 = λ_r = 0;
        else
           Set  D_r = D,  x_4 = m + D_r,  x_5 = x_4 + D_r;          Lemma 1, (ii)
--         Set  z_4 = x_4 * (1 + (1 - x_4)/(s * (x_4 - m)));
           If  (x_5 ≥ 1  or  (s - 6) * x_4 - a + 3 < 0)  then
●●            Set  x_5 = z_4;
              Set  x_4 = (m + x_5)/2,  D_r = x_4 - m;              Lemma 1, (i), (iii)
           Endif
           Set  f_5 = f(x_5, a, b, m);
           Set  λ_r = x_5 * (1 - x_5) / (s * (x_5 - m));
        Endif                                                      z_5 = x_5 + λ_r
        Set  f_4 = f(x_4, a, b, m);
        Set  p_1 = f_2 * (m - x_1),  p_2 = f_4 * (x_5 - m) + p_1;
        Set  p_3 = f_1 * λ_l + p_2,   p_4 = f_5 * λ_r + p_3;
```

<u>Generation:</u>

1. Case Distinction:

   `Generate` $U \sim U(0,1)$ `and Set` $U = U * p_4$;
   `If` $(U \leq p_1)$ `then Goto 2.` [Left Center Region];
   `else`
   `If` $(U \leq p_2)$ `then Goto 4.` [Right Center Region];
   `else`
   `If` $(U \leq p_3)$ `then Goto 6.` [Left Tail];
   `else Goto 7.` [Right Tail];
   `Endif`

2. Left Center Region — Immediate acceptance of $X \in (x_1, m)$:

   `Set` $Y = U/D_l - f_2$; $Y \sim U(-f_2, +f_2)$
   `If` $(Y \leq 0)$ `then Deliver` $X = m - U/f_2$; $X \in (x_2, m)$
   `If` $(Y \leq f_1)$ `then Deliver` $X = x_2 - Y/f_1 * D_l$; $X \in (x_1, x_2)$

3. Left Center Region — Squeeze acceptance of $X \in (x_1, x_2)$ or Acceptance–Rejection of its counterpart $V \in (x_2, m)$:

   `Generate` $U \sim U(0,1)$ `and Set` $W = U * D_l$; $W \sim U(0, D_l)$
   `Set` $X = x_2 - W$, $V = x_2 + W$;
   `--` `If` $(Y \leq f_2 * (X - z_2) / (x_2 - z_2))$ `then Deliver` X; $Y \leq L(X)$
   `Set` $W = f_2 + f_2 - Y$; $W \sim U(f_2, 2 f_2 - f_1)$
   `If` $(W < 1)$ `then`
   `--` `If` $(W \leq f_2 + (1 - f_2) * U)$ `then Deliver` V; $W \leq L(V)$
   `If` $(W \leq f(V, a, b, m))$ `then Deliver` V; $W \leq f(V)$
   `Endif`
   `Goto 8.` [Acceptance Test]; Rejection of V

4. Right Center Region — Immediate acceptance of $X \in (m, x_5)$:

   `Set` $Y = (U - p_1)/D_r - f_4$; $Y \sim U(-f_4, +f_4)$
   `If` $(Y \leq 0)$ `then Deliver` $X = m + (U - p_1)/f_4$; $X \in (m, x_4)$
   `If` $(Y \leq f_5)$ `then Deliver` $X = x_4 + Y/f_5 * D_r$; $X \in (x_4, x_5)$

5. Right Center Region — Squeeze acceptance of $X \in (x_4, x_5)$ or Acceptance–Rejection of its counterpart $V \in (m, x_4)$:

   `Generate` $U \sim U(0,1)$ `and Set` $W = U * D_r$; $W \sim U(0, D_r)$
   `Set` $X = x_4 + W$, $V = x_4 - W$;
   `--` `If` $(Y \leq f_4 * (z_4 - X) / (z_4 - x_4))$ `then Deliver` X; $Y \leq L(X)$
   `Set` $W = f_4 + f_4 - Y$; $W \sim U(f_4, 2 f_4 - f_5)$
   `If` $(W < 1)$ `then`
   `--` `If` $(W \leq f_4 + (1 - f_4) * U)$ `then Deliver` V; $W \leq L(V)$
   `If` $(W \leq f(V, a, b, m))$ `then Deliver` V; $W \leq f(V)$
   `Endif`
   `Goto 8.` [Acceptance Test]; Rejection of V

6. Left Tail — Squeeze acceptance of $X \in (-\infty, x_1]$, computed by inversion:

```
    Set U = (U − p2)/(p3 − p2);                              U ~ U(0, 1)
    Set V = log(U), X = x1 + λl * V;
    If (X ≤ 0) then Goto 1. [Continue];                      Fast rejection of X
    Generate Y ~ U(0, 1) and Set Y = Y * U;                  Y ~ U(0, h(X)/f1)
 -- If (Y ≤ 1 + V) then Deliver X;                           Y ≤ L(X)/f1
    Set Y = Y * f1 and Goto 8. [Acceptance Test];            Y ~ U(0, h(X))
 7. Right Tail — Squeeze acceptance of X ∈ [x5, +∞), computed by inversion:
    Set U = (U − p3)/(p4 − p3);                              U ~ U(0, 1)
    Set V = log(U), X = x5 − λr * V;
    If (X ≥ 1) then Goto 1. [Continue];                      Fast rejection of X
    Generate Y ~ U(0, 1) and Set Y = Y * U;                  Y ~ U(0, h(X)/f5)
 -- If (Y ≤ 1 + V) then Deliver X;                           Y ≤ L(X)/f5
    Set Y = Y * f5 and Goto 8. [Acceptance Test];            Y ~ U(0, h(X))
 8. Acceptance Test — Acceptance of X, whenever Y ≤ f(X):
    If (Y ≤ f(X, a, b, m)) then Deliver X;
                           else Goto 1. [Continue];          Rejection of X
```

Remark. In *Initialization* the number of case distinctions is reduced by setting $x_2 = \frac{m}{2}$ for $a \leq 1$ only. For the remaining part of case (i) in Lemma 1 $x_2 = (z_2 + m)/2$ from case (iii) is substituted. This is legitimate, since $(z_2 + m)/2 \geq \frac{m}{2}$ (x_2 remains a valid abscissa of P_l). Equally we substitute $x_4 = (m + z_4)/2$ in case (i) whenever $b > 1$.

5 Characteristics of Algorithms B4PE and BPRS

The fastest existing procedures for generating beta(p, q) variates with $p > 1$ and $q > 1$ are algorithms B4PE (see Schmeiser & Babu [11]) and B11, part of algorithm BSA dealing with the case $p > 1$, $q > 1$ (see Sakasegawa [10]). Sakasegawa's method is based purely on acceptance–rejection, therefore it requires at least two uniforms per beta variate. Algorithm B4PE does contain immediate acceptance with one uniform; for some parameter combinations (p, q) it needs slightly fewer than two uniforms per variate (see Table 2) — we shall see that in this respect our algorithm BPRS is more efficient.

First of all, we need the average number α of samples X from $h(x)$, until X is accepted:

$$\alpha = f_b(m) \left(f_1 \lambda_l + f_2(m - x_1) + f_4(x_5 - m) + f_5 \lambda_r \right). \tag{20}$$

The probability $P(I)$ of accepting X immediately, given that X is accepted, is the rescaled area of the four rectangles R_i:

$$P(I) = f_b(m) \Big((f_1 + f_2)(m - x_2) + (f_4 + f_5)(x_4 - m) \Big). \tag{21}$$

With these two quantities the expected number of uniforms N for one beta variate becomes

$$N = 2\alpha - P(I) - f_b(m) \left(f_1 \lambda_l e^{-x_1/\lambda_l} + f_5 \lambda_r e^{-(1 - x_5)/\lambda_r} \right). \tag{22}$$

For $p \to \infty$ and $q \to \infty$ the characteristics in (21) and (22) tend to

$$\lim_{p,q\to\infty} P(I) = \frac{2}{\sqrt{2\pi}}\left(e^{-1/2} + e^{-2}\right) = 0.5919\ldots, \qquad \lim_{p,q\to\infty} N = \frac{6}{\sqrt{2\pi e}} = 1.4518\ldots.$$

For fixed p and $q \to 1$ we have

$$\lim_{q\to 1} P(I) = \begin{cases} p/2^p & 1 < p \le 2 \\ 0 & 2 < p \end{cases},$$

$$\lim_{q\to 1} N = \begin{cases} 3p/2^p & 1 < p \le 2 \\ p\left(2 - e^{-(p-1)}\right)/(p-1) & 2 < p \end{cases}.$$

The limiting values of $P(I)$ and N are discontinuous at $p = 2$, since the special beta density $\lim_{q\to 1} f_b(x) = p\,x^{p-1}$ changes its curvature in $(0, 1)$ at $p = 2$ from concave to convex.

Our algorithm BPRS is now compared with B4PE in respect to $P(I)$ (Table 1) and N (Table 2).

Table 1: Probability $P(I)$ of accepting X immediately, given that X is accepted.

Algorithm B4PE

$p \backslash q$	1.1	1.5	2.5	5.0	1000
1.1	0.000				
1.5	0.000	0.000			
2.5	0.495	0.347	0.439		
5.0	0.492	0.438	0.510	0.572	
1000	0.460	0.441	0.505	0.565	0.592

Algorithm BPRS

$p \backslash q$	1.1	1.5	2.5	5.0	1000
1.1	0.515				
1.5	0.538	0.551			
2.5	0.628	0.563	0.579		
5.0	0.511	0.608	0.591	0.533	
1000	0.393	0.535	0.589	0.606	0.592

Table 2: Average number N of uniforms needed to generate one beta variate.

Algorithm B4PE

$p \backslash q$	1.1	1.5	2.5	5.0	1000
1.1	2.120				
1.5	2.588	2.546			
2.5	2.022	2.212	2.186		
5.0	2.034	2.069	2.047	1.939	
1000	1.970	2.019	2.009	1.926	1.892

Algorithm BPRS

$p \backslash q$	1.1	1.5	2.5	5.0	1000
1.1	1.545				
1.5	1.613	1.654			
2.5	1.453	1.578	1.622		
5.0	1.724	1.526	1.537	1.486	
1000	1.694	1.576	1.551	1.506	1.452

6 Computational Results

In this section the algorithms BPRB and BPRS are compared with the uniformly fast algorithms BB (Cheng [5]), B11 (Sakasegawa [10]) and B4PE (Schmeiser & Babu [11]) in terms of execution times, initialization times and lengths of compiled codes.

All algorithms were written as C routines, compiled and linked in Turbo C 2.0 with the instruction set of the 80286 and 80287 processors. Uniform random numbers were generated by the multiplicative congruential C–generator `krand` with modulus $m = 2^{54}$ and multiplicator $a = 2783377641436325$ (see Stadlober & Kremer [12]). This generator has a long period of 2^{52} and produces random numbers in double precision, the normal precision of C. On a PC/AT 486–33MHz it takes about 22 μsec to generate one uniform which is approximately the time for calculating $y = e^x$. The elementary operations addition, subtraction and multiplication need about 0.5 μsec. The square root requires about 3.5 μsec, and the logarithm about 16 μsec.

Table 3 displays the execution times (best times are in bold face, second best in italics). The observed times are based on samples of one million variates for each parameter combination. Our algorithm BPRS seems to be faster than its competitors unless one parameter is very small (near 1) and the other one is large (greater than 4). In this case the times of B4PE are slightly better. The execution times of B4PE are very high for $p, q \leq 2$. This is caused by the absence of immediate acceptance regions, and the rather poor fit of the uniform hat function. In this parameter region BPRS is followed by B11 which dominates B4PE if one parameter is not too small and not too large. Our simpler method (without squeezes) BPRB is about 20% slower than BPRS, whereas the generation times of BB exceed the times of the fastest method by factors close to 2.

Initialization times (set–up costs) are important, if the values of p or q change frequently. The figures in the center of Table 4 indicate very low set–up costs for BB. The higher costs for the other four algorithms vary with the size of the shape parameters. Values for three different cases (with 0, 1, and 2 points of inflection) are listed in Table 4; the symbol '$\sim$' signifies average values. Memory requirements are also displayed; it is clear that BB is much simpler than the faster methods BPRB, B4PE, BPRS and B11.

For applications in which speed, a low number of uniforms and even performance are important, our proposal BPRS can be recommended. If simplicity and small set–up time are the dominant considerations, BB could be the first choice.

Table 3: Execution times [μsec/variate] of algorithms BB (Cheng), B11 (Sakasegawa), B4PE (Schmeiser & Babu), BPRB and BPRS (Zechner & Stadlober).

$p \backslash q$	1.1	1.2	1.5	2.0	2.5	3.0	5.0	10.0	1000
1.1	124 86 98 *83* **69**								
1.2	125 86 101 *85* **69**	125 86 102 *85* **70**					1st row: 2nd row: 3rd row: 4th row: 5th row:	BB B11 B4PE BPRB BPRS	
1.5	130 *84* 111 88 **70**	128 *85* 109 88 **71**	127 *86* 111 90 **72**						
2.0	137 *72* 129 91 **62**	133 *76* 123 91 **66**	129 *81* 119 92 **71**	129 *80* 123 94 **71**					
2.5	142 *71* 74 81 **58**	138 *73* 77 86 **59**	132 *76* 87 89 **65**	130 *77* 99 92 **68**	131 *76* 83 91 **67**				
3.0	146 *75* *75* 87 **67**	141 *74* 76 82 **61**	134 *76* 83 88 **62**	131 *76* 93 90 **66**	131 *75* 79 90 **66**	132 *74* 76 89 **65**			
5.0	154 *81* **79** 102 82	149 *79* *79* 92 **72**	140 *78* 82 85 **65**	134 *77* 89 87 **64**	132 *75* 78 87 **63**	132 *74* *74* 88 **62**	133 74 *71* 90 **60**		
10.0	161 *84* **81** 105 85	156 82 *80* 96 **77**	146 *80* 83 88 **70**	139 *79* 89 87 **67**	135 *77* 79 86 **65**	134 *76* *76* 86 **63**	132 74 *71* 85 **61**	134 74 *70* 86 **59**	
1000	169 85 **81** 103 *84*	164 83 *81* 96 **78**	154 *82* 83 90 **72**	145 *80* 89 88 **69**	141 *79* 82 87 **68**	138 *78* 79 86 **66**	133 *76* 77 84 **64**	129 *75* *75* 85 **61**	130 75 *68* 83 **59**

Table 4: Initialization times [μsec] and memory requirements [bytes] of BB (Cheng), B11 (Sakasegawa), B4PE (Schmeiser & Babu), BPRB and BPRS (Zechner & Stadlober).

Algorithm	Initialization times			Memory requirements of compiled code
	$p,\ q \le 2$	$q \le 2 < p$	$p,\ q > 2$	
BB	20	20	20	1711
B11	151	~ 270	~ 387	7930
B4PE	52	~ 185	~ 324	6811
BPRB	135	~ 227	~ 316	6377
BPRS	139	~ 232	~ 324	7181

7 Appendix

The reflection (1), illustrated in Figure 1 of Section 2.1, demands that x_2 has to be at least $\frac{m}{2}$. This guarantees that the transformed top region $T_l{}'$ to the left of x_2 is inside the range $(0,\ 1)$ of the beta density. x_2 also has to be greater than or equal to the left point of inflection x_{wl}. This follows from (10), since for $z = 0$ one has $C(x_2) \le 0$ which is satisfied, if and only if $x_{wl} \le x_2 \le m$. Hence

$$x_2 \ge \max\left(\frac{m}{2}\,,\ x_{wl}\right) \tag{23}$$

is a general precondition for abscissas x_2 of valid reflection points.

Proof of Lemma 1

Case (i):

Let $x_{wl} \le \frac{m}{2}$. We show that $x_2 = \frac{m}{2}$ is feasible for $a \le 2$.
For the proof x_2 is replaced with $\frac{m}{2}$ in (10). After some calculation we arrive at

$$4(s-1)(3m-2)z^2 - m(2-m)(ms+3m-4) \ge 0\,, \qquad 0 \le z \le \frac{m}{2}\,. \tag{24}$$

Now three different cases are considered.

(a) If $s \le 1$, then no point of inflection exists, and $f(x)$ is concave in $(0,\ 1)$. Hence, $x_2 = \frac{m}{2}$ is obviously valid.

(b) If $s > 1$ and $m < \frac{2}{3}$, then the left hand side of (24) is strictly decreasing. Therefore it attains its minimum at the right end point $z = \frac{m}{2}$, and condition (24) reduces to $4m(1-m)(2-ms) \ge 0$. This shows that $x_2 = \frac{m}{2}$ is valid, provided that $ms \le 2$, i. e. $a \le 2$.

(c) If $s > 1$ and $m \geq \frac{2}{3}$, then the left hand side of (24) is increasing. Therefore it attains its minimum at the left end point $z = 0$, and (24) reduces to the single condition $-m(2-m)(ms+3m-4) \geq 0$. It follows that $x_2 = \frac{m}{2}$ is valid, provided that $ms + 3m - 4 \leq 0$. But this condition is equivalent to $x_{wl} \leq \frac{m}{2}$ which is assumed.

Combining the results of (a), (b), and (c) we can state that $x_2 = \frac{m}{2}$ is valid, whenever $x_{wl} \leq \frac{m}{2}$ and $a \leq 2$ or, using (14), whenever

$$a \leq \min\left(\frac{1-b+\sqrt{b^2+14b+1}}{2}\,,\ 2\right), \qquad b > 0\,. \tag{25}$$

If $x_2 = \frac{m}{2}$ is valid, it is relatively easy to see that all x_2 between $\frac{m}{2}$ and m are also feasible. The formal proof is omitted.

Case (ii):

Let $x_{wl} \geq \frac{m}{2}$. Then (23) demands that $x_2 \geq x_{wl}$. Again we only show that $x_2 = x_{wl}$ is valid for the part of the parameter region which is described by condition $(s-6)x_{wl} - ms + 3 \leq 0$. The argumentation is as follows.

If x_{wl} in (14) is substituted for x_2, condition (10) reduces to

$$A(x_{wl})z^2 - B(x_{wl}) \leq 0\,, \qquad 0 \leq z \leq m - x_{wl}\,, \tag{26}$$

since x_{wl} substituted into (13) yields $C(x_{wl}) = 0$. We consider two cases.

(a) $A(x_{wl}) < 0$ or, equivalently from (11) and (14), $x_{wl} < 1 - m$:

The left hand side of (26) is strictly decreasing. Therefore it attains its maximum at the left end point $z = 0$, and (26) reduces to the single condition $-B(x_{wl}) \leq 0$. Using (14) in the form $(s-1){x_{wl}}^2 - 2m(s-1)x_{wl} + m(ms-1) = 0$ finally yields

$$-\,B(x_{wl}) \;=\; m(1-m)\;((s-6)x_{wl} - ms + 3) \;\leq\; 0. \tag{27}$$

(b) $A(x_{wl}) \geq 0$ or, equivalently, $x_{wl} \geq 1 - m$:

Now the left hand side of (26) is increasing. Therefore it attains its maximum at the right end point $z = m - x_{wl}$, and (26) reduces to the single condition

$$A(x_{wl})(m - x_{wl})^2 - B(x_{wl}) \;=\; m(1-m)\;((s-5)x_{wl} - ms + m + 2) \;\leq\; 0 \tag{28}$$

which holds for all $x_{wl} \geq 1 - m$ and $x_{wl} \geq \frac{m}{2}$. This can be easily checked by discussing the two cases $s \geq 5$ and $s < 5$, and by observing that there is no x_{wl} which fulfils $A(x_{wl}) \geq 0$ in the case $m \leq \frac{1}{2}$.

$A(x_{wl}) \geq 0$ implies $-B(x_{wl}) \;\leq\; A(x_{wl})(m-x_{wl})^2 - B(x_{wl}) \;\leq\; 0$, so (28) can be replaced by (27) which is now the only condition in both cases (a) and (b). Hence, x_{wl} is valid, whenever

$$x_{wl} \geq \frac{m}{2} \quad \text{ and } \quad (s-6)x_{wl} - ms + 3 \leq 0\,. \tag{29}$$

According to (14), the first condition in (29) is equivalent to $a^2+(b-1)a-4b \geq 0$ which is fulfilled, whenever

$$a \geq a_1 = \frac{1-b+\sqrt{b^2+14b+1}}{2}, \qquad b > 0\,. \tag{30}$$

The second condition in (29) is equivalent to $(a+b-6)\sqrt{ab} + 3\,(a-b)\sqrt{a+b-1} \geq 0$ which is satisfied, whenever

$$a \geq a_2 = \frac{-(b+3)^2+(b-3)\sqrt{b^2+54b+9}}{2\,(b-9)}, \qquad b > 0\,. \tag{31}$$

This is verified by an elementary derivation, considering the four cases $a \geq b,\ b > 3$; $a \geq b,\ b \leq 3$; $a < b,\ a+b \geq 6$ and $a < b,\ a+b < 6$.

Case (iii):

Let x_{wl} be the left point of inflection (14), z_2 the point at which the tangent of $f(x)$ to x_{wl} crosses the x–axis (15), and ${x_2}^{(hlf)} = (z_2+m)/2$ our suggested bound on ${x_2}^{(min)}$.

It is easy to verify that $a_1 \geq a_2$ for $0 \leq b \leq 1$. Therefore, if $a \leq 2$, $a \geq a_1$, and $0 \leq b \leq 1$, we have $a \geq a_1 \geq a_2$ and this is covered by case (ii). $a \leq 2$, $a \geq a_1$, and $b > 1$ is not possible, since $a_1 > 2$ for $b > 1$. Finally, $a \leq 2$ and $a \leq a_1$ is case (i), and we have eliminated all cases $a \leq 2$. The remaining combinations of parameters a, b are bounded by $2 < a < \max(a_1,\ a_2)$, $b > 0$. Furthermore, if $b \leq 1$ then $a_1 \leq 2$ and $a_2 \leq 2$, and if $b > 1$ then $a_1 < a_2 \leq 9$. Therefore the remaining region is described more tightly by

$$2 \,<\, a \,<\, a_2 \,=\, \frac{-(b+3)^2+(b-3)\sqrt{b^2+54b+9}}{2\,(b-9)} \leq 9\,, \qquad b > 1\,. \tag{32}$$

In order to prove the feasibility of our proposal $x_2 = {x_2}^{(hlf)}$, we compare it with ${x_2}^{(min)}$ which is obtained from (10) as follows.

Using the polynomials in (11), (12), and (13), we define

$$P(z) = A(x_2)z^4 - B(x_2)z^2 + C(x_2) \quad \text{and} \quad D_P(x_2) = B^2(x_2) - 4A(x_2)C(x_2)$$

and assume that $A(x_2) \leq 0$ (a feasible x_2 with this property exists because of $A(x_{wl}) < 0$ in the region (32)). Furthermore we have $C(x_2) \leq 0$ from (23). Consider three cases, dependent on the signs of $B(x_2)$ and $D_P(x_2)$.

(a) Let $B(x_2) \geq 0$. Then $P(z) \leq 0$ for all z, and x_2 is valid.

(b) Let $B(x_2) < 0$ and $D_P(x_2) > 0$. Then $P(z) = 0$ has exactly two positive roots. Hence, $P(z) \leq 0$ for $0 \leq z \leq m - x_2$, whenever $P(m-x_2) \leq 0$ and $P'(m-x_2) > 0$. Further analysis shows that the minimum x_2 obtained from these two conditions is either $\geq m$ or it leads to $D_P(x_2) < 0$ which contradicts the assumption $D_P(x_2) > 0$.

(c) Let $B(x_2) < 0$ and $D_P(x_2) \le 0$. Then $P(z)$ does not change its sign. Since $C(x_2) < 0$, we have $P(z) < 0$ for all z, i. e. x_2 is valid.

From the above we conclude that ${x_2}^{(min)}$ fulfils at least one of the two conditions

$$B(x_2) \ge 0 \qquad \text{or} \quad D_P(x_2) \le 0\,, \qquad x_L < x_2 < m\,, \tag{33}$$

where $x_L = \max\left(\frac{m}{2},\, x_{wl}\right)$. $B(x_2) = 0$ implies $D_P(x_2) \le 0$, due to $D_P(x_2) = B^2(x_2) - 4\,A(x_2)C(x_2) \le B^2(x_2)$. Observing that $B(x_L) < 0$ (otherwise (a) applies leading to case (i) or case (ii)), we end up with the following simple rule for the computation of ${x_2}^{(min)}$:

Solve $B^2(x_2) - 4\,A(x_2)C(x_2) = 0$ *and return the smallest root* $x_2 > \max\left(\frac{m}{2},\, x_{wl}\right)$.

Table 5: Values c_{min} (34), corresponding to ${x_2}^{(min)}$, and values c_{hlf} (34), corresponding to our suggested bound ${x_2}^{(hlf)} = (z_2 + m)/2$. (Value 0 indicates case (ii)).

$b \setminus a$	2.0	2.1	2.2	2.5	3.0	4.0	5.0	7.0	9.0
1.0	0 0.167								
1.1	0.023 0.181	0 0.160							
1.2	0.043 0.193	0.005 0.173	0 0.154			not in the region (32)			
1.5	0.087 0.221	0.049 0.202	0.018 0.185	0 0.138		case (ii)			
2.0	0.134 0.250	0.099 0.233	0.068 0.218	0.004 0.176	0 0.118				
3.0	0.184 0.282	0.154 0.267	0.126 0.254	0.057 0.217	0.000 0.167	0 0.092			
5.0	0.225 0.309	0.200 0.297	0.177 0.285	0.115 0.254	0.040 0.210	0 0.146			
10	0.258 0.331	0.237 0.320	0.217 0.310	0.164 0.283	0.095 0.247	0.015 0.194	0 0.155		
20	0.275 0.342	0.256 0.332	0.238 0.323	0.191 0.299	0.128 0.267	0.047 0.220	0.008 0.187	0 0.141	
50	0.286 0.349	0.268 0.340	0.251 0.332	0.207 0.309	0.149 0.280	0.073 0.238	0.029 0.208	0.000 0.168	0 0.142
∞	0.293 0.354	0.276 0.345	0.260 0.337	0.218 0.316	0.164 0.289	0.092 0.250	0.049 0.224	0.009 0.189	0 0.167

For the display of the numerical results we introduce the quantities

$$c_{min} = \frac{{x_2}^{(min)} - x_{wl}}{m - x_{wl}} \qquad \text{and} \qquad c_{hlf} = \frac{{x_2}^{(hlf)} - x_{wl}}{m - x_{wl}}\,. \tag{34}$$

It is clear that $0 \le c_{min},\ c_{hlf} < 1$, and $c_{min} = 0$ signifies ${x_2}^{(min)} = x_{wl}$. With the results in Table 5 the case (iii) is concluded, but the figures also show that ${x_2}^{(hlf)}$ is a relatively crude (but simple) choice for the abscissas of valid reflection points P_l.

Acknowledgement

The authors would like to thank the referee for careful reading and helpful suggestions that significantly improved the presentation of the paper.

References

[1] Ahrens, J. H., Dieter, U.: Computer methods for sampling from gamma, beta, Poisson and binomial distributions. Computing *12*, 223 – 246 (1974).

[2] Atkinson, A. C.: A family of switching algorithms for the computer generation of beta random variables. Biometrika *66*, 141 – 145 (1979).

[3] Atkinson, A. C., Whittaker, J.: A switching algorithm for the generation of beta random variables with at least one parameter less than 1. J. Roy. Statist. Soc. A *139*, 462 – 467 (1976).

[4] Atkinson, A. C., Whittaker, J.: Algorithm AS 134: The generation of beta random variables with one parameter greater than one and one parameter less than 1. Applied Statistics *28*, 90 – 93 (1979).

[5] Cheng, R. C. H.: Generating beta variates with nonintegral shape parameters. Comm. ACM *21*, 317 – 322 (1978).

[6] Devroye, L.: Non–Uniform Random Variate Generation. Springer, New York, 1986.

[7] Dieter, U., Ahrens, J. H.: Acceptance–rejection techniques for sampling from the gamma and beta distributions. Techn. Report *83*, Dep. Statist., Stanford Univ., Stanford, California, 1974.

[8] Jöhnk, M. D.: Erzeugung von betaverteilten und gammaverteilten Zufallszahlen. Metrika *8*, 5 – 15 (1964).

[9] Minh, Do Le: Generating gamma variates. ACM Trans. on Math. Software *14*, 261 – 266 (1988).

[10] Sakasegawa, H.: Stratified rejection and squeeze method for generating beta random numbers. Ann. Inst. Statist. Math. *35*, 291 – 302 (1983).

[11] Schmeiser, B. W., Babu, A. J. G.: Beta variate generation via exponential majorizing functions. Oper. Research *28*, 917 – 926 (1980). (Errata: Oper. Research *31*, 802, 1983).

[12] Stadlober, E., Kremer, R.: Sampling from discrete and continuous distributions with C–rand. In: Simulation and Optimization (Pflug, G., Dieter, U. eds.), Springer Lecture Notes Econom. Math. Systems *371*, 154 – 162 (1992).

H. Zechner and E. Stadlober
Institut für Statistik
Techn. Universität Graz
Lessingstraße 27
A–8010 Graz, Austria
e–mail: fstat@ftug.dnet.tu–graz.ac.at