```
            }
            if (err < EPS) break;
            odd=!odd;
        }
        if (k > MAXIT) nrerror("maxits exceeded in cisi");
    }
    *si=sums;
    *ci=sumc+log(t)+EULER;
  }
  if (x < 0.0) *si = -(*si);
}
```

CITED REFERENCES AND FURTHER READING:

Stegun, I.A., and Zucker, R. 1976, *Journal of Research of the National Bureau of Standards*, vol. 80B, pp. 291–311; 1981, *op. cit.*, vol. 86, pp. 661–686.

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 5 and 7.

6.10 Dawson's Integral

Dawson's Integral $F(x)$ is defined by

$$F(x) = e^{-x^2} \int_0^x e^{t^2}\, dt \tag{6.10.1}$$

The function can also be related to the complex error function by

$$F(z) = \frac{i\sqrt{\pi}}{2} e^{-z^2} \left[1 - \mathrm{erfc}(-iz)\right]. \tag{6.10.2}$$

A remarkable approximation for $F(x)$, due to Rybicki [1], is

$$F(z) = \lim_{h\to 0} \frac{1}{\sqrt{\pi}} \sum_{n\ \mathrm{odd}} \frac{e^{-(z-nh)^2}}{n} \tag{6.10.3}$$

What makes equation (6.10.3) unusual is that its accuracy increases *exponentially* as h gets small, so that quite moderate values of h (and correspondingly quite rapid convergence of the series) give very accurate approximations.

We will discuss the theory that leads to equation (6.10.3) later, in §13.11, as an interesting application of Fourier methods. Here we simply implement a routine based on the formula.

It is first convenient to shift the summation index to center it approximately on the maximum of the exponential term. Define n_0 to be the even integer nearest to x/h, and $x_0 \equiv n_0 h$, $x' \equiv x - x_0$, and $n' \equiv n - n_0$, so that

$$F(x) \approx \frac{1}{\sqrt{\pi}} \sum_{\substack{n'=-N\\ n'\ \mathrm{odd}}}^{N} \frac{e^{-(x'-n'h)^2}}{n'+n_0}, \tag{6.10.4}$$

Sample page from NUMERICAL RECIPES IN C: THE ART OF SCIENTIFIC COMPUTING (ISBN 0-521-43108-5)
Copyright (C) 1988-1992 by Cambridge University Press. Programs Copyright (C) 1988-1992 by Numerical Recipes Software.
Permission is granted for internet users to make one paper copy for their own personal use. Further reproduction, or any copying of machine-readable files (including this one) to any server computer, is strictly prohibited. To order Numerical Recipes books, diskettes, or CDROMs
visit website http://www.nr.com or call 1-800-872-7423 (North America only), or send email to trade@cup.cam.ac.uk (outside North America).

where the approximate equality is accurate when h is sufficiently small and N is sufficiently large. The computation of this formula can be greatly speeded up if we note that

$$e^{-(x'-n'h)^2} = e^{-x'^2} e^{-(n'h)^2} \left(e^{2x'h}\right)^{n'}. \tag{6.10.5}$$

The first factor is computed once, the second is an array of constants to be stored, and the third can be computed recursively, so that only two exponentials need be evaluated. Advantage is also taken of the symmetry of the coefficients $e^{-(n'h)^2}$ by breaking the summation up into positive and negative values of n' separately.

In the following routine, the choices $h = 0.4$ and $N = 11$ are made. Because of the symmetry of the summations and the restriction to odd values of n, the limits on the `do` loops are 1 to 6. The accuracy of the result in this `float` version is about 2×10^{-7}. In order to maintain relative accuracy near $x = 0$, where $F(x)$ vanishes, the program branches to the evaluation of the power series [2] for $F(x)$, for $|x| < 0.2$.

```
#include <math.h>
#include "nrutil.h"
#define NMAX 6
#define H 0.4
#define A1 (2.0/3.0)
#define A2 0.4
#define A3 (2.0/7.0)

float dawson(float x)
Returns Dawson's integral F(x) = exp(-x^2) ∫_0^x exp(t^2)dt for any real x.
{
    int i,n0;
    float d1,d2,e1,e2,sum,x2,xp,xx,ans;
    static float c[NMAX+1];
    static int init = 0;           Flag is 0 if we need to initialize, else 1.

    if (init == 0) {
        init=1;
        for (i=1;i<=NMAX;i++) c[i]=exp(-SQR((2.0*i-1.0)*H));
    }
    if (fabs(x) < 0.2) {           Use series expansion.
        x2=x*x;
        ans=x*(1.0-A1*x2*(1.0-A2*x2*(1.0-A3*x2)));
    } else {                       Use sampling theorem representation.
        xx=fabs(x);
        n0=2*(int)(0.5*xx/H+0.5);
        xp=xx-n0*H;
        e1=exp(2.0*xp*H);
        e2=e1*e1;
        d1=n0+1;
        d2=d1-2.0;
        sum=0.0;
        for (i=1;i<=NMAX;i++,d1+=2.0,d2-=2.0,e1*=e2)
            sum += c[i]*(e1/d1+1.0/(d2*e1));
        ans=0.5641895835*SIGN(exp(-xp*xp),x)*sum;       Constant is 1/√π.
    }
    return ans;
}
```

Sample page from NUMERICAL RECIPES IN C: THE ART OF SCIENTIFIC COMPUTING (ISBN 0-521-43108-5)
Copyright (C) 1988-1992 by Cambridge University Press. Programs Copyright (C) 1988-1992 by Numerical Recipes Software.
Permission is granted for internet users to make one paper copy for their own personal use. Further reproduction, or any copying of machine-readable files (including this one) to any server computer, is strictly prohibited. To order Numerical Recipes books, diskettes, or CDROMs
visit website http://www.nr.com or call 1-800-872-7423 (North America only), or send email to trade@cup.cam.ac.uk (outside North America).

Other methods for computing Dawson's integral are also known [2,3].

CITED REFERENCES AND FURTHER READING:

Rybicki, G.B. 1989, *Computers in Physics*, vol. 3, no. 2, pp. 85–87. [1]

Cody, W.J., Pociorek, K.A., and Thatcher, H.C. 1970, *Mathematics of Computation*, vol. 24, pp. 171–178. [2]

McCabe, J.H. 1974, *Mathematics of Computation*, vol. 28, pp. 811–816. [3]

6.11 Elliptic Integrals and Jacobian Elliptic Functions

Elliptic integrals occur in many applications, because any integral of the form

$$\int R(t,s)\,dt \tag{6.11.1}$$

where R is a rational function of t and s, and s is the square root of a cubic or quartic polynomial in t, can be evaluated in terms of elliptic integrals. Standard references [1] describe how to carry out the reduction, which was originally done by Legendre. Legendre showed that only three basic elliptic integrals are required. The simplest of these is

$$I_1 = \int_y^x \frac{dt}{\sqrt{(a_1+b_1t)(a_2+b_2t)(a_3+b_3t)(a_4+b_4t)}} \tag{6.11.2}$$

where we have written the quartic s^2 in factored form. In standard integral tables [2], one of the limits of integration is always a zero of the quartic, while the other limit lies closer than the next zero, so that there is no singularity within the interval. To evaluate I_1, we simply break the interval $[y,x]$ into subintervals, each of which either begins or ends on a singularity. The tables, therefore, need only distinguish the eight cases in which each of the four zeros (ordered according to size) appears as the upper or lower limit of integration. In addition, when one of the b's in (6.11.2) tends to zero, the quartic reduces to a cubic, with the largest or smallest singularity moving to $\pm\infty$; this leads to eight more cases (actually just special cases of the first eight). The sixteen cases in total are then usually tabulated in terms of Legendre's standard elliptic integral of the 1st kind, which we will define below. By a change of the variable of integration t, the zeros of the quartic are mapped to standard locations on the real axis. Then only two dimensionless parameters are needed to tabulate Legendre's integral. However, the symmetry of the original integral (6.11.2) under permutation of the roots is concealed in Legendre's notation. We will get back to Legendre's notation below. But first, here is a better way:

Carlson [3] has given a new definition of a standard elliptic integral of the first kind,

$$R_F(x,y,z) = \frac{1}{2}\int_0^\infty \frac{dt}{\sqrt{(t+x)(t+y)(t+z)}} \tag{6.11.3}$$

Sample page from NUMERICAL RECIPES IN C: THE ART OF SCIENTIFIC COMPUTING (ISBN 0-521-43108-5)
Copyright (C) 1988-1992 by Cambridge University Press. Programs Copyright (C) 1988-1992 by Numerical Recipes Software.
Permission is granted for internet users to make one paper copy for their own personal use. Further reproduction, or any copying of machine-readable files (including this one) to any server computer, is strictly prohibited. To order Numerical Recipes books, diskettes, or CDROMs
visit website http://www.nr.com or call 1-800-872-7423 (North America only), or send email to trade@cup.cam.ac.uk (outside North America).