Interpolation and Extrapolation

We sometimes know the value of function $f(x)$ at a set of points $x_1, x_2, \cdots, x_N$, but we don't have an analytic expression for $f(x)$ that lets us calculate its value at arbitrary point. For example, the $f(x_i)$'s might result from some physical measurement or from long numerical calculation that cannot be cast into a simple functional form. The task now is to estimate $f(x)$ for arbitrary x by, in some sense, drawing a smooth curve through (and perhaps beyond) the x_i. If the desired x in between the largest and smallest of the x_i's, the problem is called *interpolation*; if x is outside that range, it is called *extrapolation*, which is considerably more hazardous.

Interpolation is the process of obtaining a value from a graph or table that is located between major points given, or between

data points plotted. A ratio process is usually used to obtain the value.

Extrapolation is the process of obtaining a value from a chart or graph that extends beyond the given data. The "trend" of the data is extended past the last point given and an estimate made of the value.

5 Interpolation

Interpolation is related to, but distinct from, function approximation. The task consists of finding an approximate (but easily computable) function

to use in place of a more complicated one. In the case of interpolation, you are given the function f at points not of your own choosing. For the case of function approximation, you are allowed to compute the function f at any desired points for the purpose of developing your approximation.

Conceptually, the interpolation process has two stages: (1) Fit an interpolating function to the data points provided; (2) Evaluate the interpolating function at the target point x. However, this two-stage method is generally not the best way to proceed in practice. Typically it is computationally less efficient, and more susceptible to round off error, than methods which construct a functional estimate $f(x)$ directly from the N tabulated every time one is desired.

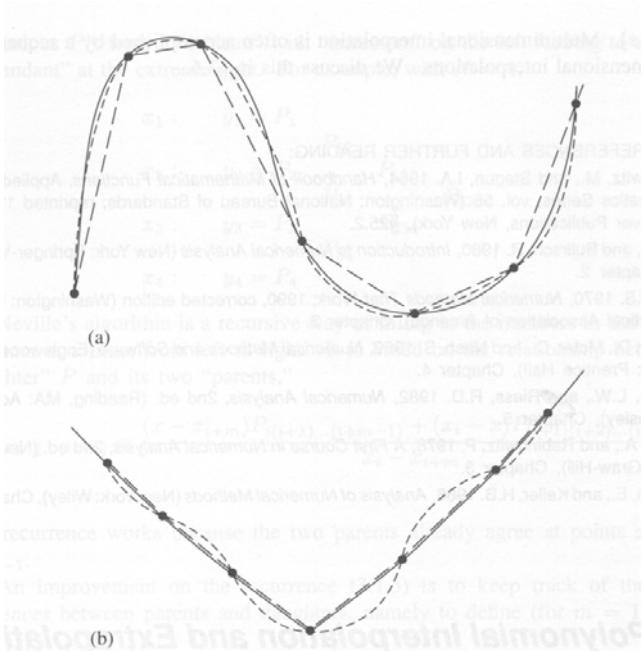


Figure 1: (a) A smooth function (solid line) is more accurately interpolated by a high-order polynomial than a low-order polynomial. (b) A function with sharp corners or rapidly changing higher derivatives is less accurately approximated by a high-order polynomial, which is too stiff than by a low-order polynomial. Even some smooth functions, such as exponential or rational functions, can badly approximated by high-order polynomials.

5.1 Polynomial interpolation

It is well known that there is a unique line through any two points, and there is a unique quadratic line through any three points etc.

5.1.1 Lagrangian polynomial interpolation

In interpolation we are interested in finding the more likely value of y for a given x using as input a table of values of y at a finite number of points x_0, $x_1, \cdots, x_N$. In general we do not have any knowledge of the relationship between y and x other than the $N+1$ pairs of value. Furthermore, there is usually no control over the values of x_i where those of y are given and in general, we have no reason to assume that they are equally spaced.

In general a continuous function $f(x)$ in a finite interval $x = [a, b]$ can always be fitted by a polynomial $P(x)$. This is known as the Weierstras convergence theorem. Our interest here is to find a polynomial approximation for $f(x)$, given to us in the form of $N + 1$ pairs of numbers $\{x_i, f(x_i)\}$ for $i = 0, 1, 2, \ldots, N$. The interpolating polynomial of degree $N - 1$ through N points is given explicitly by

$$P(x) = \sum_{k=0}^{N} p_k(x) f(x_k)$$

where

$$p_k(x) = \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_N)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_N)}.$$

$$p_k(x_l) = \delta_{kl}$$

Since $p_k(x_k) = 1$, $P(x_k) = f(x_k)$. The constructed function goes through all the points $\{x_i, f(x_i)\}$. Such a scheme is known as the Lagrangian polynomial interpolation of $f(x)$. The polynomial $p_k(x)$ is of degree N since the numerator is product of N terms each of which has the form $(x - x_i)$ for $i = 0$ to N except for $i = k$. We shall assume that no two values of x_i are equal to each other, otherwise there will be singularity in $p_k(x)$.

For example, we have $\{x_1, f(x_1)\}$, $\{x_2, f(x_2)\}$, and $\{x_3, f(x_3)\}$.

$$
\begin{aligned}
p_1(x) &= \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} \\
p_2(x) &= \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)}
\end{aligned}
$$

$$p_3(x) = \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)}$$

The final result is

$$P(x) = p_1(x)f(x_1) + p_2(x)f(x_2) + p_3(x)f(x_3).$$

5.1.2 Neville's algorithm

A polynomial of degree m may also be built in the form of a power series with $m + 1$ coefficients, a_0, a_1, $\cdots$, a_m,

$$P_m(x) = \sum_{k=0}^{m} a_k x^k.$$

If our function $f(x)$ has values given at $N + 1$ points, it is possible to construct a polynomial of this form, with degree N, in such a way that

$P_N(x)$ goes through each of the points where the values of the function are known. However, for the purpose of computer calculation, the Lagrange method is not most suitable. Furthermore, it is not easy in such an approach to make an estimate of the errors associated with the calculated result.

In interpolation, our interest is purely local. For this reason, the polynomial need only to fit the known values of the function in a small region where the results are needed. This is different from curve fitting. For this reason, there is often no point of making use of too many points in an interpolation, It complicates the calculations without necessities improving the reliability of the final result.

Consider an example of finding the approximate value of a function at x using the known values at five nearby points. Let the abscissas of five

points be x_1, x_2, x_3, x_4, and x_5. The corresponding values of y at these points are,

$$f_i \equiv f(x_i),$$

respectively. Although the method outlined below is more general, we shall assume, for the sake of simplifying the explanantion, that x_3 is the nearest point to x and x lies between x_3 and x_4.

Through two points: Assume the point x is between x_3 and x_4, and close to x_3. As the zero-order approximation, the value of $f(x)$ is f_3. For a better estimation, we make a linear interpolation.

$$f_{34}(x) = \frac{1}{x_3 - x_4}\{(x - x_4)f_3 - (x - x_3)f_4\}.$$

Through three points: A further improvement may be achieved by making quadratic interpolation. In addition to x_3 and x_4, we also include x_2 as an

input. There are several ways to do this. The most convenient one for the purpose of programming a computer is to adopt Neville's algorithm and make a linear interpolation between the values of f_{23} and f_{34}. This gives us the result

$$f_{234}(x) = \frac{1}{x_2 - x_4}\{(x - x_4)f_{23} - (x - x_2)f_{34}\}$$

For $x = x_3$, $f_{23} = f_{34} = f_3 \rightarrow f_{234} = f_3$.

Through four points:

$$f_{1234}(x) = \frac{1}{x_1 - x_4}\{(x - x_4)f_{123} - (x - x_1)f_{234}\}$$

Through five points:

$$f_{12345}(x) = \frac{1}{x_1 - x_5}\{(x - x_5)f_{1234} - (x - x_1)f_{2345}\}$$

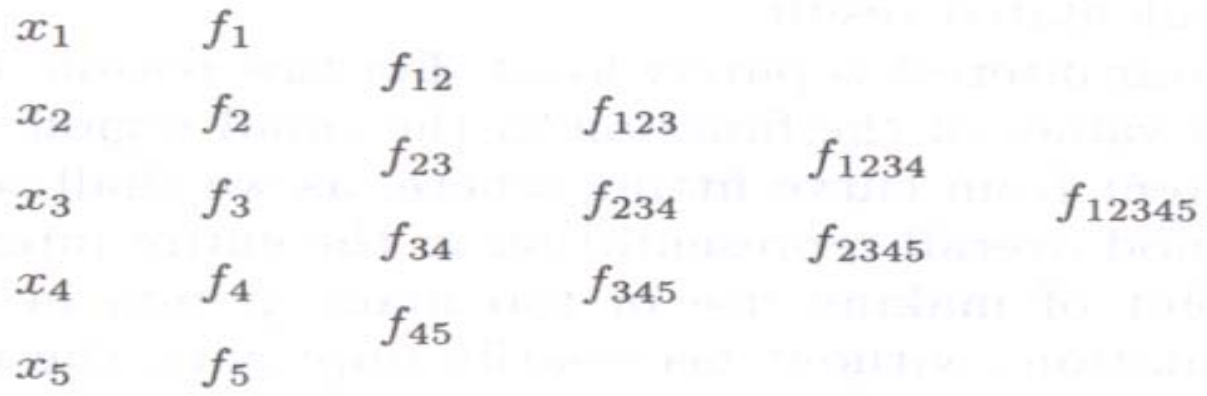


Figure 2: Neville's method of interpolation from five known values.

Recursion relation:

$$f_{ij\cdots kl} = \frac{1}{x_i - x_l}\{(x - x_l)f_{ij\cdots k} - (x - x_i)f_{j\cdots kl}\}$$

.

Improvement on the algorithm

Some improvements are made as follows. We define downward and upward functions

$$\begin{aligned} D_{i,l} &\equiv f_{i,i+1,\cdots,i+l} - f_{i,i+1,\cdots,i+l-1} \\ U_{i,l} &\equiv f_{i,i+1,\cdots,i+l} - f_{i+1,i+1,\cdots,i+l} \end{aligned}$$

with the lowest order of $l = 1$

$$U_{i,1} = D_{i,1} = f_i.$$

Using the recursion relation for $f_{i,i+1,\cdots,i+l}$, we have

$$\begin{aligned} D_{i,l+1} &= \frac{x_i - x}{x_i - x_{i+l+1}}\{D_{i+1,l} - U_{i,l}\} \\ U_{i,l+1} &= \frac{x_{i+l+1} - x}{x_i - x_{i+l+1}}\{D_{i+1,l} - U_{i,l}\} \end{aligned}$$

Since both $U_{i,l+1}$ and $D_{i,l+1}$ are correction to the previous order of approximation, we must make a choice as to which of these two quantities to adopt at each order.

In the zero-order approximation, we take $f(x) = f_i$. If $x > x_i$, the next order approximation comes from $f_{i-1,i}$. Since the difference between f_{i-1} and f_i is given by $U_{i-1,2}$, we shall use $U_{i-1,2}$ to generate the next order correction in the present theme. The difference between $f_{i-1,i}$ and $f_{i-1,i,i+1}$ is given by $D_{i-1,3}$, and we shall make use of $D_{i-1,3}$ next. Continuing the procedure, we come to conclusion that, we start from $f_i = D_{i,1}$ and successively corrections are $U_{i-1,2}$, $D_{i-1,3}$, $U_{i-2,4}$, $D_{i-2,5}$, and so on. A general scheme to obtain successive corrections to the interpolation value of $f(x)$ for $x > x_i$ may now be constructed in the following way. We start with f_i and alternate between using U and D for each higher order. Every time we use U we decrease the value of first subscript i by 1 and increase the value of the second subscript by an equal amount.

A similar approach can also be obtained for $x < x_i$. In this case we start with f_i and alternate between using D and U, with the value i decreased by 1 every time and we encounter an U. A summary of Neville's algorithm is given in Table.

Box 3-1 Subroutine NVLLE(X_IN,F_IN,N,X,FX,DF)
Neville's algorithm for interpolation

Argument list:
- X_IN(I): Input abscissa.
- F_IN(I): Input values of $f(x)$ at x =X_IN(I).
- N: Number of input points.
- X: Value of x to be calculated.
- FX: Returned value of $f(x)$.
- DF: Estimated error for FX.

Initialization: (in the calling program)
- (a) Store a table of the values of $\{x_i, f(x_i)\}$ for $i = 1$ to N.
- (b) Input the value of x.

1. Define U and D of Eq. (3-10) and let IDX= i to indicate the x_i nearest to x.
2. Make an initial estimate using only the table entry nearest to x.
   Determine if x is above x_{IDX} (UP = .TRUE.) or below (UP = .FALSE.).
3. Improving the estimate by higher-order interpolations:
   - (a) Let the order $L = 1$.
   - (b) If UP is .TRUE.:
     - (i) Decrease IDX by 1.
     - (ii) If IDX$\geq$ 1, increase the estimate using U_{IDX}.
     - (iii) Otherwise, set IDX=1 and use D_{IDX} instead.
     - (iv) Change UP to .FALSE. and go to step (d).
   - (c) If UP is .FALSE.:
     - (i) Increase the estimate using D_{IDX} if IDX $> (N - L)$.
     - (ii) Otherwise, set IDX equal to $(N - L)$ and use D_{IDX}.
     - (iii) Change UP to .TRUE..
   - (d) Increase L by 1 and go back to step (b) until L reaches the maximum.
4. Return the interpolated value and the last improvement as the estimate of error.

Figure 3:

5.2 Interpolation using rational function

In general the use of polynomials in approximations works best for smooth function without any singularity either within the range of interest or near one of the end points. Unfortunately these conditions are not met by a large class of function, such as $\tan x$ near $x = \pi/2$ and $\ln x$ for samll values of x. Thus some functions are not well approximated by polynomials, but are well approximated by rational function, that is quotients of polynomial. Let us again assume the function $f(x)$ in the interval $x = [a, b]$ is given to us in terms of $N + 1$ pairs of values $(x_i, f(x_i))$ $(i = 0, 1, \cdots N)$. For convenience we shall arrange the values in ascending order according to x. As such a table, we can write the rational function for interpolation in the following way,

$$R_{\mu\nu s}(x) = \frac{P_{\mu\nu s}(x)}{Q_{\mu\nu s}(x)} = \frac{p_{0,\mu\nu s} + p_{1,\mu\nu s}x + \cdots + p_{\mu,\mu\nu s}x^{\mu}}{q_{0,\mu\nu s} + q_{1,\mu\nu s}x + \cdots + q_{\nu,\mu\nu s}x^{\nu}}$$

The subscript s is needed to distinguish the different independent rational functions that can be constructed. Set $q_{0,\mu\nu s} = 1$, the function has only $\mu + \nu + 1$ unknown p's and q's, we must have

$$N + 1 = \mu + \nu + 1.$$

It is therefore possible to make $R_{\mu\nu s}(x)$ equal to $f(x)$ at $N + 1$ points. We shall choose $R_{\mu\nu s}(x)$ in such a way that

$$R_{\mu\nu s}(x) = f_i$$

at $x = x_s, x_{s+1}, \cdots x_{s+\mu+\nu}$. In other words, the function $R_{\mu\nu s}(x) - f(x)$ has $(\mu + \nu + 1)$ nodes and they occur at $x = x_s$, x_{s+1}, $\cdots$, $x_{s+\mu+\nu}$. As a result, rational functions having the same degrees μ and ν but differing in the location of their first node may be distinguished by the label s. Furthermore, the numerator polynomial $P_{\mu\nu s}(x)$ and denominator polynomial $Q_{\mu\nu s}(x)$ for a given μ, ν, and s are related to each other, since in

order to have roots at the given location, it is impossible to change one of the two in a nontrivial way without to modify the other. For this reason, coeffcients $p_{k,\mu\nu s}$ of the numerator polynomial $P_{\mu\nu s}(x)$ and $q_{k,\mu\nu s}$ of the numerator polynomial $Q_{\mu\nu s}(x)$ are labeled by four indicies k, the power of x it is associated with, as well as μ, ν, and s.

In specifying a rational function interpolating function, you must give a desired order of both the numerator and denominator. Bulirsch and Stoer found an algorithm of the Neville type which performs rational function interpolation on tabulated data. The Bulirsch and Stoer algorithm produces the so-called diagonal rational function with $\mu = \nu = m/2$ for even m or $\mu = (m-1)/2$ and $\nu = (m+1)/2$ for odd m. The derivation of the algorithm is fairly straightforward but some what too tedious to repeat here. Starting with $R_{0s} = f_s$ and $R_{ms} = 0$ for $m < 0$, Stoer and Bulirsch

give a recurrence relation

$$R_{m,s} = R_{m-1,s+1} + \frac{R_{m-1,s+1}(x) - R_{m-1,s}(x)}{\left(\frac{x-x_s}{x-x_{s+m}}\right)\left(1 - \frac{R_{m-1,s+1}(x)-R_{m-1,s}(x)}{R_{m-1,s+1}(x)-R_{m-2,s+1}(x)}\right) - 1}.$$

Recursion relation between difference:

From the previous section, it is advantageous for computational purpose to use instead recursion relations for the differences between rational functions of different degree. Fur this purpose, we define two difference functions:

$$\Delta_{m,s} = R_{m,s}(x) - R_{m-1,s}(x)$$

$$\Theta_{m,s} = R_{m,s}(x) - R_{m-1,s+1}(x)$$

Notice that

$$\begin{aligned}\Delta_{m+1,s} - \Theta_{m+1,s} &= R_{m,s+1}(x) - R_{m,s}(x) \\ \Delta_{m,s+1} - \Theta_{m,s} &= R_{m,s+1}(x) - R_{m,s}(x)\end{aligned}$$

Thus

$$\begin{aligned}\Theta_{m,s} &= R_{m,s}(x) - R_{m-1,s+1}(x) \\ &= \frac{R_{m-1,s+1}(x) - R_{m-1,s}(x)}{\frac{x-x_s}{x-x_{s+m}}(1 - \frac{R_{m-1,s+1}(x)-R_{m-1,s}(x)}{R_{m-1,s+1}(x)-R_{m-2,s+1}(x)}) - 1} \\ &= \frac{(\Delta_{m,s+1} - \Theta_{m,s})\Delta_{m,s+1}}{\frac{x-x_s}{x-x_{s+m}}\Theta_{m,s} - \Delta_{m,s+1}}\end{aligned}$$

$$\Delta_{m+1,s} = \Theta_{m+1,s} + \Delta_{m,s+1} - \Theta_{m,s}$$

$$= \frac{\frac{x-x_s}{x-x_{s+m}}\Theta_{m,s}(\Delta_{m,s+1}-\Theta_{m,s})}{\frac{x-x_s}{x-x_{s+m}}\Theta_{m,s}-\Delta_{m,s+1}}$$

By starting with $\Delta_{0,s} = \Theta_{0,s} = f_s$, we have now amethod for rational function interpolation that is very familiar to Neville's algorithm for polynomial interpolation. This is true also for the strategy for making a choice between $\Delta_{m,s}$ and $\Theta_{m,s}$ to improve the interpolated result at each stage.

5.3 Fourier transformation

A periodic function is one that repeats itself after a fixed interval. For example, if

$$f(x) = f(x + 2L)$$

the function $f(x)$ has a period of $2L$. One important property of such a function is that it can be expressed in terms of a Fourier series

$$f(x) = \frac{a_0}{2} + \sum_{m=1}^{+\infty} \left(a_m \cos \frac{m\pi}{L} x + b_m \sin \frac{m\pi}{L} x \right).$$

Using the orthogonality relations between trigonometry functions, the coefficients a_m and b_m are given by

$$\begin{aligned} a_m &= \frac{1}{L} \int_{-L}^{+L} f(x) \cos \frac{m\pi}{L} x dx \\ b_m &= \frac{1}{L} \int_{-L}^{+L} f(x) \sin \frac{m\pi}{L} x dx \end{aligned}$$

Alternatively, we can make use of exponential functions,

$$f(x) = \sum_{l=-\infty}^{+\infty} g_l e^{i\pi l x/L}$$

$$g_l = \frac{1}{2L}\int_{-L}^{+L} f(x)e^{-i\pi lx/L}dx$$

If we take the limit $L \to +\infty$, the summation index m becomes a continuous variable.

Discrete Fourier transform: Assume $f(x)$ is given in terms of N values, $f(x_1)$, $f(x_2)$, $\cdots$, $f(x_N)$ at $x = x_1, x_2, \cdots, x_N$. The periodic boundary condition is

$$f(x_i) = f(x_{i+N})$$

with $2L = x_{i+N} - x_i$.

$$f_k = f(x_k) = \sum_{l=1}^{N} \alpha^{lk} g_l \to f = Ag$$

$$g = A^{-1}f = \frac{1}{N}\sum_{l=1}^{N} \alpha^{-lk} f(x_k)$$

Fast Fourier transform: When the number of available points N is equal to 2^n algorithm of the fast Fourier transform usually works better. Make use of the identities,

$$\alpha^N = e^{i2\pi} = 1, \alpha^{N/2} = -1, \alpha^{N/4} = i,$$

we can simplify the transform matrix A.

5.4 Inverse interpolation

In interpolation, we are interested to find the value of $f(x)$ at a point x within an interval $[a, b]$. On many occasions, we are interested in its

opposite question to find the value x for a given value $f(x)$. This is the problem of inverse interpolation and is, in general, different from that interpolation. There are two main reasons for the difference. First, in interpolation, the value of $f(x)$ is assumed to be singly valued; otherwise the problem is not well defined. Second, most of interpolation techniques approximate the function $f(x)$ by a polynomial or rational function in x. The fact that such an approximation can be made does not guarantee that x may be represented by a polynomial or rational function in terms of $y = f(x)$.

The problem of inverse interpolation may be posed in the following way. Given $f(x)$, we wish to find the value of x for which

$$f(x) - C = 0.$$

To have a unique solution, it is necessary that $f(x)$ is not a constant equal to C in any finite size region of x. If the $f(x)$ is monotonic, the standard interpolation algorithm may be used. In general, a rational function interpolation is preferred here to ensure better results.

5.4.1 The bisection method.

Let us define a new function

$$\phi(x) = f(x) - C.$$

The problem is transferred into one of searching for the zero of $\phi(x)$ in the range of $x = [a, b]$. For the convenience of discussion, we shall assume that there is only one zero in the interval and $\phi(a) < 0$ and $\phi(b) > 0$.

Box 3-7 Inverse interpolation using bisection.
Find the only root of $\phi(x)$ in an interval

Initialization:

(a) Let ϵ be the required accuracy.

(b) Input a as the lower limit of the range and b the upper limit.

1. Check whether $\phi(a) < 0$ and $\phi(b) > 0$ (increasing function) or $\phi(a) > 0$ and $\phi(b) < 0$ (decreasing function).
2. Find the midpoint $x_m = \frac{1}{2}(a + b)$.
   (a) If $\phi(x_m) > 0$:
      (i) replace b by x_m for an increasing function, or
      (ii) replace a by x_m for a decreasing function.
   (b) If $\phi(x_m) < 0$:
      (i) replace a by x_m for an increasing function, or
      (ii) replace b by x_m for a decreasing function.
   (c) If $\phi(x_m) = 0$, exit. The root of $\phi(x)$ is at x_m.
3. Repeat step 2 if $(b - a) > \epsilon$. Otherwise, return $x = \frac{1}{2}(a + b)$.

Figure 4:

To start with, we take the middle point of a and b,

$$x_m = \frac{a+b}{2}.$$

Assume ϵ the required accuracy. If $\phi(x_m) < \epsilon$, the zero of $\phi(x)$ must be in the interval $[x_m, b]$. However, if $\phi(x_m) < \epsilon$, the zero of $\phi(x)$ must be in the interval $[x_m, b]$. In either case, the range of our search is reduced to half of the original. The procedure is repeated for the new interval until we reach at $\phi(x_m) = \epsilon$. The final middle point is the root we search.

Write a program to find x for $\sin x = 1/2$ in the range of $[0, \pi/2]$.

5.4.2 Newton's method of interpolation

5.4.3 Bessel's formula for finite difference

5.5 Cubic spline

A spline is a piece of flexible wood or plastic that can be bent into arbitrary smooth shapes. In the days before computers, it was commonly used in tracing a smooth curve between points in a sheet of graph paper. For this reason, the word is now associated with numerical techniques that perform the same function. The basic idea is to use a third-degree polynomial to approximate the relation between the dependent and independent variables,

similar to what we did in interpolation. For most cases, it is adequate to use a third-degree polynomial of the form,

$$f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

and hence we name it as the cubic spline.

Given a tabulated function $f_i = f(x_i)$, $i = 0, 1, \cdots, N$, focus attention on one particular interval between x_j and x_{j+1}. Linear interpolation in that interval gives the interpolation formula

$$f(x) = f_i \lambda_i(x) + f_{i+1} \omega_i(x)$$

where

$$\begin{aligned}\lambda_i(x) &= \frac{x_{i+1}-x}{x_{i+1}-x_i} \\ \omega_i(x) &= 1-\lambda_i(x)\end{aligned}$$

$\lambda_i(x) = 1$ and $\omega_i(x) = 0$ at $x = x_i$. $\lambda_i(x) = 0$ and $\omega_i(x) = 1$ at $x = x_{i+1}$. The goal of cubic spline is to get an interpolation formula that is smooth in the first derivative, and continuous in the second derivative, both within an interval and at its boundaries. One possible way to satisfy these requirements is to use

$$\begin{aligned}f(x)_{[x_i,x_{i+1}]} &= f_i\lambda_i(x) + f_{i+1}\omega_i(x) \\ &\quad + f_i''\frac{(x_{i+1}-x_i)^2}{6}\{\lambda_i^3(x)-\lambda_i(x)\} \\ &\quad + f_{i+1}''\frac{(x_{i+1}-x_i)^2}{6}\{\omega_i^3(x)-\omega_i(x)\}\end{aligned}$$

where f'' will be identified later as the values of the second-order derivative of $f(x)$ at x_i and x_{i+1}, respectively.j

The first-order derivative

$$\begin{aligned}\left.\frac{df}{dx}\right|_{[x_i,x_{i+1}]} = & \frac{f_{i+1}-f_i}{x_{i+1}-x_i} \\ & -f_i''\frac{(x_{i+1}-x_i)^2}{6}\{3\lambda_i^2(x)-1\} \\ & +f_{i+1}''\frac{(x_{i+1}-x_i)^2}{6}\{3\omega_i^2(x)-1\}\end{aligned}$$

The second-order derivative

$$\left.\frac{d^2f}{dx^2}\right|_{[x_i,x_{i+1}]} = f_i''\lambda_i(x)+f_{i+1}''\omega_{i+1}(x).$$

Since $\lambda_i(x_i) = 1$ and $\omega_{i+1}(x_i) = 0$, we have

$$\left.\frac{d^2 f}{dx^2}\right|_{[x_{i-1},x_i]} = \left.\frac{d^2 f}{dx^2}\right|_{[x_i,x_{i+1}]} = f_i''$$

Therefore the second-order derivative is continuous at the point $x = x_i$.
The continuity of the first-order derivative at $x = x_i$

$$\left.\frac{df}{dx}\right|_{[x_{i-1},x_i]} = \left.\frac{df}{dx}\right|_{[x_i,x_{i+1}]}$$

leads to

$$(x_i - x_{i-1})f_{i-1}'' + 2(x_{i+1} - x_{i-1})f_i'' + (x_{i+1} - x_i)f_{i+1}'' = 6\left(\frac{f_{i+1} - f_i}{x_{i+1} - x_i} - \frac{f_i - f_{i-1}}{x_i - x_{i-1}}\right)$$

There are three unknown quantities in these equations, f_{i-1}'', f_i'', f_{i+1}''.
Consider all boundaries in the interval, there are $N - 1$ linear equations in the $N + 1$ unknown f_i''. Therefore there is a two-parameter family of possible solutions.

However, since the function $f(x)$ is unknown, it is difficult in general for us to have an idea on what the second derivative should be anywhere in the region of interest. For a unique solution, we need to specify two further conditions, typically taken as boundary conditions at x_0 and x_N. The most common ways of doing this are either (1) set one or both of f_0'' and f_N'' equal to zero, giving so called *natural cubic spline*, or (2) set either of f_0'' and f_N''. to values to make the first derivative of interpolating function have a specified value on either or both boundaries.

Once f_0'' and f_N'' are determined, we can obtain the other $N-1$ f'' by

solving a set of tridiagonal equations.

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & 0 & \cdots & 0 \\ 0 & a_3 & b_3 & c_3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & b_{N-2} & c_{N-2} \\ 0 & \cdots & 0 & 0 & a_{N-1} & b_{N-1} \end{pmatrix} \begin{pmatrix} f_1'' \\ f_2'' \\ f_3'' \\ \vdots \\ f_{N-2}'' \\ f_{n-1}'' \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{N-2} \\ d_{N-1} \end{pmatrix}$$

where

$$a_i = \begin{cases} 0 & for \quad i = 1 \\ x_i - x_{i-1} & for \quad i = 2, 3, \cdots, N-1 \end{cases}$$

$$b_i = 2(x_i - x_{i-1})$$

$$c_i = \begin{cases} 0 & for \quad i = N-1 \\ x_i - x_{i-1} & for \quad i = 2, 3, \cdots, N-1 \end{cases}$$

$$d_i = \begin{cases} 6\left(\frac{f_2-f_1}{x_2-x_1} - \frac{f_1-f_0}{x_1-x_0}\right) - (x_1 - x_0)f_0'' & for \quad i = 1 \\ 6\left(\frac{f_{i+1}-f_i}{x_{i+1}-x_i} - \frac{f_i-f_{i-1}}{x_i-x_{i-1}}\right) & for \quad i = 2, 3, \cdots, N-2 \\ 6\left(\frac{f_N-f_{N-1}}{x_N-x_{N-1}} - \frac{f_{N-1}-f_{N-2}}{x_{N-1}-x_{N-2}}\right) - (x_N - x_{N-1})f_N'' & for \quad i = N-1 \end{cases}$$

Since the square matrix involved has a tridiagonal form, the solution can be obtained using the method of Gaussian elimination.

Once all the values of f_0'' are known, we have a complete set of continuous, third order polynomial approximation to $f(x)$.

Example: Cubic Spline (See Mathnote.nb)

5.6 Problem

1. If a function is polynomial of degree 2 have the form

$$f(x) = a_0 + a_1 x + a_2 x^2$$

show that the second-order difference of Neville's algorithm is exact. Express a_i in terms of notations of Neville's algorithm.

2. Using the Neville's method find the value of x for $f(x) = 0.0$.

-0.2	0	0.2	0.4	0.6	0.8	1.0	1.2
-0.7328	-0.7071	-0.6528	-0.3981	0.721	3.1165	8.4372	18.0797

3. Apply a cubic spline to the periodic function given by the following

nine pairs of values:

x	0	$\pi/2$	π	$3\pi/2$	2π	$5\pi/2$	3π	$7\pi/2$	4π
$f(x)$	0	1	0	1	0	1	0	1	0

Using $f''(x = 0) = f''(x = 4\pi) = 0$ as the two additional conditions required. These set of points can also be fitted by a since function. What is the differences between the results of a cubic spline and $\sin x$.

6 Extrapolation

Extrapolation is the process of deducing the approximate values of a function $f(x)$ outside the interval $x = [x_0, x_N]$, where the known values are.

It is inherently a very delicate operation, since the available information says very little about the behavior of $f(x)$ out side $[x_0, x_N]$. For example, a common practice to describe a minimum in a potential is to use a parabola. For our purpose here, we can approximate the function around some point $x = x_0$ near the minimum in the form

$$f(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2$$

With three or more values of $f(x)$ available around x_0, we can have a fairly good description of the function in the neighbourhood. On the other hand we have no way of predicting that, for example, there a maximum beyond $x = x_5$ and the way the function increases in value for $x < x_1$. The example amplifies the fact that extroplation can only be carried out reliably in regions close to the interval where the known values are.

Since extrapolation is also based on an approximate form of unknown function $f(x)$, *the basic philosophy of any method of extrapolation is*

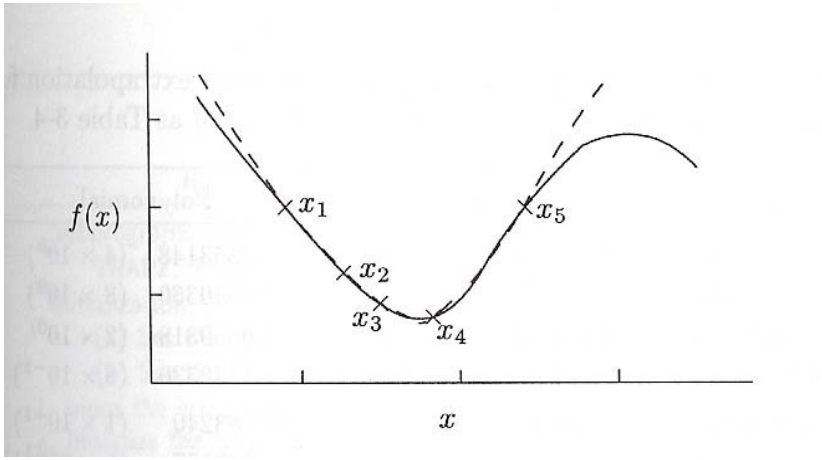


Figure 5: Example illustrating the danger of extrapolation.

essentially the same as that for interpolation. Consequently, it is possible to make some minor adjustments to the algorithm for interpolation and adapt the same techniques for extrapolation.

6.1 Revisit the calculation of π: The ancient Chinese method

To estimate the ratio of the circumference to the diameter of a circle, the π value, by measuring to a very high accuracy the ratio of the length of the side of a polygon to its diameter. If the length of the side of a regular polygon with n sides is denoted as l_n and the diameter is taken as the unit of length, then the approximation of π is given by

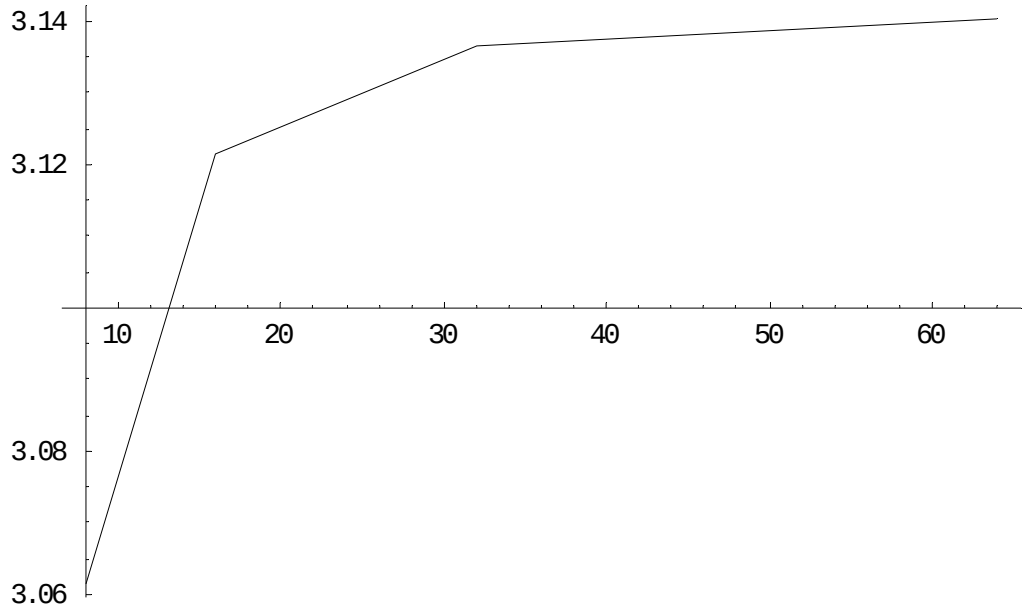
$$\pi_n = nl_n.$$

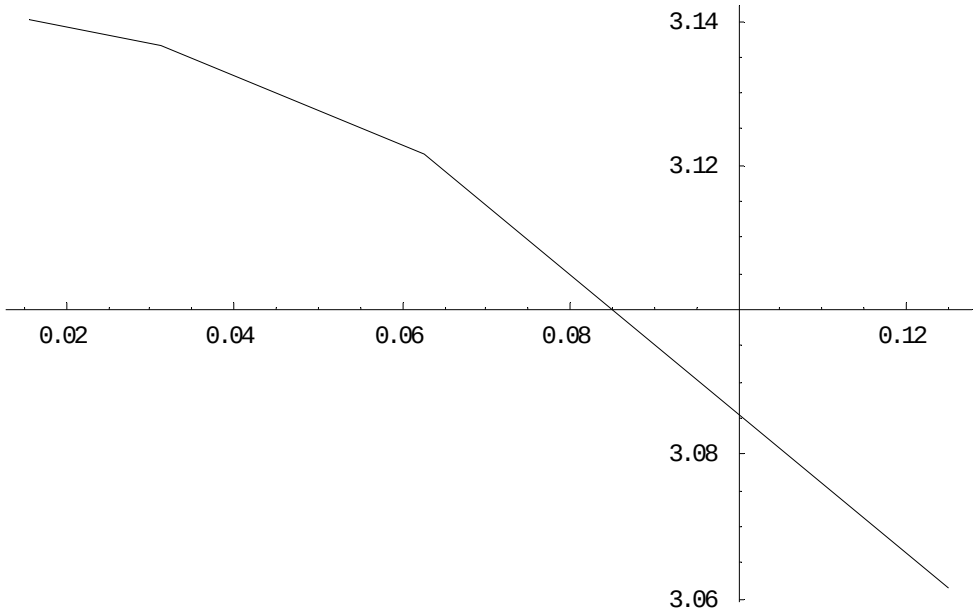
The exact π value is the limit of π_n as $n \to +\infty$. The value of π_n obtained from the measurements or calculation on the polygons can be formally written as

$$\pi_n = \pi_\infty + \frac{a_1}{n} + \frac{a_2}{n^2} + \frac{a_3}{n^3} + \cdots$$

n	8	16	32	64	$\cdots$
π_n	3.061467	3.121445	3.136548	3.140331	$\cdots$

From the four data, one can truncate the expansion at the third order and solve the equation set to obtain the value of π_∞. The approximation of π is $\pi_\infty = 3.141583$. The extrapolated value can accurately measure π_n with higher value of n.


3.14
3.12
3.08
3.06
10
20
30
40
50
60


3.14
3.12
3.08
3.06
0.02
0.04
0.06
0.08
0.12

6.2 Richardson extrapolation

An important application of extrapolation techniques is found in the method of Richardson's deferred approach to the limit. We shall use numerical integration as an illustration of the method. We know that the errors in numerical integration are proportional to powers in the step size h and one way to improve the accuracy is to reduce h. Here we shall make use of extrapolation techniques to achieve the same goal. For this purpose it is convenient to express the integral in terms of the Eular-MacLaurin summation formula,

$$\begin{aligned} I &= \int_a^b f(x)dx \\ &= h\left(\frac{1}{2}f_0 + f_1 + \cdots + f_{N-1} + \frac{1}{2}f_N\right) - \frac{B_2}{2!}h^2\{f'(b) - f'(a)\} - \cdots \end{aligned}$$

$$
\begin{aligned}
& -\frac{B_{2k}}{(2k)!}h^{2k}\{f^{(2k-1)}(b) - f^{(2k-1)}(a)\} - \cdots \\
= \; & h\left(\frac{1}{2}f_0 + f_1 + \cdots + f_{N-1} + \frac{1}{2}f_N\right) \\
& -\frac{h^2}{12}\{f'(b) - f'(a)\} + \frac{h^4}{720}\{f''(b) - f''(a)\} + \cdots
\end{aligned}
$$

where we have divided the range $[a, b]$ into N equal subinterval, each of size h. The first term in the final form is the result given by the trapezoidal rule of integration

$$
I(h) = h\left(\frac{1}{2}f_0 + f_1 + \cdots + f_{N-1} + \frac{1}{2}f_N\right).
$$

In the limit of zero step size, all the other terms vanish and we have the equality,

$$
I = \lim_{h \to 0} I(h).
$$

In a numerical calculation, it is not possible to evaluate $I(h)$ for $h = 0$. However there is nothing preventing us to extrapolate its value to the limit $h \to 0$.

To start with, we need to construct a difference table in $I(h)$ for different step sizes. Up to terms involving h^2, the integral can be written in the form

$$I_N = I(h) - \frac{1}{12}\{f'(b) - f'(a)\}h^2$$

$$I_{2N} = I(h/2) - \frac{1}{12}\{f'(b) - f'(a)\}(h/2)^2$$

From these two equations, we can eliminate the unknown factor $f'(b) - f'(a)$ and obtain the result

$$I_{N,2N} = (4I_{2N} - I_N)/3$$

$$\begin{aligned} I_{N,2N} &= \frac{4}{3}I(h/2) - \frac{1}{3}I(h) \\ &= I(h/2) + \frac{1}{3}\{I(h/2) - I(h)\} \end{aligned}$$

This is a better approximation than that given by $I(h)$ as errors of the order of h^2 have been eliminated. Similarly, we have

$$I_{2N,4N} = I(h/4) + \frac{1}{3}\{I(h/4) - I(h/2)\}$$

The error in both equations is of the order of h^4. This can be seen from the following,

$$\begin{aligned} I &\approx I_{N,2N} + \frac{1}{720}\{f''(b) - f''(a)\}\frac{1}{4}h^4 \\ I &\approx I_{2N,4N} + \frac{1}{720}\{f''(b) - f''(a)\}\frac{1}{4}\left(\frac{h}{2}\right)^4 \end{aligned}$$

Furthermore by eliminating terms that depends on the unknown factor

$\{f''(b) - f''(a)\}$ from these two equations, we have

$$I_{N,2N,4N} = I_{2N,4N} + \frac{1}{2^4 - 1}\{I_{2N,4N} - I_{N,2N}\}$$

This result is improved in the accuracy by order h^2. It is obvious that we can repeat the process and achieve even better accuracies. For example by making use of $I(h/2)$, $I(h/4)$, and $I(h/6)$, we obtain $I_{2N,4N,6N}$. This, together with $I_{N,2N,4N}$, enables us to eliminate errors of order h^6. The general recursion relation is

$$I_{2^i N,2^{i+2}N,\cdots 2^{i+m-2}N,2^{i+m}N} = I_{2^{i+2}N,\cdots 2^{i+m-2}N,2^{i+m}N}$$
$$+\frac{1}{2^m - 1}\{I_{2^{i+2}N,\cdots 2^{i+m-2}N,2^{i+m}N} - I_{2^i N,2^{i+2}N,\cdots 2^{i+m-2}N}\}$$

In this way the terms up to order h^m has been eliminated.

Example: Calculation of the error function

6.3 Romberg integration

Another way of using extrapolation techniques for numerical integration is the method of Romberg. The difference from Richardson's deferred approach to the linit may be seen by expressiong $I(h)$ as a power series in h^2

$$I(h) = a_0 + a_1 h^2 + a_2 h^4 + \cdots + a_k h^{2k} + \cdots$$

To identify the coefficients a_k in the expression, we shall rewrite the Euler-Maclaurin formular in the following form,

$$I(h) = I + \frac{h^2}{12}\left\{f^{(1)}(b) - f^{(2)}(a)\right\} - \frac{h^4}{720}\left\{f^{(2)}(b) - f^{(2)}(a)\right\} + \cdots$$

where I is the exact value as

$$I = \lim_{h \to 0} I(h)$$

What we are interested is a_0. To extrapolate for this value we need again a number of $I(h)$ calculated with different step sizes:

$$h_k = \frac{b-a}{n_k}$$

for $n_1 < n_2 < n_3 < \cdots$. This is the essence of the Romberg method of the integration. Conceptually it is simpler than the Richardson's deferred approach to the limit and the method allows more freedom in choosing the step size.

7 Special Functions

A large number of mathematical functions have been developed over the years because of the needs in physics and engineering. I shall not try to

make any attempt to cover the all range of possible applications for any special function. Instead I choose Hermite polynomial as the eigenfunction of simple harmonic oscillator in quantum mechanics for its intrinsic usefulness and introduction of the certain computational techniques.

We have already encountered a special function in the example "calculation of the period of pendulum" – the complete elliptic integral of the first kind.

$$K(\alpha^2) = \int_0^1 \frac{dz}{(1-z^2)(1-\alpha^2 z^2)}$$

7.1 Hermite polynomial and harmonic oscillator

We consider a simple harmonic oscillator in a one-dimensional case. The basic Hamiltonian is

$$H = \frac{p^2}{2m} + \frac{m}{2}\omega^2 x^2$$

where x and p are Hermitian since they are physically dynamic variables. The quantum condition is

$$[x, p] = xp - px = i\hbar$$

As the first step to solve the problem, we evaluate the commutator basket of x and p with H

$$[x, H] = \frac{1}{2m}[x, p^2] = \frac{i\hbar}{m}p$$

$$[p, H] = \frac{m\omega^2}{2}[p, x^2] = -i\hbar m\omega^2 x$$

We have

$$[x \pm i\frac{p}{m\omega}, H] = \pm\hbar\omega(x \pm i\frac{p}{m\omega})$$

As

$$[x + i\frac{p}{m\omega}, x - i\frac{p}{m\omega}] = \frac{2\hbar}{m\omega}$$

we take

$$a = \left(\frac{m\omega}{2\hbar}\right)^{1/2}(x + i\frac{p}{m\omega});\ a^\dagger = \left(\frac{m\omega}{2\hbar}\right)^{1/2}(x - i\frac{p}{m\omega})$$

such that

$$[a, a^\dagger] = 1.$$

Furthermore,

$$a^\dagger a = \frac{m\omega}{2\hbar}(x - i\frac{p}{m\omega})(x + i\frac{p}{m\omega}) = \frac{H}{\hbar\omega} - \frac{1}{2}$$

Thus

$$H = \hbar\omega(N + \frac{1}{2})$$

where

$$N = a^\dagger a,$$

Obviously,

$$[H, N] = 0$$

$i.e.$, N and H can be diagonalized simultaneously. We denote the eigenkets of N by its eigenvalues n. So

$$N\,|n\rangle = n\,|n\rangle$$

and

$$H\left|n\right\rangle = (n + \frac{1}{2})\hbar\omega\left|n\right\rangle$$

What's n?

We first note that

$$[a, N] = aa^\dagger a - a^\dagger aa = [a, a^\dagger]a = a$$

$$[a^\dagger, N] = -a^\dagger$$

As a result,

$$\begin{aligned} Na^\dagger\left|n\right\rangle &= \left\{[N, a^\dagger] + a^\dagger N\right\}\left|n\right\rangle \\ &= (n+1)a^\dagger\left|n\right\rangle \end{aligned}$$

Likewise,

$$Na\left|n\right\rangle = (n-1)a\left|n\right\rangle$$

Thus $a^{\dagger}\left|n\right\rangle$ is also an eigenket of N with eigenvalue $n+1$ (increased by one). $a\left|n\right\rangle$ is an eigenket of N with eigenvalue $n-1$ (decreased by one). So we call a the decreasing operator and $a^{\dagger}$ the increasing operator. From the point of view of energy, the increase (decrease) of n by one amounts to the creation (annihilation) of one quantum unit of energy. Hence the terms "creation and annihilation operator" for $a^{\dagger}$ and a are deemed appropriate. Since $a\left|n\right\rangle$ is an eigenket with $n-1$. We write

$$\begin{aligned} a\left|n\right\rangle &= C_n\left|n-1\right\rangle \\ (\left\langle n\right| a^{+})(a\left|n\right\rangle) &= \left\langle n-1\right|(C_n^{*}C_n)\left|n-1\right\rangle \\ &\Downarrow \\ \left\langle n\left|a^{+}a\right|n\right\rangle &= n = |C_n|^2 \geq 0 \end{aligned}$$

We take C_n real. then,

$$a\,|n\rangle = n^{1/2}\,|n-1\rangle$$

Likewise,

$$a^{\dagger}\,|n\rangle = (n+1)^{1/2}\,|n+1\rangle$$

Suppose we keep on applying a to both sides of the equation,

$$a^2\,|n\rangle = [n(n-1)]^{1/2}\,|n-2\rangle\,;$$

$$a^3\,|n\rangle = [n(n-1)(n-2)]^{1/2}\,|n-3\rangle\,.$$

The sequence of the eigenket must terminate at $n = 0$ as $n \geq 0$. If n is a non-integer, the sequence shall not terminate and leads to eigenkets with negative eigenvalues. This is in contradiction with $n \geq 0$. To conclude n is a nonnegative integer. The lowest energy state, i.e. the ground state of

the oscillator is

$$E_{n=0} = \frac{1}{2}\hbar\omega$$

The eigenstate	eigenvalue
$\|0\rangle$	$\frac{1}{2}\hbar\omega$
$\|1\rangle = a^\dagger \|0\rangle$	$\frac{3}{2}\hbar\omega$
$\|2\rangle = \frac{1}{2^{1/2}}(a^\dagger)^2 \|0\rangle$	$\frac{5}{2}\hbar\omega$
$\|3\rangle = \frac{1}{(3!)^{1/2}}(a^\dagger)^3 \|0\rangle$	$\frac{7}{2}\hbar\omega$
$\vdots$	$\vdots$
$\|n\rangle = \frac{1}{(n!)^{1/2}}(a^\dagger)^n \|0\rangle$	$(n+\frac{1}{2})\hbar\omega$

Position and Momentum

$$x = \left(\frac{\hbar}{2m\omega}\right)^{1/2}(a + a^+)$$

$$p = i\left(\frac{m\hbar\omega}{2}\right)^{1/2}(-a+a^{+})$$

From the orthonormal conditions

$$\begin{aligned}\left\langle n'\left|a\right|n\right\rangle &= n^{1/2}\delta_{n'n-1}\\ \left\langle n'\left|a^{\dagger}\right|n\right\rangle &= [n+1]^{1/2}\,\delta_{n'n+1}\end{aligned}$$

The matrix elements of x and p are

$$\begin{aligned}\left\langle n'\left|x\right|n\right\rangle &= \left(\frac{\hbar}{2m\omega}\right)^{1/2}\\ &\quad (n^{1/2}\delta_{n'n-1}+(n-1)^{1/2}\,\delta_{n'n+1})\\ \left\langle n'\left|p\right|n\right\rangle &= i\left(\frac{m\omega\hbar}{2}\right)^{1/2}\\ &\quad (-n^{1/2}\delta_{n'n-1}+(n-1)^{1/2}\,\delta_{n'n+1})\end{aligned}$$

In the Position Representation: the wave function

We first derive $\langle x|0\rangle$

$$\begin{aligned} a\left|0\right\rangle &= 0 \\ &\Longrightarrow \left\langle x\left|a\right|0\right\rangle = 0 \\ &\Longrightarrow \int dx' \left\langle x\left|a\right|x'\right\rangle \left\langle x'\left|0\right.\right\rangle = 0 \end{aligned}$$

Recall

$$a = \left(\frac{m\omega}{2\hbar}\right)^{1/2}(x + i\frac{p}{m\omega})$$

$$\begin{aligned} \left\langle x\left|x\right|x'\right\rangle &= x\delta(x-x') \\ \left\langle x\left|p\right|x'\right\rangle &= -i\hbar\delta(x-x')\frac{\partial}{\partial x} \\ &\Longrightarrow (x + \frac{\hbar}{m\omega}\frac{d}{dx})\left\langle x|0\right\rangle = 0 \end{aligned}$$

Take $\hbar/m\omega = x_0^2$. The solution of the equation is

$$\langle x|0\rangle = \frac{1}{\pi^{\frac{1}{4}} x_0^{1/2}} \exp(-\frac{1}{2}\frac{x^2}{x_0^2})$$

In general

$$\begin{aligned}
\langle x|n\rangle &= \left\langle x \left| \frac{(a^\dagger)^n}{(n!)^{1/2}} \right| 0 \right\rangle = \left\langle x \left| \frac{(a^\dagger)^n}{(n!)^{1/2}} \right| x \right\rangle \langle x|0\rangle \\
&= \frac{x_0^{-\frac{n+1}{2}}}{\pi^{\frac{1}{4}} (2^n n!)^{1/2}} (x - x_0^2 \frac{d}{dx})^n e^{-\frac{x^2}{2x_0^2}}
\end{aligned}$$

7.1.1 Harmonic oscillator wave function and Hermite polynomial

The solution for a given n can be written in the form

$$\langle x|n\rangle = \frac{1}{\pi^{\frac{1}{4}}\left(2^n n!\right)^{1/2}} e^{-\frac{x^2}{2x_0^2}} H_n(\frac{x}{x_0})$$

where $H_n(\rho)$ is Hermite polynomial of degree n. The Hermite polynomials are the solution of

$$\frac{d^2}{d\rho^2} H_n(\rho) - 2\rho \frac{d}{d\rho} H_n(\rho) + 2nH_n(\rho) = 0$$

The first several Hermite polynomials are

$$\begin{aligned} H_0(\rho) &= 1 \\ H_1(\rho) &= 2\rho \end{aligned}$$

$$\begin{aligned} H_2(\rho) &= 4\rho^2 - 1 \\ H_3(\rho) &= 8\rho^3 - 12\rho \end{aligned}$$

A recursion relation is

$$H_{n+1}(\rho) = 2\rho H_n(\rho) - 2nH_{n-1}(\rho)$$

To calculate the function, it may be expressed as a power series consisting of even or odd powers of ρ,

$$H_n(\rho) = \sum_{k=0,\text{ or } 1}^{n} a_{n,k}\rho^k$$

where

$$a_{n,k+2} = \frac{2k - 2n}{(k+1)(k+2)} a_{n,k}$$

k is even (or odd) if n is even (or odd). In terms of $a_{n,k}$, the recursion relation becomes

$$a_{n+1,k} = 2a_{n,k-1} - 2na_{n-1,k}$$

From the fact that $H_0(\rho) = 1$, $H_1(\rho) = 2\rho$, we have the starting values

$$\begin{aligned} a_{0,0} &= 1 \\ a_{1,0} &= 0 \\ a_{1,1} &= 2 \end{aligned}$$

and $a_{n,k} = 0$ for $k > n$. From these it is possible to find the coefficient $a_{2,0}$ and $a_{2,2}$ for $H_2(\rho)$. Once the coefficient for $H_2(\rho)$ are known, we can use them with those of $H_1(\rho)$ to produce those for $H_3(\rho)$.

Two ways:

1.

$$H_n(\rho) = \sum_{k=0,\text{ or } 1}^{n} a_{n,k}\rho^k$$

where

$$a_{n+1,k} = 2a_{n,k-1} - 2na_{n-1,k}$$

2.

$$H_{n+1}(\rho) = 2\rho H_n(\rho) - 2nH_{n-1}(\rho)$$

Problem: From the recursion relation, calculate the coefficients for $H_{10}(\rho)$ and $H_{11}(\rho)$.

7.2 Other special functions

Legendre polynomials and spherical harmonics: the Helmholtz equation or the Schrodinger equation in the three dimensioanl space or the eigenfunc-

tion of the angular momentum

$$\bigtriangledown^2\varphi(\mathbf{r}) + k^2\varphi(\mathbf{r}) = 0$$

Spherical Bessel functions: Schrodinger equation

$$\frac{d}{dr}(r^2\frac{dR_l}{dr}) + (k^2r^2 - l(l+1))R_l = 0$$

Laguerre polynomials: Coumb potential and the hydrogen-like atom

7.2.1 Error Function

Error integral and error function is defined by the relation

$$\mathsf{erf}(x) = \frac{2}{\pi^{1/2}}\int_0^x \mathsf{exp}[-t^2]dt.$$

The normalization constant is taken in such a way that

$$\mathrm{erf}(x = \infty) = 1.$$

From this condition we obtain a relation with the complementary error function is

$$\mathrm{erf}\, c(x) = 1 - \mathrm{erf}(x) = \frac{2}{\pi^{1/2}} \int_x^\infty \exp[-t^2] dt.$$

and

$$P(x) = \frac{1}{(2\pi)^{1/2}} \int_{-\infty}^{x} \exp[-t^2/2] dt = \frac{1 + \mathrm{erf}(x/2^{1/2})}{2}.$$

From this condition we obtain a relation with the complementary error function,

$$t = \frac{1}{1 + px}$$

where $p = 0.47047$ the error function may be approximate with an uncertainty of $\epsilon \leq 2.5 \times 10^{-5}$ is given by the expression

$$\text{erf}(x) = 1 - (a_1 t + a_2 t^2 + a_3 t^3) e^{-x^2}$$

where

$$a_1 = 0.3480242; a_2 = -0.0958798; a_3 = 0.7478556.$$

A more accurate form with uncertainty $\epsilon \leq 1.5 \times 10^{-7}$ is given by the expression

$$\text{erf}(x) = 1 - (a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5) e^{-x^2}$$

with $t = 1/(1 + 0.3275911x)$

$$\begin{aligned} a_1 &= 0.254829592; a_2 = -0.284496736; a_3 = 0.1.421413741; \\ a_4 &= -1.453152027; a_5 = 2.061405429 \end{aligned}$$

7.2.2 Gamma function

The gamma function is defined in terms of integral,

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

By integrating by parts

$$\Gamma(z+1) = \int_0^\infty e^{-t} t^z dt = -e^{-t} t^z \Big|_0^{+\infty} + z \int_0^\infty e^{-t} t^{z-1} dt = z\Gamma(z)$$

we have a recursion relation,

$$\Gamma(z+1) = z\Gamma(z).$$

Note that $\Gamma(1) = 1$, for a positive integer n, we have $\Gamma(n+1) = n!$.

For a small x, using polynomial approximations,

$$\Gamma(1+x) \approx 1 + \sum_{k=0}^{5} a_k x^k$$

find the coefficients a_k.

x	$\Gamma(1+x)$
0.0	1.0
0.1	0.95135
0.2	0.91816
0.3	0.89747
0.4	0.88726
0.5	0.88622
0.6	0.89351
0.7	0.90863
0.8	0.93138
0.9	0.96176

An efficient way to evaluate gamma functions for large values of the argument is to use Sterling's formula,

$$\Gamma(z) = (2\pi)^{1/2} e^{-z} z^{z-1/2} \{1 + \frac{1}{12z} + \frac{1}{288z^2} - \frac{139}{51840z^3} - \frac{571}{2488320z^4} + \cdots\}$$