Hart, J.F., et al. 1968, *Computer Approximations* (New York: Wiley).
Hastings, C. 1955, *Approximations for Digital Computers* (Princeton: Princeton University Press).
Luke, Y.L. 1975, *Mathematical Functions and Their Approximations* (New York: Academic Press).

6.1 Gamma Function, Beta Function, Factorials, Binomial Coefficients

The gamma function is defined by the integral

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt \tag{6.1.1}$$

When the argument z is an integer, the gamma function is just the familiar factorial function, but offset by one,

$$n! = \Gamma(n+1) \tag{6.1.2}$$

The gamma function satisfies the recurrence relation

$$\Gamma(z+1) = z\Gamma(z) \tag{6.1.3}$$

If the function is known for arguments $z > 1$ or, more generally, in the half complex plane $\mathrm{Re}(z) > 1$ it can be obtained for $z < 1$ or $\mathrm{Re}\,(z) < 1$ by the reflection formula

$$\Gamma(1-z) = \frac{\pi}{\Gamma(z)\sin(\pi z)} = \frac{\pi z}{\Gamma(1+z)\sin(\pi z)} \tag{6.1.4}$$

Notice that $\Gamma(z)$ has a pole at $z = 0$, and at all negative integer values of z.

There are a variety of methods in use for calculating the function $\Gamma(z)$ numerically, but none is quite as neat as the approximation derived by Lanczos [1]. This scheme is entirely specific to the gamma function, seemingly plucked from thin air. We will not attempt to derive the approximation, but only state the resulting formula: For certain integer choices of γ and N, and for certain coefficients $c_1, c_2, \ldots, c_N$, the gamma function is given by

$$\begin{aligned}\Gamma(z+1) = {} & (z+\gamma+\tfrac{1}{2})^{z+\frac{1}{2}} e^{-(z+\gamma+\frac{1}{2})} \\ & \times \sqrt{2\pi}\left[c_0 + \frac{c_1}{z+1} + \frac{c_2}{z+2} + \cdots + \frac{c_N}{z+N} + \epsilon\right] \quad (z > 0)\end{aligned} \tag{6.1.5}$$

You can see that this is a sort of take-off on Stirling's approximation, but with a series of corrections that take into account the first few poles in the left complex plane. The constant c_0 is very nearly equal to 1. The error term is parametrized by ϵ. For $\gamma = 5$, $N = 6$, and a certain set of c's, the error is smaller than $|\epsilon| < 2 \times 10^{-10}$. Impressed? If not, then perhaps you will be impressed by the fact that (with these

same parameters) the formula (6.1.5) and bound on ϵ apply for the *complex* gamma function, *everywhere in the half complex plane Re* $z > 0$.

It is better to implement $\ln\Gamma(x)$ than $\Gamma(x)$, since the latter will overflow many computers' floating-point representation at quite modest values of x. Often the gamma function is used in calculations where the large values of $\Gamma(x)$ are divided by other large numbers, with the result being a perfectly ordinary value. Such operations would normally be coded as subtraction of logarithms. With (6.1.5) in hand, we can compute the logarithm of the gamma function with two calls to a logarithm and 25 or so arithmetic operations. This makes it not much more difficult than other built-in functions that we take for granted, such as $\sin x$ or e^x:

```
      FUNCTION gammln(xx)
      REAL gammln,xx
          Returns the value ln[Γ(xx)] for xx > 0.
      INTEGER j
      DOUBLE PRECISION ser,stp,tmp,x,y,cof(6)
          Internal arithmetic will be done in double precision, a nicety that you can omit if five-figure
          accuracy is good enough.
      SAVE cof,stp
      DATA cof,stp/76.18009172947146d0,-86.50532032941677d0,
     *  24.01409824083091d0,-1.231739572450155d0,.1208650973866179d-2,
     *  -.5395239384953d-5,2.5066282746310005d0/
      x=xx
      y=x
      tmp=x+5.5d0
      tmp=(x+0.5d0)*log(tmp)-tmp
      ser=1.000000000190015d0
      do 11 j=1,6
          y=y+1.d0
          ser=ser+cof(j)/y
      enddo 11
      gammln=tmp+log(stp*ser/x)
      return
      END
```

How shall we write a routine for the factorial function $n!$? Generally the factorial function will be called for small integer values (for large values it will overflow anyway!), and in most applications the same integer value will be called for many times. It is a profligate waste of computer time to call `exp(gammln(n+1.0))` for each required factorial. Better to go back to basics, holding `gammln` in reserve for unlikely calls:

```
      FUNCTION factrl(n)
      INTEGER n
      REAL factrl
C     USES gammln
          Returns the value n! as a floating-point number.
      INTEGER j,ntop
      REAL a(33),gammln                Table to be filled in only as required.
      SAVE ntop,a
      DATA ntop,a(1)/0,1./             Table initialized with 0! only.
      if (n.lt.0) then
          pause 'negative factorial in factrl'
      else if (n.le.ntop) then         Already in table.
          factrl=a(n+1)
      else if (n.le.32) then           Fill in table up to desired value.
          do 11 j=ntop+1,n
```

```
        a(j+1)=j*a(j)
    enddo 11
    ntop=n
    factrl=a(n+1)
else                              Larger value than size of table is required. Actually, this big
    factrl=exp(gammln(n+1.))      a value is going to overflow on many computers, but no
endif                             harm in trying.
return
END
```

A useful point is that `factrl` will be *exact* for the smaller values of `n`, since floating-point multiplies on small integers are exact on all computers. This exactness will not hold if we turn to the logarithm of the factorials. For binomial coefficients, however, we must do exactly this, since the individual factorials in a binomial coefficient will overflow long before the coefficient itself will.

The binomial coefficient is defined by

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad 0 \le k \le n \tag{6.1.6}$$

```
    FUNCTION bico(n,k)
    INTEGER k,n
    REAL bico
C   USES factln
    Returns the binomial coefficient (n k) as a floating-point number.
    REAL factln
    bico=nint(exp(factln(n)-factln(k)-factln(n-k)))
    return  The nearest-integer function cleans up roundoff error for smaller values of n and k.
    END
```

which uses

```
    FUNCTION factln(n)
    INTEGER n
    REAL factln
C   USES gammln
    Returns ln(n!).
    REAL a(100),gammln
    SAVE a
    DATA a/100*-1./                              Initialize the table to negative values.
    if (n.lt.0) pause 'negative factorial in factln'
    if (n.le.99) then                            In range of the table.
        if (a(n+1).lt.0.) a(n+1)=gammln(n+1.)    If not already in the table, put it in.
        factln=a(n+1)
    else
        factln=gammln(n+1.)                      Out of range of the table.
    endif
    return
    END
```

If your problem requires a series of related binomial coefficients, a good idea is to use recurrence relations, for example

$$\begin{aligned}\binom{n+1}{k} &= \frac{n+1}{n-k+1}\binom{n}{k} = \binom{n}{k} + \binom{n}{k-1} \\ \binom{n}{k+1} &= \frac{n-k}{k+1}\binom{n}{k}\end{aligned} \tag{6.1.7}$$

Finally, turning away from the combinatorial functions with integer valued arguments, we come to the beta function,

$$B(z,w) = B(w,z) = \int_0^1 t^{z-1}(1-t)^{w-1}dt \tag{6.1.8}$$

which is related to the gamma function by

$$B(z,w) = \frac{\Gamma(z)\Gamma(w)}{\Gamma(z+w)} \tag{6.1.9}$$

hence

```
      FUNCTION beta(z,w)
      REAL beta,w,z
C     USES gammln
        Returns the value of the beta function B(z,w).
      REAL gammln
      beta=exp(gammln(z)+gammln(w)-gammln(z+w))
      return
      END
```

CITED REFERENCES AND FURTHER READING:

Abramowitz, M., and Stegun, I.A. 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, Volume 55 (Washington: National Bureau of Standards; reprinted 1968 by Dover Publications, New York), Chapter 6.

Lanczos, C. 1964, *SIAM Journal on Numerical Analysis*, ser. B, vol. 1, pp. 86–96. [1]

6.2 Incomplete Gamma Function, Error Function, Chi-Square Probability Function, Cumulative Poisson Function

The incomplete gamma function is defined by

$$P(a,x) \equiv \frac{\gamma(a,x)}{\Gamma(a)} \equiv \frac{1}{\Gamma(a)}\int_0^x e^{-t}t^{a-1}dt \qquad (a>0) \tag{6.2.1}$$

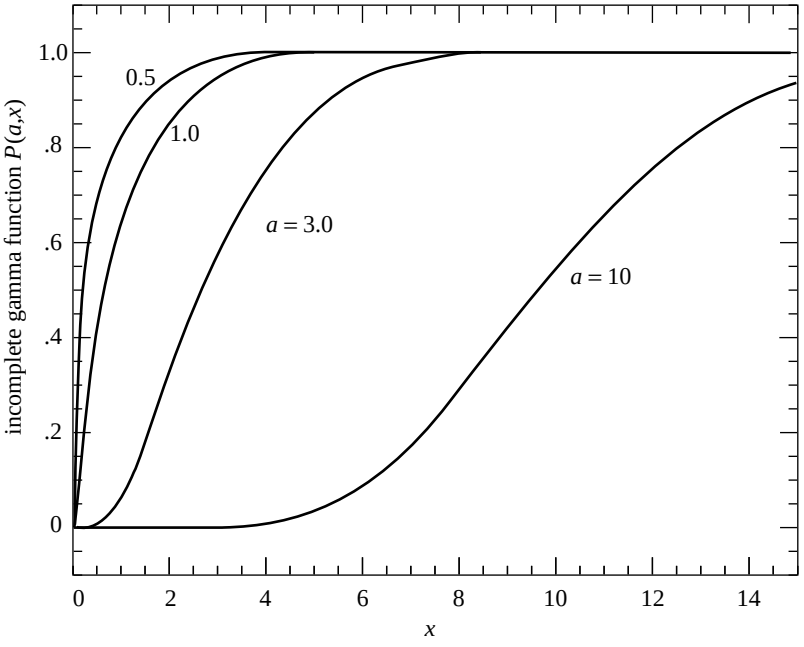

Figure 6.2.1. The incomplete gamma function $P(a,x)$ for four values of a.

It has the limiting values

$$P(a,0)=0 \qquad \text{and} \qquad P(a,\infty)=1 \tag{6.2.2}$$

The incomplete gamma function $P(a,x)$ is monotonic and (for a greater than one or so) rises from "near-zero" to "near-unity" in a range of x centered on about $a-1$, and of width about $\sqrt{a}$ (see Figure 6.2.1).

The complement of $P(a,x)$ is also confusingly called an incomplete gamma function,

$$Q(a,x)\equiv 1-P(a,x)\equiv \frac{\Gamma(a,x)}{\Gamma(a)}\equiv \frac{1}{\Gamma(a)}\int_x^\infty e^{-t}t^{a-1}dt \qquad (a>0) \tag{6.2.3}$$

It has the limiting values

$$Q(a,0)=1 \qquad \text{and} \qquad Q(a,\infty)=0 \tag{6.2.4}$$

The notations $P(a,x)$, $\gamma(a,x)$, and $\Gamma(a,x)$ are standard; the notation $Q(a,x)$ is specific to this book.

There is a series development for $\gamma(a,x)$ as follows:

$$\gamma(a,x)=e^{-x}x^a\sum_{n=0}^{\infty}\frac{\Gamma(a)}{\Gamma(a+1+n)}x^n \tag{6.2.5}$$

One does not actually need to compute a new $\Gamma(a+1+n)$ for each n; one rather uses equation (6.1.3) and the previous coefficient.

A continued fraction development for $\Gamma(a,x)$ is

$$\Gamma(a,x) = e^{-x}x^a\left(\frac{1}{x+}\ \frac{1-a}{1+}\ \frac{1}{x+}\ \frac{2-a}{1+}\ \frac{2}{x+}\cdots\right) \qquad (x>0) \qquad (6.2.6)$$

It is computationally better to use the even part of (6.2.6), which converges twice as fast (see §5.2):

$$\Gamma(a,x) = e^{-x}x^a\left(\frac{1}{x+1-a-}\ \frac{1\cdot(1-a)}{x+3-a-}\ \frac{2\cdot(2-a)}{x+5-a-}\ \cdots\right) \qquad (x>0) \qquad (6.2.7)$$

It turns out that (6.2.5) converges rapidly for x less than about $a+1$, while (6.2.6) or (6.2.7) converges rapidly for x greater than about $a+1$. In these respective regimes each requires at most a few times $\sqrt{a}$ terms to converge, and this many only near $x=a$, where the incomplete gamma functions are varying most rapidly. Thus (6.2.5) and (6.2.7) together allow evaluation of the function for all positive a and x. An extra dividend is that we never need compute a function value near zero by subtracting two nearly equal numbers. The higher-level functions that return $P(a,x)$ and $Q(a,x)$ are

```
      FUNCTION gammp(a,x)
      REAL a,gammp,x
C     USES gcf,gser
        Returns the incomplete gamma function P(a,x).
      REAL gammcf,gamser,gln
      if(x.lt.0..or.a.le.0.)pause 'bad arguments in gammp'
      if(x.lt.a+1.)then             Use the series representation.
        call gser(gamser,a,x,gln)
        gammp=gamser
      else                          Use the continued fraction representation
        call gcf(gammcf,a,x,gln)
        gammp=1.-gammcf             and take its complement.
      endif
      return
      END
```

```
      FUNCTION gammq(a,x)
      REAL a,gammq,x
C     USES gcf,gser
        Returns the incomplete gamma function Q(a,x) ≡ 1 − P(a,x).
      REAL gammcf,gamser,gln
      if(x.lt.0..or.a.le.0.)pause 'bad arguments in gammq'
      if(x.lt.a+1.)then             Use the series representation
        call gser(gamser,a,x,gln)
        gammq=1.-gamser             and take its complement.
      else                          Use the continued fraction representation.
        call gcf(gammcf,a,x,gln)
        gammq=gammcf
      endif
      return
      END
```

The argument gln is returned by both the series and continued fraction procedures containing the value $\ln\Gamma(a)$; the reason for this is so that it is available to you if you want to modify the above two procedures to give $\gamma(a,x)$ and $\Gamma(a,x)$, in addition to $P(a,x)$ and $Q(a,x)$ (cf. equations 6.2.1 and 6.2.3).

The procedures gser and gcf which implement (6.2.5) and (6.2.7) are

```
      SUBROUTINE gser(gamser,a,x,gln)
      INTEGER ITMAX
      REAL a,gamser,gln,x,EPS
      PARAMETER (ITMAX=100,EPS=3.e-7)
C     USES gammln
        Returns the incomplete gamma function P(a,x) evaluated by its series representation as
        gamser. Also returns ln Γ(a) as gln.
      INTEGER n
      REAL ap,del,sum,gammln
      gln=gammln(a)
      if(x.le.0.)then
          if(x.lt.0.)pause 'x < 0 in gser'
          gamser=0.
          return
      endif
      ap=a
      sum=1./a
      del=sum
      do 11 n=1,ITMAX
          ap=ap+1.
          del=del*x/ap
          sum=sum+del
          if(abs(del).lt.abs(sum)*EPS)goto 1
      enddo 11
      pause 'a too large, ITMAX too small in gser'
1     gamser=sum*exp(-x+a*log(x)-gln)
      return
      END
```

```
      SUBROUTINE gcf(gammcf,a,x,gln)
      INTEGER ITMAX
      REAL a,gammcf,gln,x,EPS,FPMIN
      PARAMETER (ITMAX=100,EPS=3.e-7,FPMIN=1.e-30)
C     USES gammln
        Returns the incomplete gamma function Q(a,x) evaluated by its continued fraction repre-
        sentation as gammcf. Also returns ln Γ(a) as gln.
        Parameters: ITMAX is the maximum allowed number of iterations; EPS is the relative accu-
        racy; FPMIN is a number near the smallest representable floating-point number.
      INTEGER i
      REAL an,b,c,d,del,h,gammln
      gln=gammln(a)
      b=x+1.-a                          Set up for evaluating continued fraction by modified
      c=1./FPMIN                            Lentz's method (§5.2) with b0 = 0.
      d=1./b
      h=d
      do 11 i=1,ITMAX                   Iterate to convergence.
          an=-i*(i-a)
          b=b+2.
          d=an*d+b
          if(abs(d).lt.FPMIN)d=FPMIN
          c=b+an/c
          if(abs(c).lt.FPMIN)c=FPMIN
          d=1./d
          del=d*c
```

```
        h=h*del
        if(abs(del-1.).lt.EPS)goto 1
    enddo 11
    pause 'a too large, ITMAX too small in gcf'
1   gammcf=exp(-x+a*log(x)-gln)*h       Put factors in front.
    return
    END
```

Error Function

The error function and complementary error function are special cases of the incomplete gamma function, and are obtained moderately efficiently by the above procedures. Their definitions are

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{6.2.8}$$

and

$$\mathrm{erfc}(x) \equiv 1 - \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt \tag{6.2.9}$$

The functions have the following limiting values and symmetries:

$$\mathrm{erf}(0) = 0 \qquad \mathrm{erf}(\infty) = 1 \qquad \mathrm{erf}(-x) = -\mathrm{erf}(x) \tag{6.2.10}$$

$$\mathrm{erfc}(0) = 1 \qquad \mathrm{erfc}(\infty) = 0 \qquad \mathrm{erfc}(-x) = 2 - \mathrm{erfc}(x) \tag{6.2.11}$$

They are related to the incomplete gamma functions by

$$\mathrm{erf}(x) = P\left(\frac{1}{2}, x^2\right) \qquad (x \geq 0) \tag{6.2.12}$$

and

$$\mathrm{erfc}(x) = Q\left(\frac{1}{2}, x^2\right) \qquad (x \geq 0) \tag{6.2.13}$$

Hence we have

```
    FUNCTION erf(x)
    REAL erf,x
C   USES gammp
        Returns the error function erf(x).
    REAL gammp
    if(x.lt.0.)then
        erf=-gammp(.5,x**2)
    else
        erf=gammp(.5,x**2)
    endif
    return
    END
```

```
      FUNCTION erfc(x)
      REAL erfc,x
C     USES gammp,gammq
        Returns the complementary error function erfc(x).
      REAL gammp,gammq
      if(x.lt.0.)then
        erfc=1.+gammp(.5,x**2)
      else
        erfc=gammq(.5,x**2)
      endif
      return
      END
```

If you care to do so, you can easily remedy the minor inefficiency in `erf` and `erfc`, namely that $\Gamma(0.5) = \sqrt{\pi}$ is computed unnecessarily when `gammp` or `gammq` is called. Before you do that, however, you might wish to consider the following routine, based on Chebyshev fitting to an inspired guess as to the functional form:

```
      FUNCTION erfcc(x)
      REAL erfcc,x
        Returns the complementary error function erfc(x) with fractional error everywhere less than
        1.2 × 10^−7.
      REAL t,z
      z=abs(x)
      t=1./(1.+0.5*z)
      erfcc=t*exp(-z*z-1.26551223+t*(1.00002368+t*(.37409196+
     *      t*(.09678418+t*(-.18628806+t*(.27886807+t*(-1.13520398+
     *      t*(1.48851587+t*(-.82215223+t*.17087277)))))))))
      if (x.lt.0.) erfcc=2.-erfcc
      return
      END
```

There are also some functions of *two* variables that are special cases of the incomplete gamma function:

Cumulative Poisson Probability Function

$P_x(< k)$, for positive x and integer $k \geq 1$, denotes the *cumulative Poisson probability* function. It is defined as the probability that the number of Poisson random events occurring will be between 0 and $k-1$ *inclusive*, if the expected mean number is x. It has the limiting values

$$P_x(< 1) = e^{-x} \qquad P_x(< \infty) = 1 \tag{6.2.14}$$

Its relation to the incomplete gamma function is simply

$$P_x(< k) = Q(k, x) = \texttt{gammq}\,(k, x) \tag{6.2.15}$$

Chi-Square Probability Function

$P(\chi^2|\nu)$ is defined as the probability that the observed chi-square for a correct model should be less than a value χ^2. (We will discuss the use of this function in Chapter 15.) Its complement $Q(\chi^2|\nu)$ is the probability that the observed chi-square will exceed the value χ^2 by chance *even* for a correct model. In both cases ν is an integer, the number of degrees of freedom. The functions have the limiting values

$$P(0|\nu) = 0 \qquad P(\infty|\nu) = 1 \tag{6.2.16}$$

$$Q(0|\nu) = 1 \qquad Q(\infty|\nu) = 0 \tag{6.2.17}$$

and the following relation to the incomplete gamma functions,

$$P(\chi^2|\nu) = P\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right) = \texttt{gammp}\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right) \tag{6.2.18}$$

$$Q(\chi^2|\nu) = Q\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right) = \texttt{gammq}\left(\frac{\nu}{2}, \frac{\chi^2}{2}\right) \tag{6.2.19}$$

CITED REFERENCES AND FURTHER READING:

Abramowitz, M., and Stegun, I.A. 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, Volume 55 (Washington: National Bureau of Standards; reprinted 1968 by Dover Publications, New York), Chapters 6, 7, and 26.

Pearson, K. (ed.) 1951, *Tables of the Incomplete Gamma Function* (Cambridge: Cambridge University Press).

6.3 Exponential Integrals

The standard definition of the exponential integral is

$$E_n(x) = \int_1^\infty \frac{e^{-xt}}{t^n} dt, \qquad x > 0, \quad n = 0, 1, \ldots \tag{6.3.1}$$

The function defined by the principal value of the integral

$$\mathrm{Ei}(x) = -\int_{-x}^{\infty} \frac{e^{-t}}{t}\, dt = \int_{-\infty}^{x} \frac{e^t}{t}\, dt, \qquad x > 0 \tag{6.3.2}$$

is also called an exponential integral. Note that $\mathrm{Ei}(-x)$ is related to $-E_1(x)$ by analytic continuation.

The function $E_n(x)$ is a special case of the incomplete gamma function

$$E_n(x) = x^{n-1}\Gamma(1-n, x) \tag{6.3.3}$$

We can therefore use a similar strategy for evaluating it. The continued fraction — just equation (6.2.6) rewritten — converges for all $x > 0$:

$$E_n(x) = e^{-x}\left(\frac{1}{x+}\,\frac{n}{1+}\,\frac{1}{x+}\,\frac{n+1}{1+}\,\frac{2}{x+}\cdots\right) \tag{6.3.4}$$

We use it in its more rapidly converging even form,

$$E_n(x) = e^{-x}\left(\frac{1}{x+n-}\,\frac{1\cdot n}{x+n+2-}\,\frac{2(n+1)}{x+n+4-}\cdots\right) \tag{6.3.5}$$

The continued fraction only really converges fast enough to be useful for $x \gtrsim 1$. For $0 < x \lesssim 1$, we can use the series representation

$$E_n(x) = \frac{(-x)^{n-1}}{(n-1)!}[-\ln x + \psi(n)] - \sum_{\substack{m=0\\ m\neq n-1}}^{\infty} \frac{(-x)^m}{(m-n+1)m!} \tag{6.3.6}$$

The quantity $\psi(n)$ here is the digamma function, given for integer arguments by

$$\psi(1) = -\gamma, \qquad \psi(n) = -\gamma + \sum_{m=1}^{n-1}\frac{1}{m} \tag{6.3.7}$$

where $\gamma = 0.5772156649\ldots$ is Euler's constant. We evaluate the expression (6.3.6) in order of ascending powers of x:

$$\begin{aligned} E_n(x) = &-\left[\frac{1}{(1-n)} - \frac{x}{(2-n)\cdot 1} + \frac{x^2}{(3-n)(1\cdot 2)} - \cdots + \frac{(-x)^{n-2}}{(-1)(n-2)!}\right] \\ &+ \frac{(-x)^{n-1}}{(n-1)!}[-\ln x + \psi(n)] - \left[\frac{(-x)^n}{1\cdot n!} + \frac{(-x)^{n+1}}{2\cdot(n+1)!} + \cdots\right] \end{aligned} \tag{6.3.8}$$

The first square bracket is omitted when $n = 1$. This method of evaluation has the advantage that for large n the series converges before reaching the term containing $\psi(n)$. Accordingly, one needs an algorithm for evaluating $\psi(n)$ only for small n, $n \lesssim 20 - 40$. We use equation (6.3.7), although a table look-up would improve efficiency slightly.

Amos [1] presents a careful discussion of the truncation error in evaluating equation (6.3.8), and gives a fairly elaborate termination criterion. We have found that simply stopping when the last term added is smaller than the required tolerance works about as well.

Two special cases have to be handled separately:

$$\begin{aligned} E_0(x) &= \frac{e^{-x}}{x} \\ E_n(0) &= \frac{1}{n-1}, \qquad n > 1 \end{aligned} \tag{6.3.9}$$

The routine expint allows fast evaluation of $E_n(x)$ to any accuracy EPS within the reach of your machine's word length for floating-point numbers. The only modification required for increased accuracy is to supply Euler's constant with enough significant digits. Wrench [2] can provide you with the first 328 digits if necessary!

```
FUNCTION expint(n,x)
INTEGER n,MAXIT
REAL expint,x,EPS,FPMIN,EULER
PARAMETER (MAXIT=100,EPS=1.e-7,FPMIN=1.e-30,EULER=.5772156649)
    Evaluates the exponential integral E_n(x).
    Parameters: MAXIT is the maximum allowed number of iterations; EPS is the desired rel-
    ative error, not smaller than the machine precision; FPMIN is a number near the smallest
    representable floating-point number; EULER is Euler's constant γ.
INTEGER i,ii,nm1
REAL a,b,c,d,del,fact,h,psi
nm1=n-1
if(n.lt.0.or.x.lt.0..or.(x.eq.0..and.(n.eq.0.or.n.eq.1)))then
    pause 'bad arguments in expint'
else if(n.eq.0)then                 Special case.
    expint=exp(-x)/x
else if(x.eq.0.)then                Another special case.
    expint=1./nm1
else if(x.gt.1.)then                Lentz's algorithm (§5.2).
    b=x+n
    c=1./FPMIN
    d=1./b
    h=d
    do 11 i=1,MAXIT
        a=-i*(nm1+i)
        b=b+2.
        d=1./(a*d+b)                Denominators cannot be zero.
        c=b+a/c
        del=c*d
        h=h*del
        if(abs(del-1.).lt.EPS)then
            expint=h*exp(-x)
            return
        endif
    enddo 11
    pause 'continued fraction failed in expint'
else                                Evaluate series.
    if(nm1.ne.0)then                Set first term.
        expint=1./nm1
    else
        expint=-log(x)-EULER
    endif
    fact=1.
    do 13 i=1,MAXIT
        fact=-fact*x/i
        if(i.ne.nm1)then
            del=-fact/(i-nm1)
        else
            psi=-EULER              Compute ψ(n).
            do 12 ii=1,nm1
                psi=psi+1./ii
            enddo 12
            del=fact*(-log(x)+psi)
        endif
        expint=expint+del
        if(abs(del).lt.abs(expint)*EPS) return
    enddo 13
```

```
        pause 'series failed in expint'
    endif
    return
    END
```

A good algorithm for evaluating Ei is to use the power series for small x and the asymptotic series for large x. The power series is

$$\mathrm{Ei}(x) = \gamma + \ln x + \frac{x}{1 \cdot 1!} + \frac{x^2}{2 \cdot 2!} + \cdots \tag{6.3.10}$$

where γ is Euler's constant. The asymptotic expansion is

$$\mathrm{Ei}(x) \sim \frac{e^x}{x}\left(1 + \frac{1!}{x} + \frac{2!}{x^2} + \cdots\right) \tag{6.3.11}$$

The lower limit for the use of the asymptotic expansion is approximately $|\ln \mathtt{EPS}|$, where EPS is the required relative error.

```
    FUNCTION ei(x)
    INTEGER MAXIT
    REAL ei,x,EPS,EULER,FPMIN
    PARAMETER (EPS=6.e-8,EULER=.57721566,MAXIT=100,FPMIN=1.e-30)
        Computes the exponential integral Ei(x) for x > 0.
        Parameters: EPS is the relative error, or absolute error near the zero of Ei at x = 0.3725;
        EULER is Euler's constant γ; MAXIT is the maximum number of iterations allowed; FPMIN
        is a number near the smallest representable floating-point number.
    INTEGER k
    REAL fact,prev,sum,term
    if(x.le.0.) pause 'bad argument in ei'
    if(x.lt.FPMIN)then                  Special case: avoid failure of convergence test be-
        ei=log(x)+EULER                     cause of underflow.
    else if(x.le.-log(EPS))then         Use power series.
        sum=0.
        fact=1.
        do 11 k=1,MAXIT
            fact=fact*x/k
            term=fact/k
            sum=sum+term
            if(term.lt.EPS*sum)goto 1
        enddo 11
        pause 'series failed in ei'
1       ei=sum+log(x)+EULER
    else                                Use asymptotic series.
        sum=0.                          Start with second term.
        term=1.
        do 12 k=1,MAXIT
            prev=term
            term=term*k/x
            if(term.lt.EPS)goto 2       Since final sum is greater than one, term itself ap-
            if(term.lt.prev)then            proximates the relative error.
                sum=sum+term            Still converging: add new term.
            else
                sum=sum-prev            Diverging: subtract previous term and exit.
                goto 2
            endif
        enddo 12
2       ei=exp(x)*(1.+sum)/x
    endif
```

```
      return
      END
```

CITED REFERENCES AND FURTHER READING:

Stegun, I.A., and Zucker, R. 1974, *Journal of Research of the National Bureau of Standards*, vol. 78B, pp. 199–216; 1976, *op. cit.*, vol. 80B, pp. 291–311.

Amos D.E. 1980, *ACM Transactions on Mathematical Software*, vol. 6, pp. 365–377 [1]; also vol. 6, pp. 420–428.

Abramowitz, M., and Stegun, I.A. 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, Volume 55 (Washington: National Bureau of Standards; reprinted 1968 by Dover Publications, New York), Chapter 5.

Wrench J.W. 1952, *Mathematical Tables and Other Aids to Computation*, vol. 6, p. 255. [2]

6.4 Incomplete Beta Function, Student's Distribution, F-Distribution, Cumulative Binomial Distribution

The incomplete beta function is defined by

$$I_x(a,b) \equiv \frac{B_x(a,b)}{B(a,b)} \equiv \frac{1}{B(a,b)} \int_0^x t^{a-1}(1-t)^{b-1} dt \qquad (a,b>0) \qquad (6.4.1)$$

It has the limiting values

$$I_0(a,b) = 0 \qquad I_1(a,b) = 1 \qquad (6.4.2)$$

and the symmetry relation

$$I_x(a,b) = 1 - I_{1-x}(b,a) \qquad (6.4.3)$$

If a and b are both rather greater than one, then $I_x(a,b)$ rises from "near-zero" to "near-unity" quite sharply at about $x = a/(a+b)$. Figure 6.4.1 plots the function for several pairs (a,b).

The incomplete beta function has a series expansion

$$I_x(a,b) = \frac{x^a(1-x)^b}{aB(a,b)} \left[1 + \sum_{n=0}^{\infty} \frac{B(a+1,n+1)}{B(a+b,n+1)} x^{n+1} \right], \qquad (6.4.4)$$

but this does not prove to be very useful in its numerical evaluation. (Note, however, that the beta functions in the coefficients can be evaluated for each value of n with just the previous value and a few multiplies, using equations 6.1.9 and 6.1.3.)

The continued fraction representation proves to be much more useful,

$$I_x(a,b) = \frac{x^a(1-x)^b}{aB(a,b)} \left[\frac{1}{1+} \frac{d_1}{1+} \frac{d_2}{1+} \cdots \right] \qquad (6.4.5)$$

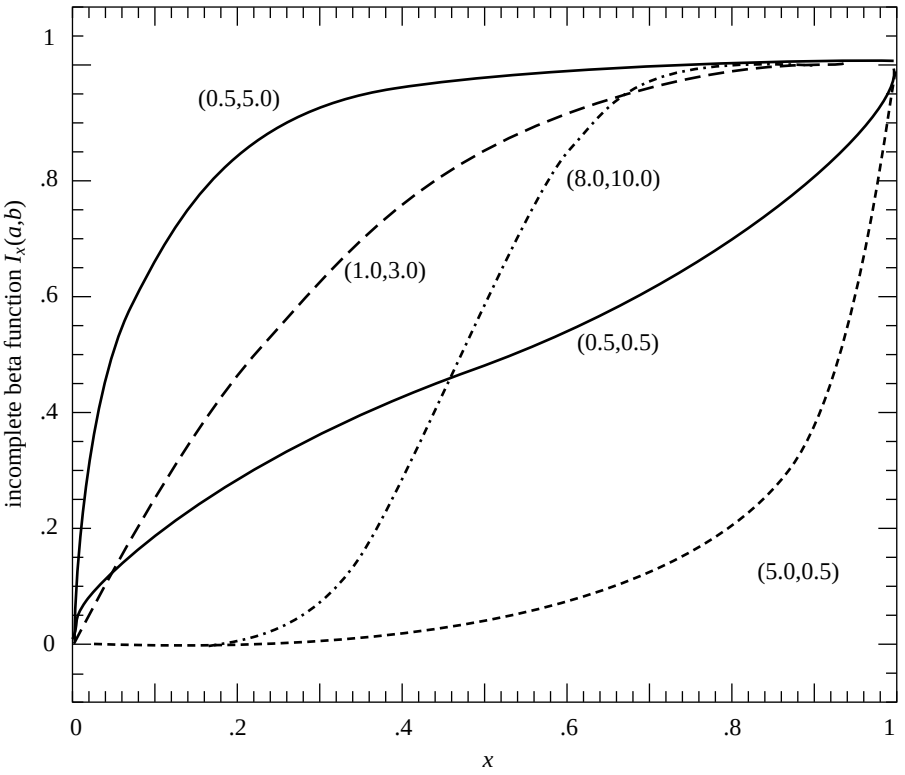

Figure 6.4.1. The incomplete beta function $I_x(a,b)$ for five different pairs of (a,b). Notice that the pairs $(0.5,5.0)$ and $(5.0,0.5)$ are related by reflection symmetry around the diagonal (cf. equation 6.4.3).

where

$$d_{2m+1} = -\frac{(a+m)(a+b+m)x}{(a+2m)(a+2m+1)}$$
$$d_{2m} = \frac{m(b-m)x}{(a+2m-1)(a+2m)} \tag{6.4.6}$$

This continued fraction converges rapidly for $x < (a+1)/(a+b+2)$, taking in the worst case $O(\sqrt{\max(a,b)})$ iterations. But for $x > (a+1)/(a+b+2)$ we can just use the symmetry relation (6.4.3) to obtain an equivalent computation where the continued fraction will also converge rapidly. Hence we have

```
      FUNCTION betai(a,b,x)
      REAL betai,a,b,x
C     USES betacf,gammln
        Returns the incomplete beta function Ix(a,b).
      REAL bt,betacf,gammln
      if(x.lt.0..or.x.gt.1.)pause 'bad argument x in betai'
      if(x.eq.0..or.x.eq.1.)then
        bt=0.
      else                                    Factors in front of the continued fraction.
        bt=exp(gammln(a+b)-gammln(a)-gammln(b)
     *      +a*log(x)+b*log(1.-x))
      endif
      if(x.lt.(a+1.)/(a+b+2.))then            Use continued fraction directly.
```

```
        betai=bt*betacf(a,b,x)/a
        return
    else
        betai=1.-bt*betacf(b,a,1.-x)/b      Use continued fraction after making the symme-
        return                                  try transformation.
    endif
    END
```

which utilizes the continued fraction evaluation routine

```
    FUNCTION betacf(a,b,x)
    INTEGER MAXIT
    REAL betacf,a,b,x,EPS,FPMIN
    PARAMETER (MAXIT=100,EPS=3.e-7,FPMIN=1.e-30)
        Used by betai: Evaluates continued fraction for incomplete beta function by modified
        Lentz's method (§5.2).
    INTEGER m,m2
    REAL aa,c,d,del,h,qab,qam,qap
    qab=a+b                                 These q's will be used in factors that occur in the
    qap=a+1.                                    coefficients (6.4.6).
    qam=a-1.
    c=1.                                    First step of Lentz's method.
    d=1.-qab*x/qap
    if(abs(d).lt.FPMIN)d=FPMIN
    d=1./d
    h=d
    do 11 m=1,MAXIT
        m2=2*m
        aa=m*(b-m)*x/((qam+m2)*(a+m2))
        d=1.+aa*d                           One step (the even one) of the recurrence.
        if(abs(d).lt.FPMIN)d=FPMIN
        c=1.+aa/c
        if(abs(c).lt.FPMIN)c=FPMIN
        d=1./d
        h=h*d*c
        aa=-(a+m)*(qab+m)*x/((a+m2)*(qap+m2))
        d=1.+aa*d                           Next step of the recurrence (the odd one).
        if(abs(d).lt.FPMIN)d=FPMIN
        c=1.+aa/c
        if(abs(c).lt.FPMIN)c=FPMIN
        d=1./d
        del=d*c
        h=h*del
        if(abs(del-1.).lt.EPS)goto 1        Are we done?
    enddo 11
    pause 'a or b too big, or MAXIT too small in betacf'
1   betacf=h
    return
    END
```

Student's Distribution Probability Function

Student's distribution, denoted $A(t|\nu)$, is useful in several statistical contexts, notably in the test of whether two observed distributions have the same mean. $A(t|\nu)$ is the probability, for ν degrees of freedom, that a certain statistic t (measuring the observed difference of means) would be smaller than the observed value if the means were in fact the same. (See Chapter 14 for further details.) Two means are

significantly different if, e.g., $A(t|\nu) > 0.99$. In other words, $1 - A(t|\nu)$ is the significance level at which the hypothesis that the means are equal is disproved.

The mathematical definition of the function is

$$A(t|\nu) = \frac{1}{\nu^{1/2} B(\frac{1}{2}, \frac{\nu}{2})} \int_{-t}^{t} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} dx \tag{6.4.7}$$

Limiting values are

$$A(0|\nu) = 0 \qquad A(\infty|\nu) = 1 \tag{6.4.8}$$

$A(t|\nu)$ is related to the incomplete beta function $I_x(a, b)$ by

$$A(t|\nu) = 1 - I_{\frac{\nu}{\nu+t^2}}\left(\frac{\nu}{2}, \frac{1}{2}\right) \tag{6.4.9}$$

So, you can use (6.4.9) and the above routine `betai` to evaluate the function.

F-Distribution Probability Function

This function occurs in the statistical test of whether two observed samples have the same variance. A certain statistic F, essentially the ratio of the observed dispersion of the first sample to that of the second one, is calculated. (For further details, see Chapter 14.) The probability that F would be as *large* as it is if the first sample's underlying distribution actually has *smaller* variance than the second's is denoted $Q(F|\nu_1, \nu_2)$, where ν_1 and ν_2 are the number of degrees of freedom in the first and second samples, respectively. In other words, $Q(F|\nu_1, \nu_2)$ is the significance level at which the hypothesis "1 has smaller variance than 2" can be rejected. A small numerical value implies a very significant rejection, in turn implying high confidence in the hypothesis "1 has variance greater or equal to 2."

$Q(F|\nu_1, \nu_2)$ has the limiting values

$$Q(0|\nu_1, \nu_2) = 1 \qquad Q(\infty|\nu_1, \nu_2) = 0 \tag{6.4.10}$$

Its relation to the incomplete beta function $I_x(a, b)$ as evaluated by `betai` above is

$$Q(F|\nu_1, \nu_2) = I_{\frac{\nu_2}{\nu_2 + \nu_1 F}}\left(\frac{\nu_2}{2}, \frac{\nu_1}{2}\right) \tag{6.4.11}$$

Cumulative Binomial Probability Distribution

Suppose an event occurs with probability p per trial. Then the probability P of its occurring k *or more* times in n trials is termed a *cumulative binomial probability*, and is related to the incomplete beta function $I_x(a, b)$ as follows:

$$P \equiv \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j} = I_p(k, n-k+1) \tag{6.4.12}$$

For n larger than a dozen or so, `betai` is a much better way to evaluate the sum in (6.4.12) than would be the straightforward sum with concurrent computation of the binomial coefficients. (For n smaller than a dozen, either method is acceptable.)

CITED REFERENCES AND FURTHER READING:

Abramowitz, M., and Stegun, I.A. 1964, *Handbook of Mathematical Functions*, Applied Mathematics Series, Volume 55 (Washington: National Bureau of Standards; reprinted 1968 by Dover Publications, New York), Chapters 6 and 26.

Pearson, E., and Johnson, N. 1968, *Tables of the Incomplete Beta Function* (Cambridge: Cambridge University Press).

6.5 Bessel Functions of Integer Order

This section and the next one present practical algorithms for computing various kinds of Bessel functions of integer order. In §6.7 we deal with fractional order. In fact, the more complicated routines for fractional order work fine for integer order too. For integer order, however, the routines in this section (and §6.6) are simpler and faster. Their only drawback is that they are limited by the precision of the underlying rational approximations. For full double precision, it is best to work with the routines for fractional order in §6.7.

For any real ν, the Bessel function $J_\nu(x)$ can be defined by the series representation

$$J_\nu(x) = \left(\frac{1}{2}x\right)^\nu \sum_{k=0}^{\infty} \frac{(-\frac{1}{4}x^2)^k}{k!\Gamma(\nu+k+1)} \tag{6.5.1}$$

The series converges for all x, but it is not computationally very useful for $x \gg 1$.

For ν *not* an integer the Bessel function $Y_\nu(x)$ is given by

$$Y_\nu(x) = \frac{J_\nu(x)\cos(\nu\pi) - J_{-\nu}(x)}{\sin(\nu\pi)} \tag{6.5.2}$$

The right-hand side goes to the correct limiting value $Y_n(x)$ as ν goes to some integer n, but this is also not computationally useful.

For arguments $x < \nu$, both Bessel functions look qualitatively like simple power laws, with the asymptotic forms for $0 < x \ll \nu$

$$\begin{aligned} J_\nu(x) &\sim \frac{1}{\Gamma(\nu+1)}\left(\frac{1}{2}x\right)^\nu \qquad \nu \geq 0 \\ Y_0(x) &\sim \frac{2}{\pi}\ln(x) \\ Y_\nu(x) &\sim -\frac{\Gamma(\nu)}{\pi}\left(\frac{1}{2}x\right)^{-\nu} \qquad \nu > 0 \end{aligned} \tag{6.5.3}$$