A program for calculating the incomplete gamma function

By Eric Kostlan and Dmitry Gokhman
November 1987. *Key words and phrases:* Incomplete gamma function, continued fraction. Work supported by MSRI and Mathematics Dept., University of California, Berkeley, CA.

We describe an algorithm to calculate the incomplete gamma function for complex arguments. The algorithm combines a power series with a continued fraction, in order to converge quickly for a wide range of argument values. In principle, arbitrary precision may be obtained. Such an algorithm is convenient since several well known functions are special cases of the incomplete gamma function. We include a double precision FORTRAN 77 implementation.

§0. Introduction

In this paper we present a general algorithm, along with a working FORTRAN 77 implementation, to calculate the incomplete gamma function $\Gamma(\alpha,x)$. Special cases of the incomplete gamma function include the exponential and logarithm integrals and the error function (or probability integral), as well as the classical χ^2 distributions. Since the algorithm accepts complex arguments, the sine, cosine and Fresnel integrals can be calculated. Other special cases include Dawson's integral and an integral compiled by Abramowitz (See Appendix.). Therefore, when producing code to calculate special functions, the incomplete gamma function with complex arguments is an appropriate level of generality with which to work. Unfortunately, there seem to be no standard library functions for computing the incomplete gamma function, nor are there standard library functions for computing the gamma function for complex arguments. There is a standard UNIX FORTRAN 77 [13] library function 'dgamma' — 'lgamma' in the C language — to calculate the gamma function $\Gamma(\alpha,0)$, but unlike many FORTRAN library functions, there is no complex version. UNIX MACSYMA [12] suffers from essentially the same defect as the FORTRAN and C function libraries. Numerical evaluation with complex arguments will not result in simplification, and although MACSYMA can recognize the error function, it doesn't recognize the incomplete gamma function in its full generality. ACM Algorithm 14 [1] calculates the incomplete gamma for a restricted range of the parameters α and x on the complex plane. In particular, as x cannot be zero, his algorithm does not compute the gamma function. ACM Algorithm 435 [2] calculates the incomplete gamma function for real α and positive real x. Observe that ACM Algorithm 435 must be supplied with a gamma function calculator. Both these algorithms use the continued fraction expansion for the incomplete gamma function, but they calculate the continued fraction by repeated division, instead of using a two term linear recursion relation. The SAS routines [7] can be used to calculate both the incomplete gamma function $\Gamma(\alpha,x)$ and its inverse with respect to x for real parameters. The IMSL Library [5] contains the error function and the complete gamma function, but the incomplete gamma function, in its full generality, can only be calculated for real single precision arguments. The algorithm we implement here is essentially algebraic and generally convergent. Therefore arbitrary precision can, at least in principle, be obtained. Furthermore, the continued fraction approximation is in some sense optimal if Re(x) is large.

§1. The algorithm

$$\Gamma(\alpha,x) = \frac{e^{-x}x^{\alpha}}{\mathrm{h}(\alpha,x)} ,$$

where

$$\mathrm{h}(\alpha,x) = x + \cfrac{1-\alpha}{1+\cfrac{1}{x+\cfrac{2-\alpha}{1+\cfrac{2}{x+\cfrac{3-\alpha}{1+\ \cdots}}}}} .$$

See [3] 8.358. See [14] for a convergence proof.

Efficient calculation of $h(\alpha,x)$ is possible because consecutive terms of a continued fraction can be computed by means of a two term linear recursion relation (See [6], chapter 1.1). The recursion relation of interest to us is given by the following lemma.

Lemma 1: *Let* $\{p_k\},\{q_k\}, k=0,1,2,...$ *be sequences satisfying the second order homogeneous linear difference equation*

$$y_{k+2} = \begin{cases} xy_{k+1} + \frac{k+2}{2}y_k & \textit{for even } k \\ y_{k+1} + \left(\frac{k+3}{2} - \alpha\right)y_k & \textit{for odd } k \end{cases}$$

and the initial conditions $p_0=x$, $p_1=x+1-\alpha$, $q_0=1$, $q_1=1$. *Then* p_k/q_k *is the k-th term in the continued fraction for* $h(\alpha,x)$.

The value of $\mathrm{Re}(\alpha-x)$ can be reduced by using the following lemma.

Lemma 2:

$$\Gamma(\alpha+1,x) = \alpha\Gamma(\alpha,x) + x^{\alpha}e^{-x}$$

or, equivalently,

$$x/\mathrm{h}(\alpha+1,x) = \alpha/\mathrm{h}(\alpha,x) + 1$$

whenever the expression is defined.

See [3] 8.356.2. As our code concentrates on $h(\alpha,x)$, it is the second equation of lemma 2 that is of interest to us.

If x is near the negative real axis we can move x to some new value using the following lemma.

Lemma 3:

$$\Gamma(\alpha,y) - \Gamma(\alpha,x) = \sum_{n=0}^{\infty} \frac{(-1)^n (x^{\alpha+n} - y^{\alpha+n})}{n!(\alpha+n)}$$

whenever the expression is defined.

See [3] 8.354.2. In our code $y=1$ to facilitate computation. If (α,x) is near $(-n,0)$, where n is a nonnegative integer, but if x is *not equal to zero*, then the n–th term in the sum in the above lemma can become something like zero over zero to the computer. We can extend the domain of our routine closer to $(-n,0)$ if we replace this term with $-\log(x)-(\alpha+n)\log(x)^2/2$. Essentially, we are helping the computer calculate $(1-x^{\alpha+n})/(\alpha+n)$ accurately and recognize what the limit of this expression is, as α tends to $-n$. We are left with three intrinsic problems that any algorithm for $\Gamma(\alpha,x)$ must ultimately face: First, for $x=0$ and α a nonpositive integer, $\Gamma(\alpha,x)=\infty$. Second, unless α is an integer, $\Gamma(\alpha,x)$ is a multi-valued function with a branch point at $x=0$. Third, as we approach ∞ along certain directions, computation of $\Gamma(\alpha,x)$ is limited because of the inability of the computer to calculate x^{α} accurately.

§2. Certification

We compared the code with tables of special functions that are special cases of the incomplete gamma function. See the appendix for the pertinent definitions. Calculations were performed on a VAX 11/780. The operating system was UNIX [13]. Note that the largest real number is approximately $1.7014118346046\times10^{38}$. Values of $\log\Gamma(x+iy,0)$ for $x=0(0.1)10$, $y=0(0.1)10$ are given to 12 decimal places in [11]. If $x\geq2$ our code will deflate x until $x<1$, and thus our primary concern was for $x<2$. For $x<2$, the values for $x=0(0.1)1.9$, $y=0(0.1)2(1)10$ were checked. For $x\geq2$, 150 values were checked, including $x=10$, $y=0(0.1)10$. Agreement was generally perfect, with some disagreement in the 12^{th} place. For $y>7.7$, there was occasional disagreement in the 11^{th} place. For example, the table gives $\mathrm{Re}[\log\Gamma(10+10i,0)]=8.236131750449$, whereas the code gives $\mathrm{Re}[\log\Gamma(10+10i,0)]=8.2361317504863$. In [10], values of $-\mathrm{Ei}(-x-iy)$ for $x=-3.1(0.1)0$, $y=0(0.1)3.1$ are given to 10 decimal places. 60 values were checked. Five values disagreed in the last place. The greatest discrepancy occured at $x=-0.4$, $y=0.7$: the table gives a real part of -0.6484417133 whereas the code gives -0.64844171335203. This book also gives 6 decimal place values of $-e^{-x-iy}\,\mathrm{Ei}(-x-iy)$ for $x=-20(1)20$, $y=0(1)20$. 50 values were checked; agreement was perfect. Values of $\mathrm{ci}(x)$ and $\mathrm{si}(x)$ for $x=10(0.01)100$ to 10 decimal places may be found in [9]. 50 values were checked with the code, including 10(10)100. No discrepancies were found. The probability integral is compiled in [4] table 7.1. The table gives 10 decimal place values of $\Phi(x)$ for $x=0(0.01)2$. All values were checked; no discrepancies were found. The Fresnel integrals are compiled in [4] table 7.7. The table gives 7 decimal place values of $C(\sqrt{2\pi}x/2)$ and $S(\sqrt{2\pi}x/2)$ for $x=0(0.02)5$. All 251 values were checked. The code reproduced all the values of C, but for S it disagreed with the table at 5 values of x: 1.24, 1.76, 1.96, 3.48, and 3.70. The greatest discrepancy occurs at 1.96, where the table gives a value of 0.3483830 but the code gives a value of 0.34838294063437. Dawson's integral is compiled in [4] table 7.5. The table gives 10 decimal place values of $\mathrm{daw}(x)$ for $x=0(0.02)2$. All values were checked; no discrepancies were found. The chi-square distribution is compiled in [4] table 26.7. The table gives 5 decimal place values of $F_{\chi_k^2}(x)$ for $k = 1(1)30$, $x = 0.001(0.001)0.01(0.01)0.1(0.1)2(0.2)10(0.5)20(1)40(2)76$. Of these values 2682 are not equal to 0 or 1 to 5 decimal places. All values were checked. For 68 of the values, the last place of the table differed from the values produced by the code by one digit. The remaining values where reproduced exactly. The integral studied by Abramowitz is compiled in [4] table 7.6. The table gives 7 decimal place values of abr(x) for x=0(0.02)1.7(0.04)2.3. Of these 101 values, all but 5 were reproduced exactly. For example, whereas the table gives abr(0.14)=0.1566711, the code gives 0.15667104724645. Similar discrepancies occur when x equals 0.52, 0.74, 0.98, and 1.28.

§3. The code

Here we give the FORTRAN 77 code. The continued fraction is calculated in the routine ‘cdhs’. The rest of the code is required to shift α and x into a range in which ‘cdhs’ works efficiently. The shift in α is performed by ‘cdh’. The shift from $\Gamma(\alpha,x)$ to $\Gamma(\alpha,1)$ for x near the negative real axis is performed by the main routine ‘cdig’. Certain features of the code remain arbitrary. For example, the distance from the negative real axis is measured using the L^1-norm. Thus the following code may be far from the optimal implementation of the algorithm described in this paper. In particular, varying the variables *xlim* and *ibuf* of ‘cdig’, *buf* of ‘cdh’, and *tol1, tol2* of ‘cdhs’ may improve performance of the code. The variables *error* and *ilim* of ‘cdhs’ control precision. We have set *ilim* to a high value to ensure high precision at the possible expense of time. The variables *xlim* of ‘cdig’ and *buf* of ‘cdh’ control the shifting of x and α, respectively. The routine ‘dnrm’ calculates the L^1-norm on the complex plane.

```
      double complex function cdig(alpha,x)
c --- Written By Eric Kostlan & Dmitry Gokhman
c --- March  1986
      double complex alpha,x,cdh
      double complex re,one,p,q
      double precision xlim,zero,dnrm
      data re,one/0.36787944117144232,1./
      data xlim,zero/1.,0./ibuf/34/
c --- If x is near the negative real axis, then shift to x=1.
      if(dnrm(x).lt.xlim.or.dreal(x).lt.zero.and.
     *   dabs(dimag(x)).lt.xlim)then
         cdig=re/cdh(alpha,one)
         ilim=dreal(x/re)
         do 1 i=0,ibuf-ilim
            call term(alpha,x,i,p,q)
            cdig=cdig+p*q
    1    continue
      else
```

```
         cdig=cdexp(-x+alpha*cdlog(x))/cdh(alpha,x)
      endif
      return
      end
c
      subroutine term(alpha,x,i,p,q)
c --- Calculate p*q = -1**i(1-x**(alpha+i))/(alpha+i)i! carefully.
      double complex alpha,x,p,q,ci,alphai
      double complex zero,one,two,cdlx
      double precision tol,xlim,dnrm
      data zero,one,two/0.,1.,2./tol/3.d-7/xlim/39./
      if(i.eq.0)q=one
      ci=i
      alphai=alpha+ci
      if(x.eq.zero)then
         p=one/alphai
         if(i.ne.0)q=-q/ci
         return
      endif
      cdlx=cdlog(x)
c --- If (1-x**alphai)=-x**alphai on the computer,
c --- then change the inductive scheme to avoid overflow.
      if(dreal(alphai*cdlx).gt.xlim.and.i.ne.0)then
         p=p*(alphai-one)/alphai
         q=-q*x/ci
         return
      endif
      if(dnrm(alphai).gt.tol)then
         p=(one-x**alphai)/alphai
      else
         p=-cdlx*(one+cdlx*alphai/two)
      endif
      if(i.ne.0)q=-q/ci
      return
      end
c
      double complex function cdh(alpha,x)
c --- Written By Eric Kostlan & Dmitry Gokhman
c --- March  1986
      double complex alpha,x,cdhs
      double complex one,term,sum,cn,alpha1
      double precision buf
      data one/1./buf/0./
c --- If Re(alpha-x) is too big, shift alpha.
      n=dreal(alpha-x)-buf
      if(n.gt.0)then
         cn=n+1
         alpha1=alpha-cn
         term=one/x
         sum=term
         do 1 i=1,n
            cn=n-i+1
            term=term*(alpha1+cn)/x
            sum=term+sum
```

```
    1     continue
            sum=sum+term*alpha1/cdhs(alpha1,x)
            cdh=one/sum
         else
            cdh=cdhs(alpha,x)
         endif
         return
         end
c
      double complex function cdhs(alpha,x)
c --- Written By Eric Kostlan & Dmitry Gokhman
c --- March  1986
      double complex zero,half,one,alpha,x
      double complex p0,q0,p1,q1,r0,r1,ci,factor
      double precision tol1,tol2,error,dnrm
      data zero,half,one/0.,0.5,1./
      data tol1,tol2,error/1.d10,1.d-10,5.d-18/ilim/100000/
      q0=one
      q1=one
      p0=x
      p1=x+one-alpha
      do 1 i=1,ilim
         ci=i
         if(p0.ne.zero.and.q0.ne.zero.and.q1.ne.zero)then
            r0=p0/q0
            r1=p1/q1
            if(dnrm(r0-r1).le.dnrm(r1)*error)then
               cdhs=r1
               return
            endif
c --------- Occasionally renormalize the sequences to avoid over(under)flow.
            if(dnrm(p0).gt.tol1.or.dnrm(p0).lt.tol2.or.
     *         dnrm(q0).gt.tol1.or.dnrm(q0).lt.tol2)then
               factor=p0*q0
               p0=p0/factor
               q0=q0/factor
               p1=p1/factor
               q1=q1/factor
            endif
         endif
         p0=x*p1+ci*p0
         q0=x*q1+ci*q0
         p1=p0+(ci+one-alpha)*p1
         q1=q0+(ci+one-alpha)*q1
    1 continue
c --- If the peripheral routines are written correctly,
c --- the following four statements should never be executed.
      print*,'cdhs:  *** Warning: i >',ilim
      print*,'cdhs:  *** r0,r1= ',r0,r1
      cdhs=half*(r0+r1)
      return
      end
c
      double precision function dnrm(z)
```

```
double complex z
dnrm=dabs(dreal(z))+dabs(dimag(z))
return
end
```

Appendix

Here we present standard formulas that relate the incomplete gamma function and other special functions.

Integral Definitions:

The incomplete gamma function ([3] 8.350.2, 8.310.1):

$$\Gamma(\alpha,x) = \int_x^\infty t^{\alpha-1} e^{-t}\, dt \ , \ \Gamma(\alpha) = \Gamma(\alpha,0).$$

The complementary incomplete gamma function is defined by

$$\gamma(\alpha,x) = \Gamma(x) - \Gamma(\alpha,x).$$

Related to the incomplete gamma function are the *Boehmer integrals* ([8] 39:12:1, 39:12:2):

$$S(x;\alpha) = \int_x^\infty t^{\alpha-1} sin(t) dt,$$

$$C(x;\alpha) = \int_x^\infty t^{\alpha-1} cos(t) dt.$$

Precisely we have

$$\Gamma(\alpha,ix) = C(x;\alpha) + iS(x;\alpha) \ , \ \Gamma(\alpha,-ix) = C(x;\alpha) - iS(x;\alpha).$$

As the algorithm described above works for complex arguments, the Boehmer integrals may be calculated as well.

Special Cases:

The exponential and logarithm integrals are defined as follows:

$$\mathrm{Ei}(x) = \int_{-\infty}^x \frac{e^t}{t} dt \ , \ \mathrm{li}(x) = \int_0^x \frac{dt}{\ln t}.$$

Note that $\mathrm{Ei}(x) = \mathrm{li}(e^x)$. These functions are related to the incomplete gamma function by the following equations ([3] 8.359):

$$\mathrm{Ei}(x) = -\Gamma(0,-x) \ , \ \mathrm{li}(x) = -\Gamma(0,\ln\frac{1}{x}).$$

The sine and cosine integrals ([3] 8.230)

$$\mathrm{si}(x) = -\int_x^\infty \frac{\sin t}{t} dt \ , \ \mathrm{ci}(x) = -\int_x^\infty \frac{\cos t}{t} dt ,$$

are related to the exponential integral by ([3] 8.233.1)

$$\mathrm{Ei}(ix) = \mathrm{ci}(x) + i\ \mathrm{si}(x)\ ,\ \mathrm{Ei}(-ix) = \mathrm{ci}(x) - i\ \mathrm{si}(x).$$

They are Boehmer integrals:

$$\mathrm{ci}(x) = C(x;0)\ ,\ \mathrm{si}(x) = S(x;0).$$

The notation $\mathrm{Ci}(x)=\mathrm{ci}(x)$ and $\mathrm{Si}(x)=\pi/2+\mathrm{si}(x)$ is also used.

The error function, or probability integral, and the Fresnel integrals ([3] 8.250) are defined by

$$erf(x) = \Phi(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2} dt \ ,\ S(x) = \frac{2}{\sqrt{2\pi}}\int_0^x \sin t^2 dt \ ,\ C(x) = \frac{2}{\sqrt{2\pi}}\int_0^x \cos t^2 dt.$$

They are related to the incomplete gamma function as follows ([3] 8.256):

$$\Phi(x) = 1 - \frac{1}{\sqrt{\pi}}\Gamma(\frac{1}{2},x^2).$$

$$C(x) + iS(x) = \frac{\sqrt{i}}{\sqrt{2}}\Phi(\frac{x}{\sqrt{i}})\ ,\ C(x) - iS(x) = \frac{1}{\sqrt{2i}}\Phi(x\sqrt{i}\,).$$

The Fresnel integrals are related to the Boehmer integrals ([8] 39:12:5, 39:12:6):

$$C(x;\frac{1}{2}) = \sqrt{2\pi}\,[\frac{1}{2} - C(\sqrt{x}\,)]\ ,\ S(x;\frac{1}{2}) = \sqrt{2\pi}\,[\frac{1}{2} - S(\sqrt{x}\,)].$$

A related function is Dawson's integral ([8] 42:0:1):

$$\mathrm{daw}(x) = e^{-x^2}\int_0^x e^{t^2} dt \ = \frac{-i\sqrt{\pi}}{2} e^{-x^2}\Phi(ix) = \frac{-i}{2} e^{-x^2}[\sqrt{\pi} - \Gamma(\frac{1}{2},-x^2)].$$

The chi-square distribution ([4] table 26.7) is defined by

$$F_{\chi_k^2}(x) = \frac{1}{2^{k/2}\Gamma(k/2)}\int_0^x t^{(k-2)/2} e^{-t/2} dt.$$

The chi-square distribution is related to the incomplete gamma function by the following equation:

$$F_{\chi_k^2}(x) = 1 - \frac{\Gamma(k/2,x/2)}{\Gamma(k/2)}\ .$$

We also have as a special case an integral studied by Abramowitz ([4] table 7.6), which we denote $\mathrm{abr}(x)$:

$$\mathrm{abr}(x) = \frac{3}{\Gamma(1/3)}\int_0^x e^{-t^3} dt \ = 1 - \frac{\Gamma(1/3,x^3)}{\Gamma(1/3)}\ .$$

References

1. Beam, A. Algorithm 14 Complex Exponential Integral. *Collected Algorithms from ACM 1* (1960).

2. Fullerton, W. Algorithm 435 Modified Incomplete Gamma Function. *Collected Algorithms from ACM 3* (1970).

3. Gradshteyn, I.S., Ryzhik, I.M. *Table of Integrals, Series and Products.* Gosudarstvennoe Izdatel'stvo Fiziko-Matematicheskoy Literatury, Moscow (1963); Corrected enlarged edition: Academic Press, New York (1980).

4. *Handbook of Mathematical Functions.* National Bureau of Standards Applied Mathematics Series 55 (1964).

5. *IMSL Library User's Manual.* IMSL LIB-0009, edition 9 (June 1982).

6. Khovanskii, A. N. *The Application of Continued Fractions and Their Generalizations to Problems in Approximation Theory.* Gosudarstvennoe Izdatel'stvo Tekhniko-Teoreticheskoy Literatury, Moscow (1956); P. Noordhoff, Ltd., Groningen (1963).

7. *SAS User's Guide: Basics, Version 5 Edition.* SAS Institute Inc. (1985).

8. Spanier, J. and Oldham, K. *An Atlas of Functions.* Harper & Row, Washington (1987).

9. *Table of Sine and Cosine Integrals for Arguments from 10 to 100.* National Bureau of Standards Applied Standards Applied Mathematics Series 32 (1954).

10. *Tables of the Exponential Integral for Complex Arguments.* National Bureau of Standards Applied Standards Applied Mathematics Series 51 (1958).

11. *Table of the Gamma Function for Complex Arguments.* National Bureau of Standards Applied Standards Applied Mathematics Series 34 (1954).

12. *UNIX MACSYMA (release 309.2).* Massachusetts Institute of Technology (1976, 1984); enhancements by Symbolics, Inc. (1984).

13. *UNIX Programmer's Manual (4.2 Berkeley Software Distribution, Virtual VAX–11 Version).* Computer Science Division, Dept. of EECS, UC Berkeley (August 1983).

14. Wall, S.H. *Analytic Theory of Continued Fractions.* D. van Nostrand Company, Inc., Toronto (1948).