A perfectly obvious property of this symbol, that we will use repeatedly, is

$$[x^n]\{x^a f(x)\} = [x^{n-a}]f(x). \tag{1.2.7}$$

Another property of this symbol is the convention that if β is any real number, then

$$[\beta x^n]f(x) = (1/\beta)[x^n]f(x), \tag{1.2.8}$$

so, for instance, $[x^n/n!]e^x = 1$ for all $n \geq 0$.

Before we move on to the next example, here is a summary of the method of generating functions as we have used it so far.

THE METHOD

Given: a recurrence formula that is to be solved by the method of generating functions.

1. Make sure that the set of values of the free variable (say n) for which the given recurrence relation is true, is clearly delineated.
2. Give a name to the generating function that you will look for, and write out that function in terms of the unknown sequence (e.g., call it $A(x)$, and define it to be $\sum_{n\geq 0} a_n x^n$).
3. Multiply both sides of the recurrence by x^n, and sum over all values of n for which the recurrence holds.
4. Express both sides of the resulting equation explicitly in terms of your generating function $A(x)$.
5. Solve the resulting equation for the unknown generating function $A(x)$.
6. If you want an exact formula for the sequence that is defined by the given recurrence relation, then attempt to get such a formula by expanding $A(x)$ into a power series by any method you can think of. In particular, if $A(x)$ is a rational function (quotient of two polynomials), then success will result from expanding in partial fractions and then handling each of the resulting terms separately.

1.3 A three term recurrence

Now let's do the Fibonacci recurrence

$$F_{n+1} = F_n + F_{n-1}. \qquad (n \geq 1; F_0 = 0;\ F_1 = 1). \tag{1.3.1}$$

Following 'The Method,' we will solve for the generating function

$$F(x) = \sum_{n\geq 0} F_n x^n.$$

To do that, multiply (1.3.1) by x^n, and sum over $n \geq 1$. We find on the left side

$$F_2x + F_3x^2 + F_4x^3 + \cdots = \frac{F(x) - x}{x},$$

and on the right side we find

$$\{F_1x + F_2x^2 + F_3x^3 + \cdots\} + \{F_0x + F_1x^2 + F_2x^3 + \cdots\} = \{F(x)\} + \{xF(x)\}.$$

(Important: Try to do the above yourself, without peeking, and see if you get the same answer.) It follows that $(F - x)/x = F + xF$, and therefore that the unknown generating function is now known, and it is

$$F(x) = \frac{x}{1 - x - x^2}.$$

Now we will find some formulas for the Fibonacci numbers by expanding $x/(1 - x - x^2)$ in partial fractions. The success of the partial fraction method is greatly enhanced by having only linear (first degree) factors in the denominator, whereas what we now have is a quadratic factor. So let's factor it further. We find that

$$1 - x - x^2 = (1 - xr_+)(1 - xr_-) \qquad (r_\pm = (1 \pm \sqrt{5})/2)$$

and so

$$\begin{aligned}\frac{x}{1 - x - x^2} &= \frac{x}{(1 - xr_+)(1 - xr_-)} \\ &= \frac{1}{(r_+ - r_-)}\left(\frac{1}{1 - xr_+} - \frac{1}{1 - xr_-}\right) \\ &= \frac{1}{\sqrt{5}}\left\{\sum_{j\geq 0} r_+^j x^j - \sum_{j\geq 0} r_-^j x^j\right\},\end{aligned}$$

thanks to the magic of the geometric series. It is easy to pick out the coefficient of x^n and find

$$F_n = \frac{1}{\sqrt{5}}(r_+^n - r_-^n) \qquad (n = 0, 1, 2, \ldots) \tag{1.3.3}$$

as an explicit formula for the Fibonacci numbers F_n.

This example offers us a chance to edge a little further into what generating functions can tell us about sequences, in that we can get not only the exact answer, but also an approximate answer, valid when n is large. Indeed, when n is large, since $r_+ > 1$ and $|r_-| < 1$, the second term in (1.3.3) will be minuscule compared to the first, so an extremely good approximation to F_n will be

$$F_n \sim \frac{1}{\sqrt{5}}\left(\frac{1 + \sqrt{5}}{2}\right)^n. \tag{1.3.4}$$

But, you may ask, why would anyone want an approximate formula when an exact one is available? One answer, of course, is that sometimes exact answers are fearfully complicated, and approximate ones are more revealing. Even in this case, where the exact answer isn't very complex, we can still learn something from the approximation. The reader should take a few moments to verify that, by neglecting the second term in (1.3.3), we neglect a quantity that is never as large as 0.5 in magnitude, and consequently not only is F_n approximately given by (1.3.4), it is *exactly* equal to the integer nearest to the right side of (1.3.4). Thus consideration of an approximate formula has found us a simpler exact formula!

1.4 A three term boundary value problem

This example will differ from the previous ones in that the recurrence relation involved does not permit the direct calculation of the members of the sequence, although it does determine the sequence uniquely. The situation is similar to the following: suppose we imagine the Fibonacci recurrence, together with the additional data $F_0 = 1$ and $F_{735} = 1$. Well then, the sequence $\{F_n\}$ would be uniquely determined, but you wouldn't be able to compute it directly by recurrence because you would not be in possession of the two consecutive values that are needed to get the recurrence started.

We will consider a slightly more general situation. It consists of the recurrence

$$au_{n+1} + bu_n + cu_{n-1} = d_n \qquad (n = 1, 2, \ldots, N-1; u_0 = u_N = 0) \quad (1.4.1)$$

where the positive integer N, the constants a, b, c and the sequence $\{d_n\}_{n=1}^{N-1}$ are given in advance. The equations (1.4.1) determine the sequence $\{u_i\}_0^N$ uniquely, as we will see, and the method of generating functions gives us a powerful way to attack such boundary value problems as this, which arise in numerous applications, such as the theory of interpolation by spline functions.

To begin with, we will define two generating functions. One of them is our unknown $U(x) = \sum_{j=0}^{N} u_j x^j$, and the second one is $D(x) = \sum_{j=1}^{N-1} d_j x^j$, and it is regarded as a known function (did we omit any given values of the d_j's, like d_0? or d_N? Why?).

Next, following the usual recipe, we multiply the recurrence (1.4.1) by x^n and sum over the values of n for which the recurrence is true, which in this case means that we sum from $n = 1$ to $N-1$. This yields

$$a \sum_{n=1}^{N-1} u_{n+1} x^n + b \sum_{n=1}^{N-1} u_n x^n + c \sum_{n=1}^{N-1} u_{n-1} x^n = \sum_{n=1}^{N-1} d_n x^n.$$

If we now express this equation in terms of our previously defined generating functions, it takes the form

$$\frac{a}{x}\{U(x) - u_1 x\} + bU(x) + cx\{U(x) - u_{N-1}x^{N-1}\} = D(x). \qquad (1.4.2)$$

Next, with only a nagging doubt because u_1 and u_{N-1} are unknown, we press on with the recipe, whose next step asks us to solve (1.4.2) for the unknown generating function $U(x)$. Now that isn't too hard, and we find at once that

$$\{a + bx + cx^2\}U(x) = x\{D(x) + au_1 + cu_{N-1}x^N\}. \tag{1.4.3}$$

The unknown generating function $U(x)$ is now known except for the two still-unknown constants u_1 and u_{N-1}, but (1.4.3) suggests a way to find them, too. There are two values of x, call them r_+ and r_-, at which the quadratic polynomial on the left side of (1.4.3) vanishes. Let us suppose that $r_+^N \neq r_-^N$, for the moment. If we let $x = r_+$ in (1.4.3), we obtain one equation in the two unknowns u_1, u_{N-1}, and if we let $x = r_-$, we get another. The two equations are

$$\begin{aligned} au_1 + (cr_+^N)u_{N-1} &= -D(r_+) \\ au_1 + (cr_-^N)u_{N-1} &= -D(r_-). \end{aligned} \tag{1.4.4}$$

Once these have been solved for u_1 and u_{N-1}, equation (1.4.3) then gives $U(x)$ quite explicitly and completely. We leave the exceptional case where $r_+^N = r_-^N$ to the reader.

Here is an application* of these results to the theory of spline interpolation.

Suppose we are given a table of values $y_0, y_1, \ldots, y_n$ of some function $y(x)$, at a set of equally spaced points $t_i = t_0 + ih$ $(0 \le i \le n)$. We want to construct a smooth function $S(x)$ that fits the data, subject to the following conditions:

(i) Within each interval (t_i, t_{i+1}) $(i = 0, \ldots, n-1)$ our function $S(x)$ is to be a cubic polynomial (a different one in each interval!);

(ii) The functions $S(x)$, $S'(x)$ and $S''(x)$ are to be continuous on the whole interval $[t_0, t_n]$;

(iii) $S(t_i) = y_i$ for $i = 0, \ldots, n$.

A function $S(x)$ that satisfies these conditions is called a *cubic spline.* Suppose our unknown spline $S(x)$ is given by $S_0(x)$, if $x \in [t_0, t_1]$, $S_1(x)$, if $x \in [t_1, t_2]$,…, $S_{n-1}(x)$, if $x \in [t_{n-1}, t_n]$, and we want now to determine all of the cubic polynomials $S_0, \ldots, S_{n-1}$. To do this we have $2n$ interpolatory conditions

$$S_{i-1}(t_i) = y_i = S_i(t_i) \quad (i = 1, \ldots, n-1);\ S_0(t_0) = y_0;\ S_{n-1}(t_n) = y_n \tag{1.4.5}$$

along with $2n - 2$ continuity conditions

$$S'_{i-1}(t_i) = S'_i(t_i);\ S''_{i-1}(t_i) = S''_i(t_i) \qquad (i = 1, \ldots, n-1). \tag{1.4.6}$$

* This application is somewhat specialized, and may be omitted at a first reading.

There are altogether $4n-2$ conditions to satisfy. We have n cubic polynomials to be determined, each of which has 4 coefficients, for a total of $4n$ unknown parameters. Since the conditions are linear, such a spline $S(x)$ surely exists and we can expect it to have two free parameters. It is conventional to choose these so that $S(x)$ has a point of inflection at t_0 and at t_n.

Now here is the solution. The functions $S_i(x)$ are given by

$$S_i(x) = \frac{1}{6h}\big(z_i(t_{i+1}-x)^3 + z_{i+1}(x-t_i)^3 + (6y_{i+1}-h^2z_{i+1})(x-t_i) \\ + (6y_i - h^2 z_i)(t_{i+1}-x)\big) \qquad (i=0,1,\ldots,n-1), \tag{1.4.7}$$

provided that the numbers $z_1,\ldots,z_{n-1}$ satisfy the simultaneous equations

$$z_{i-1}+4z_i+z_{i+1} = \frac{6}{h^2}(y_{i+1}-2y_i+y_{i-1}) \qquad (i=1,2,\ldots,n-1) \tag{1.4.8}$$

in which $z_0 = z_n = 0$. It is easy to check this, by substituting $x = t_i$ and $x = t_{i+1}$ into (1.4.7) to verify that (1.4.5) and (1.4.6) are satisfied. Hence it remains only to solve the equations (1.4.8).

The system of equations (1.4.8) is of the form (1.4.1), hence we can find the solutions from (1.4.3), (1.4.4). To do this, begin with the given set of points $\{(t_i,y_i)\}_{i=0}^n$, through which we wish to interpolate. Use them to write down

$$D(x) = \frac{6}{h^2}\sum_{i=1}^{n-1}(y_{i+1}-2y_i+y_{i-1})x^i. \tag{1.4.9}$$

Since $(a,b,c)=(1,4,1)$ in this example, we have $r_\pm = -2\pm\sqrt{3}$. Now our unknown generating function $U(x)$ is given by (1.4.3), which reads as

$$U(x) = \frac{x(D(x)+z_1+z_{n-1}x^n)}{(1+4x+x^2)}, \tag{1.4.10}$$

in which the unknown numbers z_1, z_{n-1} are determined by the requirement that the right side of (1.4.10) be a polynomial, or equivalently by the two equations (1.4.4), which become

$$\begin{aligned} z_1 + (\sqrt{3}-2)^n z_{n-1} &= -D(\sqrt{3}-2) \\ z_1 + (-\sqrt{3}-2)^n z_{n-1} &= -D(-\sqrt{3}-2). \end{aligned} \tag{1.4.11}$$

When we know $U(x)$, which is, after all, $\sum_{i=1}^{n-1} z_i x^i$, we can read off its coefficients to find the z's, and use them in (1.4.7) to find the interpolating spline.

Example.

Now let's try an example with real live numbers in it. Suppose we are trying to fit the powers of 2 by a cubic spline on the interval $[0, 5]$. Our input data are $y_i = 2^i$ for $i = 0, 1, \ldots, 5$, $h = 1$, and $n = 5$. From (1.4.9) we find that $D(x) = 6x(1+2x+4x^2+8x^3)$. Then we solve (1.4.11) to find that $z_1 = 204/209$ and $z_4 = 2370/209$. Next (1.4.10) tells us that

$$U(x) = \frac{204\,x}{209} + \frac{438\,x^2}{209} + \frac{552\,x^3}{209} + \frac{2370\,x^4}{209},$$

and now we know all of the z_i's. Finally, (1.4.7) tells us the exact cubic polynomials that form the spline. For example, $S_0(x)$, which lives on the subinterval $[0, 1]$, is

$$S_0(x) = 1 + \frac{175}{209}x + \frac{34}{209}x^3.$$

Note that $S_0(0) = 1$ and $S_0(1) = 2$, so it correctly fits the data at the endpoints of its subinterval, and that $S_0''(0) = 0$, so the fit will have an inflection point at the origin. The reader is invited to find all of the $S_i(x)$ $(i = 0, 1, \ldots, 5)$, in this example, and check that they smoothly fit into each other at the points $1, 2, 3, 4$, in the sense that the functions and their first two derivatives are continuous. ■

One reason why you might like to fit some numerical data with a spline is because you want to integrate the function that the data represent. Integration of (1.4.7) from $t_i = ih$ to $t_{i+1} = (i+1)h$ shows that

$$\int_{t_i}^{t_{i+1}} S_i(x)dx = \frac{h}{2}(y_i + y_{i+1}) - \frac{h^3}{24}(z_i + z_{i+1}). \tag{1.4.12}$$

Thus, fitting some data by a spline and integrating the spline amounts to numerical integration by the trapezoidal rule with a third order correction term. If we sum (1.4.12) over $i = 0, \ldots, n-1$ we get for the overall integral,

$$\int_0^{nh} S(x)dx = \text{trap} - \frac{h}{12}(y_0 - y_1 - y_{n-1} + y_n) - \frac{h^3}{72}(z_1 + z_{n-1}) \tag{1.4.13}$$

in which 'trap' is the trapezoidal rule, and z_1, z_{n-1} satisfy (1.4.11).

Interpolation by spline functions is an important subject. It occurs in the storage of computer fonts, such as the one that you are now reading. Did you ever wonder how the shapes of the letters in the fonts are actually stored in a computer? One way is by storing the parameters of spline functions that fit the contours of the letters in the font.