Recursive Formulas, Fast Algorithm and Its Implementation of Partial Derivatives of the Beta Function

†H.Z. Qin[1], Youmin Lu[2] and Nina Shang[1]

[1]Institute of Applied Mathematics, Shandong University of Technology Zibo, Shandong, P. R. China
[2]Department of Mathematics and Computer Science Bloomsburg University Bloomsburg, PA 17815, USA
†Corresponding author: $qinhz\_000@163.com$

Abstract

In this paper, the values of Beta function $B(x,y)$ at $(-n,y),(x,-m),(-n,-m)$ for $n,m = 0,1,2,\cdots,x,y \neq 0,1,2,\cdots$ are redefined and some recurrence formulas on the partial derivatives $B_{p,q}(x,y) = \frac{\partial^{q+p}}{\partial x^p \partial y^q}B(x,y)$ of the Beta function are established in Mathematica,where p,q are the positive integers, and x,y are complex numbers, When $x = n, n+\frac{1}{2}, y = m, m+\frac{1}{2}$ and $n,m = 0,\pm1,\pm2,\cdots$, $B_{p,q}(x,y)$ can be expressed as Riemann zeta function. We provide a fast algorithm, give its implementation in Mathematica, obtain closed forms of many generalized integrals and achieve high-precision calculation of these integrals.

Keywords: Riemann zeta function; Beta Function; Partial derivatives of the Beta Function; high-precision.

Introduction

In Mathematical software such as Mathematica, Maple and Matlab there are special functions, and the Beta function is one of them. By partial derivatives of the Beta function some generalized integral can be calculated. For example

$$\int_0^1 t^{x-1}(1-t)^{y-1}\ln^p t \ln^q(1-t)dt = B_{p,q}(x,y), \tag{1.1}$$

where $B_{p,q}(x,y) = \frac{\partial^{p+q}}{\partial x^p \partial y^q}B(x,y)$. However, we note that although the following integral exists

$$\int_0^1 t^{-2}(1-t)^{-2}\ln^p t \ln^q(1-t)dt \tag{1.2}$$

for integer $p,q \geq 2$, but $B_{p,q}(-1,-1) = \infty(D[D[Beta[xx,yy],\{xx,p\}]/.xx \to -1\{yy,q\}]/.yy \to -1)$ in Mathematica. By Mathematica symbolic integral, the closed form of the integral (1.2) can also be obtained for smaller p,q, but very time consuming, and the closed form of the integral (1.2) are difficult to obtain for larger integer p,q. By closed form, we mean that the integral can be expressed analytically in terms of a finite number of Riemann zeta functions and some constant π and the Euler-Mascheroni constant γ, etc.

The Beta function was the first known scattering amplitude in string theory, first conjectured by Gabriele Veneziano. It also occurs in the theory of the preferential attachment process, a type of stochastic urn process[1,2], the supersymmetric gauge theories[3] and other physical[4-5].

For $B_{p,q}(x,y)$ we have established a recurrence formula by the neutrix calculus[6-8]. In this article, in Mathematica, we give the function DBeta of the calculating $B_{p,q}(x,y)$ for positive integers p and q and complex numbers x and y. Through a number of examples show that our program is very effective is better in the calculation of the closed form and the numerical integration.

In the following sections, we introduce additional definitions of the Beta function, some recurrence formulas and an algorithm for calculating the values of partial derivatives of the Beta function.

Software Summary

Manuscript title: Remark on Beta Function and it's Partial Derivatives in mathematca.

Authors: Huizeng Qin, Youmin Lu Nina Shang

Title of program: BetaAll (for computing the Beta Function $B(x, y)$ in all complex values of x and y), DBeta (for computing partial derivatives $\frac{\partial^{p+q}}{\partial x^p \partial y^q} B(x, y)$ of the Beta Function in all complex values of x and y).

Licensing provisions: None

Computer: ACPI Multiprocessor PC.

Operating system: Microsoft windows XP, but does not depend on the particular operating system.

Programming language used: Mathematica 9

Memory required to execute with typical data: 2 Megabytes.

CPC Library Classification: 6.5 Software including Parallel Algorithms

Solution method: For the partial derivatives of the Beta Function, the recurrence formulas (2.5),(2.6), (2.7)-(2.9) and (2.12) in this paper are employed. BetaAll is composed of the following five key subprograms: DBeta, PolyGammaAmend, DPochhammer, DBeta1 and DBeta2. BetaAll is based on the formulas (2.2)-(2.4) and Beta in Mathematica. PolyGammaAmend is based on the formulas (2.13)-(2.20). DPochhammer is based on the formula (2.11). DBeta1 is based on the formulas (2.5) and (2.6) for x and $x + y \neq 0, -1, -2, \cdots$. DBeta2 is based on the formulas (2.7)-(2.9) and (2.12) for $x = -1, -2, \cdots$ or $y = -1, -2, \cdots$ or $x+y = -1, -2, \cdots$.

Nature of the problem: The Beta function $B(x, y)$ is a very important special function. Many mathematical softwares have defined inherent function (for example $Beta[x, y]$ in Mathematica) for computing the Beta function $B(x, y)$. However, wnen $x = -1, -2, \cdots$ or $y = -1, -2, \cdots$, $Beta[x, y]$ is not defined in Mathematica, and similar problem exists in other mathematics software. In addition, it is possible to use symbolic deferentiation and integration in Mathematica to obtain the partial derivatives of the Beta function, but it is very inefficient in speed and can rarely get the closed forms(it cannot get the closed form although it exists). Therefore, we give an algorithm that calculates the values of Beta function and its partial derivetives in the entire complex plane. In this way, one can obtain the closed forms of all integrals that can be expressed in terms of partial derivatives of Beta function.

Typical running time: The running time of BetaAll depends strongly on p, q, x, y and the number of bits required by computation precision. BetaAll is 30-10000 times faster than Integrate in Mathematica. As the number of bits for precision increases, the advantage of BetaAll becomes more significant.

The purpose of the program design: This process is designed to calculate the values of the partial derivatives of the Beta function. Thus, it can be used to achieve fast and high-precision calculation of generalized integrals that can be represented in terms of partial derivatives of the Beta function, regardless of computing power. The speed of this process is far superior to Integrate and NIntegrate in Mathematica.

Additional Definition and a Recurrence Formula of Partial derivatives of the Beta Function

The values of x and y must be real and non-negative for the Beta function $B(x,y)$ in Matlab. Although they may be complex in Mathematica and Maple, the definitions there

$$B(-n,y)=\infty, B(x,-m)=\infty, B(-n,-m)=\infty, n,m=0,1,2,\cdots, \tag{2.1}$$

where x and y is not an integer, lead to the following unreasonable results:

$$B(-1,\frac{1}{2})=\infty,\ B(-\frac{3}{2},\frac{1}{2})=0, B(-1,\frac{5}{2})=\infty,\ B(-\frac{3}{2},\frac{5}{2})=\pi.$$

To remedy this problem, it is necessary to modify (2.1). For this reason, we do give some additional definitions and results[9-12].

For $B(x,y)$ the following definitions are given for $x>0, y>0$ and $n,m=1,2,\cdots$:

$$\begin{aligned} B(n,-m)=B(-m,n)&=\sum_{l=0,l\neq m}^{n-1} C_{n-1}^{l}\frac{(-1)^l}{l-m}, m=0,1,2,\cdots,n=1,2,\cdots \\ &=\begin{cases}\frac{(-1)^m(m-1)!(n-m)!}{n!}, n=1,2,\cdots,m, m=1,2,\cdots \\ \frac{(-1)^n(m-1)!(H_n-H_{m-n-1})}{n!(m-n-1)!}, n=m+1,m+2,\cdots,m=1,2,\cdots,n\end{cases}\end{aligned} \tag{2.2}$$

$$\begin{gathered} B(-n,y)=(-1)^n C_{y-1}^{n}\left((y-n-1)B_{0,1}(y-n-1,1)+H_n\right), \\ y\neq 0,-1,-2,\cdots, \\ B(x,-m)=B(-m,x), x\neq 0,-1,-2,\cdots, \end{gathered} \tag{2.3}$$

where $H_n=\sum\limits_{l=1}^{n}\frac{1}{l}$, and

$$B(-n,-m)=-\sum_{i=0}^{m-1}\begin{pmatrix} n+i \\ i\end{pmatrix}\frac{1}{m-i}-\sum_{j=0}^{n-1}\begin{pmatrix} m+j \\ j\end{pmatrix}\frac{1}{n-j}. \tag{2.4}$$

We obtain the following three groups of recurrence formulas of $B_{p,q}(x,y)$.

I. For integers $q,p\geq 1$ and complex numbers x,y satisfying $x,y,x+y\neq 0,-1,-2,\cdots$,

$$\begin{aligned} B_{0,q}(x,y)&=\sum_{j=0}^{q-1}C_{q-1}^{j}\left(\psi^{(q-1-j)}(y)-\psi^{(q-1-j)}(x+y)\right)B_{0,j}(x,y), \\ B_{p,q}(x,y)&=\sum_{j=0}^{q-1}C_{q-1}^{j}\left(\psi^{(q-1-j)}(y)-\psi^{(q-1-j)}(x+y)\right)B_{p,j}(x,y) \\ &\quad -\sum_{k=0}^{p-1}C_{p}^{k}\sum_{j=0}^{q-1}C_{q-1}^{j}\psi^{(p+q-1-k-j)}(x+y)B_{k,j}(x,y). \end{aligned} \tag{2.5}$$

or

$$\begin{aligned} B_{p,0}(x,y)&=\sum_{k=0}^{p-1}C_{p-1}^{k}(\psi^{(p-1-k)}(x)-\psi^{(p-1-k)}(x+y))B_{k,0}(x,y), \\ B_{p,q}(x,y)&=\sum_{k=0}^{p-1}C_{p-1}^{k}(\psi^{(p-1-k)}(x)-\psi^{(p-1-k)}(x+y))B_{k,q}(x,y) \\ &\quad -\sum_{j=0}^{q-1}C_{q}^{j}\sum_{k=0}^{p-1}C_{p-1}^{k}\psi^{(p+q-1-k-j)}(x+y)B_{k,j}(x,y) \end{aligned} \tag{2.6}$$

where $\psi(x)$ is the digamma function defined by

$$\psi(x)=\frac{d}{dx}\ln\Gamma(x)=-\gamma-\frac{1}{x}+\sum_{l=1}^{\infty}\left(\frac{1}{l}-\frac{1}{l+x}\right).$$

II. For integers $q,p,n,m\geq 0$ and complex numbers x,y satisfying $x,y\neq 0,-1,-2,\cdots$,

$$\begin{aligned}B_{p,q}(-n,y)=&\tfrac{1}{(p+1)a_{n+1,1}(-n)}\sum_{u=0}^{p+1}C_{p+1}^{u}\sum_{v=0}^{q}C_{q}^{v}a_{n+1,p+q+1-u-v}(y-n)B_{u,v}(1,y)\\&-\tfrac{1}{(p+1)a_{n+1,1}(-n)}\sum_{u=0}^{p-1}C_{p+1}^{u}a_{n+1,p+1-u}(-n)B_{u,q}(-n,y).\end{aligned}\tag{2.7}$$

$$\begin{aligned}B_{p,q}(x,-m)=&\tfrac{1}{(q+1)a_{m+1,1}(-m)}\sum_{u=0}^{p}C_{p}^{u}\sum_{v=0}^{q+1}C_{q+1}^{v}a_{m+1,p+q+1-u-v}(x-m)B_{u,v}(x,1)\\&-\tfrac{1}{(q+1)a_{m+1,1}(-m)}\sum_{v=0}^{q-1}C_{q+1}^{v}a_{m+1,q+1-v}(-m)B_{p,v}(-m,y)\end{aligned}.\tag{2.8}$$

and

$$\begin{aligned}B_{p,q}(-n,-m)=&\tfrac{(-1)^{n+m}}{(q+1)(p+1)n!m!}\sum_{u=0}^{p+1}C_{p+1}^{u}\sum_{v=0}^{q+1}C_{q+1}^{v}a_{n+m+2,p+q+2-u-v}(-n-m)B_{u,v}(1,1)\\&-\tfrac{(-1)^{n}}{(p+1)n!}\sum_{u=0}^{p-1}C_{p+1}^{u}a_{n+1,p+1-u}(-n)B_{u,q}(-n,-m)\\&-\tfrac{(-1)^{m}}{(q+1)m!}\sum_{v=0}^{q-1}C_{q+1}^{v}a_{m+1,q+1-v}(-m)B_{u,v}(-n,-m)\\&-\tfrac{(-1)^{n+m}}{(q+1)(p+1)n!m!}\sum_{u=0}^{p-1}C_{p+1}^{u}\sum_{v=0}^{q-1}C_{q+1}^{v}a_{n+1,p+1-u}(-n)a_{m+1,q+1-v}(-m)B_{u,v}(-n,-m).\end{aligned}\tag{2.9}$$

where

$$a_{n,i}(x)=\frac{d^{i}}{dx^{i}}\left(x\right)_{n}=i!\sum_{k=i}^{n}C_{k}^{i}(-1)^{n-k}s(n,k)x^{k-i},i=1,2,\cdots,\tag{2.10}$$

$$\left(x\right)_{n}=x(x+1)\cdots(x+n-1)=\sum_{k=1}^{n}(-1)^{n-k}s(n,k)x^{k},\tag{2.11}$$

and $s(n,k)$ is the Stirling number of the first kind.

III. For integers $q,p,n,m\geq 0$ and complex numbers x,y satisfying $x+y=0,-1,-2,\cdots, Rex\neq 0,-1,-2,\cdots$, we have the following recurrence relations

$$\begin{aligned}B_{p,q}(x,y)=&\tfrac{1}{(x)_n(y)_m}\sum_{u=0}^{p}C_{p}^{u}\sum_{v=0}^{q}C_{q}^{v}a_{n+m,p+q-u-v}(x+y)B_{u,v}(x+n,y+m)\\&-\tfrac{1}{(y)_m}\sum_{v=0}^{q-1}C_{q}^{v}a_{m,q-v}(y)B_{p,v}(x,y)-\tfrac{1}{(x)_n}\sum_{u=0}^{p-1}C_{p}^{u}a_{n,p-u}(x)B_{u,q}(x,y)\\&-\tfrac{1}{(x)_n(y)_m}\sum_{u=0}^{p-1}C_{p}^{u}\sum_{v=0}^{q-1}C_{q}^{v}a_{n,p-u}(x)a_{m,q-v}(y)B_{u,v}(x,y).\end{aligned}\tag{2.12}$$

Now we are ready to consider the closed form of $B_{p,q}(x,y)$. It is well-known that the digamma function $\psi(x)$ has the following identities:

$$\psi(n+x)=\psi(x)+\sum_{l=0}^{n-1}\frac{1}{(l+x)},\psi(x-n)=\psi(x)+\sum_{l=1}^{n}\frac{1}{(l-x)}, \tag{2.13}$$

$$\psi^{(k)}(x)=k!(-1)^{k+1}\zeta(k+1,x),k>0, \tag{2.14}$$

and

$$\begin{gathered}\psi^{(k)}(n+x)=k!(-1)^{k+1}\zeta(k+1,x)+(-1)^k k!\sum_{l=0}^{n-1}\frac{1}{(l+x)^{k+1}},k>0\\ \psi^{(k)}(x-n)=k!(-1)^{k+1}\zeta(k+1,x)+k!\sum_{l=1}^{n}\frac{1}{(l-x)^{k+1}},k>0,\end{gathered} \tag{2.15}$$

where $\zeta(s,x)$ is the Hurwitz zeta function defined by

$$\zeta(s,x)=\sum_{l=0}^{\infty}\frac{1}{(l+x)^s},\zeta(s,0)=\zeta(s,1).$$

The Hurwitz zeta function $\zeta(s,x)$ also has the following identity

$$\begin{gathered}\zeta(s,n+x)=\zeta(s,x)-\sum_{l=0}^{n-1}\frac{1}{(l+x)^s},\zeta(s,-n+x)=\zeta(s,x)+\sum_{l=1}^{n}\frac{1}{(x-l))^s},\\ \zeta(s,\tfrac{1}{2})=(2^s-1)\zeta(s).\end{gathered} \tag{2.16}$$

particularly[13],

$$\zeta(k,0)=\left\{\begin{array}{c}\gamma,k=1\\ \zeta(k),k>1\end{array}\right.,\zeta(k,\frac{1}{2})=\left\{\begin{array}{c}\gamma+2\ln 2,k=1\\ \left(2^k-1\right)\zeta(k),k>1\end{array}\right., \tag{2.17}$$

and

$$\begin{gathered}\left.\begin{array}{c}\zeta(2n+1,\frac{1}{3})\\ \zeta(2n+1,\frac{2}{3})\end{array}\right\}=\frac{3^{2n+1}-1}{2}\zeta(2n+1)\\ \pm\frac{\sqrt{3}}{2\pi}\left(\left(2n+2+3^{2n+2}\right)\zeta(2n+2)-2\sum_{l=0}^{n-1}3^{2n-2l}\zeta(2n-2l)\zeta(2l+2)\right)\end{gathered} \tag{2.18}$$

$$\begin{gathered}\left.\begin{array}{c}\zeta(2n+1,\frac{1}{4})\\ \zeta(2n+1,\frac{3}{4})\end{array}\right\}=2^{2n}(2^{2n+1}-1)\zeta(2n+1)\\ \pm\frac{1}{2\pi}\left(2n+2+4^{2n+2}\right)\zeta(2n+2)-2\sum_{l=0}^{n-1}4^{2n-2l}\zeta(2n-2l)\zeta(2l+2)\end{gathered} \tag{2.19}$$

$$\begin{gathered}\left.\begin{array}{c}\zeta(2n+1,\frac{1}{6})\\ \zeta(2n+1,\frac{5}{6})\end{array}\right\}=\frac{6^{2n+1}-3^{2n+1}-2^{2n+1}+1}{2}\zeta(2n+1)\\ \pm\frac{1}{2\sqrt{3}\pi}\left(6^{2n+2}-3^{2n+2}\right)\zeta(2n+2)-2\sum_{l=0}^{n-1}\left(6^{2n-2l}-3^{2n-2l}\right)\zeta(2n-2l)\zeta(2l+2)\end{gathered} \tag{2.20}$$

where $\zeta(1)=\gamma$.

Remark If $x=\pm n,\frac{1}{2}\pm n,y=\pm m,\frac{1}{2}\pm m,n,m=0,1,2,\cdots$, $B_{p,q}(x,y)$ certainly has closed form. If $x,y=\frac{1}{3}\pm n,\frac{1}{4}\pm n,\frac{1}{6}\pm n,n=0,1,2,\cdots$, $B_{p,q}(x,y)$ may have closed form. Otherwise, $B_{p,q}(x,y),p,q>1$ does not seem to have closed form.

Algorithms for Calculating $B_{p,q}(x,y)$ and comparison with the symbolic(numerical) integration in Mathematica

Algorithm

The source code $BetaAll[x,y]$ and $DBeta[x,y,p,q,all]$ that calculates the values of $B(x,y)$ and $B_{p,q}(x,y)$ is placed in the file beta.nb(Mathematica file format). $DBeta[x,y,p,q,all]$ calls five key subprograms BetaAll[x,y], PolyGammaAmend[k,x], DPochhammer[k,x], DBeta1[x,y,p,q,all] and DBeta2[x,y,p,q,all]. The following is our specific algorithm.

1) To obtain closed form, PolyGammaAmend performs the calculation of (2.13)-(2.18) when x is a real number or $x = a \pm n, a = 0, \frac{1}{2}, \frac{1}{3}, \frac{2}{3}, \frac{1}{4}, \frac{3}{4}, \frac{1}{6}, \frac{5}{6}, n = 0, 1, 2, 3, \cdots$. Otherwise, PolyGamma from Mtathematica replaces PolyGammaAmend.

2) BetaAll does the calculation of (2.2)-(2.4) when $x$or $y = 0, -1, -2$. Otherwise, Beta from Mtathematica replaces BetaAll.

3) DPochhammer[k, x] does calculation of (2.10).

4) DBeta1[x, y, p, q, all] is to calculate the values of $B_{p,q}(x,y)$ by using (2.5) and (2.6) when $x, y, x+y \neq 0, -1, -2, \cdots$. The parameter all indicates whether all the values of $B_{i,j}(x,y), i = 0, 1, 2, \cdots, p, j = 0, 1, 2, \cdots q$ are displayed or only the value of $B_{p,q}(x,y)$ is displayed depending on all is positive or zero.

5) DBeta[x, y, p, q, all] calls BetaDl[x, y, p, q, all] directly when $x, y, x+y \neq 0, -1, -2, \cdots$. Otherwise, DBeta[$x, y, p, q, all$] calls DBeta2[$x, y, p, q, all$] that calculates the values of $B_{p,q}(x,y)$ by using (2.7)-(2.9) and (2.12) and calling two subprograms BetaAll[x, y, p, q, all] and DPochhammer[k, x].

The above algorithm is run in the mathematics symbolic computation system. If the numerical calculation, we will be in front of the source code to add a "N", for example, change BetaAll to NBetaAll and the above algorithm is run in the specified precision Prec. Therefore, in beta.nb there are a public constants: Prec, which is for the calculation precision.

Comparison with the symbolic(numerical) integration in Mathematica

In order to show how much more efficient of DBeta and NDBeta is than the corresponding programs of Mathematica, we apply it, the Mathematica symbolic integration (Integrate) and the Mathematica numerical integration (NIntegrate) to a couple more integrals with different parameters and display the running results in Tables 1 and 2.

Table 1. Comparison of Several Algorithms

x,y	p,q		Time	p,q		Time	p,q		Time
		I	113.1787		I	137.9672		I	159.0118
$2,2$	$4,4$	B	0.046800	$6,4$	B	0.078001	$6,6$	B	0.156001
		BD	0.873606		DB	2.199614		DB	6.162039
		I	57.08076		I	87.87536		I	123.6307
$2,-\frac{5}{2}$	$3,3$	B	0.483603	$4,5$	B	1.341609	$5,5$	B	5.132432
		BD	6.162039		BD	80.32491		BD	187.9500
		I	106.6266		I	126.7664		I	127.8272
$-1,\frac{5}{2}$	$3,4$	B	1.887612	$4,5$	B	5.226033	$5,5$	B	17.61251
		BD	$*$0.031200		BD	$*$0.062400		BD	$*$0.046800
		I	59.29598		I	69.42044		I	161.2114
$-1,-1$	$2,2$	B	0.062400	$3,2$	B	0.312002	$4,4$	B	3.915625
		BD	$*$0.046800		BD	$*$0.031200		BD	$*$0.078001

In Table 1, letters I, B and BD represent Integrate[$t^{x-1}(1-t)^{y-1}Log[t]^pLog[1-t]^q$, $[t, 0, 1\}$],DBeta-

$[x, y, p, q, 0]$ and $D[D[Beta[xx, yy], \{xx, p\}]/.xx \to x, \{yy, q\}]/.yy \to y]$, respectively. The data shows that B is much more efficient in time than I and BD, and the rate ranges from 7 to 2400. An asterisk in front the time consumed indicates that the algorithm is valid.

Table 2 Comparison of NDBeta$[x, y, p, q, 0]$ and
NIntegrate[$t^{\text{x-1}}(1-t)^{y-1}$Log[t]^{p}Log[1-t]^q, $\{t, 0, 1\}, WorkingPrecision- > Prec$]

x, y	p, q		T$_{32}, rr$	T$_{64}, rr$	T$_{128}, rr$	T$_{256}, rr$
$2, 2$	$6, 6$	NI	$0.046800, 10^{-32}$	$0.156001, 10^{-65}$	$0.499203, 10^{-129}$	$1.856412, 10^{-257}$
		NB	$0.015600, 10^{-47}$	$0.031200, 10^{-100}$	$0.015600, 10^{-207}$	$0.015600, 10^{-420}$
$-\frac{5}{2}, -\frac{7}{3}$	$6, 6$	NI	$0.062400, 10^{-32}$	$0.280802, 10^{-65}$	$0.795605, 10^{-129}$	$2.761218, 10^{-257}$
		NB	$0.031200, 10^{-34}$	$0.031200, 10^{-87}$	$0.031200, 10^{-194}$	$0.031200, 10^{-407}$
$-4, -\frac{5}{2}$	$6, 6$	NI	$0.062400, 10^{-32}$	$0.202801, 10^{-65}$	$0.717605, 10^{-117}$	$2.511616, 10^{-257}$
		NB	$0.062400, 10^{-39}$	$0.062400, 10^{-92}$	$0.062400, 10^{-199}$	$0.062400, 10^{-412}$
$-4, -5$	$6, 6$	NI	$0.046800, 10^{-32}$	$0.187201, 10^{-64}$	$0.577204, 10^{-129}$	$2.402415, 10^{-257}$
		NB	$0.093601, 10^{-38}$	$0.093601, 10^{-91}$	$0.093601, 10^{-198}$	$0.093601, 10^{-411}$

In this table, NI and NB represent

$$NIntegrate[t^{x-1}(1-t)^{y-1}Log[t]^p Log[1-t]^q, \{t, 0, 1\}, WorkingPrecision \to Prec]$$

and NDBeta$[x, y, p, q, 0]$, respectively. In order to reduce the accumulated calculation accuracy take [5Prec/3]. The subindex of T indicates the computing accuracy requirement and rr is the relative error. From Table 2, we see that the running time of NDBeta$[x, y, p, q, 0]$ is not substantially affected by the specified accuracy and much smaller than NIntegrate$[t^{x-1}(1-t)^{y-1}Log[t]^p Log[1-t]^q, [t, 0, 1\}$,WorkingPrecision-¿Prec] and its efficiency is more significant especially in high-precision. It is noteworthy that the relative error of the BetaD$[x, y, p, q, 0]$ is always less than the specified one.

Partial derivatives of the Beta Function used in some generalized integral calculation

Many generalized integrals can be expressed in terms of Beta function and its partial derivatives. Thus, a faster and high accuracy algorithm for calculating the values of the Beta function and its partial derivatives can speed up and increase the accuracy of the calculation of generalized integrals. There are many identities in this respect. For Example [14],

$$\begin{array}{c}
\int_0^1 t^{x-1}(1-t)^{y-1}dt = B(x, y) \qquad [Rex, Rey > 0] \\
\int_0^\infty \frac{t^{x-1}}{(1+t)^{x+y}}dt = B(x, y) \qquad [Rex > 0, Rey > 0] \\
\int_{-1}^1 \frac{(1+t)^{2x-1}(1-t)^{2y-1}}{(1+t^2)^{x+y}}dt = 2^{x+y-2}B(x, y) \quad [Rex > 0, Rey > 0] \\
\int_0^1 \frac{t^{x-1}+t^{y-1}}{(1+t)^{x+y}}dt = \int_1^\infty \frac{t^{x-1}+t^{y-1}}{(1+t)^{x+y}}dt = B(x, y) \quad [Rex, Rey > 0] \\
\int_0^1 \frac{(1+t)^{x-1}(1-t)^{y-1}+(1+t)^{y-1}(1-t)^{x-1}}{2^{x+y-1}}dt = B(x, y) \quad [Rex > 0, Rey > 0] \\
\int_0^{\frac{\pi}{2}} \sin^{2x-1} t \cos^{2y-1} t dt = \frac{1}{2}B(x, y) \qquad [Rex, Rey > 0] \\
\int_{-\infty}^\infty \frac{e^{2iyt}}{(2\cosh t)^{2x}}dt = \frac{B(x+iy, x-iy)}{2} \quad [Rex > 0, y \text{ is a real}] \\
\int_{-\infty}^\infty \frac{e^{-2yt}}{(2\cosh t)^{2x}}dt = \frac{B(x-y, x+y)}{2} \quad [Rex > 0, Rex > |Rey|] \\
\int_0^\infty \frac{\cosh 2yt}{(2\cosh zt)^{2x}}dt = \frac{B(x+\frac{y}{z}, x-\frac{y}{z})}{4z} \quad [Rex > |Rey|] \\
\int_0^1 (1-t^z)^{x-1}t^{y-1}dt = \frac{1}{z}B(x, \frac{y}{z}) \quad \left[Re\frac{y}{z}, Rex > 0\right] \\
\int_0^\infty e^{-xt}\sinh^y zt dt = \frac{1}{2^{y+1}z}B(\frac{x}{2z} - \frac{y}{2}, y+1) \quad [Rez > -1, Rezy > 0] \\
\int_0^\infty \frac{e^{-xt}}{\cosh^{2y+1} xt}dt = \frac{2^{2y-2}}{x}B(y, y) - \frac{1}{2xy} \quad [x, y > 0] \\
\int_0^\infty e^{-xt}(\cosh zt - 1)^y dt = \frac{B(\frac{x}{z}-y, 2y+1)}{2^y z} \quad \left[Rey > -\frac{1}{2}, Rex > Rezy, Rez > 0\right]
\end{array} \tag{4.1}$$

and

$$\int_0^1 \frac{\left(\frac{t}{t+z}\right)^x\left(\frac{1-t}{t+z}\right)^y}{t(1-t)}dt = \begin{cases} \frac{B(x,y)}{z^y(1+z)^x}, Rex, Rey > 0, Re(x+y) < 1, -1 < z < 0 \\ \left(\frac{1}{z}\right)^y\left(\frac{1}{1+z}\right)^x B(x,y),\ Rex, Rey > 0, z \notin [-1,0] \end{cases}$$

$$\int_0^{\frac{\pi}{2}} \frac{\left(\frac{\cos^2 t}{\cos^2 t+z}\right)^x\left(\frac{\sin^2 t}{\cos^2 t+z}\right)^y}{\sin t\cos t}dt = \begin{cases} \frac{B(x,y)}{2z^y(1+z)^x}, Rex, Rey > 0, Re(x+y) < 1, -1 < z < 0 \\ \frac{1}{2}\left(\frac{1}{z}\right)^y\left(\frac{1}{1+z}\right)^x B(x,y),\ Rex, Rey > 0, z \notin [-1,0] \end{cases} \tag{4.2}$$

By the means of (4.1) and (4.2), we can express many generalized integrals in terms of partial derivatives of the Beta function. We give several examples here.

1) When p and q are non-negative integers, $p + Rey > 0$ and $q + Rex > 0$, we have

$$\int_0^1 \left(\begin{array}{c} (-1)^q t^{x-1}(1+t)^{-x-y}\ln^p \frac{t}{1+t}\ln^q(1+t) \\ +(-1)^p t^{y-1}(1+t)^{-x-y}\ln^q \frac{t}{1+t}\ln^p(1+t) \end{array} \right) dt = B_{p,q}(x,y), \tag{4.3}$$

and

$$\int_0^1 \frac{t^{x-1}}{(1+t)^{2x}}\ln^p \frac{t}{1+t}\ln^p(1+t)dt = \frac{(-1)^p}{2}B_{p,p}(x,x). \tag{4.4}$$

2) When p and q are non-negative integers,we have

$$\begin{array}{c} \int_0^1 \frac{1}{t(1-t)}\left(\frac{t}{t+z}\right)^x\left(\frac{1-t}{t+z}\right)^y \ln^p \frac{t}{t+z}\ln^q \frac{1-t}{t+z}dt \\ = \left(\frac{1}{z}\right)^y\left(\frac{1}{1+z}\right)^x \sum\limits_{j=0}^{p} C_p^j \ln^{p-j}(\frac{1}{1+z}) \sum\limits_{k=0}^{q} C_q^k \ln^{q-k}\frac{1}{z}B_{j,k}(x,y). \end{array} \tag{4.5}$$

for $Rex, Rey > 0, z \notin [-1,0]$.

3) When p and q are non-negative integers, and $Rex > |y|$, y is real, we have

$$\begin{array}{c} \int_0^\infty \frac{t^{2q}\cosh 2yt\ln^p\cosh t}{(2\cosh zt)^{2x}}\ln^p(2\cosh zt)\,dt \\ = \frac{1}{2^{p+2q+2}z^{2q+1}} \sum\limits_{j=0}^{p} C_p^j \sum\limits_{k=0}^{2q}(-1)^k C_{2q}^k B_{j+2q-k,p-j+k}(x+\frac{y}{z}, x-\frac{y}{z}) \end{array} \tag{4.6}$$

and

$$\begin{array}{c} \int_0^\infty \frac{t^{2q+1}\sinh 2yt\ln^p(2\cosh zt)}{(2\cosh zt)^{2x}}dt \\ = \frac{1}{2^{p+2q+3}z^{2q+2}} \sum\limits_{j=0}^{p} C_p^j \sum\limits_{k=0}^{2q+1}(-1)^k C_{2q+1}^k B_{j+2q+1-k,p-j+k}(x+\frac{y}{z}, x-\frac{y}{z}). \end{array} \tag{4.7}$$

4) When p and q are non-negative integers, $\alpha > 0$ and $Rex > |Imy|$, we have

$$\begin{array}{c} \int_{-\infty}^\infty \frac{t^q e^{-2yt}\ln^p(2\cosh t)}{(2\cosh t)^{2x}}dt = \frac{(-1)^{p+q}}{2^{p+q+1}} \sum\limits_{j=0}^{p} C_p^j \sum\limits_{k=0}^{q}(-1)^k C_q^k B_{k+j,p+q-k-j}(x-y, x+y) \\ p, q \text{ are integer},\ p, q \geq 0,\ Rex > |Rey|\,. \end{array} \tag{4.8}$$

5) When p and q are non-negative integers, $Rez > 0, p + Re\frac{y}{z}$ and $q + Rex > 0$, we have

$$\int_0^1 (1-t^z)^{x-1}t^{y-1}\ln^p(1-t^z)\ln^q t dt = \frac{1}{z^{q+1}}B_{p,q}(x, \frac{y}{z}). \tag{4.9}$$

6) Letting $t = \sin^2 u$ or $\cos^2 u$ in (4.5), we have

$$\begin{array}{l} \int_0^{\frac{\pi}{2}} \frac{\left(\frac{\cos^2 t}{\cos^2 t+z}\right)^x \left(\frac{\sin^2 t}{\cos^2 t+z}\right)^y}{\sin t \cos t} \ln^p \frac{\cos^2 t}{\cos^2 t+z} \ln^q \frac{\sin^2 t}{\cos^2 t+z} dt \\ = \frac{1}{2} \left(\frac{1}{z}\right)^y \left(\frac{1}{1+z}\right)^x \sum\limits_{j=0}^{p} C_p^j \ln^{p-j}\left(\frac{1}{1+z}\right) \sum\limits_{k=0}^{q} C_q^k \ln^{q-k} \frac{1}{z} B_{j,k}(x,y), \end{array} \tag{4.10}$$

for integer, $p, q \geq 0, Rex, Rey > 0, z \notin [-1, 0]$.

7) When p and q are non-negative integer, $q + Rex > 0$ and $Rey > 0$, we have

$$\int_0^{\infty} t^{x-1}(1+t)^{-x-y} \ln^p \frac{t}{1+t} \ln^q(1+t) dt = (-1)^q B_{p,q}(x,y). \tag{4.11}$$

For $x = \pm n, \frac{1}{2} \pm n, y = \pm m, \frac{1}{2} \pm m, n, m = 0, 1, 2, \cdots$, $B_{p,q}(x,y)$ and $B_{p,q}(x+y, x-y)$ always exists closed form, so the generalized integral (4.3)-(4.11), which also exist closed form. However, the use of symbolic integration (Integrate) in Mathematica, closed forms of these integrals are difficult to obtain. For example, in Mathematica we have the following results for the generalized integral (4.3).

$$\begin{array}{c} x = 2; y = 1/2; p = 1; q = 1; \\ Timing[s1 = Integrate[\frac{t\hat{}(x-1)Log[\frac{t}{1+t}]\hat{}p*Log[\frac{1}{1+t}]\hat{}q}{(1+t)\hat{}(x+y)} + \frac{p*t\hat{}(y-1)Log[\frac{t}{1+t}]\hat{}q*Log[\frac{1}{1+t}]\hat{}p}{(1+t)\hat{}(x+y)}, \{t, 0, 1\}]] \\ Timing[s2 = Simplify[DBeta[x, y, p, q, 0]]] \\ N[s1 - s2, Prec] \\ \{96.736220, \frac{1}{27}(320 - 116\sqrt{2}] - 30\pi^2 + 72ArcSin[\sqrt{2}]^2 - 72ArcSinh[1] - 27Log[2]^2 - \\ 4i\sqrt{2}HypergeometricPFQ[\{-\frac{3}{2}, -\frac{3}{2}, -\frac{3}{2}, \frac{1}{2}\}, \{-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}\}, 2] - \\ 12i\pi(-13 + Log[8]) - 64Log[8] - 72IArcSin[\sqrt{2}](1 + Log[4] + 2Log[4 - 2\sqrt{2}]) - \\ 96Log[-1 + \sqrt{2}] + 108Log[2]Log[1 + \sqrt{2}] + 144Log[1 + \sqrt{2}]^2 - \\ 144Log[1 + \sqrt{2}]Log[2 + \sqrt{2}] - 216PolyLog[2, -\frac{1}{\sqrt{2}}] + \\ 288Log[1 + \sqrt{2}] + 216PolyLog[2, 1 - \sqrt{2}] + 72PolyLog[2, -3 + 2\sqrt{2}])\} \\ \{0., -\frac{2}{27}(9\pi^2 + 16(-10 + Log[64]))\} \\ 0. * 10^{-111} + 0. * 10^{-112}i \end{array}$$

When $p, q > 1$, the use of the symbolic integration even above complex can not be obtained.

However, the right-hand sides($(4.*)_R$) of the equations (4.3)-(4.11) give a high accuracy and fast algorithm to calculate the integrals of the left-hand side($(4.*)_L$). In order to verify the correctness of the formulas (4.3)-(4.11) and further show the high accuracy and time efficiency of our algorithm, the following numerical results are given in Mathematica.

Table 3 Comparison of numerical integration for (4.3)-(4.5)

	p, q, x, y	T_{32}, rr	T_{64}, rr	T_{128}, rr	Integral value
$(4.3)_L$	$4, 4, -2, i-3$	$0.1716, 10^{-32}$	$0.5616, 10^{-64}$	$1.5600, 10^{-128}$	$-1.554939\cdots$
$(4.3)_R$		$0.0156, 10^{-46}$	$0.0156, 10^{-99}$	$0.0156, 10^{-206}$	$+1.779627\cdots i$
$(4.4)_L$	$4, 4, -\frac{5}{2}+i, -3+i$	$0.1404, 10^{-32}$	$0.3900, 10^{-64}$	$1.2636, 10^{-128}$	$-2.114631\cdots$
$(4.4)_R$		$0.0156, 10^{-50}$	$0.0156, 10^{-102}$	$0.0156, 10^{-210}$	$-2.863698\cdots i$
$(4.5)_L$, z=$\frac{1}{2}$	$3, 2, \frac{1}{3}, \frac{1}{5}$	$0.1248, 10^{-14}$	$0.2028, 10^{-23}$	$0.4836, 10^{-27}$	$-279.808345\cdots$
$(4.5)_R$		$0., 10^{-51}$	$0.0156, 10^{-104}$	$0.0156, 10^{-211}$	
$(4.5)_L$, z= -2	$3, 3, \frac{1}{3}, \frac{1}{4}$	$0.1872, 10^{-13}$	$0.4056, 10^{-18}$	$0.7800, 10^{-27}$	$-54904.915\cdots$
$(4.5)_R$		$0., 10^{-51}$	$0.0156, 10^{-104}$	$0.0156, 10^{-211}$	$+28376.28\cdots i$

In particular, for the left integral($(4.5)_L$) of the formula (4.5), the error of numerical integration always exists regardless of the calculation precision. When the numerical computation in Mathematica is used for calculating values of integrals, the error does not much improve no matter how the accuracy requirement is increased.

It is noteworthy that we found symbol integration and numerical integration of inconsistent results in Mathematica.

$$z = -1/2; x = 1/4; y = 1/5; Prec = 32;$$
$$Timing[s1 = NIntegrate[\left(\frac{t}{z+t}\right) \hat{} x \left(\frac{1-t}{z+t}\right) \hat{} y \frac{1}{t(1-t)}, \{t, 0, 1\}, WorkingPrecision- > Prec]]$$
$$Timing[s2 = N[Integrate[\left(\frac{t}{z+t}\right) \hat{} x \left(\frac{1-t}{z+t}\right) \hat{} y \frac{1}{t(1-t)}, \{t, 0, 1\}], Prec]]$$
$$Timing[s0 = \frac{1}{z \hat{} y} \left(\frac{1}{1+z}\right) \hat{} y * NBetaAll[x, y]]$$
$$\{s1 - s0, s2 - s0, s3 - s0\}$$
$$\{0.124801, 9.3442494430451904923601953855976 + 6.7887335956884325541842812213085I\}$$
$$\{1.419609, 9.3462960217122886259805243287725 - 6.7904815391012934935031911000193I\}$$
$$\{0., 9.34629602171228862598052432877250731935215923310015 09-$$
$$6.7904815391012934935031911000192799983188576389449694I\}$$
$$\{-0.0020465786670981336203289431749 + 13.579215134789726047687472321327 8i,$$
$$0. \times 10^{-32} + 0. \times 10^{-32}\}$$

Conclusions

By giving additional definition of the Beta function, the domain of the Beta function has been extended to the entire complex plane. In the entire complex plane, we have established recursive formulas on the partial derivatives of the Beta function. Applying these recursive formulas, we give the conditions of the closed form of the generalized integral, which are expressed in terms of partial derivatives of the Beta function. And for the numerical calculation, calculation speed and accuracy have some improvements.

References

[1] http://en.wikipedia.org/wiki/Gabriele_Veneziano#cite_note-5.

[2] Riddhi D., Beta Function and its Applications, http://sces.phys.utk.edu/˜moreo/mm08/Riddi.pdf.

[3] V.A. Novikov, M.A. Shifman,A.I. Vanshtein and V.i. Zakhrov, The Beta Function in Supersymmetric Gauge Theories. Instantons Versus Traditional Approach,Physics Letters, Volume 166B,number 3 16 January (1986) ,329–333.

[4] Akio Morita, Haruyo Koiso, Yukiyoshi Ohnishi, and Katsunobu Oide, Measurement and correction of on-and off-momentum beta functions at KEKB, Phys. Rev. ST Accel. Beams 10, 072801 Published 27 July 2007

[5] G.Benfatto G.Gallavottie, A.Procacci B.Scoppola, Beta Function and Schwinger Functions for a Many Fermions System in One Dimension. Anomaly of the Fermi Surface, Commun. Math.Phys. 160,93–171 (1994)

[6] Y. Jack Ng, H. van Dam, Neutrix calculus and finite quantum field theory, J. Phys. A 38 (2005) 317–323.

[7] Y. Jack Ng, H. van Dam, An application of neutrix calculus to quantum field theory, Int.J.Mod.Phys. A21 (2006) 297–312

[8] Youn-Sha Chan, Albert C. Fannjiang, Glaucio H. Paulino, and Bao-Feng Feng, Finite Part Integrals and Hypersingular Kernels,DCDIS A Supplement, Advances in Dynamical Systems, Vol. 14(S2)264–269

[9] N. Shang and H. Qin, The Closed Form on a Kind of Log-cosine and Log-sine Integral, Journal of Mathematics in Practice and Theory,Volume 42, No.23, December 2012 234–246.

[10] N. Shang, A. Li, Z. Sun, and H. Qin,A Note on the Beta Function And Some Properties of Its Partial Derivatives, IAENG International Journal of Applied Mathematics, 44:4, IJAM_44_4_06.

[11] Z. Sun, H. Qin,A. Li, Extension of the partial derivatives of the incomplete beta function for complex values, Applied Mathematics and Computation 275(2016)63–71.

[12] A Li, Z Sun, H Qin,The Algorithm and Application of the Beta Function and Its Partial Derivatives, Engineering Letters, 23:3, EL_23_3_04

[13] Huizeng Qin, Nina shang, Aijuan Li, Some identities on the Hurwitz zeta function and the extended Euler sums. Integral Transforms and Special Functions. iFirst, 2012,1–21.

[14] I.S.Gradshteyn I. M. Ryzhik,Table of Integrals Series and Products, Seventh Ediyion,Academic Press is an imprint of Elsevier, 908–909