

# 小变量情况下第一类整数阶 Bessel 函数的计算

冯佳计, 贾晓伟, 沈建琪

(上海理工大学理学院, 上海 200093)

**摘 要:** 在计算第一类整数阶 Bessel 函数时, 后向递推算法稳定高效. 然而, 起始点的选取必须有足够高的阶数, 并且需要进行归一化处理. 本文对 Taylor 级数展开算法进行研究, 并对其级数展开规律、计算精度, 以及求和项与参数间的关系进行了讨论. 此外, 本文利用指数形式, 极大扩展了该算法的可计算范围. 与 du Toit 算法、MATLAB 和 Mathematica 应用软件的计算结果比较显示, 本文的算法具有较高的准确性和稳定性.

**关键词:** Bessel 函数; Taylor 级数展开; 指数扩展

**中图分类号:** TP301.6

**文献标识码:** A

**文章编号:** 0372-2112 (2016)11-2720-06

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2016.11.022

## Computation of the Integer Order Bessel Functions of First Kind with Small Arguments

FENG Jia-ji, JIA Xiao-wei, SHEN Jian-qi

(College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China)

**Abstract:** Algorithm based on the backward recurrence for computing the integer order Bessel functions of the first kind is stable and efficient. However, the orders of the starting points should be high enough and the normalization is required. In this paper, we introduce an algorithm based on the Taylor series expansion (TSE), in which all the quantities involved are expressed in the exponential format so as to expand the numeric range of calculation. Comparison against du Toit's algorithm as well as MATLAB and Mathematica shows that our algorithm is stable and reliable.

**Key words:** Bessel function; Taylor series expansion; exponential scaling

## 1 引言

第一类整数阶 Bessel 函数  $J_n(z)$  在物理学、力学、电磁学、环境科学等领域都有着重要的用途<sup>[1,2]</sup>. 由于应用广泛, 大量文献致力于讨论它的性质并给出了多种不同的算法<sup>[3-16]</sup>. 其中后向递推算法选取两个连续的高阶函数作为起始点赋任意值并后向递推计算, 最后做归一化处理<sup>[9-11]</sup>. 这种算法稳定且高效. 然而, 为了确保计算的顺利进行, 后向递推算法起始点的阶数必须足够高, 需要更多的计算机资源和更大的计算量. 在采用后向递推算法计算 Bessel 函数时, 可考虑采用 Taylor 级数展开法精确计算出起始阶数的函数值以提高计算效率. 原则上, Bessel 函数的 Taylor 级数展开对所有变量均收敛. 然而, 其收敛速度却依赖于阶数  $n$  和变量  $z$  的大小. 在阶数  $n$  很小、参数  $z$  很大的情况下, 需要计算

的级数展开项增多, 且在计算过程中有可能会丢失有效数字, 导致计算效率降低、计算结果出现偏差.

本文针对 Bessel 函数  $J_n(z)$  的 Taylor 级数展开计算进行讨论. 考虑到 Bessel 函数  $J_n(z)$  满足关系式  $J_{-n}(z) = (-1)^n J_n(z)$ 、 $J(-z) = (-1)^n J_n(z)$  和  $J_n(z^*) = J_n^*(z)$ <sup>[4]</sup>, 我们只考虑  $n \geq 0$ 、 $\text{Re}(z) \geq 0$  和  $\text{Im}(z) \geq 0$  的情况.

## 2 递推算法

整数阶 Bessel 函数满足前向递推关系

$$B_{n+1}(z) = \frac{2n}{z} B_n(z) - B_{n-1}(z) \quad (1)$$

或者后向递推关系

$$B_{n-1}(z) = \frac{2n}{z} B_n(z) - B_{n+1}(z) \quad (2)$$

对于给定参数  $z$ , 前向递推从两个连续低阶级数开

始计算,而后向递推从两个连续高阶级数开始计算.像大多数递推过程一样,上述公式易受传播误差的影响<sup>[7]</sup>.

当  $z$  为实数或者虚部很小 ( $|\operatorname{Im} z| \leq 1.0$ ) 的复数时, Bessel 函数的绝对值  $|J_n(z)|$  在  $n < 0.9|z|$  范围变化不大,而在  $n \geq 0.9|z|$  范围随阶数的增大快速衰减.对于  $n < 0.9|z|$  的范围,前向递推或者后向递推都能用来计算  $J_n(z)$ ;但当  $n > 0.9|z|$  时,则只能采用后向递推法<sup>[5,9]</sup>.如果  $z$  的虚部较大, Bessel 函数的绝对值  $|J_n(z)|$  在  $n \geq 0$  范围内随阶数的增大迅速衰减,只能采用后向递推法计算.

用后向递推方法计算 Bessel 函数  $J_n(z)$  需从两个连续的高阶级数  $B_q(z)$  和  $B_{q+1}(z)$  开始,  $B_q(z)$  和  $B_{q+1}(z)$  的值可以任意选取, du. Toit 建议  $B_q(z) = 1$  和  $B_{q+1}(z) = 0$ <sup>[9]</sup>.然而,阶数  $q$  必须同时满足  $\operatorname{env}|J_{q+1}(z)| \leq 10^{-p/2} \operatorname{env}|J_N(z)|$  和  $\operatorname{env}|2J_q(z)| \leq 10^{-p} \operatorname{env}|J_0(z)|$  以确保收敛,其中  $p$  为所需的有效位数,  $N$  是所需计算的最高阶数,  $\operatorname{env}|x|$  表示函数的包络线.如果  $N \leq |z|$ ,  $q$  的最小值需满足

$$q_{\min} = \begin{cases} |z| + 10.26|z|^{0.341015} + 1.8, & |z| \leq 25 \\ |z| + 6.6362|z|^{0.342481} + 0.4, & |z| > 25 \end{cases} \quad (3)$$

若  $N > |z|$ ,  $q$  值的选取则相对繁琐<sup>[5,11]</sup>.完成  $B_n(z)$  的计算后,需进行归一化处理得到相应的函数值  $J_n(z)$ .针对不同的变量值,采用不同的归一化方法.当  $|z| \leq 25$  时,  $B_n(z)$  归一化处理为

$$S = \begin{cases} B_0 + 2 \sum_{m=1}^{q/2} B_{2m}, & |\operatorname{Im}(z)| \leq 1 \\ [B_0 + 2 \sum_{m=1}^{q/2} (-1)^m B_{2m}] \sec(z) & |\operatorname{Im}(z)| > 1 \\ \text{or } [2 \sum_{m=1}^{q/2} (-1)^m B_{2m+1}] \csc(z), & \end{cases} \quad (4)$$

当  $|z| > 25$  时,使用  $S = B_0(z)/J_0(z)$  或  $S = B_1(z)/J_1(z)$  归一化效率更高.使用时应避免  $J_0(z)$  和  $J_1(z)$  的零点.

从以上讨论可知,后向递推法从两个任意取值  $B_q(z)$  和  $B_{q+1}(z)$  开始计算,不仅需要计算起始阶数  $q$  以及从阶数  $q$  到阶数  $N$  的函数  $B_n(z)$ ,还需对  $B_n(z)$  做归一化处理.因此,为了使计算更加高效,在使用后向递推方法时,建议采用 Taylor 级数展开法预先计算  $B_q(z)$  和  $B_{q+1}(z)$  的精确值.

### 3 Bessel 函数的 Taylor 级数展开

整数阶 Bessel 函数  $J_n(z)$  的 Taylor 级数展开式为:

$$J_n(z) = \left(\frac{z}{2}\right)^n \sum_{k=0}^{\infty} \frac{(-1)^k (z^2/4)^k}{k!(n+k)!} \quad (5)$$

令

$$J_n(z) = \frac{1}{n!} \left(\frac{z}{2}\right)^n \sum_{k=0}^{\infty} J_n^k \quad (6)$$

$$J_n^k(z) = \frac{(-1)^k n! (z^2/4)^k}{k! (n+k)!} \quad (7)$$

则有递推关系:

$$J_n^k = J_n^{k-1} \cdot \frac{-z^2}{4k(n+k)} \quad (8)$$

其中,起始值  $J_n^0 = 1$ . 对于固定的变量  $z$  和阶数  $n$ , 求和项的绝对值  $|J_n^k(z)|$  的峰值位置为:

$$k_{\text{peak}} = 0.5(\sqrt{|z|^2 + n^2} - n) + 0.5 \quad (9)$$

数值计算结果如图 1 和图 2 所示. 求和项  $|J_n^k(z)|$  随着  $k$  的增大,有可能经历一个增大的过程,在  $k_{\text{peak}}$  处达到最大值,然后单调下降. 求和项出现上升过程的条件为  $k_{\text{peak}} \geq 1$ , 即  $(|z|/2)^2 \geq n + 1$ . 反过来说,当  $(|z|/2)^2 < n + 1$  时,求和项是单调下降的(对应图 1 中的单调下降区). 如果变量  $|z|$  和阶数  $n$  足够大,存在近似关系:

$$|z| < 2\sqrt{n+1} \quad (10)$$

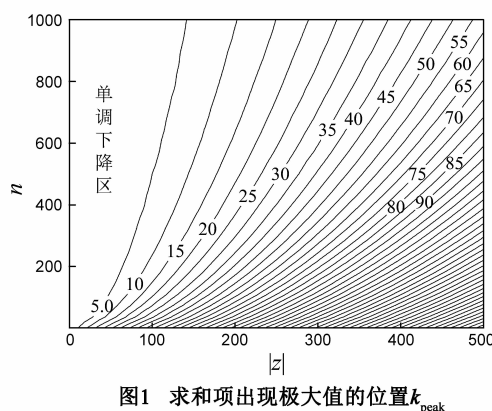
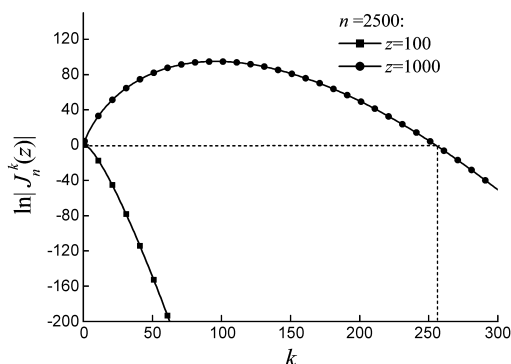


图1 求和项出现极大值的位置  $k_{\text{peak}}$

如图 2 中的  $n = 2500$  和  $|z| = 100$  所示. 此时, 整数阶 Bessel 函数  $J_n(z)$  的级数展开形式(即公式(5))是快速收敛的. 对于  $|z| > 2\sqrt{n+1}$  的情况, 求和项先递增、达到极大值、然后递减, 如图 2 中的  $n = 2500$  和  $|z| = 1000$ . Taylor 级数收敛明显变慢.

从公式(7)可以看出, 求和项的极大值  $|J_n^{k_{\text{peak}}}(z)|$  随着  $k_{\text{peak}}$  的增大而增大. 图 2 的算例中, 对于  $n = 2500$  和  $|z| = 1000$ ,  $|J_n^{k_{\text{peak}}}(z)| \approx 1.6 \times 10^{41}$ , 而 Bessel 函数  $J_{2500}(1000) \approx 7.65 \times 10^{-709}$ . 这说明 Taylor 级数展开式各求和项存在大数相减的情况. 从数值计算的角度来说, 二个大数相减存在有效数字丢失的危险, 严重时可导致最终结果失去意义. 因此, 除了考虑收敛速度外, 还需研究各求和项之间的相互关系.

对于复变量的一般情况  $z = |z|e^{i\delta_z}$ , Taylor 级数展开

图2 求和项的变化规律 $\ln|J_n^k(z)| \sim k$ 

形式写成如下形式:

$$J_n(z = |z| e^{i\delta_z}) = \frac{1}{n!} \left( \frac{|z|}{2} e^{i\delta_z} \right)^n \sum_{k=0}^{\infty} J_n^k(z) \quad (11)$$

$$J_n^k(z) = \frac{n!}{k!} \frac{(|z|^2/4)^k}{(n+k)!} e^{ik(2\delta_z - \pi)} \quad (12)$$

考虑二种特殊情况  $z_1 = |z|$  (实数) 和  $z_2 = |z| e^{i\frac{\pi}{2}}$  (虚数):

$$J_n(z_1) = \frac{1}{n!} \left( \frac{|z|}{2} \right)^n \sum_{k=0}^{\infty} \frac{(-1)^k n! (|z|^2/4)^k}{k! (n+k)!} \quad (13)$$

$$J_n(z_2) = \frac{1}{n!} \left( \frac{|z|}{2} \right)^n e^{i\frac{n\pi}{2}} \sum_{k=0}^{\infty} \frac{n! (|z|^2/4)^k}{k! (n+k)!} \quad (14)$$

公式(13)中的求和项在计算过程中出现正负交替的情况,如果  $k_{\text{peak}}$  比较大,必然导致数值比较大的数字相减并导致有效数字丢失从而计算结果变差.反之,公式(14)中的求和项一直为正,不会出现公式(13)的情况.因此,在这种情况下,即使  $k_{\text{peak}}$  比较大,也能得到合理的计算结果,但求和项比较多、收敛较慢.

图3给出了相位角接近0度和接近90度情况下求和项在复平面的分布情况.其中,  $|z|=20$ ,  $n=5$ ,由公式(9)可知  $k_{\text{peak}} = 7 \sim 8$ .可以看出,  $\delta_z \rightarrow \pi/2$  时,各求和项之间的相位角变化较小,级数求和得到的值  $\left| \sum_{k=0}^{\infty} J_n^k \right|$  属于累加性质;  $\delta_z \rightarrow 0$  时,连续二个求和项在复平面上的分布近似反向,在求和过程中出现连续项相减的情况,级数求和得到的值  $\left| \sum_{k=0}^{\infty} J_n^k \right|$  远小于  $|J_n^{k_{\text{peak}}}|$ ,导致大数相减有效数字丢失的问题.

考察 Bessel 函数的 Taylor 级数展开中有效数字丢失的问题对于得到可靠的计算结果是非常重要的.为此,需要得到可靠的  $\left| \sum_{k=0}^{\infty} J_n^k \right|$  和  $|J_n^{k_{\text{peak}}}|$  值.  $\left| \sum_{k=0}^{\infty} J_n^k \right|$  的计算采用后向递推法得到  $J_n(z)$ . 则有

$$T = \ln \left| \sum_{k=0}^{\infty} J_n^k \right| = \ln |J_n(z)| - \ln \left| \frac{1}{n!} \left( \frac{|z|}{2} \right)^n \right| \quad (15)$$

同时令  $S = \ln |J_n^{k_{\text{peak}}}|$ ,  $T > S$  表示求和项累加起主导作用,

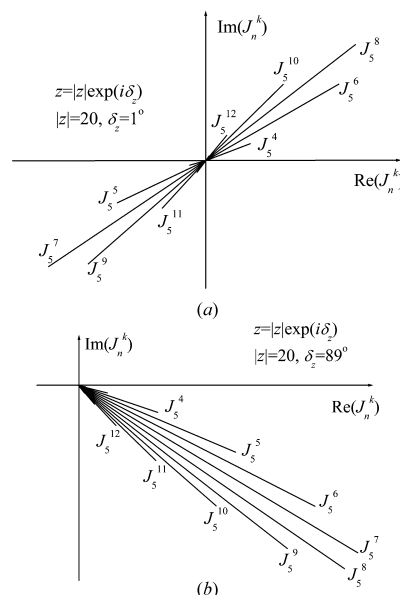


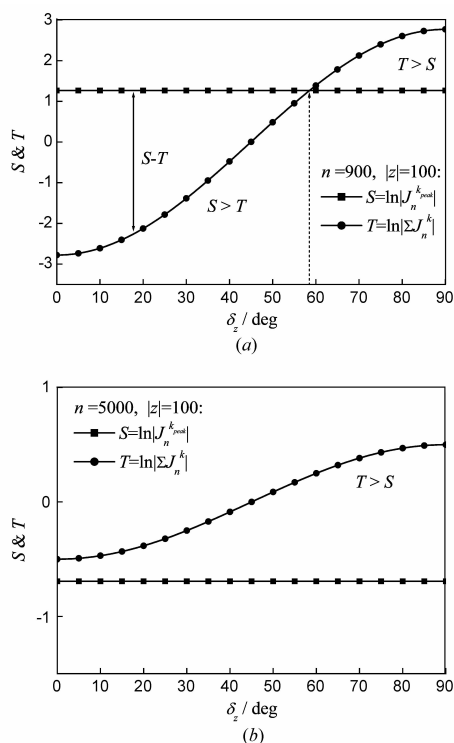
图3 不同相位角情况下求和项在复平面的分布情况

反之则表示求和项以相减为主.当  $T \ll S$  时,有效数字存在严重丢失的情况.图4给出了相位角  $\delta_z$  的变化导致级数求和过程的变化.其中  $|z|=100$ ,级数分别为  $n=900$  和  $n=5000$ .可以看出,当满足公式(10)时,  $T > S$ ,所有求和过程均是累加的.反之,随着相位角的增大,求和项  $T = \left| \sum_{k=0}^{\infty} J_n^k \right|$  逐步增大(经历了一个与  $S = \ln |J_n^{k_{\text{peak}}}|$  交叉的过程).在相位角较小的范围内,求和项以相减为主;当相位角较大时,求和项以累加为主.在曲线交叉点左侧,  $S > T$ ,表示有效数字丢失.当  $S - T$  较大时,有效数字丢失严重.数值计算表明:当  $S - T < 18$ ,所得结果的前7位得到保证;当  $S - T \geq 30$  时,级数展开方法所得结果完全不可信.

以上讨论表明:(1)当  $(|z|/2)^2 < n+1$  时,级数求和项依次递减,级数收敛速度较快, Taylor 级数展开式在所有的相位角范围内求解均有效.(2)当  $(|z|/2)^2 > n+1$  时,级数求和项经历一个递增然后下降的过程,在  $k_{\text{peak}} = 0.5(\sqrt{|z|^2 + n^2} - n)$  处为最大值.级数收敛速度较慢,而且相位角对级数求和项结果的影响变得明显.当相位角较大时,求和项以累加为主,所得结果不存在有效数字丢失,计算结果可信;反之,大数相减的情况导致有效数字丢失,计算结果会严重偏离其真实值.

在采用 Taylor 级数展开式时,并不一定要求严格满足  $(|z|/2)^2 < n+1$ . 但应该根据变量  $z = |z| e^{i\delta}$  的实际情况确定一个阶数  $n$  的可信范围.如前所述,当  $S - T < 18$  时计算结果的前7位可保证.以下的讨论以此为依据.

相位角  $\delta_z = 0$  时有效数字丢失最严重,因此在该相

图4 相位角 $\delta_z$ 的变化导致级数求和的变化规律

位角情况下满足  $S - T < 18$  的所有阶数  $n_{\text{cal}}$  对所有的相位角计算均是可信的 (见图 5). 可以看出,  $n_{\text{cal}} \approx |z|^2/40 - 10$  表示满足  $n > n_{\text{cal}}$  时, 对于任意相位角  $\delta_z$  的计算结果保证至少 7 位有效数, 这与  $n > |z|^2/4 - 1$  的要求降低了大约 10 倍.

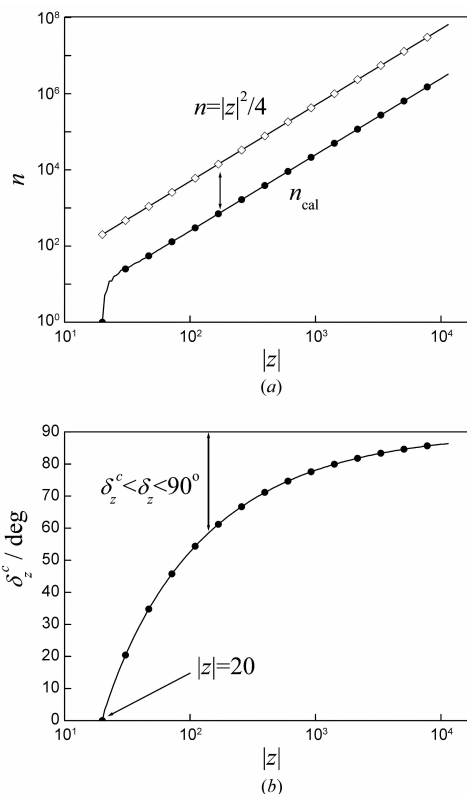
图 5 同时给出了最低阶数  $n=0$  对应的可计算相位角临界值  $\delta_z^c$  ( $\delta_z^c < \delta_z < 90^\circ$ ). 可以看出, 当  $|z| < 20$  时, Taylor 级数展开式对阶数  $n$  和相位角  $\delta_z$  没有限制. 但如果  $|z| > 20$ , 其阶数  $n$  和相位角  $\delta_z$  存在某种特定的关系. 对于某个固定的阶数  $n$  和  $|z|$ , 存在一个临界相位角  $\delta_z$ ; 反之, 对于某一个相位角  $\delta_z$  和  $|z|$ , 其阶数  $n$  必须大于一个特定的值.

图 6 给出了  $|z| = 600$  时阶数  $n$  和相位角  $\delta_z$  的关系, 以曲线为界, 上部区域表示 Taylor 级数展开式适用的范围, 下部区域中计算得到的结果其有效位数低于 7 位. 显然, 如果提高对有效位数的要求, 则图中的曲线将会向上移动, 对应的可计算范围缩小.

在  $20 < |z| < 10^4$  范围内对多个不同的复变量  $z$  进行了数值计算, 并进行曲线拟合, 得到如下经验公式:

$$n^c = \begin{cases} \frac{1}{2} \left\{ \frac{|z|^2}{|z|-1} e^{a \sqrt{1.21 - (\frac{\delta_z}{\delta_z^c})^2}} - |z| + 1 \right\}, & \delta_z < \delta_z^c \\ 1, & \delta_z \geq \delta_z^c \end{cases} \quad (16)$$

$$a = \ln(|z| - 0.5232|z|^{0.3} + 20.49) - 2.997 \quad (17)$$

图5 不同 $|z|$ 对应的最低阶数 $n_{\text{cal}}$ (对应 $\delta_z=0$ )和临界相位角 $\delta_z^c=0$ (对应 $n=0$ )

$$\delta_z^c = \arctan(0.5923 \ln |z| - 1.886) \times 62.18 + 14.49 \quad (18)$$

该经验公式表明, 对于复变量  $z = |z|e^{i\delta_z}$ , 存在一个临界相位角  $\delta_z^c$ . 如果  $\delta_z^c \leq \delta_z \leq 90^\circ$ , 则对于所有  $n \geq 1$  阶数的 Bessel 函数, Taylor 级数展开式适用; 反之, 如果  $0^\circ \leq \delta_z < \delta_z^c$ , 则 Taylor 级数展开式只适用于  $n > n^c$  的阶数. 需要注意, 经验公式给出的范围内, 保证 7 位有效数.

此外, 采用 Taylor 级数展开式时, 求和项应在适当的阶数  $k_{\text{max}}$  终止. 图 2 的曲线说明,  $k_{\text{max}}$  至少应满足  $k_{\text{max}} > 2k_{\text{peak}}$ . 此外, 可通过精度控制, 在满足  $|J_n^{k_{\text{max}}}| / \left| \sum_{k=0}^{k_{\text{max}}} J_n^k \right| < \varepsilon$  时终止累加, 其中  $\varepsilon$  是一个小数, 譬如  $\varepsilon = 10^{-20}$ .

#### 4 复数的大数计算方法

在复变量情况下, Bessel 函数值及其 Taylor 级数展开式的求和项在一个很大的动态范围内变化. 为了保证计算的顺利进行, 需要对中间计算量的数值范围进行扩展. 为此, 所有变量均采用复数形式储存并参与运算.

在高阶情况下, Taylor 级数展开式的系数  $\frac{1}{n!} \left( \frac{z}{2} \right)^n$  会出现溢出. 可采用复数的表达方式避免.

$$\frac{1}{n!} \left( \frac{z}{2} \right)^n \Rightarrow \left\{ n \ln \frac{|z|}{2} - \sum_{i=1}^n \ln t_i, n \delta_z \right\} \quad (19)$$

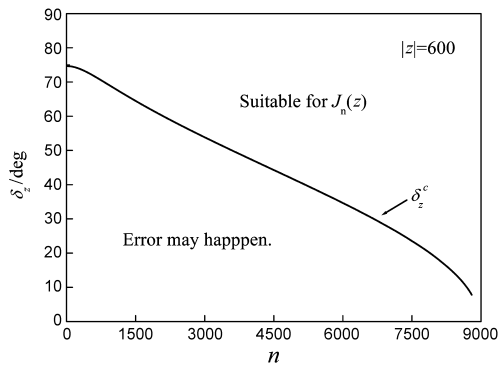


图6  $|z|=600$ 时, 各个阶数 $n$ 对应的相位角 $\delta_z$ 可计算范围  
(曲线以上部分)

## 5 数值计算结果和分析

为了验证本文提出的算法,我们将所得结果与 du

表1  $J_n(z)$  数据比较

算例	参数	du. Toit 算法	MATLAB 软件	Mathematica 软件	级数展开算法
1	$n = 1, z = 1.0$	0.440050585744936	0.440050585744934	0.440050585744934	0.440050585744934
2	$n = 100, z = i1.0$	$8.47367400814057 \times 10^{-189}$	$8.47367400813812 \times 10^{-189}$	$8.47367400813825 \times 10^{-189}$	$8.47367400813780 \times 10^{-189}$
3	$n = 5, z = 6.0$	0.362087074887192	0.362087074887173	0.362087074887172	0.362087074887172
4	$n = 5, z = i6.0$	$i7.96846774239639$	$i7.96846774239626$	$i7.96846774239626$	$i7.96846774239625$
5	$n = 20, z = 6.0$	$9.29639840900654 \times 10^{-10}$	$9.29639840900663 \times 10^{-10}$	$9.29639840900667 \times 10^{-10}$	$9.29639840900665 \times 10^{-10}$
6	$n = 20, z = i6.0$	$2.19107550354601 \times 10^{-9}$	$2.19107550354598 \times 10^{-9}$	$2.19107550354599 \times 10^{-9}$	$2.19107550354598 \times 10^{-9}$
7	$n = 280, z = 20.0$	$4.17596714311458 \times 10^{-286}$	$4.17596714308330 \times 10^{-286}$	$4.17596714308307 \times 10^{-286}$	$4.17596714308398 \times 10^{-286}$
8	$n = 45,$ $z = 39.39 + i6.949$	$-2.37447088205862 \times 10^{-2}$ $-i3.34493113795567 \times 10^{-2}$	$-2.37447088205834 \times 10^{-2}$ $-i3.34493113795577 \times 10^{-2}$	$-2.37447088205830 \times 10^{-2}$ $-i3.34493113795569 \times 10^{-2}$	$-2.37447087253296 \times 10^{-2}$ $-i3.34493114358264 \times 10^{-2}$
9	$n = 45,$ $z = 6.949 + i39.39$	$-5.17417139474068 \times 10^5$ $-i3.22406108081521 \times 10^5$	$-5.17417139474096 \times 10^5$ $-i3.22406108081539 \times 10^5$	$-5.17417139474096 \times 10^5$ $-i3.22406108081539 \times 10^5$	$-5.17417139474090 \times 10^5$ $-i3.22406108081540 \times 10^5$
10	$n = 100,$ $z = 10.0 + i10.0$	$-8.38447157414137 \times 10^{-74}$ $+i4.52662428160093 \times 10^{-74}$	$-8.38447157413478 \times 10^{-74}$ $+i4.52662428159888 \times 10^{-74}$	$-8.38447157413476 \times 10^{-74}$ $+i4.52662428159886 \times 10^{-74}$	$-8.38447157413476 \times 10^{-74}$ $+i4.52662428159874 \times 10^{-74}$
11	$n = 180,$ $z = 10.0 + i1.0$	$4.10877851496173 \times 10^{-204}$ $-i5.58875798501633 \times 10^{-204}$	$4.10877851497847 \times 10^{-204}$ $-i5.58875798503520 \times 10^{-204}$	$4.10877851497890 \times 10^{-204}$ $-i5.58875798503516 \times 10^{-204}$	$4.10877851498001 \times 10^{-204}$ $-i5.58875798503783 \times 10^{-204}$
12	$n = 180,$ $z = 10.0 + i10.0$	$-3.86931746132398 \times 10^{-177}$ $+i1.09691545788579 \times 10^{-177}$	$-3.86931746132608 \times 10^{-177}$ $+i1.09691545788509 \times 10^{-177}$	$-3.86931746132656 \times 10^{-177}$ $+i1.09691545788523 \times 10^{-177}$	$-3.86931746132691 \times 10^{-177}$ $+i1.09691545788510 \times 10^{-177}$
13	$n = 400,$ $z = 28.28 + i28.28$	下溢出	下溢出	下溢出	$2.06257678665981 \times 10^{-349}$ $-i3.19258210839493 \times 10^{-349}$
14	$n = 400,$ $z = 39.39 + i6.949$	下溢出	下溢出	$1.43426117842300 \times 10^{-349}$ $+i5.90078876392000 \times 10^{-350}$	$1.434261178423350 \times 10^{-349}$ $+i5.90078876391940 \times 10^{-350}$
15	$n = 400,$ $z = 6.949 + i39.39$	下溢出	下溢出	$4.82692121728000 \times 10^{-349}$ $-i8.88023842994000 \times 10^{-349}$	$4.82692121728580 \times 10^{-349}$ $-i8.88023842994444 \times 10^{-349}$
16	$n = 1000, z = 18.0$	下溢出	下溢出	下溢出	$4.00609355929885 \times 10^{-1614}$
17	$n = 1000,$ $z = 18.0 + i50.0$	下溢出	下溢出	$9.82458841560000 \times 10^{-1144}$ $-i4.50214268157000 \times 10^{-1144}$	$9.82458841558480 \times 10^{-1144}$ $-i4.50214268156110 \times 10^{-1144}$
18	$n = 1999,$ $z = 0.0 + i85.0$	下溢出	下溢出	下溢出	$-i2.09886430751377 \times 10^{-2477}$

Toit 的后向递推法、MATLAB 和 Mathematica 软件的计算结果进行了比较,如表 1 所示. 本程序采用 C/C++ 语言以双精度变量编写,在个人电脑上运行(配置为 3.00GHz CPU 和 2GB RAM). 在算例 1~7 中,变量  $|z| \leq 20$ , Bessel 函数的阶数  $n$  不受限制. Taylor 级数展开计算结果与其它软件结果吻合良好. 算例 8 和 9 中  $|z| = 40, n = 45$ , 根据经验公式(16~18), 临界相位角为  $\delta_z^c = 32.55^\circ$ , 算例 8 和 9 的相位角  $\delta_z$  分别为  $10^\circ$  和  $80^\circ$ . 因此,算例 9 的计算结果具有较高的精度,而算例 8 只能保证 7 位有效数. 表 1 中 Taylor 级数展开计算结果与其它软件结果的比较说明了这种情况. 算例 10~18 中,复变量  $z$  与阶数  $n$  均满足公式(10), Taylor 级数展开式求和项单调递减,因此计算收敛较快且所得结果精度较高. 在部分算例中现有软件输出结果为零(即出现了下溢出),本文算法由于采用了指数扩展仍可得到合理的数值.

## 6 结论

本文介绍了第一类整数阶 Bessel 函数的 Taylor 级数展开算法. 为了保证高速收敛并且具有较高的计算精度, Taylor 级数展开应在  $n \geq |z|^2/4 - 1$  条件下使用; 在保证 7 位有效数计算精度的情况下, 给出了 Taylor 级数展开式适用范围的经验公式; 此外, 在算法中采用了指数形式以扩展计算数值范围. 计算结果与现有的软件进行了比较, 表明该算法具有较好的效果.

递推算法与级数展开算法结合使用可有效提高计算效率. 此外, 本文中给出的算法也可以应用于计算非整数阶 Bessel 函数以及改进的 Bessel 函数.

### 参考文献

- [1] G Gouesbet, G Gréhan. Generalized Lorenz-Mie Theories [M]. Springer-Verlag Berlin Heidelberg, 2011.
- [2] JA Lock. Improved Gaussian beam-scattering algorithm [J]. Applied Optics, 1995, 34 (3): 599 – 570.
- [3] GN Watson. A Treatise on the Theory of Bessel Functions [M]. Cambridge University Press, Cambridge, 1958.
- [4] M Abramowitz, IA Stegun. Handbook of Mathematical Functions [M]. 10th ed., National Bureau of Standards, Washington, DC, 1972.
- [5] S Zhang, J Jin. Computation of Special Functions [M]. Wiley, New York, 1996.
- [6] M Goldstein, RM Thaler. Recurrence techniques for the calculation of Bessel functions [J]. Mathematical Tables & Other Aids to Computation, 1959, 13(66): 102 – 102.
- [7] W Gautschi. Computational aspects of three-term recurrence relations [J]. Society for Industrial and Applied Mathematics, 1967, 9(1): 24 – 28.
- [8] DE Amos. Algorithm 644: A portable package for Bessel functions of a complex argument and nonnegative order [J]. Acm Transactions on Mathematical Software, 1986, 12: 265 – 273.
- [9] CF du Toit. The numerical computation of Bessel functions of the first and second kind for integer orders and complex arguments [J]. IEEE Antennas & Propagation Magazine, 1990, 38(9): 1341 – 1349.
- [10] CF du Toit. Bessel functions  $J_n(z)$  and  $Y_n(z)$  for integer order and complex argument [J]. Computer Physics Communications, 1993, 78(1–2): 181 – 189.
- [11] CF du Toit. Evaluation of some algorithms and programs for the computation of integer-order Bessel functions of the first and second kind with complex arguments [J]. IEEE Antennas & Propagation Magazine, 1993, 35(3): 19 – 25.
- [12] KG Valeev, OY Kostinskii. Calculation of Bessel functions by using continued fractions [J]. Ukrainian Mathematical Journal, 1997, 47(12): 1949 – 1950.
- [13] HA Yousif, R Melka. Bessel function of the first kind with complex argument [J]. Computer Physics Communications, 1997, 106(3): 199 – 206.
- [14] EJ Rothwell. Computation of the logarithm of Bessel functions of complex argument [J]. Communications in Numerical Methods in Engineering, 2005, 21(10): 597 – 605.
- [15] LW Cai. On the computation of spherical Bessel functions of complex arguments [J]. Computer Physics Communications, 2011, 182(3): 663 – 668.
- [16] 张善杰, 唐汉. 任意实数阶复宗量第一类和第二类 Bessel 函数的精确计算 [J]. 电子学报, 1996, 24(3): 77 – 80.

Zhang Shanjie, Tang Han. Accurate computation of Bessel functions of the first and second kinds with arbitrary real order and complex argument [J]. Acta Electronica Sinica, 1996, 24(3): 77 – 80. (in Chinese)

### 作者简介



冯佳计 男, 硕士研究生, 1990 年 5 月生于江西省抚州市东乡县. 主要从事光学测试方面的研究.  
E-mail: jiaji19900505@163.com



沈建琪(通信作者) 男, 1965 年 7 月出生, 浙江桐乡人, 博士生导师, 分别于 1987 年和 1990 年在华东师范大学获得理学学士和硕士学位, 2003 年在德国 Cottbus 工大获得工学博士学位. 主要从事光散射颗粒测试技术研究.  
E-mail: jqshenk@163.com