Special functions

Recursion is a particularly simple way of computing some special functions. Many functions are labeled by an order or index and satisfy recursion relations with respect to this label. If the function can be computed explicitly for the few lowest orders, then the higher orders can be found from these formulas. As an example, consider the computation of the Legendre polynomials, $P_l(x)$, for $|x| \leq 1$ and $l = 0, 1, 2, \ldots$. These are important in the solution of wave equations in situations with a spherical symmetry. The recursion relation with respect to degree is

$$(l+1)P_{l+1}(x) + lP_{l-1}(x) - (2l+1)xP_l(x) = 0\,. \qquad (4.1)$$

Using the explicit values $P_0(x) = 1$ and $P_1(x) = x$, forward recursion in l yields P_l for any higher value of l required. The following FORTRAN program accomplishes this for any value for x and l input.

```
C chap4a.for
20      PRINT *, ' Enter x, l (l .lt. 0 to stop)'
        READ *, X,L
        IF (L .LT. 0) THEN
            STOP
```

Legendre polynomial

$$P_l(x) = \frac{1}{2^l l!}\frac{d^l}{dx^l}(x^2-1)^l$$

$l = 0,1,2,\ldots\ldots$

```
      ELSE IF (L .EQ. 0) THEN
          PL=0.                          !explicit form for L=0
      ELSE IF (L .EQ. 1) THEN
          PL=X                           !explicit form for L=1
      ELSE
          PM=1.                          !values to start recursion
          PZ=X
          DO 10 IL=1,L-1                 !loop for forward recursion
             PP=((2*IL+1)*X*PZ-IL*PM)/(IL+1) !Eq. 4.1
             PM=PZ                       !roll values
             PZ=PP
10        CONTINUE
          PL=PZ
      END IF
      PRINT *,X,L,PL
      GOTO 20
      END
```

This code works with no problems, and the results agree with the values given in the tables to the arithmetic precision of the computer. We can also compute the derivatives of the Legendre polynomials with this algorithm using the relation

$$(1-x^2)P_l' = -lxP_l + lP_{l-1}\ . \tag{4.2}$$

Other sets of orthogonal polynomials, such as Hermite and Laguerre, can be treated similarly.

As a second example, consider the cylindrical Bessel functions, $J_n(x)$ and $Y_n(x)$, which arise as the regular and irregular solutions to wave equations in cylindrical geometries. These functions satisfy the recursion relation

$$C_{n-1}(x) + C_{n+1}(x) = \frac{2n}{x}C_n(x)\ , \tag{4.3}$$

where C_n is either J_n or Y_n. To use these recursion relations in the forward direction, we need the values of C_0 and C_1. These are most easily obtained from the polynomial approximations given in [Ab64], formulas 9.4.1–3. For $|x| < 3$, we have

$$\begin{aligned} J_0(x) = {} & 1 - 2.2499997\,y^2 + 1.2656208y^4 \\ & - 0.3163866y^6 + 0.0444479y^8 - 0.039444y^{10} \\ & + 0.0002100y^{12} + \epsilon\ ; \quad |\epsilon| \le 5\times 10^{-8}\ , \end{aligned} \tag{4.4a}$$

Compute $P_l(x)$ & $P_l'(x)$ for l=0 to l=5
(x=-1 to +1)
Plot $P_l(x)$ & $P_l'(x)$ vs. x

■ **Exercise 4.3** The regular and irregular spherical Bessel functions, j_l and n_l, satisfy the recursion relation

$$s_{l+1} + s_{l-1} = \frac{2l+1}{x} s_l \ ,$$

where s_l is either j_l or n_l. The explicit formulas for the few lowest orders are

$$j_0 = \frac{\sin x}{x} \ ; \ j_1 = \frac{\sin x}{x^2} - \frac{\cos x}{x} \ ; \ j_2 = \left(\frac{3}{x^3} - \frac{1}{x}\right) \sin x - \frac{3}{x^2} \cos x \ ,$$

and

$$n_0 = -\frac{\cos x}{x} \ ; \ n_1 = -\frac{\cos x}{x^2} - \frac{\sin x}{x} \ ; \ n_2 = \left(-\frac{3}{x^3} + \frac{1}{x}\right) \cos x - \frac{3}{x^2} \sin x \ .$$

At $x = 0.5$, the exact values of the functions of order 2 are

$$n_2 = -25.059923 \ ; \ j_2 = 1.6371107 \times 10^{-2} \ .$$

Show that n_2 can be calculated either by explicit evaluation or by forward recursion and convince yourself that the latter method will work for all l and x. Investigate the calculation of $j_2(0.5)$ by forward recursion, explicit evaluation, and by backward recursion and show that the first two methods can be quite inaccurate. Can you see why? Thus, even if explicit expressions for a function are available, the stability of backward recursion can make it the method of choice.

Our discussion has illustrated some pitfalls in computing the some commonly used special functions. Specific methods useful for other functions can be found in the appropriate chapters of [Ab64].

4.2 Gaussian quadrature

In Chapter 1, we discussed several methods for computing definite integrals that were most convenient when the integrand was known at a series of equally spaced lattice points. While such methods enjoy widespread use, especially when the integrand involves a numerically-generated solution to a differential equation, more efficient quadrature schemes exist if we can evaluate the integrand for arbitrary abscissae. One of the most useful of these is Gaussian quadrature.

Consider the problem of evaluating

$$I = \int_{-1}^{1} f(x)\, dx \ .$$

The formulas discussed in Chapter 1 were of the form

$$I \approx \sum_{n=1}^{N} w_n f(x_n) \,, \tag{4.8}$$

where

$$x_n = -1 + 2\frac{(n-1)}{(N-1)}$$

are the equally spaced lattice points. Here, we are referring to the "elementary" quadrature formulas [such as (1.9), (1.11), or (1.13a,b)], and not to compound formulas such as (1.12). For example, for Simpson's rule (1.11), $N = 3$ and

$$x_1 = -1 \,,\ x_2 = 0 \,,\ x_3 = 1 \,;\ w_1 = w_3 = \frac{1}{3} \,,\ w_2 = \frac{4}{3} \,.$$

From the derivation of Simpson's rule, it is clear that the formula is exact when f a polynomial of degree 3 or less, which is commensurate with the error estimate given in Eq. (1.11). More generally, if a quadrature formula based on a Taylor series uses N points, it will integrate exactly a polynomial of degree $N-1$ (degree N if N is odd). That is, the N weights w_n can be chosen to satisfy the N linear equations

$$\int_{-1}^{1} x^p \, dx = \sum_{n=1}^{N} w_n x_n^p \,; \quad p = 0, 1, \ldots, N-1 \,. \tag{4.9}$$

(When N is odd, the quadrature formula is also exact for the odd monomial x^N.)

A greater precision for a given amount of numerical work can be achieved if we are willing to give up the requirement of equally-spaced quadrature points. That is, we will choose the x_n in some optimal sense, subject only to the constraint that they lie within the interval $[-1, 1]$. We then have $2N$ parameters at our disposal in constructing the quadrature formula (the N x_n's and the N w_n's), and so we should be able to choose them so that Eq. (4.9) is satisfied for p ranging from 0 to $2N-1$. That is, the quadrature formula using only N carefully chosen points can be made exact for polynomials of degree $2N-1$ or less. This is clearly more efficient than using equally-spaced abscissae.

To see how to best choose the x_n, we consider the Legendre polynomials, which are orthogonal on the interval $[-1, 1]$:

$$\int_{-1}^{1} P_i(x) P_j(x) \, dx = \frac{2}{2i+1} \delta_{ij} \,. \tag{4.10}$$

$$\int_{-h}^{h} f(x) \, dx = \frac{h}{3}(f_1 + 4f_0 + f_{-1}) + \mathcal{O}(h^5) \,. \tag{1.11}$$

This is Simpson's rule, which can be seen to be accurate to two orders higher than the

It is easily shown that P_i is a polynomial of degree i with i roots in the interval $[-1,1]$. Any polynomial of degree $2N-1$ or less then can be written in the form

$$f(x) = Q(x)P_N(x) + R(x)\,,$$

where Q and R are polynomials of degree $N-1$ or less. The exact value of the required integral (4.8) is then

$$I = \int_{-1}^{1} (QP_N + R)\,dx = \int_{-1}^{1} R\,dx\,, \tag{4.11}$$

where the second step follows from the orthogonality of P_N to all polynomials of degree $N-1$ or less. If we now take the x_n to be the N zeros of P_N, then application of (4.8) gives (exactly)

$$I = \sum_{n=1}^{N} w_n\left[Q(x_n)P_N(x_n) + R(x_n)\right] = \sum_{n=1}^{N} w_n R(x_n)\,. \tag{4.12}$$

It remains to choose the w_n so that R (a polynomial of degree $N-1$ or less) is integrated exactly. That is, the w_n satisfy the set of linear equations (4.9) when the x_n are the zeros of P_N. It can be shown that w_n is related to the derivative of P_N at the corresponding zero. Specifically,

$$w_n = \frac{2}{(1-x_n^2)[P_N'(x_n)]^2}\,.$$

$$w_n = \frac{2(1-x_n^2)}{(n+1)^2[P_{n+1}(x_n)]^2}$$

This completes the specification of what is known as the Gauss-Legendre quadrature formula. Note that it can be applied to any definite integral between finite limits by a simple linear change of variable. That is, for an integral between limits $x = a$ and $x = b$, a change of variable to

$$t = -1 + 2\frac{(x-a)}{(b-a)}$$

reduces the integral to the form required. Other non-linear changes of variable that make the integrand as smooth as possible will also improve the accuracy.

Other types of orthogonal polynomials provide useful Gaussian quadrature formulas when the integrand has a particular form. For example, the Laguerre polynomials, L_i, which are orthogonal on the interval $[0,\infty]$ with the weight function e^{-x}, lead to the Gauss-Laguerre quadrature formula

$$\int_0^{\infty} e^{-x} f(x)\,dx \approx \sum_{n=1}^{N} w_n f(x_n)\,, \tag{4.13}$$

where the x_n are the roots of L_N and the w_n are related to the values of L_{N+1} at these points. Similarly, the Hermite polynomials provide Gauss-Hermite quadrature formulas for integrals of the form

$$\int_{-\infty}^{\infty} e^{-x^2} f(x)\, dx \ .$$

These Gaussian quadrature formulas, and many others, are given in Section 25.4 of [Ab64], which also contains tables of the abscissae and weights.

In the practical application of Gaussian quadrature formulas, one does not need to write programs to calculate the abscissae and weights. Rather, there are usually library subroutines that can be used to establish arrays containing these numbers. For example, subroutine QUAD in Example 4 (see Appendix B) establishes the Gauss-Legendre abscissae and weights for many different values of N.

As a general rule, Gaussian quadrature is the method of choice when the integrand is smooth, or can be made smooth by extracting from it a function that is the weight for a standard set of orthogonal polynomials. We must, of course, also have the ability to evaluate the integrand at the required abscissae. If the integrand varies rapidly, we can compound the basic Gaussian quadrature formula by applying it over several subintervals of the range of integration. Of course, when the integrand can be evaluated only at equally-spaced abscissae (such as when it is generated by integrating a differential equation), then formulas of the type discussed in Chapter 1 must be used.

As an illustration of Gaussian quadrature, consider using a 3-point Gauss-Legendre quadrature to evaluate the integral

$$I = \int_0^3 (1+t)^{\frac{1}{2}}\, dt = 4.66667 \ . \tag{4.14}$$

Making the change of variable to

$$x = -1 + \frac{2}{3}t$$

results in

$$I = \frac{3}{2}\int_{-1}^{1} \left(\frac{3}{2}x + \frac{5}{2}\right)^{\frac{1}{2}} dx \ . \tag{4.15}$$

For $N = 3$, the Gauss-Legendre abscissae and weights are

$$x_1 = -x_3 = 0.774597 \ , \ \ x_2 = 0 \ ; \ \ w_1 = w_3 = 0.555556 \ , \ \ w_2 = 0.888889 \ .$$

Assignment

Straightforward evaluation of the quadrature formula (4.8) then results in $I = 4.66683$, while a Simpson's rule evaluation of (4.14) with $h = 1.5$ gives 4.66228. Gaussian quadrature is therefore more accurate than Simpson's rule by about a factor of 27, yet requires the same number of evaluations of the integrand (three).

■ **Exercise 4.4** Consider the integral

$$\int_{-1}^{1} (1-x^2)^{1/2}\, dx = \frac{\pi}{2}\,.$$

Evaluate this integral using some of the quadrature formulas discussed in Chapter 1 and using Gauss-Legendre quadrature. Note that the behavior of the integrand near $x = \pm 1$ is cause for some caution. Compare the accuracy and efficiency of these various methods for different numbers of abscissae. Note that this integral can be evaluated exactly with a "one-point" Gauss-Chebyshev quadrature formula of the form

$$\int_{-1}^{1} (1-x^2)^{1/2} f(x)\, dx = \sum_{n=1}^{N} w_n f(x_n)\,,$$

with

$$x_n = \cos\frac{n}{N+1}\pi\,;\quad w_n = \frac{\pi}{N+1}\sin^2\frac{n}{N+1}\pi\,.$$

(See Section 25.4.40 of [Ab64].)

4.3 Born and eikonal approximations to quantum scattering

In this example, we will investigate the Born and eikonal approximations suitable for describing quantum-mechanical scattering at high energies, and in particular calculate the scattering of fast electrons (energies greater than several 10's of eV) from neutral atoms. The following project deals with the exact partial-wave solution of this problem.

Extensive discussions of the quantum theory of scattering are given in many texts (see, for example, [Me68], [Ne66], or [Wu62]); we will only review the essentials here. For particles of mass m and energy

$$E = \frac{\hbar^2}{2m}k^2 > 0\,,$$

scattering from a central potential $V(r)$ is described by a wave function

Steps

1. Function for calculating Legendre polynomial of degree N
2. Function for calculating number of nodes of Legendre polynomial in the specified interval and the corresponding values of abscissae. Also calculate the weights for those values of abscissae.
3. Evaluate the integral using the functional form, roots and weights.