Bessel Functions of Fractional Order

William H. Press and Saul A. Teukolsky

Citation: Computers in Physics **5**, 244 (1991); doi: 10.1063/1.4822982
View online: https://doi.org/10.1063/1.4822982
View Table of Contents: https://aip.scitation.org/toc/cip/5/2
Published by the American Institute of Physics

NUMERICAL RECIPES

Bessel Functions Of Fractional Order

William H. Press and Saul A. Teukolsky

This is the first of a two-part column on the subject of Bessel functions of fractional order. In this part we will give routines for the ordinary Bessel functions, and (as an application) for spherical Bessel functions. In the next issue, we will conclude with modified Bessel functions and (another application) Airy functions.

Many algorithms have been proposed for numerically computing Bessel functions. For integer orders there are several good ways.[1-3] Fractional orders, however, present difficulties. Many of the proposed methods are simply inaccurate, especially in regions where the functions are nearly singular. Often the methods involve awkward schemes to find starting points for backward recurrences. By contrast, the routines given here are robust; we recommend them wholeheartedly. If you like, you can use the routines as black boxes and skip the rest of this column, which presents the theory behind them.

The basic idea is *Steed's method*,[4-6] which was originally developed for Coulomb wave functions. The method calculates J_ν, J'_ν, Y_ν, and Y'_ν simultaneously, and so involves four relations among these functions. Three of the relations come from two continued fractions, one of which is complex. The fourth is provided by the Wronskian relation

$$W \equiv J_\nu Y'_\nu - Y_\nu J'_\nu = 2/\pi x. \tag{1}$$

The first continued fraction, CF1, is a standard one that can be derived from the recurrence relation:

$$f \equiv \frac{J'_\nu}{J_\nu} = \frac{\nu}{x} - \frac{J_{\nu+1}}{J_\nu} = \frac{\nu}{x} - \frac{1}{2(\nu+1)/x-}\;\frac{1}{2(\nu+2)/x-}\cdots. \tag{2}$$

Forward evaluation of the continued fraction by one of the methods of Ref. 7 is essentially equivalent to backward recurrence of the recurrence relation. The rate of convergence of CF1 is determined by the position of the *turning point* $x_{\rm tp} = \sqrt{\nu(\nu+1)} \approx \nu$, beyond which the Bessel functions become oscillatory. If $x \lesssim x_{\rm tp}$, convergence is very rapid. If $x \gtrsim x_{\rm tp}$, then each iteration of the continued fraction effectively increases ν by 1 until $x \lesssim x_{\rm tp}$; thereafter rapid convergence sets in. Thus the number of iterations of CF1 is of order x for large x. In the routine bessjy in Box 1 we set the maximum allowed number of iterations to 10 000. For larger x, you can use the usual asymptotic expressions for Bessel functions.

One can show that the sign of J_ν is the same as the sign of the denominator of CF1 once it has converged.

The complex continued fraction CF2 is defined by

$$p + iq \equiv \frac{J'_\nu + iY'_\nu}{J_\nu + iY_\nu} = -\frac{1}{2x} + i + \frac{i}{x}\;\frac{(1/2)^2 - \nu^2}{2(x+i)+}\;\frac{(3/2)^2-\nu^2}{2(x+2i)+}\cdots. \tag{3}$$

(We defer the derivation of CF2 to the analogous case of modified Bessel functions, in our next column.) This continued fraction converges rapidly for $x \gtrsim x_{\rm tp}$, while convergence fails as $x \to 0$. We will have to adopt a special method for small x. For x not too small, we can ensure that $x \gtrsim x_{\rm tp}$ by a stable recurrence of J_ν and J'_ν downwards to a value $\nu = \mu \lesssim x$, thus yielding the ratio f_μ at this lower value of ν. This is the stable direction for the recurrence relation. The initial values for the recurrence are

$$J_\nu = \text{arbitrary}, \quad J'_\nu = f_\nu J_\nu, \tag{4}$$

with the sign of J_ν chosen from the sign of the denominator of CF1. Choosing the initial value of J_ν very small minimizes the possibility of overflow during the recurrence. The recurrence relations are

$$J_{\nu-1} = (\nu/x)J_\nu + J'_\nu,$$
$$J'_{\nu-1} = [(\nu-1)/x]J_{\nu-1} - J_\nu. \tag{5}$$

Once CF2 has been evaluated at $\nu = \mu$, then with the Wronskian (1) we have enough relations to solve for all four quantities. The formulas are simplified by introducing the quantity

$$\gamma \equiv (p - f_\mu)/q. \tag{6}$$

Then

$$J_\mu = \pm\{W/[q + \gamma(p - f_\mu)]\}^{1/2}, \tag{7}$$
$$J'_\mu = f_\mu J_\mu, \tag{8}$$
$$Y_\mu = \gamma J_\mu, \tag{9}$$
$$Y'_\mu = Y_\mu(p + q/\gamma). \tag{10}$$

The sign of J_μ in (7) is chosen to be the same as the sign of the initial J_ν in (4).

Once all four functions have been determined at the value $\nu = \mu$, we can find them at the original value of ν.

William H. Press is a professor of astronomy and physics at Harvard University. Saul A. Teukolsky is a professor of physics and astronomy at Cornell University.

Box 1

```
      SUBROUTINE bessjy(x,xnu,rj,ry,rjp,ryp)
      INTEGER MAXIT
      REAL rj,rjp,ry,ryp,x,xnu,XMIN
      DOUBLE PRECISION EPS,FPMIN,PI
      PARAMETER(EPS=1.e-10,FPMIN=1.e-30,MAXIT=10000,XMIN=2.,
     *     PI=3.141592653589793d0)
C     USES cheb
```

Returns the Bessel functions $\texttt{rj} = J_\nu$, $\texttt{ry} = Y_\nu$ and their derivatives $\texttt{rjp} = J'_\nu$, $\texttt{ryp} = Y'_\nu$, for positive x and for $\texttt{xnu} = \nu \geq 0$. The relative accuracy is within one or two significant digits of EPS, except near a zero of one of the functions, where EPS controls its absolute accuracy. FPMIN is a number close to the machine's smallest floating point number. All internal arithmetic is in double precision. To convert the entire routine to double precision, change the REAL declaration above and decrease EPS to 10^{-16}. Also convert the subroutine cheb.

```
      INTEGER i,isign,l,nl
      DOUBLE PRECISION a,b,br,bi,c,cr,ci,d,del,del1,den,di,dlr,dli,
     *     dr,e,f,fact,fact2,fact3,ff,gam,gam1,gam2,gammi,gampl,h,
     *     p,pimu,pimu2,q,r,rjl,rjl1,rjmu,rjp1,rjpl,rjtemp,ry1,
     *     rymu,rymup,rytemp,sum,sum1,temp,w,x2,xi,xi2,xmu,xmu2
      if(x.le.0..or.xnu.lt.0.) pause 'bad arguments in bessjy'
      if(x.lt.XMIN)then          nl is the number of downward recurrences of the J's and
        nl=int(xnu+.5d0)            upward recurrences of Y's. xmu lies between -1/2 and
      else                          1/2 for x < XMIN, while it is chosen so that x is greater
        nl=max(0,int(xnu-x+1.5d0))  than the turning point for x >= XMIN.
      endif
      xmu=xnu-nl
      xmu2=xmu*xmu
      xi=1.d0/x
      xi2=2.d0*xi
      w=xi2/PI                   The Wronskian.
      isign=1                    Evaluate CF1 by modified Lentz's method. isign keeps track
      h=xnu*xi                      of sign changes in the denominator.
      if(h.lt.FPMIN)h=FPMIN
      b=xi2*xnu
      d=0.d0
      c=h
      do 11 i=1,MAXIT
        b=b+xi2
        d=b-d
        if(abs(d).lt.FPMIN)d=FPMIN
        c=b-1.d0/c
        if(abs(c).lt.FPMIN)c=FPMIN
        d=1.d0/d
        del=c*d
        h=del*h
        if(d.lt.0.d0)isign=-isign
        if(abs(del-1.d0).lt.EPS)go to 1
      enddo 11
      pause 'x too large in bessjy; try asymptotic expansion'
1     continue
      rjl=isign*FPMIN            Initialize J_ν and J'_ν for downward recurrence.
      rjpl=h*rjl
      rjl1=rjl                   Store values for later rescaling.
      rjp1=rjpl
      fact=xnu*xi
      do 12 l=nl,1,-1
        rjtemp=fact*rjl+rjpl
        fact=fact-xi
        rjpl=fact*rjtemp-rjl
        rjl=rjtemp
      enddo 12
      if(rjl.eq.0.d0)rjl=EPS
      f=rjpl/rjl                 Now have unnormalized J_μ and J'_μ.
      if(x.lt.XMIN) then         Use series.
        x2=.5d0*x
        pimu=PI*xmu
        if(abs(pimu).lt.EPS)then
          fact=1.d0
        else
          fact=pimu/sin(pimu)
        endif
        d=-log(x2)
        e=xmu*d
        if(abs(e).lt.EPS)then
          fact2=1.d0
        else
          fact2=sinh(e)/e
        endif
        call cheb(xmu,gam1,gam2,gampl,gammi)   Chebyshev evaluation of Γ1 and Γ2.
        ff=2.d0/PI*fact*(gam1*cosh(e)+gam2*fact2*d)   f0.
        e=exp(e)
        p=e/(gampl*PI)                             p0.
        q=1.d0/(e*PI*gammi)                        q0.
        pimu2=0.5d0*pimu
        if(abs(pimu2).lt.EPS)then
          fact3=1.d0
        else
          fact3=sin(pimu2)/pimu2
        endif
        r=PI*pimu2*fact3*fact3
        c=1.d0
        d=-x2*x2
        sum=ff+r*q
        sum1=p
        do 13 i=1,MAXIT
          ff=(i*ff+p+q)/(i*i-xmu2)
          c=c*d/i
          p=p/(i-xmu)
          q=q/(i+xmu)
          del=c*(ff+r*q)
          sum=sum+del
          del1=c*p-i*del
          sum1=sum1+del1
          if(abs(del).lt.(1.d0+abs(sum))*EPS)go to 2
        enddo 13
        pause 'bessy series failed to converge'
2       continue
        rymu=-sum
        ry1=-sum1*xi2
        rymup=xmu*xi*rymu-ry1
        rjmu=w/(rymup-f*rymu)    Equation (13).
      else                       Evaluate CF2 by modified Lentz's method. Complex arith-
        a=.25d0-xmu2                metic done with real variables for portability in double preci-
        p=-.5d0*xi                  sion.
        q=1.d0
        br=2.d0*x
        bi=2.d0
        fact=a*xi/(p*p+q*q)
        cr=br+q*fact
        ci=bi+p*fact
        den=br*br+bi*bi
        dr=br/den
        di=-bi/den
        dlr=cr*dr-ci*di
        dli=cr*di+ci*dr
        temp=p*dlr-q*dli
        q=p*dli+q*dlr
        p=temp
        do 14 i=2,MAXIT
          a=a+2*(i-1)
          bi=bi+2.d0
          dr=a*dr+br
          di=a*di+bi
          if(abs(dr)+abs(di).lt.FPMIN)dr=FPMIN
          fact=a/(cr*cr+ci*ci)
          cr=br+cr*fact
          ci=bi-ci*fact
          if(abs(cr)+abs(ci).lt.FPMIN)cr=FPMIN
          den=dr*dr+di*di
          dr=dr/den
          di=-di/den
          dlr=cr*dr-ci*di
          dli=cr*di+ci*dr
          temp=p*dlr-q*dli
          q=p*dli+q*dlr
          p=temp
          if(abs(dlr-1.d0)+abs(dli).lt.EPS)go to 3
        enddo 14
        pause 'cf2 failed in bessjy'
3       continue
        gam=(p-f)/q              Equations (6) - (10).
        rjmu=sqrt(w/((p-f)*gam+q))
        rjmu=sign(rjmu,rjl)
        rymu=rjmu*gam
        rymup=rymu*(p+q/gam)
        ry1=xmu*xi*rymu-rymup
      endif
      fact=rjmu/rjl
      rj=rjl1*fact               Scale original J_ν and J'_ν.
      rjp=rjp1*fact
      do 15 i=1,nl               Upward recurrence of Y_ν.
        rytemp=(xmu+i)*xi2*ry1-rymu
        rymu=ry1
        ry1=rytemp
      enddo 15
      ry=rymu
      ryp=xnu*xi*rymu-ry1
      return
      END
```

For J_ν and J'_ν, simply scale the values in (4) by the ratio of (7) to the value found after applying the recurrence (5). The quantities Y_ν and Y'_ν can be found by starting with the values in (9) and (10) and using the stable upwards recurrence

$$Y_{\nu+1} = (2\nu/x)Y_\nu - Y_{\nu-1}, \tag{11}$$

together with the relation

$$Y'_\nu = (\nu/x)Y_\nu - Y_{\nu+1}. \tag{12}$$

Now turn to the case of small x, when CF2 is not suitable. Temme[8] has given a good method of evaluating Y_ν and $Y_{\nu+1}$, and hence Y'_ν from (12), by series expansions that accurately handle the singularity as $x \to 0$. The expansions only work for $|\nu| \leqslant 1/2$, and so now the recurrence (5) is used to evaluate f at a value $\nu = \mu$ in this interval. Then one calculates J_μ from

$$J_\mu = W/(Y'_\mu - Y_\mu f) \tag{13}$$

and J'_μ from (8). The values at the original value of ν are determined by scaling as before, and the Y's are recurred up as before.

Temme's series are

$$Y_\nu = -\sum_{k=0}^{\infty} c_k g_k, \quad Y_{\nu+1} = -\frac{2}{x}\sum_{k=0}^{\infty} c_k h_k. \tag{14}$$

Here,

$$c_k = (-x^2/4)^k/k! \tag{15}$$

while the coefficients g_k and h_k are defined in terms of quantities p_k, q_k, and f_k that can be found by recursion:

$$\begin{aligned} g_k &= f_k + (2/\nu)\sin^2(\nu\pi/2)q_k, \\ h_k &= -kg_k + p_k, \\ p_k &= p_{k-1}/(k-\nu), \\ q_k &= q_{k-1}/(k+\nu), \\ f_k &= (kf_{k-1} + p_{k-1} + q_{k-1})/(k^2 - \nu^2). \end{aligned} \tag{16}$$

Box 2

```
      SUBROUTINE cheb(x,gam1,gam2,gampl,gammi)
      INTEGER NUSE1,NUSE2
      DOUBLE PRECISION gam1,gam2,gammi,gampl,x
      PARAMETER(NUSE1=5,NUSE2=5)
C     USES chebev
        Evaluates Γ1 and Γ2 by Chebyshev expansion for |x| ≤ 1/2. Also returns 1/Γ(1 + x) and
        1/Γ(1 − x). If converting to double precision, set NUSE1 = 7, NUSE2 = 8.
      REAL xx,c1(7),c2(8),chebev
      SAVE c1,c2
      DATA c1/-1.142022680371172d0,6.516511267076d-3,
     *     3.08709017308d-4,-3.470626964d-6,6.943764d-9,
     *     3.6780d-11,-1.36d-13/
      DATA c2/1.843740587300906d0,-.076852840844786d0,
     *     1.271927136655d-3,-4.971736704d-6,-3.3126120d-8,
     *     2.42310d-10,-1.70d-13,-1.d-15/
      xx=8.d0*x*x-1.d0               Multiply x by 2 to make range be −1 to 1, and then ap-
      gam1=chebev(-1.,1.,c1,NUSE1,xx)    ply transformation for evaluating even Cheby-
      gam2=chebev(-1.,1.,c2,NUSE2,xx)    shev series.
      gampl=gam2-x*gam1
      gammi=gam2+x*gam1
      return
      END
```

Box 3

```
      SUBROUTINE sphbes(n,x,sj,sy,sjp,syp)
      INTEGER n
      REAL sj,sjp,sy,syp,x
C     USES bessjy
        Returns spherical Bessel functions j_n(x), y_n(x), and their derivatives j'_n(x), y'_n(x) for integer
        n.
      REAL factor,order,rj,rjp,ry,ryp,RTPIO2
      PARAMETER(RTPIO2=1.2533141)
      if(n.lt.0.or.x.le.0.)pause 'bad arguments in sphbes'
      order=n+0.5
      call bessjy(x,order,rj,ry,rjp,ryp)
      factor=RTPIO2/sqrt(x)
      sj=factor*rj
      sy=factor*ry
      sjp=factor*rjp-sj/(2.*x)
      syp=factor*ryp-sy/(2.*x)
      return
      END
```

The initial values for the recurrences are

$$\begin{aligned} p_0 &= (1/\pi)(x/2)^{-\nu}\Gamma(1+\nu), \\ q_0 &= (1/\pi)(x/2)^{\nu}\Gamma(1-\nu), \\ f_0 &= \frac{2}{\pi}\frac{\nu\pi}{\sin\nu\pi}\left[\cosh\sigma\Gamma_1(\nu) + \frac{\sinh\sigma}{\sigma}\ln\left(\frac{2}{x}\right)\Gamma_2(\nu)\right], \end{aligned} \tag{17}$$

with

$$\begin{aligned} \sigma &= \nu\ln(2/x), \\ \Gamma_1(\nu) &= \frac{1}{2\nu}\left(\frac{1}{\Gamma(1-\nu)} - \frac{1}{\Gamma(1+\nu)}\right), \\ \Gamma_2(\nu) &= \frac{1}{2}\left(\frac{1}{\Gamma(1-\nu)} + \frac{1}{\Gamma(1+\nu)}\right). \end{aligned} \tag{18}$$

The whole point of writing the formulas in this way is that the potential problems as $\nu \to 0$ can be controlled by evaluating $\nu\pi/\sin\nu\pi$, $\sinh\sigma/\sigma$, and Γ_1 carefully. In particular, Temme gives Chebyshev expansions for $\Gamma_1(\nu)$ and $\Gamma_2(\nu)$. We have rearranged his expansion for Γ_1 to be explicitly an even series in ν. Since the Chebyshev polynomials satisfy $T_{2n}(x) = T_n(2x^2-1)$, we can evaluate a series of even Chebyshev polynomials in the same way[1–3] we evaluate an ordinary series, but with the argument x replaced by $2x^2 - 1$. The routine is shown in Box 2.

The routine assumes $\nu \geqslant 0$. For negative ν you can use the reflection formulas

$$\begin{aligned} J_{-\nu} &= \cos\nu\pi\, J_\nu - \sin\nu\pi\, Y_\nu, \\ Y_{-\nu} &= \sin\nu\pi\, J_\nu + \cos\nu\pi\, Y_\nu. \end{aligned} \tag{19}$$

The routine also assumes $x > 0$. For $x < 0$ the functions are in general complex, but expressible in terms of functions with $x > 0$. For $x = 0$, Y_ν is singular.

Spherical Bessel Functions

For integer n, these are defined by

$$j_n(x) = \sqrt{\pi/2x}\, J_{n+1/2}(x),$$
$$y_n(x) = \sqrt{\pi/2x}\, Y_{n+1/2}(x). \tag{20}$$

They can be evaluated by a call to bessjy, and the derivatives can safely be found from the derivatives of equation (20). This is done in the routine sphbes in Box 3.

Note that in the continued fraction CF2 in (3) just the first term survives for $\nu = 1/2$. Thus one can make a very simple algorithm for spherical Bessel functions along the lines of bessjy by always recursing j_n down to $n = 0$, setting p and q from the first term in CF2, and then recursing y_n up. No special series is required near $x = 0$. However, bessjy is already so efficient that we have not bothered to provide an independent routine for spherical Bessels. ■

In our next column: Modified Bessel functions and Airy functions.

References

1. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing* (Cambridge U. P., New York, 1986).
2. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge U. P., New York, 1988).
3. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in Pascal: The Art of Scientific Computing* (Cambridge U. P., New York, 1989).
4. A. R. Barnett, D. H. Feng, J. W. Steed, and L. J. B. Goldfarb, Comp. Phys. Comm. **8**, 377 (1974).
5. A. R. Barnett, Comp. Phys. Comm. **21**, 297 (1981).
6. I. J. Thompson and A. R. Barnett, J. Comp. Phys. **64**, 490 (1986).
7. W. H. Press and S. A. Teukolsky, Comput. Phys. **2**, 88 (1988).
8. N. M. Temme, J. Comp. Phys. **21**, 343 (1976).