Mathematics of C functions for beta inverse etc

Stig Rosenlund
2017-01-07

These functions compute the beta-, binomial, F- and t-distribution functions (cumulative distribution functions, CDFs) and their inverses. I rely heavily on AoS (Abramowitz & Stegun) and NR (Numerical Recipes in C). The mathematics I describe is for the inverses. For the CDF:s, read the references! The reference list does not include all relevant original work. I do not intend to publish this as a peer-reviewed scientific article.

Complete listings are given of the functions in companion files, as Pdf for reading and printing and as text for direct use, ready for compile.

1. Beta distribution inverse

The beta distribution function is

$$I_x(a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1}dt, \quad 0 \le x \le 1;\ a > 0,\ b > 0 \tag{1.1}$$

Some important statistical functions can be expressed in it, see AoS p. 945 and NR, §6.4, p. 226 ff.

I call the inverse function $I^{-1}(\cdot,a,b)$, ie if $I_x(a,b) = p$, then $x = I^{-1}(p,a,b)$.

It holds $I^{-1}(0,a,b) = 0$, $I^{-1}(1,a,b) = 1$ and $I^{-1}(p,a,b) = 1 - I^{-1}(1-p,b,a)$.

I will describe how to try inverse quadratic search to find the solution $I^{-1}(p,a,b)$. For this to succeed a good initial approximation is needed.

1.1. Beta inverse approximations: betadistinv_ap()

1.1.1. *AoS (26.7.5) p. 949 with my modification – the t-distribution*

This can be used as an approximation if one of a and b is 0.5 and the other one a multiple of 0.5. The inverses of the beta- and t-distributions are related this way. Let the t-distribution function with ν degrees of freedom be $F_t(x\,|\,\nu)$ with inverse $F_t^{-1}(p\,|\,\nu)$.

For r a multiple of $\frac{1}{2}$, set

$$q_1(p,r) = F_t^{-1}(1 - \frac{p}{2} \mid 2r) \tag{1.2}$$

$$q_2(p,r) = I^{-1}(2-2p, r, \frac{1}{2}) \tag{1.3}$$

Then

$$I^{-1}(p,r,\frac{1}{2}) = \frac{2r}{q_1(p,r)^2 + 2r} \tag{1.4}$$

$$I^{-1}(p,\frac{1}{2},r) = \frac{q_1(1-p,r)^2}{q_1(1-p,r)^2 + 2r} \tag{1.5}$$

and

$$F_t^{-1}(p \mid \nu) = \sqrt{\frac{\nu(1-q_2(p,\nu/2))}{q_2(p,\nu/2)}},\ p \ge \frac{1}{2} \tag{1.6}$$

E-mail: stig.ingvar.rosenlund@gmail.com. Web: Google Stig Rosenlund and bypass the Finnish artist.

$$F_t^{-1}(p \mid \nu) = -F_t^{-1}(1-p \mid \nu),\ p < \frac{1}{2} \tag{1.7}$$

Replace $q_1(p,r)$ by my improvement of the AoS (26.7.5) approximation in Section 4.

1.1.2. *AoS (26.5.22) p. 945*

This is rather unneccessary to include here, but since I have taken the time to write it down I might as well render it. The formula says, for a and b not too small

$$y = \Phi^{-1}(1–p) \text{ (normal inverse)} \qquad h = 2\left(\frac{1}{2a-1} + \frac{1}{2b-1}\right)^{-1} \qquad \lambda = \frac{y^2-3}{6} \tag{1.8}$$

$$w = \frac{y(h+\lambda)^{0.5}}{h} - \left(\frac{1}{2b-1} - \frac{1}{2a-1}\right)\left(\lambda + \frac{5}{6} - \frac{2}{3h}\right) \tag{1.9}$$

$$I^{-1}(p,a,b) \approx \frac{a}{a+be^{2w}} \tag{1.10}$$

I use the approximation of Φ^{-1} by Moro.

1.1.3. *AoS (26.6.14) p. 947 – the F-distribution*

AoS p. 947 refer to the general formula (26.5.22) for approximation of the inverse of the F distribution. However, (26.6.14) can give a better approximation for some ν_1, ν_2 and p. Let $F(x \mid \nu_1, \nu_2)$ be the F-distribution CDF with ν_1 and ν_2 degrees of freedom. The formula is expressed in 1 - CDF, which translated to CDF reads

$$F(x\,|\,\nu_1,\nu_2) \approx \Phi(y), \qquad y = \frac{\sqrt{(2\nu_2-1)\frac{\nu_1}{\nu_2}x} - \sqrt{2\nu_1-1}}{\sqrt{1+\frac{\nu_1}{\nu_2}x}} \tag{1.11}$$

$F^{-1}(p\,|\,\nu_1,\nu_2)$ is the solution x to the equation $F(x\,|\,\nu_1,\nu_2) = p$.

Set

$$w = \Phi^{-1}(p) \tag{1.12}$$

$$t = w^2 \tag{1.13}$$

$$u = \sqrt{\frac{\nu_1}{\nu_2}x} \tag{1.14}$$

$$c_1 = 2\nu_1 - 1 \tag{1.15}$$

$$c_2 = 2\nu_2 - 1 \tag{1.16}$$

An approximation of $F^{-1}(p\,|\,\nu_1,\nu_2)$ is then the solution in x of $y = \Phi^{-1}(p)$. Thus

$$\frac{\sqrt{c_2}u - \sqrt{c_1}}{\sqrt{1+u^2}} = w \tag{1.17}$$

$$\sqrt{c_2}u - \sqrt{c_1} = w\sqrt{1+u^2} \tag{1.18}$$

$$(\sqrt{c_2}u - \sqrt{c_1})^2 = t(1+u^2) \tag{1.19}$$

$$c_2u^2 - 2\sqrt{c_1c_2}u + c_1 = t + tu^2 \tag{1.20}$$

$$c_2u^2 - tu^2 - 2\sqrt{c_1c_2}u + c_1 - t = 0 \tag{1.21}$$

$$(c_2 - t)u^2 - 2\sqrt{c_1c_2}u = t - c_1 \tag{1.22}$$

$$u^2 - \frac{2\sqrt{c_1c_2}}{c_2 - t}u + \frac{c_1c_2}{(c_2 - t)^2} = \frac{t - c_1}{c_2 - t} + \frac{c_1c_2}{(c_2 - t)^2} \tag{1.23}$$

$$\left(u - \frac{\sqrt{c_1c_2}}{c_2 - t}\right)^2 = \frac{t - c_1}{c_2 - t} + \frac{c_1c_2}{(c_2 - t)^2} \tag{1.24}$$

$$u = \frac{\sqrt{c_1c_2}}{c_2 - t} \pm \sqrt{\frac{t - c_1}{c_2 - t} + \frac{c_1c_2}{(c_2 - t)^2}} \tag{1.25}$$

$$u = \frac{1}{c_2 - t}\left[\sqrt{c_1c_2} \pm \sqrt{c_2t - t^2 - c_1c_2 + c_1t + c_1c_2}\right] \tag{1.26}$$

$$u = \frac{1}{c_2 - t}\left[\sqrt{c_1c_2} \pm \sqrt{c_2t - t^2 + c_1t}\right] \tag{1.27}$$

$$u = \frac{1}{c_2 - t}\left[\sqrt{c_1c_2} \pm \sqrt{t}\sqrt{c_1 + c_2 - t}\right] \tag{1.28}$$

$$u = \sqrt{\frac{\nu_1}{\nu_2}x} = \frac{1}{c_2 - w^2}\left[\sqrt{c_1c_2} \pm w\sqrt{c_1 + c_2 - w^2}\right] \tag{1.29}$$

$$x = \frac{\nu_2}{\nu_1}\left\{\frac{1}{c_2 - w^2}\left[\sqrt{c_1c_2} \pm w\sqrt{c_1 + c_2 - w^2}\right]\right\}^2 \tag{1.30}$$

Hence

$$F^{-1}(p\,|\,\nu_1, \nu_2) \approx \frac{\nu_2}{\nu_1}\left\{\frac{1}{c_2 - w^2}\left[\sqrt{c_1c_2} \pm w\sqrt{c_1 + c_2 - w^2}\right]\right\}^2 \tag{1.31}$$

where $w = \Phi^{-1}(p),\ c_1 = 2\nu_1 - 1,\ c_2 = 2\nu_2 - 1.$

Here $\pm$ shall be $+$, but the formula should only be used for $p \le 0.5$. For $p > 0.5$ we use the reflection formula AoS (26.6.9) p. 946-947. Thus, if $2\nu_1 + 2\nu_2 - 2 - w^2 \ge 0$, let

$$G(p\,|\,\nu_1, \nu_2) = \frac{\nu_2}{\nu_1}\left\{\frac{1}{2\nu_2 - 1 - w^2}\left[\sqrt{(2\nu_1 - 1)(2\nu_2 - 1)} + w\sqrt{2\nu_1 + 2\nu_2 - 2 - w^2}\right]\right\}^2 \tag{1.32}$$

If $2\nu_1 + 2\nu_2 - 2 - w^2 < 0$ let $G(p\,|\,\nu_1, \nu_2) = 0$.

$$F^{-1}(p\,|\,\nu_1, \nu_2) \approx G(p\,|\,\nu_1, \nu_2), \quad p \le 0.5 \tag{1.33}$$

$$F^{-1}(p\,|\,\nu_1, \nu_2) \approx \frac{1}{G(1 - p\,|\,\nu_2, \nu_1)}, \quad p > 0.5 \tag{1.34}$$

$I^{-1}(p, a, b)$ can be deduced from $F^{-1}()$ for a and b multiples of 0.5. It holds

$$F(x\,|\,\nu_1, \nu_2) = 1 - I_{\frac{\nu_2}{\nu_2 + \nu_1 x}}\left(\frac{\nu_2}{2}, \frac{\nu_1}{2}\right) \tag{1.35}$$

from which follows, for a and b multiples of 0.5,

$$I^{-1}(p, a, b) = \frac{a}{a + bF^{-1}(1 - p\,|\,2b, 2a)} \tag{1.36}$$

The following are the criteria in simple C-code for deciding whether to use (1.33) or not, for $p \le 0.5$. They have been found by crude experimentation. At this point in the function the t-distribution has already been used and the function returned, if possible.

```
// method = 1 is default to not use F-distribution, method = 2 to use it.
method = 1;
if ((r1=2*a) == floor(r1) && (r2=2*b) == floor(r2)) {
  if (a == 1) {
    if (b == 1) { if (p >= 0.26) method = 2; }
    else        { if (p >= 0.17) method = 2; }
  }
  else if (a == 1.5) {
    if      (b == 1.0) { if (p >= 0.10) method = 2; }
    else if (b == 1.5) { if (p >= 0.35) method = 2; else method = 1; }
    else               { if (p >= 0.22) method = 2; }
  }
  else if (a == 2.0) {
    if (b == 1) { if (p >= 0.05) method = 2; }
    else        { if (p >= 0.23) method = 2; }
  }
  else if (a == 2.5) { if (b == 1) { if (p >=   0.03) method = 2; } }
  else if (a == 3.0) { if (b == 1) { if (p >=   0.02) method = 2; } }
  else if (a <= 5.5) { if (b == 1) { if (p >=  0.001) method = 2; } }
  else if (a <= 6.5) { if (b == 1) { if (p >= 0.0001) method = 2; } }
  else { if (b == 1) method = 2; }
}
```

1.2. Inverse quadratic search for beta inverse: betadistinv()

Now skip a and b in the notation and set

$$C = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \tag{1.37}$$

$$I(x) = I_x(a,b) \tag{1.38}$$

$$H(p) = I^{-1}(p) \tag{1.39}$$

Then (try to deduce this)

$$H^{(1)}(p) = [I'(H(p))]^{-1} \tag{1.40}$$

$$H^{(2)}(p) = -[I'(H(p))]^{-3} I''(H(p)) \tag{1.41}$$

where

$$I'(x) = Cx^{a-1}(1-x)^{b-1} \tag{1.42}$$

$$I''(x) = \frac{I'(x)(2x - ax - bx + a - 1)}{x(1-x)} \tag{1.43}$$

Given p, determine an approximation $y_0 \approx H(p)$.

Put $p_0 = I(y_0)$ so that $y_0 = I^{-1}(p_0) = H(p_0)$.

Then p_0 is close to p. Put

$$h_j = H^{(j)}(p_0), \text{ where } h_0 = H(p_0) = y_0$$

We can now try inverse quadratic search.

With the three first terms in a Taylor expansion around p_0, put

$$K_0(p) = y_0 + h_1(p - p_0) + h_2\frac{(p-p_0)^2}{2}$$

Then $H(p) \approx K_0(p)$. Put

$$y_1 = K_0(p) \tag{1.44}$$

which can be a better approximation of $H(p)$.

This gives a recursive procedure.

$$\begin{aligned} y_0 &= \text{approximation of } I^{-1}(p) \\ p_k &= I(y_k) \iff y_k = I^{-1}(p_k) = H(p_k), \quad k = 0, 1, 2, \dots \\ y_{k+1} &= K_k(p) = y_k + H^{(1)}(p_k)(p - p_k) + H^{(2)}(p_k)\frac{(p - p_k)^2}{2} \end{aligned}$$

that might converge so that $\lim_{k\to\infty} y_k = I^{-1}(p)$. In the C function I break when I find that $|\, y_{k+1} - y_k \,| > 0.5\,|\, y_k - y_{k-1}|$ before convergence, ie before $|y_{k+1} - y_k\,| < \epsilon$. This is interpreted as a sign that the procedure is at that point worse than bisection or even beginning to diverge. It is not certain that $|\, y_k - H(p)\,| < \epsilon$ at convergence, either.

After this the solution $H(p)$ is brought to an end by stepping and bisection, if needed.

Remark 1.1. *Shaw, p. 45, second paragraph from the bottom, asserts that Newton-Raphson methods are highly unstable when applied to the beta distribution and not to recommend over stepping or bisection. However, my tests show the solution given above to be highly efficient. See Section 5.*

2. Binomial inverse: bindistinv()

Here the formulation of the binomial distribution in terms of the beta distribution (see the references) is used together with the CLT (Central Limit Theorem). I leave to the reader to figure out the details.

3. F-distribution inverse: fdistinv()

We use here (1.36). It gives

$$F^{-1}(p\,|\,\nu_1, \nu_2) = \frac{\nu_2}{\nu_1 I^{-1}(1 - p, \frac{\nu_2}{2}, \frac{\nu_1}{2})} - \frac{\nu_2}{\nu_1} = \frac{\nu_2}{\nu_1\left[1 - I^{-1}(p, \frac{\nu_1}{2}, \frac{\nu_2}{2})\right]} - \frac{\nu_2}{\nu_1} \tag{3.1}$$

4. t-distribution inverse

The t-distribution function with ν degrees of freedom is

$$F_t(x\,|\,\nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi}\nu^{0.5}\Gamma(\frac{\nu}{2})}\int_{-\infty}^{x}\left(1 + \frac{\tau^2}{\nu}\right)^{-\frac{\nu+1}{2}} d\tau, \quad -\infty < x < -\infty \tag{4.1}$$

I call the inverse function $F_t^{-1}(p\,|\,\nu)$, ie if $F_t(x\,|\,\nu) = p$, then $x = F_t^{-1}(p\,|\,\nu)$.

I treat $p < 0.5$. We have $F_t^{-1}(0.5) = 0$. For $p > 0.5$ we use $F_t^{-1}(p\,|\,\nu) = -F_t^{-1}(1-p\,|\,\nu)$.

4.1. t inverse approximation: tdistinv_ap()

A good and quickly computed approximation to $F_t^{-1}(p\,|\,\nu)$ is useful. I have improved on AoS (26.7.5) p. 949. In this formula we have

$$t_p = F_t^{-1}(1-p\,|\,\nu) \qquad x_p = \Phi^{-1}(1-p) \tag{4.2}$$

The approximation is stated as

$$t_p \sim x_p + \frac{g_1(x_p)}{\nu} + \frac{g_2(x_p)}{\nu^2} + \frac{g_3(x_p)}{\nu^3} + \dots \tag{4.3}$$

The $g_j(x)$ are given for $j = 1, 2, 3, 4$. They are polynomials in x. One can also define

$$t_p = F_t^{-1}(p\,|\,\nu) \qquad x_p = \Phi^{-1}(p) \tag{4.4}$$

Formula (4.3) holds also with this definition, which I will adhere to below. Here $p < 0.5$.

Let

$$G(p,\nu) = x_p + \frac{g_1(x_p)}{\nu} + \frac{g_2(x_p)}{\nu^2} + \frac{g_3(x_p)}{\nu^3} + \frac{g_4(x_p)}{\nu^4} \quad (x_p \text{ approximation by Moro}) \tag{4.5}$$

I state my problem as finding a function $h(p,\nu)$ such that $G(p,\nu) + h(p,\nu)$ is closer to t_p than $G(p,\nu)$, and such that the extra runtime to compute it is much smaller than the decrease in runtime obtained in a subsequent root finding to determine t_p with relative error $< 5^{-16}$, for example. Other forms of improvement might be to find factors for the $g_j(x_p)$. I tried to find a polynomial $g_5(x_p)$ but was not successful.

I have found that functions $a(p,\nu)$ and $q(p,\nu)$ for

$$h(p,\nu) = \frac{-a(p,\nu)(-\log(p))^{q(p,\nu)}}{\nu^5} \quad (\nu = 3, 5, 6, 7, \dots) \tag{4.6}$$

will give a very good approximation. The functions $a(,)$ and $q(,)$ are ≥ 0 and constant over intervals of p and ν. They are given in tdistinv_ap() as arrays `aarr[][]` and `qarr[][]` in the function listing. For $\nu = 1, 2, 4$ exact expressions for t_p are available.

The following tables give $h(p,\nu)$ and what it aims to be close to, namely $t_p - G(p,\nu)$, for some values of ν and p. For each ν the functions $a(,)$ and $q(,)$ are constant. And they are the same for $\nu = 30$ and 39, since 30-39 is an interval of ν which I grouped together. Table 2 thus shows that the factor $1/\nu^5$ and the analytic form in (4.6) are right.

Table 1. Fit of $h(p,\nu)$ for $\nu = 5, 15$

ν	p	$h(p,\nu)$	$t_p - G(p,\nu)$	ν	p	$h(p,\nu)$	$t_p - G(p,\nu)$
5	0.0001	-1.245×10^{-1}	-1.215×10^{-1}	15	0.0001	-4.090×10^{-4}	-4.081×10^{-4}
5	0.0002	-7.294×10^{-2}	-7.300×10^{-2}	15	0.0002	-2.493×10^{-4}	-2.498×10^{-4}
5	0.0003	-5.309×10^{-2}	-5.310×10^{-2}	15	0.0003	-1.833×10^{-4}	-1.837×10^{-4}
5	0.0004	-4.197×10^{-2}	-4.193×10^{-2}	15	0.0004	-1.459×10^{-4}	-1.461×10^{-4}
5	0.0005	-3.476×10^{-2}	-3.470×10^{-2}	15	0.0005	-1.216×10^{-4}	-1.216×10^{-4}
5	0.0006	-2.968×10^{-2}	-2.959×10^{-2}	15	0.0006	-1.043×10^{-4}	-1.042×10^{-4}
5	0.0007	-2.589×10^{-2}	-2.579×10^{-2}	15	0.0007	-9.136×10^{-5}	-9.114×10^{-5}
5	0.0008	-2.294×10^{-2}	-2.283×10^{-2}	15	0.0008	-8.126×10^{-5}	-8.095×10^{-5}
5	0.0009	-2.058×10^{-2}	-2.046×10^{-2}	15	0.0009	-7.315×10^{-5}	-7.276×10^{-5}

Table 2. Fit of $h(p,\nu)$ for $\nu = 30, 39$

ν	p	$h(p,\nu)$	$t_p - G(p,\nu)$	ν	p	$h(p,\nu)$	$t_p - G(p,\nu)$
30	0.0001	-1.234×10^{-5}	-1.221×10^{-5}	39	0.0001	-3.323×10^{-6}	-3.259×10^{-6}
30	0.0002	-7.481×10^{-6}	-7.503×10^{-6}	39	0.0002	-2.015×10^{-6}	-2.006×10^{-6}
30	0.0003	-5.519×10^{-6}	-5.530×10^{-6}	39	0.0003	-1.486×10^{-6}	-1.480×10^{-6}
30	0.0004	-4.406×10^{-6}	-4.408×10^{-6}	39	0.0004	-1.187×10^{-6}	-1.181×10^{-6}
30	0.0005	-3.678×10^{-6}	-3.673×10^{-6}	39	0.0005	-9.906×10^{-7}	-9.842×10^{-7}
30	0.0006	-3.161×10^{-6}	-3.150×10^{-6}	39	0.0006	-8.514×10^{-7}	-8.444×10^{-7}
30	0.0007	-2.773×10^{-6}	-2.757×10^{-6}	39	0.0007	-7.469×10^{-7}	-7.392×10^{-7}
30	0.0008	-2.470×10^{-6}	-2.450×10^{-6}	39	0.0008	-6.652×10^{-7}	-6.570×10^{-7}
30	0.0009	-2.226×10^{-6}	-2.204×10^{-6}	39	0.0009	-5.996×10^{-7}	-5.908×10^{-7}

The arrays have been found by tedious search through grids of (a, q), minimizing the maximum absolute deviation. The improvement from the correction is mostly better than a factor 0.01, ie if the error of the AoS approximation is ϵ, then the error with the correction is mostly $< 0.01\epsilon$. Often it is better than a factor 0.001. No assertion can however be given for a uniform bound on the maximal absolute deviation $\mid G(p,\nu) + h(p,\nu) - t_p \mid$, since for p small enough this deviation can be made arbitrarily large.

If someone finds functions $a(p,\nu)$ and $q(p,\nu)$ with a much smaller number of parameters, such as combinations of trigonometric functions or polynomials, that do not sacrifice too much accuracy, I would be happy to put them in the function.

4.2. Inverse quadratic search for t inverse: tdistinv()

I now describe inverse quadratic search to find $F_t^{-1}(p\,|\,\nu)$. Simplify the notation and set

$$C_1 = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi}\nu^{0.5}\Gamma(\frac{\nu}{2})} \tag{4.7}$$

$$C_2 = \frac{(\nu+1)\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi}\nu^{1.5}\Gamma(\frac{\nu}{2})} = \frac{\nu+1}{\nu}C_1 \tag{4.8}$$

$$F(x) = F_t(x\,|\,\nu) \tag{4.9}$$

$$H(p) = F_t^{-1}(p\,|\,\nu) \tag{4.10}$$

(The notation is partly the same as in Section 1.2.) Then

$$H^{(1)}(p) = [F'(H(p))]^{-1} \tag{4.11}$$

$$H^{(2)}(p) = -[F'(H(p))]^{-3}F''(H(p)) \tag{4.12}$$

where

$$F'(x) = C_1\left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \tag{4.13}$$

$$F''(x) = -C_2\,x\left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+3}{2}} \tag{4.14}$$

Compute an approximation $y_0 \approx H(p)$ by my formula (4.6).

Put $p_0 = F(y_0)$ so that $y_0 = F^{-1}(p_0) = H(p_0)$. Then p_0 is close to p. Put

$$h_j = H^{(j)}(p_0) \tag{4.15}$$

Then

$$h_0 = H(p_0) = y_0 \tag{4.16}$$

$$h_1 = H^{(1)}(p_0) = [F'(y_0)]^{-1} = \left[C_1\left(1 + \frac{y_0^2}{\nu}\right)^{-\frac{\nu+1}{2}}\right]^{-1} \tag{4.17}$$

$$h_2 = H^{(2)}(p_0) = -[F'(y_0)]^{-3}F''(y_0) = \frac{C_2\, y_0\Big(1+\frac{y_0^2}{\nu}\Big)^{-\frac{\nu+3}{2}}}{\Big[C_1\Big(1+\frac{y_0^2}{\nu}\Big)^{-\frac{\nu+1}{2}}\Big]^3} \tag{4.18}$$

which is reduced to

$$h_0 = y_0 \tag{4.19}$$

$$h_1 = \frac{1}{C_1}\left(1+\frac{y_0^2}{\nu}\right)^{\frac{\nu+1}{2}} \tag{4.20}$$

$$h_2 = \frac{\nu+1}{\nu C_1^2}\, y_0\left(1+\frac{y_0^2}{\nu}\right)^{\nu} \tag{4.21}$$

We can now try inverse quadratic search. With the three first terms in a Taylor expansion around p_0, put

$$K_0(p) = y_0 + h_1(p-p_0) + h_2\frac{(p-p_0)^2}{2} \tag{4.22}$$

Then $H(p) \approx K_0(p)$. Put

$$y_1 = K_0(p) \tag{4.23}$$

which can be a better approximation of $H(p)$.

This gives a recursive procedure.

$$\begin{aligned} y_0 &= \text{ approximation of } F^{-1}(p) \\ p_k &= F(y_k) \iff y_k = F^{-1}(p_k) = H(p_k), \quad k = 0,1,2,\dots \\ y_{k+1} &= K_k(p) = y_k + H^{(1)}(p_k)(p-p_k) + H^{(2)}(p_k)\frac{(p-p_k)^2}{2} \end{aligned}$$

The principle of the algorithm is the same as in Section 1.2 and the continuation is the same. I have found that it has some advantages over using (1.6) and (1.7).

5. Numerical comparison of methods for beta inverse

Here are some results for the efficiency of my algorithm for the beta inverse in Section 1.2 compared to bisection. I give the number of computations of betadist() needed for my algorithm on one hand and for bisection on the other hand. For the latter I use the approximation betadist_ap() and stepping appropriately from that to obtain left and right bounds for the solution. Taking 0 as left bound and 1 as right bound necessitates mostly more computations of betadist(), but can sometimes need a few computations less. Both variants of bisection need many more computations than my algorithm.

I computed $I^{-1}(2p, \frac{\nu}{2}, \frac{1}{2})$. These are the parameters to obtain the t inverse from the beta inverse. See (1.3), (1.6) and (1.7). N_1 is the number of computations of betadist() with my algorithm and N_2 is the number with bisection.

Table 3. Efficiency of inverse quadratic search with correction $h(p,\nu)$

ν	p	N_1	N_2	ν	p	N_1	N_2	ν	p	N_1	N_2
3	0.00000001	4	55	5	0.00000001	6	54	6	0.00000001	5	54
3	0.00000010	4	55	5	0.00000010	4	55	6	0.00000010	5	54
3	0.00000100	4	54	5	0.00000100	4	54	6	0.00000100	5	60
3	0.00001000	4	64	5	0.00001000	4	60	6	0.00001000	4	55
3	0.00010000	4	62	5	0.00010000	4	58	6	0.00010000	3	55
3	0.00100000	4	60	5	0.00100000	2	56	6	0.00100000	2	56
3	0.01000000	4	54	5	0.01000000	3	52	6	0.01000000	3	55
3	0.10000000	3	54	5	0.10000000	3	55	6	0.10000000	3	55

Tabulating $p = 0.99999999, 0.9999999, \ldots, 0.99, 0.9$ gives the same numbers. The table continues this way for higher values of ν, with $N_1 \leq 5$ for $\nu \geq 6$.

The improvement from my correction $h(p, \nu)$ in (4.6) is shown in the following table, where the results were obtained without it. (Remember that tdistinv_ap() is used by betadistinv_ap(), see Section 1.1.1.)

Table 4. Efficiency of inverse quadratic search without correction $h(p, \nu)$

ν	p	N_1	N_2	ν	p	N_1	N_2	ν	p	N_1	N_2
3	0.00000001	52	58	5	0.00000001	8	55	6	0.00000001	6	54
3	0.00000010	52	57	5	0.00000010	5	55	6	0.00000010	6	54
3	0.00000100	52	55	5	0.00000100	5	54	6	0.00000100	6	54
3	0.00001000	7	55	5	0.00001000	5	55	6	0.00001000	6	54
3	0.00010000	6	55	5	0.00010000	5	54	6	0.00010000	4	49
3	0.00100000	5	55	5	0.00100000	4	53	6	0.00100000	3	52
3	0.01000000	6	54	5	0.01000000	4	53	6	0.01000000	4	55
3	0.10000000	5	54	5	0.10000000	4	54	6	0.10000000	3	55

Of course, if you are content with somewhat less than lightning speed, you can dispense with my correction for $\nu > 3$ or $0.00001 \leq p \leq 0.99999$, ie use tdistinv_ap0().

More functions

The functions described, which I have decided to give away for free, are part of the source code for my language Rapp for actuarial mathematics, mathematical statistics and mathematics in general, in that order. All functions are designed for maximal execution speed, subject to giving double accuracy to double returns. Also there are functions for "bignum" multiprecision computing.

I have developed Rapp for 33 years, the first eight in other languages than C. You access Rapp by googling free actuarial language. At least you get a hit in November 2016. If Google changes its search criteria, then google Stig Rosenlund and pass by the artist with the same name. The Rapp manual tells what kind of functions it is. Look in particular under Proc Mbasic.

If you want access to all source code for Rapp, then please help me make Rapp open source. It is now stuck in a review process for open source, and I find all the legal details in it confusing. I am a mathematician and programmer, not a lawyer. My aim with having the Rapp code as open source is to have a maximal permissive license, so that everyone can freely use it, while at the same time preventing commercial software companies from appropriating it. My only requirement would be that my name is mentioned when the code is used.

REFERENCES

ABRAMOWITZ, M. & STEGUN, I. (1964, 9:th printing 1970). *Handbook of Mathematical functions.* Dover Publications, New York.

MORO, B. (1995). The Full Monte. *Risk* **8**(2), 57-58.

PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W.T. & FLANNERY, B. P. (1992). *Numerical Recipes in C.* Cambridge University NR, Cambridge, MA.

SHAW, W. T. (2006). Sampling Student's T distribution - use of the inverse cumulative distribution function. *Journal of Computational Finance* **9**(4), 37-73.

STIG ROSENLUND
E-mail: stig.ingvar.rosenlund@gmail.com