# Derivatives of the Incomplete Beta Function

Robert J. Boik[1]
Department of Mathematical Sciences
Montana State University — Bozeman

James F. Robison-Cox
Department of Mathematical Sciences
Montana State University — Bozeman

March 15, 1998

[1] *Address for Correspondence:* Department of Mathematical Sciences, Montana State University — Bozeman, Bozeman, Montana 59717-2400

# Derivatives of the Incomplete Beta Function

*Keywords: Censored beta; Continued fractions; Truncated beta; Truncated beta-binomial*

## Languages

FORTRAN 77, MATLAB, and S–PLUS

## Description and Purpose

The incomplete beta function is defined as

$$I_{x,p,q} = \int_0^x \frac{u^{p-1}(1-u)^{q-1}}{\mathrm{Beta}(p,q)}\, du,$$

where $\mathrm{Beta}(p,q)$ is the beta function. Dutka (1981) gave a history of the development and numerical evaluation of this function. In this article, an algorithm for computing first and second derivatives of $I_{x,p,q}$ with respect to $p$ and $q$ is described. The algorithm is useful, for example, when fitting parameters to a censored beta, truncated beta, or a truncated beta-binomial model.

## Numerical Method

The incomplete beta function can be written as a hypergeometric series in the following manner:

$$I_{x,p,q} = K_{x,p,q} \times {}_2F_1\left(1-q, 1; p+1; \frac{-x}{1-x}\right), \text{ where } K_{x,p,q} = \frac{x^p(1-x)^{q-1}}{p\,\mathrm{Beta}(p,q)} \tag{1}$$

and ${}_2F_1$ is the hypergeometric series. See Abramowitz and Stegun (1965, chapters 15 & 26) for details. Müller (1931) obtained a corresponding continued fraction representation of the above hypergeometric series. From Müller's corresponding continued fraction, Gautschi (1967) and Tretter and Walster (1976) derived the associated continued fraction representation. The $n^{\text{th}}$ approximant of the associated continued fraction representation is

$$I_{x,p,q} \approx K_{x,p,q} \left\{ 1 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \ddots + \cfrac{a_n}{b_n}}}} \right\}, \tag{2}$$

where $K_{x,p,q}$ is given in (1);

$$a_n = \begin{cases} \dfrac{pf\,(q-1)}{q\,(p+1)} & \text{if } n = 1, \\ \dfrac{p^2 f^2\,(n-1)\,(p+q+n-2)\,(p+n-1)\,(q-n)}{q^2\,(p+2n-3)\,(p+2n-2)^2\,(p+2n-1)} & \text{if } n \geq 2; \end{cases}$$

$$b_n = \frac{2(pf+2q)n^2 + 2(pf+2q)(p-1)n + pq(p-2-pf)}{q(p+2n-2)(p+2n)}; \text{ and}$$

$$f = \frac{qx}{p(1-x)}.$$

Tretter and Walster (1976, 1979, 1980) found that by monitoring subtractions and subtracting analytically when possible, the continuing fraction in (2) yields accurate results, even with extreme arguments. For example, the 21$^{\text{st}}$ approximant yielded 25 significant digits when $p = 0.5 \times 10^{20}$, $q = 5000$, and $x = 0.9999999999999999$. To obtain this accuracy, the approximants were computed in tail-to-head fashion.

The current algorithm is not intended to be used for such extreme arguments. Accordingly, tail-to-head computing is unnecessary and forward recurrence relations will be used to compute approximants in a head-to-tail fashion. Using forward recurrence relations, the $n^{\text{th}}$ approximant can be computed as

$$I_{x,p,q} \approx K_{x,p,q}\frac{A_n}{B_n}, \; n \geq 1, \tag{3}$$

where $K_{x,p,q}$ is given in (1) and the terms $A_n$ and $B_n$ are defined recursively. For $n \geq 1$, $A_n$ and $B_n$ are given by

$$A_n = a_n A_{n-2} + b_n A_{n-1} \text{ and } B_n = a_n B_{n-2} + b_n B_{n-1},$$

where $A_{-1} = 1$; $A_0 = 1$; $B_{-1} = 0$; $B_0 = 1$;

Denote the first and second derivatives of the incomplete beta function with respect to $p$ and/or $q$ by $I^p_{x,p,q}$, $I^{pp}_{x,p,q}$, $I^q_{x,p,q}$, $I^{qq}_{x,p,q}$, and $I^{pq}_{x,p,q}$. The proposed algorithm computes approximations to these derivatives by differentiating the approximants in (3). This technique was successfully used by Moore (1982) to obtain derivatives of the incomplete gamma function. For example, the $n^{\text{th}}$ approximant to $I^p_{x,p,q}$ is computed as follows:

$$I^p_{x,p,q} = \frac{\partial I_{x,p,q}}{\partial p} \approx \left(\frac{\partial K_{x,p,q}}{\partial p}\right)\frac{A_n}{B_n} + K_{x,p,q}\left\{\frac{1}{B_n}\left(\frac{\partial A_n}{\partial p}\right) - \frac{A_n}{B_n^2}\left(\frac{\partial B_n}{\partial p}\right)\right\}, \text{ where}$$

$$\frac{\partial A_n}{\partial p} = \left(\frac{\partial a_n}{\partial p}\right)A_{n-2} + a_n\left(\frac{\partial A_{n-2}}{\partial p}\right) + \left(\frac{\partial b_n}{\partial p}\right)A_{n-1} + b_n\left(\frac{\partial A_{n-1}}{\partial p}\right); \text{ and}$$

$$\frac{\partial B_n}{\partial p} = \left(\frac{\partial a_n}{\partial p}\right) B_{n-2} + a_n \left(\frac{\partial B_{n-2}}{\partial p}\right) + \left(\frac{\partial b_n}{\partial p}\right) B_{n-1} + b_n \left(\frac{\partial B_{n-1}}{\partial p}\right).$$

Expressions for the derivatives of $K_{x,p,q}$, $a_n$, and $b_n$ were obtained using MAPLE V (Char et al., 1991) and are given in the appendix. To avoid overflows due to large $p$ and/or $q$, the derivatives of $a_n$, and $b_n$ were converted to partial fractions prior to numerical evaluation. For example, $\partial b_n / \partial q$ can be evaluated as

$$\frac{\partial b_n}{\partial q} = -\frac{p^2 f}{q\,(p+2n-2)\,(p+2n)} = -\frac{f}{q} - \frac{2f(n-1)^2}{q(p+2n-2)} + \frac{2fn^2}{q(p+2n)}.$$

Also, when $x > p/(p+q)$, the incomplete beta function and its derivatives were computed using the relation

$$I_{x,p,q} = 1 - I_{1-x,q,p}.$$

# Structure

## Fortran 77

*SUBROUTINE INBEDER(X,P,Q,PSI,DER,NAPPX,ERRAPX,IFAULT)*

*Input Parameters*

| | | | |
|---|---|---|---|
| X | Real | input: | the value $x$ at which the incomplete beta function is to be evaluated |
| P | Real | input: | a shape parameter |
| Q | Real | input: | a shape parameter |
| PSI | Real array (7) | input: | vector containing log gamma functions and their derivatives<br>PSI(1) = $\ln\{\text{Beta}(p,q)\}$<br>PSI(2) = $\psi_p$, the digamma function:<br>$\psi_p = d\,\ln\{\Gamma(p)\}/d\,p$<br>PSI(3) = $\psi'_p$, the trigamma function:<br>$\psi'_p = d^2\ln\{\Gamma(p)\}/(d\,p)^2$<br>PSI(4) = $\psi_q$<br>PSI(5) = $\psi'_q$<br>PSI(6) = $\psi_{p+q}$<br>PSI(7) = $\psi'_{p+q}$ |

*Output Parameters*

| | | | |
|---|---|---|---|
| DER | Real array (6) | output: | vector containing evaluated incomplete beta function and its derivatives<br>DER(1) $= I_{x,p,q}$<br>DER(2) $= I^{p}_{x,p,q}$<br>DER(3) $= I^{pp}_{x,p,q}$<br>DER(4) $= I^{q}_{x,p,q}$<br>DER(5) $= I^{qq}_{x,p,q}$<br>DER(6) $= I^{pq}_{x,p,q}$ |
| NAPPX | Integer | output: | highest order approximant evaluated |
| ERRAPX | Real | output: | approximate maximum absolute error of computed derivatives |
| IFAULT | Integer | output: | a fault indicator<br>= 0 if completion is successful<br>= 1 if X is less than 0 or greater than 1<br>= 2 if P is less than 0<br>= 3 if Q is less than 0<br>= 4 if derivatives were set to 0 because $I_{x,p,q}$ was evaluated to 0 or 1<br>= 5 if evaluation stopped after MAXAPPX terms. |

*Constants*

Three constants, ERR, MAXAPPX, and MINAPPX, are set in a data statement. The constant ERR should be set to $10^{-j}$, where $j = \text{round}\{b_f \log_{10}(2)\} - 4$ and $b_f$ is the number of bits used to represent the fractional component of floating point numbers. For example, suppose that floating point numbers are represented using 64 bits — one bit for the sign, 11 bits for the exponent, and 52 bits for the fraction. Then ERR should be set to $10^{-12}$. If $b_f$ is not known, then $b_f$ can be computed by function N2DIGIT given in Boik (1993, pp. 571). The constant MAXAPPX controls the highest order approximant that will be evaluated. The constant MINAPPX controls the minimum order approximant that will be evaluated.

*Auxiliary Algorithms*

No auxiliary algorithms are called by subroutine INBEDER. The user, however, is required to input values for the log of the complete beta function and first and second derivatives of the log gamma function. The algorithms of Lanczos (1964), Bernardo (1976), and Schneider (1978) are included and may be used to compute the required values. Alternative algorithms for the log gamma function and its first derivative were constructed by Cody (1993). Cody's algorithms are more accuarte than the older algorithms. FORTRAN source code for these algorithms can be obtained from Netlib.

*Use of Fortran Driver Program*

As an illustration, a diary of a double precision session appears below.

```
Number of Bits for Fractional Component
of Floating Point Numbers is   52

The constant ERR should be set to  0.1D-11

To exit program, type Control-C

 Input p, q, and x
1.5 11 .001
 Highest order approximant evaluated =   3
Approximate maximum absolute error = 0.28102520E-15, Ifault = 0
 I = 0.89170111E-03,  Ip =-0.45720356E-02, Ipp = 0.23080438E-01
Iq = 0.11845673E-03, Iqq = 0.51418717E-05, Ipq =-0.53324285E-03

 Input p, q, and x
^C
```

## Matlab

*function [der,psi,nappx,errapx] = inbeder(x,p,q)*

*Input Parameters*

- $x$ Input: vector of length $k$ containing values to which beta function is to be integrated
- $p, q$ Input: beta shape parameters, either vectors with same dimension as $x$ or scalars

*Output Parameters*

| | |
|---|---|
| der | output: matrix of dimension $k \times 6$ |
| | der(:,1) $= I_{x,p,q}$ |
| | der(:,2) $= I^{p}_{x,p,q}$ |
| | der(:,3) $= I^{pp}_{x,p,q}$ |
| | der(:,4) $= I^{q}_{x,p,q}$ |
| | der(:,5) $= I^{qq}_{x,p,q}$ |
| | der(:,6) $= I^{pq}_{x,p,q}$ |
| psi | output: matrix of dimension $k \times 7$ |
| | psi(:,1) $= \log[\mathrm{Beta}(p,q)]$ |
| | psi(:,2) $= \psi_p$ |
| | psi(:,3) $= \psi'_p$ |
| | psi(:,4) $= \psi_q$ |
| | psi(:,5) $= \psi'_q$ |
| | psi(:,6) $= \psi_{p+q}$ |
| | psi(:,7) $= \psi'_{p+q}$ |
| nappx | output: highest order approximant evaluated. Iteration stops if nappx $>$ maxappx |
| errapx | output: approximate maximum absolute error of computed derivatives |

*Optional Arguments*

Function inbeder can have one or two additional input arguments. Using [der,psi,nappx] = inbeder(x,p,q,minappx,maxappx) sets parameters MINAPPX and MAXAPPX which control the minimum and maximum order approximant that will be returned. Default values are 3 and 200. Using inbeder(x,p,q,minappx) changes MINAPPX only and using inbeder(x,p,q,[ ],maxappx) changes MAXAPPX only.

*Constants*

One constant, err, is set in function inbeder. The constant is set to err $=$ eps $\times 10^4$, where eps is an internal Matlab constant.

*Additional Functions*

The following Matlab functions are called by function inbeder.

1. function derconf: computes derivatives of $a_n$ and $b_n$ with respect to $p$ and/or $q$ when $n = 1$.

2. function subd: computes derivatives of $a_n$ and $b_n$ with respect to $p$ and/or $q$ when $n \geq 2$.

3. function distinct: finds distinct entries in a vector.

4. function digam: computes the digamma function for an input vector.

5. function trigam: computes the trigamma function for an input vector.
6. function vec: stacks the elements of a matrix.
7. function devec: arranges the elements of a vector in a matrix.

These functions should be saved with filenames derconf.m, subd.m, distinct.m, digam.m, trigam.m, vec.m, and devec.m, respectively.

## S–Plus

*ibd <- inc.beta.deriv(x,p,q)*

*Input Parameters*

| | |
|---|---|
| $x$ | vector of length $k$ containing values to which beta function is to be integrated |
| $p, q$ | beta shape parameters, either vectors with same dimension as $x$ or scalars |

*Output Parameters*

The output of function inc.beta.deriv, namely ibd, is a list of 15 elements. Each of the first 13 elements in the list is a vector having the same length as $x$. Each of the last two elements in the list is a scalar.

| | |
|---|---|
| I | $= I_{x,p,q}$ |
| Ip | $= I^{p}_{x,p,q}$ |
| Ipp | $= I^{pp}_{x,p,q}$ |
| Iq | $= I^{q}_{x,p,q}$ |
| Iqq | $= I^{qq}_{x,p,q}$ |
| Ipq | $= I^{pq}_{x,p,q}$ |
| log.Beta | $= \log[\text{Beta}(p,q)]$ |
| digamma.p | $= \psi_p$ |
| trigamma.p | $= \psi'_p$ |
| digamma.q | $= \psi_q$ |
| trigamma. | $= \psi'_q$ |
| digamma.pq | $= \psi_{p+q}$ |
| trigamma.pq | $= \psi'_{p+q}$ |
| nappx | highest order approximant evaluated. Iteration stops if nappx > maxappx |
| errapx | approximate maximum absolute error of computed derivatives |

*Additional Optional Inputs*

In addition to the required arguments above, the user may specify the following additional arguments.

| | |
|---|---|
| err | approximation control variable, iteration stops when maximum relative change is less than err |
| minappx | minimum order of approximation (default is 3) |
| maxappx | maximum order of approximation (default is 200) |

*Additional Functions*

The following S–Plus functions are called by function inc.beta.deriv.

1. function digamma: computes the digamma function for an input vector.

2. function trigamma: computes the trigamma function for an input vector.

*Help Files*

Help files are included for three of the defined functions. On unix systems these files should be stored in a .Data/.Help directory with filenames inc.beta.deriv, digamma, and trigamma.

# Accuracy

Benchmarks for evaluating the accuracy of the algorithm were obtained by computing the derivatives $I^p_{x,p,q}$, $I^q_{x,p,q}$, etc by means of numerical integration. Derivatives for several test cases were computed to 32 significant digits of accuracy using Maple V (Char et al., 1991) numerical integration routines. The output was compared to output from an quadruple precision version (floating point numbers represented using 128 bits) of Fortran subroutine INBEDER. The Fortran and Maple results agreed to at least 23 significant digits, except when the exact derivative is zero. It can be shown that $I^{p,q}_{.5,p,q} = 0$ whenever $p = q$. The derivatives to 8 significant digits are displayed in Table 1.

To gage the accuracy of the Fortran (double precision), Matlab, and S–plus versions of the algorithm, an additional list of test values was used. The values were obtained by equating $I_{x,p,q}$, in turn to 0.01, 0.10, and 0.50 and solving for $x$, for each $p, q \in \{0.10, 1.0, 10000.0\}$. The 48 $x$ values were then rounded to 8 significant digits and the derivatives were computed using the quadruple precision Fortran version of INBEDER. All three version of INBEDER yielded at least seven significant digits of accuracy for each test case except when the derivative is zero.

# Applications

## Maximum Likelihood Estimation of Beta Parameters from Censored Data

Gnanadesikan, Pinkham, and Hughes (1967) described an algorithm for maximum likelihood estimation of the parameters of the beta distribution from the smallest order statistics. Ignoring additive constants, the log likelihood function given the smallest $k$ order statistics from a sample of size $n$ is the following:

$$L(p, q|x_1, \ldots, x_k) =$$
$$(p-1)\sum_{i=1}^{k} \ln(x_i) + (q-1)\sum_{i=1}^{k} \ln(1-x_i) - k\ln\{\text{Beta}(p,q)\} + (n-k)\ln\left(1 - I_{x_k,p,q}\right).$$

Using subroutine INBEDER, the log likelihood function is readily maximized using a Newton-Raphson algorithm. To solve the likelihood equations, Gnanadesikan et al approximated first derivatives of the incomplete beta function using a seven term Laplacian expansion.

Gnanadesikan et al illustrated their algorithm using a sample of size $n = 20$ from a beta distribution having parameters $p = 1.5$ and $q = 11.0$. Their results along with results based on INBEDER are given in Table 2. It is apparent from Table 2 that the Laplacian expansion used by Gnanadesikan et al is not sufficiently accurate. Their estimates do not maximize the likelihood function and are especially inaccurate when the ratio $k/n$ is small. Also, from Table 2 it appears that the maximum likelihood estimates are positively biased when $k/n$ is small. The variances and covariances of the maximum likelihood estimators can be estimated by the inverse of the observed information matrix.

## Maximum Likelihood Estimation in Truncated Beta and Truncated Beta-Binomial Distributions

The beta-binomial model is sometimes used to model binary responses that display greater variation than would be expected under a binomial model. Smith (1983) and Smith and Ridout (1995) described an algorithm for maximum likelihood estimation of the beta-binomial parameters.

A random variable $Y$ follows a beta-binomial model if conditional on $\Pi = \pi$, $Y \sim \text{Binomial}(n, \pi)$ and the marginal distribution of $\Pi$ is beta. If an investigator has reason to believe that the support of $\pi$ is restricted to a known subset of the $(0, 1)$ interval, then a truncated beta-binomial model may be more appropriate. Suppose that conditional on $\Pi = \pi$, $Y \sim \text{Binomial}(n, \pi)$ and that the marginal density of $\Pi$ is

$$h_\Pi(\pi) = \frac{\pi^{p-1}(1-\pi)^{q-1}}{\left(I_{\tau_2,p,q} - I_{\tau_1,p,q}\right)\text{Beta}(p,q)} I_{(\tau_1,\tau_2)}(\pi), \tag{4}$$

where $\tau_1$ and $\tau_2$ are known constants satisfying $0 \le \tau_1 < \tau_2 \le 1$ and $I_{(\tau_1,\tau_2)}(\pi)$ is an indicator function. It is readily shown that the marginal probability mass function (pmf) of $Y$ is the following:

$$\Pr(Y = y|p, q, n, \tau_1, \tau_2) =$$
$$\binom{n}{y} \frac{\text{Beta}(p+y, q+n-y)\left(I_{\tau_2,p+y,q+n-y} - I_{\tau_1,p+y,q+n-y}\right)}{\text{Beta}(p,q)\left(I_{\tau_2,p,q} - I_{\tau_1,p,q}\right)} I_{\{0,1,\ldots,n\}}(y). \tag{5}$$

The usual beta-binomial pmf is obtained as the limit of (5) as $\tau_1 \to 0$ and $\tau_2 \to 1$. The truncated beta-binomial model in this article (5) differs from the truncated beta-binomial model described by Tripathi et al. (1994). Their model is obtained by excluding $Y = 0$ from the conventional beta-binomial model. That is, Tripathi et al. truncate the support of the discrete random variable $Y$, whereas the model in (5) truncates the support of the continuous random variable $\Pi$. Maximum likelihood estimates of the parameters of the truncated beta distribution in (4) or of the truncated beta-binomial distribution in (5) can be computed using a Newton-Raphson algorithm.

As an illustration, $N = 30$ random variates $\pi_1, \ldots, \pi_{30}$, were generated from the truncated beta distribution having parameters $\tau_1 = 0.2$, $\tau_2 = 0.7$, $p = 3$, and $q = 2$. Conditional on $\Pi = \pi_i$, a binomial variate was generated from the Binomial$(n_i, \pi_i)$ distribution, where the sample sizes $n_i$ for $i = 1, \ldots, 30$ were 25, 50, or 75. The data appear in Table 3.

Assuming that the values $\pi_i$ for $i = 1, \ldots, N$ have been observed, the MLEs of $p$ and $q$ can be obtained by maximizing the log likelihood function

$$L(p, q|\pi_1, \ldots, \pi_N; \tau_1, \tau_2) =$$
$$(p-1)\sum_{i=1}^{N} \ln(\pi_i) + (q-1)\sum_{i=1}^{N} \ln(1-\pi_i) - N\ln\left(I_{\tau_2,p,q} - I_{\tau_1,p,q}\right) - N\ln\left\{\text{Beta}(p,q)\right\},$$

with respect to $p$ and $q$. For the data in Table 3, the MLEs are $\widehat{p} = 4.153$ and $\widehat{q} = 1.680$. The variances and covariances of the maximum likelihood estimators can be estimated by the inverse of the observed information matrix. If truncation is ignored, the MLEs of the beta parameters are $\widehat{p} = 10.052$ and $\widehat{q} = 8.328$.

Assuming that the values $y_i$ for $i = 1, \ldots, N$ have been observed, but $\pi_i$ for $i = 1, \ldots, N$ are unobservable, the MLEs of $p$ and $q$ can be obtained by maximizing the log likelihood function

$$L(p, q|n_1, \ldots, n_N; y_1, \ldots, y_N; \tau_1, \tau_2) =$$
$$(p-1)\sum_{i=1}^{N} \ln\left\{\text{Beta}(p+y_i, q+n_i-y_i)\right\} + \sum_{i=1}^{N} \ln\left(I_{\tau_2,p+y_i,q+n_i-y_i} - I_{\tau_1,p+y_i,q+n_i-y_i}\right)$$
$$-N\ln\left\{\text{Beta}(p,q)\right\} - N\ln\left(I_{\tau_2,p,q} - I_{\tau_1,p,q}\right),$$

with respect to $p$ and $q$. For the data in Table 3, the MLEs are $\widehat{p} = 6.451$ and $\widehat{q} = 4.248$. For a sample size of $n = 10$, the maximum likelihood estimates of $\Pr(Y = y)$ for $y = 0, \ldots, 10$ are given in Table 4. For comparison, estimates computed under the conventional beta-binomial model and the conventional binomial model also are reported. For the conventional beta-binomial model (ignoring truncation), the MLEs of the beta parameters are $\widehat{p} = 11.749$ and $\widehat{q} = 10.075$. For the conventional binomial model, $\hat{\pi} = 0.533$. The Kullback-Leibler distances between the true distribution and the three estimates are 0.048 (truncated beta-binomial), 0.200 (beta-binomial), and 0.447 (binomial). It is apparent that precision is lost if truncation and/or extra variation are ignored.

## References

Abramowitz, M. & Stegun, I. A. (Eds.) (1965). *Handbook of Mathematical Functions*, New York, Dover Publications, Inc.

Bernardo, J. M. (1976). Algorithm AS 103: Psi (digamma) function. *Appl. Statist.*, **25**, 315–317.

Boik, R. J. (1993). Algorithm AS 284: Null distribution of a statistic for testing sphericity and additivity: A Jacobi polynomial expansion. *Appl. Statist.*, **42**, 567–584.

Char, B. W., Geddes, K. O., Gonnet, G. H., Leong, B. L., Monagan, M. B., & Watt, S. M. (1991) *Maple V Reference Manual.* New York: Springer-Verlag.

Cody, W. J. (1993). ALGORITHM 715: SPECFUN — A portable FORTRAN package of special function routines and test drivers. *ACM Transactions on Mathematical Software*, **19**, 22–32.

Dutka, J. (1981). The incomplete beta function — A historical profile. *Archive for History of the Exact Sciences*, **24**, 11–29.

Gautschi, W. (1967). Computational aspects of three-term recurrence relations. *SIAM review*, **9**, 24–82.

Gnanadesikan, R., Pinkham, R. S., & Hughes, L. P. (1967) Maximum likelihood estimation of the parameters of the beta distribution from smallest order statistics. *Technometrics*, **9**, 607–620.

Lanczos, C. (1964). A precision approximation of the gamma function, *SIAM Journal on Numerical Analysis, Ser. B*, **1**, 86–96.

Moore, R. J. (1982). Algorithm AS 187: Derivatives of the incomplete gamma integral. *Appl. Statist.*, **32**, 330–335.

Müller, J. H. (1931). On the application of continued fractions to the evaluation of certain integrals, with special reference to the incomplete Beta function. *Biometrika*, **23**, 284–297.

Schneider, B. E. (1978). Algorithm AS 121: Trigamma function. *Appl. Statist.*, **27**, 97–99.

Smith, D. M. (1983). Algorithm AS 189: Maximum likelihood estimation of the parameters of the beta binomial distribution. *Appl. Statist.*, **32**, 196–204.

Smith, D. M. & Ridout, M. S. (1995). Comment on “Algorithm AS 189: Maximum likelihood estimation of the parameters of the beta-binomial distribution” (V32 p196-204). *Appl. Statist.*, **44** 545–547.

Tretter, M. J. & Walster, G. W. (1976). A fast algorithm for significant digit computation of the incomplete beta function for extreme values. *Proceedings of the Computer Science and Statistics* 9$^{\text{th}}$ *Annual Symposium on the Interface*, 293–295, Boston: Prindle, Weber, & Schmidt.

Tretter, M. J. & Walster, G. W. (1979). Continued fractions for the incomplete beta function: Additions and corrections. *Ann. Statist.*, **7**, 462–465.

Tretter, M. J. & Walster, G. W. (1980). Analytic subtraction applied to the incomplete gamma and beta functions. *SIAM Journal on Scientific and Statistical Computing*, **1**, 321–326.

Tripathi, R. C., Gupta, R. C., & Gurland, J. (1994). Estimation of parameters in the beta binomial model. *Ann. Inst. Statist. Math.*, **46**, 317–331.

# Appendix: Required Derivatives

## Derivatives of $K_{x,p,q}$

In the following equations, $\psi$ is the psi or digamma function and $\psi'$ is the trigamma function. That is,

$$\psi_p = \frac{\partial \ln \Gamma(p)}{\partial p} \text{ and } \psi'_p = \frac{\partial^2 \ln \Gamma(p)}{(\partial p)^2}.$$

$$\frac{\partial K_{x,p,q}}{\partial p} = K_{x,p,q}\left\{\ln(x) - p^{-1} + \psi_{p+q} - \psi_p\right\}$$

$$\frac{\partial^2 K_{x,p,q}}{(\partial p)^2} = K_{x,p,q}\left[\left\{\ln(x) - p^{-1} + \psi_{p+q} - \psi_p\right\}^2 + p^{-2} + \psi'_{p+q} - \psi'_p\right]$$

$$\frac{\partial K_{x,p,q}}{\partial q} = K_{x,p,q}\left\{\ln(1-x) + \psi_{p+q} - \psi_q\right\}$$

$$\frac{\partial^2 K_{x,p,q}}{(\partial q)^2} = K_{x,p,q}\left[\left\{\ln(1-x) + \psi_{p+q} - \psi_q\right\}^2 + \psi'_{p+q} - \psi'_q\right]$$

$$\frac{\partial^2 K_{x,p,q}}{(\partial p)(\partial q)} = K_{x,p,q}\left[\left\{\ln(x) - p^{-1} + \psi_{p+q} - \psi_p\right\}\left\{\ln(1-x) + \psi_{p+q} - \psi_q\right\} + \psi'_{p+q}\right]$$

## Derivatives of $a_n$

$$\frac{\partial a_n}{\partial p} = \begin{cases} -\dfrac{pf\,(q-1)}{q\,(p+1)^2} & \text{if } n = 1; \\[2ex] \begin{aligned} &-(n-1)\,f^2p^2\,(q-n)\Big[(-8+8p+8q)\,n^3 \\ &+\{16p^2 + (-44+20q)\,p + 26 - 24q\}\,n^2 \\ &+(10p^3 + (14q-46)\,p^2 + (-40q+66)\,p - 28 + 24q)\,n \\ &+2p^4 + (-13+3q)\,p^3 + (-14q+30)\,p^2 \\ &+(-29+19q)\,p + 10 - 8q\Big] \\ &\div\left\{q^2\,(p+2n-3)^2\,(p+2n-2)^3\,(p+2n-1)^2\right\} \end{aligned} & \text{if } n > 1 \end{cases}$$

$$\frac{\partial^2 a_n}{(\partial p)^2} = \begin{cases} 2\frac{pf\,(q-1)}{q\,(p+1)^3} & \text{if } n=1; \\ \begin{array}{l} 2\,(n-1)\,f^2p^2\,(q-n) \\ \Big[-32n^6+(-64p+32q+160)\,n^5 \\ +\{(160+160q)\,p-292-160q\}\,n^4 \\ +\{80p^3+(240q-240)\,p^2+(-640q+68)\,p \\ +332q+196\}n^3 \\ +\{70p^4+(160q-400)\,p^3+(-720q+771)\,p^2 \\ +(960q-524)\,p+43-356q\}n^2 \\ +\{24p^5+(50q-190)\,p^4+(-320q+574)\,p^3 \\ +(711q-813)\,p^2+(-640q+522)\,p \\ +195q-113\}n \\ +3p^6+(6q-30)\,p^5+(-50q+121)\,p^4 \\ +(-251+157q)\,p^3+(-231q+282)\,p^2 \\ +(-163+161q)\,p+38-43q\Big] \\ \div\Big\{q^2\,(p+2n-3)^3\,(p+2n-2)^4\,(p+2n-1)^3\Big\} \end{array} & \text{if } n\geq 2 \end{cases}$$

$$\frac{\partial\, a_n}{\partial\, q} = \begin{cases} \frac{fp}{q\,(p+1)} & \text{if } n=1; \\ \frac{p^2f^2\,(n-1)\,(p+n-1)\,(2q+p-2)}{q^2\,(p+2n-3)\,(p+2n-2)^2\,(p+2n-1)} & \text{if } n\geq 2 \end{cases}$$

$$\frac{\partial^2 a_n}{(\partial\, q)^2} = \begin{cases} 0 & \text{if } n=1; \\ 2\frac{(n-1)\,f^2p^2\,(p+n-1)}{q^2\,(p+2n-3)\,(p+2n-2)^2\,(p+2n-1)} & \text{if } n\geq 2 \end{cases}$$

$$\frac{\partial^2 a_n}{(\partial p)(\partial q)} = \begin{cases} -\dfrac{fp}{q\,(p+1)^2} & \text{if } n = 1; \\ -\,(n-1)\,f^2p^2\Big[-8n^4 + (16+16q-12p)\,n^3 \\ +\{2p^2 + (40q-4)\,p - 48q + 2\}n^2 \\ +\{7p^3 + (-32+28q)\,p^2 + (-80q+47)\,p \\ -20+48q\}n & \text{if } n \geq 2 \\ +2p^4 + (-13+6q)\,p^3 + (-28q+30)\,p^2 \\ +(-29+38q)\,p + 10 - 16q\Big] \\ \div\Big\{q^2\,(p+2n-3)^2\,(p+2n-2)^3\,(p+2n-1)^2\Big\} \end{cases}$$

## Derivatives of $b_n$

$$\frac{\partial\, b_n}{\partial\, p} = \frac{pf\left\{(-4p-4q+4)\,n^2 + \left(4p-4+4q-2p^2\right)n + p^2q\right\}}{q\,(p+2n-2)^2\,(p+2n)^2}$$

$$\begin{aligned} \frac{\partial^2 b_n}{(\partial\, p)^2} = \; & -2pf\Big[8n^4 - 16qn^3 + \Big\{-6p^2 + (12-12q)\,p + 24q - 16\Big\}\,n^2 \\ & + \{-2p^3 + 6p^2 + (-12+12q)\,p + 8 - 8q\}\,n + p^3q\Big] \\ & \div\Big\{q\,(p+2n-2)^3\,(p+2n)^3\Big\} \end{aligned}$$

$$\frac{\partial\, b_n}{\partial\, q} = -\frac{p^2 f}{q\,(p+2n-2)\,(p+2n)}$$

$$\frac{\partial^2 b_n}{(\partial\, q)^2} = 0$$

$$\frac{\partial^2 b_n}{(\partial\, p)(\partial\, q)} = \frac{pf\left(-4n^2+4n+p^2\right)}{q\,(p+2n-2)^2\,(p+2n)^2}$$

Table 1

First and second derivatives of the incomplete beta function for selected values of $p$, $q$, and $x$.

| $p$ | $q$ | $x$ | $I_{p,q,x}$ <br> $I^{pp}_{p,q,x}$ <br> $I^{qq}_{p,q,x}$ | $I^{p}_{x,p,q}$ <br> $I^{q}_{x,p,q}$ <br> $I^{pq}_{x,p,q}$ |
|---|---|---|---|---|
| 1.5 | 11.0 | 0.001 | $8.9170111 \times 10^{-04}$ | $-4.5720356 \times 10^{-03}$ |
| | | | $2.3080438 \times 10^{-02}$ | $1.1845673 \times 10^{-04}$ |
| | | | $5.1418717 \times 10^{-06}$ | $-5.3324285 \times 10^{-04}$ |
| 1.5 | 11.0 | 0.500 | $9.9861069 \times 10^{-01}$ | $-2.5501997 \times 10^{-03}$ |
| | | | $-3.5047111 \times 10^{-03}$ | $9.0824388 \times 10^{-04}$ |
| | | | $-5.8941710 \times 10^{-04}$ | $1.5603497 \times 10^{-03}$ |
| 1000.0 | 1000.0 | 0.500 | $5.0000000 \times 10^{-01}$ | $-8.9224793 \times 10^{-03}$ |
| | | | $4.4630987 \times 10^{-06}$ | $8.9224793 \times 10^{-03}$ |
| | | | $-4.4630987 \times 10^{-06}$ | $0.0000000 \times 10^{+00}$ |
| 1000.0 | 1000.0 | 0.550 | $9.9999632 \times 10^{-01}$ | $-3.6713108 \times 10^{-07}$ |
| | | | $-3.4809144 \times 10^{-08}$ | $4.0584118 \times 10^{-07}$ |
| | | | $-4.2964422 \times 10^{-08}$ | $3.8682578 \times 10^{-08}$ |

Table 2

Maximum likelihood estimates of the beta parameters from the $k$ smallest order statistics when $p = 1.5$, $q = 11.0$, and $n = 20$.

| | | Gnanadesikan et al | | | Current | | |
|---|---|---|---|---|---|---|---|
| $k$ | $k/n$ | $\hat{\alpha}$ | $\hat{\beta}$ | $\ln(L)$ | $\hat{\alpha}$ | $\hat{\beta}$ | $\ln(L)$ |
| 2 | 0.1 | 1.443 | 6.472 | 0.658 | 3.295 | 38.146 | 1.088 |
| 4 | 0.2 | 1.472 | 8.145 | 2.843 | 3.367 | 37.714 | 3.656 |
| 6 | 0.3 | 1.483 | 9.602 | 5.954 | 3.817 | 46.262 | 7.404 |
| 8 | 0.4 | 1.564 | 9.760 | 6.668 | 2.029 | 15.124 | 6.858 |
| 10 | 0.5 | 1.672 | 11.765 | 10.788 | 2.519 | 21.977 | 11.325 |
| 12 | 0.6 | 1.774 | 13.175 | 14.120 | 2.520 | 21.857 | 14.592 |
| 14 | 0.7 | 1.902 | 14.995 | 18.095 | 2.632 | 23.378 | 18.560 |
| 16 | 0.8 | 1.732 | 12.202 | 17.381 | 1.768 | 12.554 | 17.383 |
| 18 | 0.9 | 1.800 | 12.920 | 20.931 | 1.812 | 13.035 | 20.932 |
| 20 | 1.0 | 1.793 | 12.781 | 24.313 | 1.793 | 12.784 | 24.313 |

Table 3

Simulated data from truncated beta and truncated beta-binomial distributions with parameters $\tau_1 = 0.2$, $\tau_2 = 0.7$, $p = 3$, and $q = 2$.

| Sample | $\pi_i$ | $n_i$ | $y_i$ | Sample | $\pi_i$ | $n_i$ | $y_i$ | Sample | $\pi_i$ | $n_i$ | $y_i$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.4376 | 25 | 13 | 11 | 0.2379 | 50 | 15 | 21 | 0.2937 | 75 | 19 |
| 2 | 0.4408 | 25 | 10 | 12 | 0.4263 | 50 | 22 | 22 | 0.3899 | 75 | 30 |
| 3 | 0.4857 | 25 | 9 | 13 | 0.5026 | 50 | 26 | 23 | 0.4314 | 75 | 34 |
| 4 | 0.6009 | 25 | 14 | 14 | 0.5408 | 50 | 25 | 24 | 0.4883 | 75 | 37 |
| 5 | 0.6098 | 25 | 15 | 15 | 0.5459 | 50 | 21 | 25 | 0.5090 | 75 | 31 |
| 6 | 0.6211 | 25 | 19 | 16 | 0.5810 | 50 | 27 | 26 | 0.5804 | 75 | 47 |
| 7 | 0.6320 | 25 | 17 | 17 | 0.5905 | 50 | 30 | 27 | 0.6247 | 75 | 45 |
| 8 | 0.6338 | 25 | 18 | 18 | 0.5997 | 50 | 31 | 28 | 0.6459 | 75 | 47 |
| 9 | 0.6775 | 25 | 17 | 19 | 0.6405 | 50 | 30 | 29 | 0.6467 | 75 | 55 |
| 10 | 0.6795 | 25 | 16 | 20 | 0.6983 | 50 | 29 | 30 | 0.6763 | 75 | 50 |

Table 4

Maximum likelihood estimates of $\Pr(Y = y)$ from data in Table 3. MLE: estimates were computed assuming a truncated beta-binomial distribution with $\tau_1 = 0.2$ and $\tau_2 = 0.7$; $\text{MLE}_{bb}$: estimates were computed assuming a beta-binomial distribution; $\text{MLE}_{bi}$: estimates were computed assuming a binomial distribution.

| | $\Pr(Y = y)$ | | | |
|---|---|---|---|---|
| $y$ | True | MLE | $\text{MLE}_{bb}$ | $\text{MLE}_{bi}$ |
| 0 | 0.011 | 0.003 | 0.002 | 0.000 |
| 1 | 0.030 | 0.013 | 0.014 | 0.006 |
| 2 | 0.054 | 0.035 | 0.046 | 0.029 |
| 3 | 0.080 | 0.070 | 0.099 | 0.088 |
| 4 | 0.105 | 0.114 | 0.158 | 0.176 |
| 5 | 0.126 | 0.154 | 0.199 | 0.241 |
| 6 | 0.140 | 0.178 | 0.197 | 0.229 |
| 7 | 0.144 | 0.175 | 0.153 | 0.149 |
| 8 | 0.135 | 0.141 | 0.089 | 0.064 |
| 9 | 0.110 | 0.086 | 0.035 | 0.016 |
| 10 | 0.066 | 0.031 | 0.007 | 0.002 |