CITED REFERENCES AND FURTHER READING:

Erdélyi, A., Magnus, W., Oberhettinger, F., and Tricomi, F.G. 1953, *Higher Transcendental Functions*, Vol. II, (New York: McGraw-Hill). [1]

Gradshteyn, I.S., and Ryzhik, I.W. 1980, *Table of Integrals, Series, and Products* (New York: Academic Press). [2]

Carlson, B.C. 1977, *SIAM Journal on Mathematical Analysis*, vol. 8, pp. 231–242. [3]

Carlson, B.C. 1987, *Mathematics of Computation*, vol. 49, pp. 595–606 [4]; 1988, *op. cit.*, vol. 51, pp. 267–280 [5]; 1989, *op. cit.*, vol. 53, pp. 327–333 [6]; 1991, *op. cit.*, vol. 56, pp. 267–280. [7]

Bulirsch, R. 1965, *Numerische Mathematik*, vol. 7, pp. 78–90; 1965, *op. cit.*, vol. 7, pp. 353–354; 1969, *op. cit.*, vol. 13, pp. 305–315. [8]

Carlson, B.C. 1979, *Numerische Mathematik*, vol. 33, pp. 1–16. [9]

Carlson, B.C., and Notis, E.M. 1981, *ACM Transactions on Mathematical Software*, vol. 7, pp. 398–403. [10]

Carlson, B.C. 1978, *SIAM Journal on Mathematical Analysis*, vol. 9, p. 524–528. [11]

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 17. [12]

Mathews, J., and Walker, R.L. 1970, *Mathematical Methods of Physics*, 2nd ed. (Reading, MA: W.A. Benjamin/Addison-Wesley), pp. 78–79.

6.12 Hypergeometric Functions

As was discussed in §5.14, a fast, general routine for the the complex hypergeometric function ${}_2F_1(a,b,c;z)$, is difficult or impossible. The function is defined as the analytic continuation of the hypergeometric series,

$$\begin{aligned} {}_2F_1(a,b,c;z) = 1 &+ \frac{ab}{c}\frac{z}{1!} + \frac{a(a+1)b(b+1)}{c(c+1)}\frac{z^2}{2!} + \cdots \\ &+ \frac{a(a+1)\dots(a+j-1)b(b+1)\dots(b+j-1)}{c(c+1)\dots(c+j-1)}\frac{z^j}{j!} + \cdots \end{aligned} \tag{6.12.1}$$

This series converges only within the unit circle $|z| < 1$ (see [1]), but one's interest in the function is not confined to this region.

Section 5.14 discussed the method of evaluating this function by direct path integration in the complex plane. We here merely list the routines that result.

Implementation of the function `hypgeo` is straightforward, and is described by comments in the program. The machinery associated with Chapter 16's routine for integrating differential equations, `odeint`, is only minimally intrusive, and need not even be completely understood: use of `odeint` requires a common block with one zeroed variable, one subroutine call, and a prescribed format for the derivative routine `hypdrv`.

The subroutine `hypgeo` will fail, of course, for values of z too close to the singularity at 1. (If you need to approach this singularity, or the one at ∞, use the "linear transformation formulas" in §15.3 of [1].) Away from $z = 1$, and for moderate values of a, b, c, it is often remarkable how few steps are required to integrate the equations. A half-dozen is typical.

```
      FUNCTION hypgeo(a,b,c,z)
      COMPLEX hypgeo,a,b,c,z
      REAL EPS
      PARAMETER (EPS=1.e-6)                    Accuracy parameter.
C     USES bsstep,hypdrv,hypser,odeint
```

Complex hypergeometric function ${}_2F_1$ for complex a, b, c, and z, by direct integration of the hypergeometric equation in the complex plane. The branch cut is taken to lie along the real axis, Re $z > 1$.

```
      INTEGER kmax,nbad,nok
      EXTERNAL bsstep,hypdrv
      COMPLEX z0,dz,aa,bb,cc,y(2)
      COMMON /hypg/ aa,bb,cc,z0,dz
      COMMON /path/ kmax                       Used by odeint.
      kmax=0
      if (real(z)**2+aimag(z)**2.le.0.25) then Use series...
          call hypser(a,b,c,z,hypgeo,y(2))
          return
      else if (real(z).lt.0.) then             ...or pick a starting point for the path inte-
          z0=cmplx(-0.5,0.)                        gration.
      else if (real(z).le.1.0) then
          z0=cmplx(0.5,0.)
      else
          z0=cmplx(0.,sign(0.5,aimag(z)))
      endif
      aa=a                                     Load the common block, used to pass pa-
      bb=b                                         rameters "over the head" of odeint to
      cc=c                                         hypdrv.
      dz=z-z0
      call hypser(aa,bb,cc,z0,y(1),y(2))       Get starting function and derivative.
      call odeint(y,4,0.,1.,EPS,.1,.0001,nok,nbad,hypdrv,bsstep)
```

The arguments to odeint are the vector of independent variables, its length, the starting and ending values of the dependent variable, the accuracy parameter, an initial guess for stepsize, a minimum stepsize, the (returned) number of good and bad steps taken, and the names of the derivative routine and the (here Bulirsch-Stoer) stepping routine.

```
      hypgeo=y(1)
      return
      END
```

```
      SUBROUTINE hypser(a,b,c,z,series,deriv)
      INTEGER n
      COMPLEX a,b,c,z,series,deriv,aa,bb,cc,fac,temp
```

Returns the hypergeometric series ${}_2F_1$ and its derivative, iterating to machine accuracy. For cabs(z) $\leq 1/2$ convergence is quite rapid.

```
      deriv=cmplx(0.,0.)
      fac=cmplx(1.,0.)
      temp=fac
      aa=a
      bb=b
      cc=c
      do 11 n=1,1000
          fac=((aa*bb)/cc)*fac
          deriv=deriv+fac
          fac=fac*z/n
          series=temp+fac
          if (series.eq.temp) return
          temp=series
          aa=aa+1.
          bb=bb+1.
          cc=cc+1.
      enddo 11
      pause 'convergence failure in hypser'
      END
```

```
SUBROUTINE hypdrv(s,y,dyds)
REAL s
COMPLEX y(2),dyds(2),aa,bb,cc,z0,dz,z
    Derivative subroutine for the hypergeometric equation, see text equation (5.14.4).
COMMON /hypg/ aa,bb,cc,z0,dz
z=z0+s*dz
dyds(1)=y(2)*dz
dyds(2)=((aa*bb)*y(1)-(cc-((aa+bb)+1.)*z)*y(2))*dz/(z*(1.-z))
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). [1]