which is related to the gamma function by

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

hence

```
#include <math.h>

float beta(float z, float w)
Returns the value of the beta function B(z, w).
{
    float gammln(float xx);

    return exp(gammln(z)+gammln(w)-gammln(z+w));
}
```

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

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

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,

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

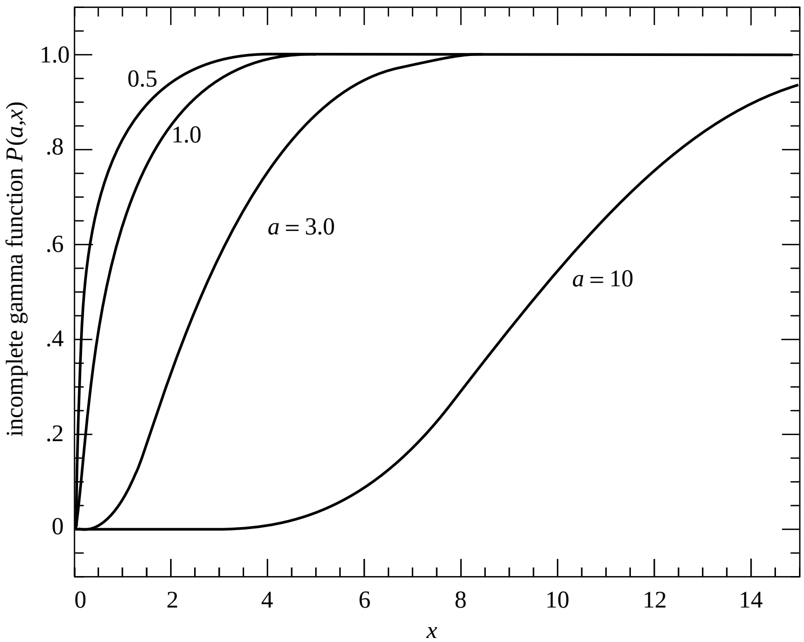

Figure 6.2.1. The incomplete gamma function $P(a,x)$ for four values of a.

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

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

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

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

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

```
float gammp(float a, float x)
Returns the incomplete gamma function P(a, x).
{
    void gcf(float *gammcf, float a, float x, float *gln);
    void gser(float *gamser, float a, float x, float *gln);
    void nrerror(char error_text[]);
    float gamser,gammcf,gln;

    if (x < 0.0 || a <= 0.0) nrerror("Invalid arguments in routine gammp");
    if (x < (a+1.0)) {                Use the series representation.
        gser(&gamser,a,x,&gln);
        return gamser;
    } else {                          Use the continued fraction representation
        gcf(&gammcf,a,x,&gln);
        return 1.0-gammcf;            and take its complement.
    }
}
```

```
float gammq(float a, float x)
Returns the incomplete gamma function Q(a, x) ≡ 1 − P(a, x).
{
    void gcf(float *gammcf, float a, float x, float *gln);
    void gser(float *gamser, float a, float x, float *gln);
    void nrerror(char error_text[]);
    float gamser,gammcf,gln;

    if (x < 0.0 || a <= 0.0) nrerror("Invalid arguments in routine gammq");
    if (x < (a+1.0)) {                Use the series representation
        gser(&gamser,a,x,&gln);
        return 1.0-gamser;            and take its complement.
    } else {                          Use the continued fraction representation.
        gcf(&gammcf,a,x,&gln);
        return gammcf;
    }
}
```

The argument `gln` is set by both the series and continued fraction procedures to 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 functions `gser` and `gcf` which implement (6.2.5) and (6.2.7) are

```
#include <math.h>
#define ITMAX 100
#define EPS 3.0e-7

void gser(float *gamser, float a, float x, float *gln)
Returns the incomplete gamma function P(a, x) evaluated by its series representation as gamser.
Also returns ln Γ(a) as gln.
{
    float gammln(float xx);
```

```
    void nrerror(char error_text[]);
    int n;
    float sum,del,ap;

    *gln=gammln(a);
    if (x <= 0.0) {
        if (x < 0.0) nrerror("x less than 0 in routine gser");
        *gamser=0.0;
        return;
    } else {
        ap=a;
        del=sum=1.0/a;
        for (n=1;n<=ITMAX;n++) {
            ++ap;
            del *= x/ap;
            sum += del;
            if (fabs(del) < fabs(sum)*EPS) {
                *gamser=sum*exp(-x+a*log(x)-(*gln));
                return;
            }
        }
        nrerror("a too large, ITMAX too small in routine gser");
        return;
    }
}
```

```
#include <math.h>
#define ITMAX 100                          Maximum allowed number of iterations.
#define EPS 3.0e-7                         Relative accuracy.
#define FPMIN 1.0e-30                      Number near the smallest representable
                                               floating-point number.
void gcf(float *gammcf, float a, float x, float *gln)
```

Returns the incomplete gamma function $Q(a, x)$ evaluated by its continued fraction representation as gammcf. Also returns $\ln \Gamma(a)$ as gln.

```
{
    float gammln(float xx);
    void nrerror(char error_text[]);
    int i;
    float an,b,c,d,del,h;

    *gln=gammln(a);
    b=x+1.0-a;                             Set up for evaluating continued fraction
    c=1.0/FPMIN;                               by modified Lentz's method (§5.2)
    d=1.0/b;                                   with b0 = 0.
    h=d;
    for (i=1;i<=ITMAX;i++) {               Iterate to convergence.
        an = -i*(i-a);
        b += 2.0;
        d=an*d+b;
        if (fabs(d) < FPMIN) d=FPMIN;
        c=b+an/c;
        if (fabs(c) < FPMIN) c=FPMIN;
        d=1.0/d;
        del=d*c;
        h *= del;
        if (fabs(del-1.0) < EPS) break;
    }
    if (i > ITMAX) nrerror("a too large, ITMAX too small in gcf");
    *gammcf=exp(-x+a*log(x)-(*gln))*h;     Put factors in front.
}
```