
1. Linear Difference Equations (with Constant Coefficients).

1.1. What a Linear Difference Equation with Constant Coefficients is. By this phrase we shall informally mean a formal equation

\[ a_nX_{i+n} + a_{n-1}X_{i+n-1} + \cdots + a_0X_i + \cdots + a_{-m}X_{i-m} = C_i \]  

(1.1)

where the \(a_j\)'s are constants, \(m\) and \(n\) are (fixed) natural numbers, \(\{C_i\}\) is a sequence whose indices run over all integers \(\geq\) a certain integer, and we want to find a sequence \(\{X_i\}\) whose indices run over all integers \(\geq\) a certain integer, such that equation (1.1) holds for all indices \(i\) for which the equation makes sense. If the sequence \(\{C_i\}\) is the all-zero sequence then (1.1) is called a homogeneous linear difference equation, and if (1.1) is given with \(C_i \neq 0\) then

\[ a_nX_{i+n} + a_{n-1}X_{i+n-1} + \cdots + a_0X_i + \cdots + a_{-m}X_{i-m} = 0 \]

(1.2)

is called the homogeneous linear difference equation associated with equation (1.1). It is intuitively clear that these equations can be solved (for any finite number of terms, as opposed to finding a “closed-form formula for \(X_i\)” ) by recursion, so one assumes that \(a_n \neq 0\)—one can “solve for \(X_{i+n}\)” —and that \(a_{-m} \neq 0\) —one “really is looking back to \(X_{i-m}\).” The order of the equation is then defined to be \(n + m\) for reasons which will be clear when we try to find the generic form of their solutions.

Equations of this kind abound in numerical analysis, e.g., in the context of estimating the size of the error made by iterating an approximation procedure, or in the theory of multi-step ODE solvers. Here is an example from probability theory, arising as a result of conditioning in something involving sequences of trials.\(^{1}\) In his exposition of the “gambler’s ruin” or “drunkard’s/random walk” process, Example 4j, p. 90 ff. in Sheldon Ross, *A First Course in Probability*, 5th ed., Prentice-Hall (1998), ISBN # 0-13-746314-6, Ross lets \(P_i\) denote the probability of winning (“breaking the bank”) on the \(i\)-th random trial (coin flip at \$1 per flip, playing against a “house” with \$\(N\) as capital), knowing that the probability of winning \$1 on an individual trial is \(p\). He conditions on the outcome of the first random trial to obtain the equation

\[ P_i = p P_{i+1} + q P_{i-1} \]  

(1.3.1)

\(^{1}\) The examples are probabilistic because these notes are adapted from some notes used in Math 477. Numerical-analytic examples have to await some numerical analysis! If you are not familiar with the probabilistic context, then look on these as being examples of the kind of difference equations that arise naturally in important mathematical situations.
which is a homogeneous linear difference equation with constant coefficients: this equation is equivalent to

\[ pX_{i+1} - X_i + qX_{i-1} = 0. \]  

It is handy to know that there is a general procedure for finding “closed-form general solutions” for homogeneous equations and some special—but frequently-occurring—inhomogeneous equations (all with constant coefficients). People who know a little about linear ordinary differential equations with constant coefficients can find the first step (b): finding enough solutions of the homogeneous equation to “adjust” the solution we guesswork, although there are some basic families), and then (if the equation was already homogeneous one skips the first step) (b): finding enough solutions of the homogeneous equation to “adjust” the solution we

1.2. Constructing the Solution You Need from Readily Available Parts. The following basic propositions are exactly parallel to the corresponding propositions about linear ordinary differential equations with constant coefficients, and are proved in just about the same way.

**Proposition:** If \( \{X_i\} \) and \( \{Y_i\} \) are two solutions of a given homogeneous linear difference equation with constant coefficients, then for any constant coefficients \( \alpha \) and \( \beta \) the sequence \( \{\alpha X_i + \beta Y_i\} \) is also a solution. If \( \{X_i\} \) is a solution of a given (not-necessarily homogeneous) linear difference equation with constant coefficients and \( \{Y_i\} \) is a solution of the homogeneous linear difference equation associated with it, then the sequence \( \{X_i + Y_i\} \) is also a solution of the given equation.

**Proof.** The whole point is the fact that the expression on the l. h. s. of equation (1.1) is a linear function of the sequence \( \{X_i\} \):

\[ a_n(\alpha X_{i+n} + \beta Y_{i+n}) + \cdots + a_m(\alpha X_{i-m} + \beta Y_{i-m}) \]

\[ = a_n\alpha X_{i+n} + a_n\beta Y_{i+n} + \cdots + a_m\alpha X_{i-m} + a_m\beta Y_{i-m} \]

\[ = \alpha(a_nX_{i+n} + \cdots + a_mX_{i-m}) + \beta(a_nY_{i+n} + \cdots + a_mY_{i-m}). \]  

From this equation it is obvious that both terms on the r. h. s. will be zero if both \( \{X_i\} \) and \( \{Y_i\} \) satisfy the homogeneous equation, and equally obvious (with \( \alpha = \beta = 1 \)) that if \( \{X_i\} \) satisfies equation (1.1) and \( \{Y_i\} \) satisfies the associated homogeneous equation, then the r. h. s. of (1.4) will equal \( C_i + 0 = C_i \), as advertised.

The work of finding the solution one wants of a particular linear difference equation with constant coefficients thus boils down to (a): finding at least one solution of the given equation (possibly by inspired guesswork, although there are some basic families), and then (if the equation was already homogeneous one skips the first step) (b): finding enough solutions of the homogeneous equation to “adjust” the solution we found in (a) to make it conform to other requirements of the problem in which the equation arose.

1.3. Finding the Parts. The most important things one finds are given by

**Proposition:** A homogeneous equation

\[ a_nX_{i+n} + a_{n-1}X_{i+n-1} + \cdots + a_0X_i + \cdots + a_mX_{i-m} = 0 \]

always has solutions of the form \( X_i = \lambda^i \), where \( \lambda \) is a(ny) root of the \( (n+m) \)-th degree polynomial equation

\[ \chi_n(\lambda) \equiv a_n\lambda^{n+m} + a_{n-1}\lambda^{n+m-1} + \cdots + a_m = 0. \]  

**Proof.** If \( \lambda \) is a root of (1.5) and for an arbitrary index \( i \in \mathbb{Z} \) one multiplies (1.5) by \( \lambda^{i-m} \), one gets

\[ a_n\lambda^{i+n} + a_{n-1}\lambda^{i+n-1} + \cdots + a_m\lambda^{i-m} = 0 \]  

which is exactly what one wants. (Note that \( a_m \neq 0 \Rightarrow \lambda \neq 0 \) for roots of (1.5).)
This result should be compared with the well-known method used to find solutions of a homogeneous linear differential equation with constant coefficients, e.g., of formally plugging $e^{rx}$ into

\[
y'' + 6y' + 5y = 0 \\
r^2 e^{rx} + 6r e^{rx} + 5e^{rx} = 0 \\
e^{rx}(r^2 + 6y + 5) = 0 \\
r^2 + 6y + 5 = (r + 5)(r + 1) = 0
\]

yielding $r = -5$ or $r = -1$ to produce two solutions $e^{-x}$ and $e^{-5x}$ of the differential equation, with the hope that the two-parameter family of solutions $y(x) = a e^{-x} + b e^{-5x}$ has enough “degrees of freedom” to enable one to satisfy whatever initial or boundary conditions have been prescribed along with the DE.

Let us try this method on Ross’ homogeneous equation

\[
P_i = p P_{i+1} + q P_{i-1}
\]

above. (It is not necessary to rewrite it in our de facto “standard form”—we’ll just plug in $P_i = \lambda^i$ and get an equivalent equation anyhow.) This plug-in gives

\[
\lambda^i = p \lambda^{i+1} + q \lambda^{i-1}, \tag{1.7.1}
\]

which after dividing out the smallest power of $\lambda$ present and doing a little transposition gives us

\[
p \lambda^2 - \lambda + q = 0. \tag{1.7.2}
\]

Obviously $\lambda = 1$ is a root of this equation, and knowing a root we can factor the l. h. s.: the equation is

\[
p \lambda^2 - \lambda + q \equiv (p \lambda - q)(\lambda - 1) = 0 \tag{1.7.3}
\]

whose roots are $\lambda = 1$ and $\lambda = q/p$ if $p \neq q$ (we shall have to give separate attention to the case $p = q = 1/2$).

Assuming that $p = q = 1/2$ does not hold, we see that the possibilities

\[
P_i = \alpha + \beta (q/p)^i, \quad \alpha, \beta \in \mathbb{R} \tag{1.8}
\]

all represent solutions of (1.3.1). Must every solution of (1.3.1) be of this form (the uniqueness question)?

Yes, because if one knows $P_0$ and $P_1$ then rewriting (1.3.1) in the form

\[
P_{i+1} = \frac{1}{p}P_i - \frac{q}{p}P_{i-1} \tag{1.9}
\]

(the cases $p = 0$ and $p = 1$ can be regarded as already solved, since the probabilities of their outcomes are trivial to determine) shows that if \{P_i\} and \{P_i'\} are two solutions of (1.3.1) that agree at $i = 0$ and $i = 1$—so the difference sequence \{P_i - P_i'\}, which satisfies the same equation, is zero for $i = 0$, 1—then the difference sequence will be zero for all indices, i.e., the sequences will be identical. So some choice of $\alpha$ and $\beta$ will result in the $P_i$’s determined by Ross’ Example 4j, and that fact enables us to skip the complicated telescoping-series argument of his formula (4.4), p. 91, and proceed directly to picking the correct solution out of this two-parameter family.

How does one find $\alpha$ and $\beta$? Use the facts that $P_0 = 0$ and $P_N = 1$. These conditions force

\[
\alpha + \beta \left(\frac{q}{p}\right)^0 = 0 \quad \alpha + \beta \left(\frac{q}{p}\right)^N = 1, \tag{1.10.1}
\]
\[ \alpha + \beta = 0 \quad \alpha + \beta \left( \frac{q}{p} \right)^N = 1 \]  
(1.10.2)

\[ \alpha = \frac{1}{1 - (q/p)^N} \quad \beta = -\frac{1}{1 - (q/p)^N} \]  
(1.10.3)

\[ P_i = \frac{1 - (q/p)^i}{1 - (q/p)^N} \]  
(1.10.4)

in agreement with Ross' formula (4.5), p. 92.\(^{(2)}\)

What about the case where \( p = q = 1/2 \), in which equation (1.7.3) above essentially became \((\lambda - 1)^2 = 0\), with only one (double) root \( \lambda_1 = 1 \)? In analogy with the case of differential equations, it can be shown that if a root \( \lambda_j \) of equation (1.5) has multiplicity \( n_j > 1 \)—so that \((\lambda - \lambda_j)^{n_j}\) divides the l. h. s. of equation (1.5), but no higher power of \((\lambda - \lambda_j)\) does—then the sequences \( \{\lambda_j^i\}_{i=0}^\infty \), \( \{i \cdot \lambda_j^{i-1}\}_{i=0}^\infty \), \( \{i(i-1) \cdot \lambda_j^{i-2}\}_{i=0}^\infty \), \( \{i(i-1)(i-n_j+1) \cdot \lambda_j^{i-n_j}\}_{i=0}^\infty \) will all satisfy the homogeneous difference equation.\(^{(3)}\) In this case, if \( p = q = 1/2 \) it is immediate that the sequence whose \( i \)-th term is \( i(= i \cdot 1^i) \) satisfies the difference equation: plug \( i \) in for \( P_i \) in \( P_i = (1/2) P_{i+1} + (1/2) P_{i-1} \). If we want the sequence \( \{\alpha + \beta i\} \) to satisfy the “boundary conditions” \( P_0 = 0 \) and \( P_N = 1 \) we are led immediately to

\[ \alpha + \beta \cdot 0 = 0 \quad \alpha + \beta \cdot N = 1 , \quad \alpha = 0 \quad \beta = \frac{1}{N} \quad \Rightarrow \quad P_i = \frac{i}{N} , \]  
(1.10.5)

also in agreement with Ross' formula (4.5), p. 92.\(^{(4)}\)

An example of a case in which the equation is not homogeneous is given by Ross’ Theoretical Exercise 15, p. 120, which leads to the inhomogeneous linear difference equation (we replace the index \( n \) by \( i \) for uniformity of notation)

\[ P_i = p (1 - P_{i-1}) + (1 - p) P_{i-1} \]

\[ P_i + (2p - 1) P_{i-1} = p . \]  
(1.11)

One can usually find a “particular solution” of a constant-coefficient linear difference equation whose r. h. s. is a polynomial in the index by seeking a polynomial in the index (if the r. h. s. is the \( i \)-th power of some constant \( r \), one tries for a polynomial in the index times the \( i \)-th power of \( r \), etc.) of the same or lower degree. This suggests we try to find a constant \( c \) that satisfies (1.11); plugging in \( c \) for both \( P_i \) and \( P_{i-1} \) in (1.11) and solving gives \( c = 1/2 \). A solution of the homogeneous equation associated with (1.11) is given by the powers of \( \lambda \) where \( \lambda + (2p - 1) = 0 \), or \( \lambda = (1 - 2p) \). Just as with linear differential equations, it is easy to see that adding multiples of the solution of the homogeneous equation to a particular solution of the inhomogeneous equation gives all possible solutions of the inhomogeneous equation. So the solution of (1.11) must have the form

\[ P_i = \frac{1}{2} + \alpha (1 - 2p)^i \quad \text{with} \]

\[ 1 - p = P_i = \frac{1}{2} + \alpha \cdot (1 - 2p) \quad \alpha \cdot (1 - 2p) = \frac{1}{2} - p \]

\[ \alpha = \frac{(1/2) - p}{1 - 2p} = \frac{1}{2} \]

\[ P_i = \frac{1 + (1 - 2p)^i}{2} . \]  
(1.12)

\(^{(2)}\) Note that we have just solved the discrete analogue of a boundary-value problem for an ordinary differential equation.

\(^{(3)}\) We shall verify this in §1.4 below, but using a non-elementary argument.

\(^{(4)}\) Detailed rigorous justification of these statements will be given below, but people may take this on faith.
1.4. A Transcendental Approach to a Basically Algebraic Problem. The solution of concrete linear difference equations over \( \mathbb{C} \) can be automated—modulo finding some numbers—and the method of automation also gives proofs of some of the bald assertions made in the preceding §§. We need to make some preliminary observations. First of all, by making a simple shift of indices the homogeneous equation (1.2) can always be “normalized” to replace “\( m \)” by zero and “\( n \)” by “\( n + m \)” (the reader can easily check the details), and (by dividing by the given \( a_n \) if necessary) one can make the further normalization that \( a_n = 1 \) (recall that we had previously assumed also that \( a_0 \neq 0 \)); these normalizations do not make any essential changes in the set of sequences satisfying the difference equation. Thus we may restrict our consideration of homogeneous equations to ones having the form

\[
X_{i+n} + a_{n-1} X_{i+n-1} + \cdots + a_0 X_i = 0
\]

(1.13)

with the initial value of the sequence index \( i \) being zero. Second, we can observe that the set of sequences \( \{X_i\}_{i=0}^{\infty} \) satisfying (1.13) is a vector space over \( \mathbb{C} \) under index-by-index addition and scalar multiplication, and indeed that it is a space of dimension \( n \). The assertion about dimension is proved by establishing an isomorphism of vector spaces; the important step is a simple mathematical induction on the index. The correspondence

\[
\mathbb{C}^n \ni \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{n-1} \end{bmatrix} \leftrightarrow \{X_i\}_{i=0}^{\infty}
\]

(1.14)

is obviously linear, with \( \leftrightarrow \) mapping onto \( \mathbb{C}^n \) and \( \rightarrow \) being 1-1, and induction on the running index of the sequence beginning at \( i = n - 1 \) shows that \( \rightarrow \) is onto and \( \leftrightarrow \) is 1-1. (The reader should work through the details.) Third, the rate-of-growth with the running index \( i \) of any solution \( \{X_i\}_{i=0}^{\infty} \) of (1.13)

is at most geometric. To make a very crude but adequate estimate: if we denote the vector

\[
\begin{bmatrix} X_i \\ X_{i+1} \\ \vdots \\ X_{i+n-1} \end{bmatrix}
\]

by \( X^{(i)} \)—so that \( X^{(0)} \) is the vector of the \( n \) initial values of \( X_1 \)—and put \( \|a\|_1 = \sum_{k=0}^{n-1} |a_k| \), then clearly

\[
\|X^{(i+1)}\|_\infty \leq (\|a\|_1 + 1) \cdot \|X^{(i)}\|_\infty,
\]

where as usual the “infinity norm” of a vector is the largest of the absolute values of its components. An easy induction shows that one must have \(|X_i| \leq (\|a\|_1 + 1)^{i-n+1} \cdot \|X^{(0)}\|_\infty\) for \( i \geq n \). In consequence, the power series in the indeterminate \( 1/\lambda \)

\[
F_X(\lambda) = \sum_{k=0}^{\infty} \frac{X_k}{\lambda^k}
\]

(1.15)

converges in the “punctured disc centered on \( \infty \)” of the form \( \{\lambda \in \mathbb{C} : |\lambda| > \|a\|_1 + 1\} \). By the uniqueness theorem for Laurent series, the correspondence \( F_X(\lambda) \leftrightarrow \{X_i\}_{i=0}^{\infty} \) is 1-1, and it is obviously linear. The function \( F_X \) is called\(^{(5)} \) the Z-transform of the sequence \( \{X_i\}_{i=0}^{\infty} \).

It seems unlikely that every power series in \( 1/\lambda \) converging in a neighborhood of \( \infty \) is the Z-transform of a solution sequence of some linear homogeneous difference equation—although it will follow from the results

---

\(^{(5)} \) This is a concession to our electrical-engineering brethren; mathematicians would call it the “generating function of the sequence” if \( 1/\lambda \) were replaced by \( \lambda \). In fact, in this context there are technical advantages in “working around \( \infty \)” instead of working around zero; the EEs have found these advantages and exploited them skillfully. The “Z” in “Z-transform” comes from the usual name of the usual indeterminate in power series; as we shall see, the name “\( \lambda \)” is very natural from the standpoint of the usual mathematical onomastic conventions.
given below that the linear space of these Z-transforms consists exactly of the rational functions on \( \mathbb{C} \mathbb{P}^1 \) that are finite at \( \infty \) and zero at \( 0 \). However, the Z-transform of any sequence that is a solution sequence of a linear homogeneous difference equation must be such a rational function, and in fact the denominator of the rational function turns out to be the polynomial \( \chi_n(\lambda) \) on the r. h. s of (1.5) above. To see this, we need only multiply \( F_X(\lambda) \) by that polynomial and watch what happens. This is most perspicaciously done term-by-term (the normalization \( a_n = 1 \) is unnecessary for this computation):

\[
\begin{align*}
a_0 F_X(\lambda) &= \sum_{k=0}^{\infty} \frac{1}{\lambda^k} a_0 X_k \\
a_1 \lambda F_X(\lambda) &= a_1 \lambda X_0 + \sum_{k=0}^{\infty} \frac{1}{\lambda^k} a_1 X_{k+1} \\
a_2 \lambda^2 F_X(\lambda) &= a_2 \lambda^2 X_0 + a_2 \lambda X_1 + \sum_{k=0}^{\infty} \frac{1}{\lambda^k} a_2 X_{k+2} \\
&\vdots \\
a_n \lambda^n F_X(\lambda) &= a_n \lambda^n X_0 + a_n \lambda^{n-1} X_1 + \cdots + a_n \lambda^1 X_{n-1} + \sum_{k=0}^{\infty} \frac{1}{\lambda^k} a_n X_{k+n}
\end{align*}
\]

where the next-to-last displayed equation was obtained by adding its predecessors, and the infinite series disappears from the last equation because the fact that \( \{X_k\}_{k=0}^{\infty} \) satisfies the difference equation has made all the coefficients of that infinite series equal zero.

The problem of finding an explicit solution of (1.13) with prescribed initial values \( X_0, \ldots, X_{n-1} \) is thus seen to be logically equivalent to that of expanding the r. h. s of (1.16) into partial fractions. By what everybody knows about expanding rational functions into the sums of their principal parts,\(^6\), if the linear factorization of the denominator is \( \chi_n(\lambda) = (\lambda - \lambda_1)^{n_1} \cdots (\lambda - \lambda_\ell)^{n_\ell} \), then the partial fractions expansion will consist of a finite linear combination of terms of the form \( \frac{1}{\lambda - \lambda_j} \), \( \cdots \), \( \frac{1}{(\lambda - \lambda_j)^{n_j}} \) for \( j = 1, \ldots, \ell \). (There is no polynomial term since \( F_X(\lambda) \) is finite at \( \infty \).) Since

\[
\frac{1}{(\lambda - \lambda_j)^q} = \frac{1}{\lambda^q} \left( \frac{1}{1 - \frac{\lambda_j}{\lambda}} \right)^q = \frac{1}{(q-1)! \lambda_j^q} \sum_{i=q}^{\infty} (i-1)(i-2) \cdots (i-q+1) \lambda_j^i
\]

we see that the solutions listed on p. 4 above are all obtainable by the Z-transform technique (up to constant factors) as solutions of (1.13); moreover, all such Z-transforms are linear combinations of series of the form (1.17). The count of the distinct solutions of this form is exactly \( \sum n_j = n \), and their linear independence follows from the linear independence of the functions \( \frac{1}{(\lambda - \lambda_j)^q} \) to which they correspond under the Z-transform. It follows that these solutions must in fact form a basis of the \( n \)-dimensional linear space of solutions of (1.13), as asserted above.

\(^6\) See, for example, L. V. Ahlfors, \textit{Complex Analysis}, 3rd ed., McGraw-Hill (1979), pp. 31–32—or realize that what you learned in second-semester calculus works just as well over the field \( \mathbb{C} \).
Since the Z-transform technique for homogeneous linear difference equations is finitistic—apart from the problems of finding the zeros of \( \chi_\lambda(\lambda) \)—it is perhaps not surprising that it has been adapted for symbolic computation. The built-in functions \( \text{ztrans}(\cdot, \cdot) \) and \( \text{iztrans}(\cdot, \cdot) \) implement the direct and inverse Z-transforms in the Symbolic Math Toolbox in the student version of Matlab.\(^7\) For example, the third-order recursion \( 2X_{i+3} - 5X_{i+2} + 4X_{i+1} - X_i = 0 \) with characteristic polynomial \( \chi_\lambda(\lambda) = (\lambda - 1/2)(\lambda - 1)^2 \) would have as a basis of its solution space the solutions whose Z-transforms are \( \frac{\lambda}{\chi_\lambda(\lambda)} \frac{\lambda^2}{\chi_\lambda(\lambda)} \) and \( \frac{\lambda^3}{\chi_\lambda(\lambda)} \). The symbolic routine \( \text{iztrans}(\cdot, \cdot) \) gave as the corresponding solutions of the recursion \( X_i = \frac{4}{2^i}, X_i = \frac{4}{2^i} - 2 + 2i \) and \( X_i = \frac{1}{2^i} + 2i \), in which the basic solutions \( X_i = 1, X_i = i \) and \( X_i = \frac{1}{2^i} \) are easy to spot.

It is not difficult to reverse the steps above and show that every rational function on \( \mathbb{CP}^1 \) of the form (in lowest terms) \( \frac{p(\lambda)}{q(\lambda)} \), where \( \deg p \leq \deg q = n \), \( q(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_0 = 0 \), \( q(0) = a_0 \neq 0 \) and \( p(0) = 0 \), is the Z-transform of a sequence \( \{X_i\}_{i=0}^\infty \) satisfying the linear homogeneous difference equation \( X_{i+n} + a_{n-1}X_{i+n-1} + \cdots + a_0X_i = 0 \) for a suitable choice of initial \( n \) values \( X_0, \ldots, X_{n-1} \). This can safely be left as an exercise for the interested reader.\(^8\)

The form of the elements of the basis \( \{\lambda_j^i\}_{i=0}^\infty \cdot \{i \cdot \lambda_j^i\}_{i=0}^\infty \cdot \{i(i-1) \cdot \lambda_j^i\}_{i=0}^\infty \cdot \ldots \cdot \{i(i-1) \cdots (i-n_j + 2) \cdot \lambda_j^i\}_{i=0}^\infty \) of solutions of (1.13), however, implies the following proposition, which is extremely important in the construction of multi-step numerical ODE solvers:

**Proposition** [Boundedness and Stability Conditions]: In order for all the solution sequences of the linear difference equation (1.13) to be (uniformly) bounded sequences, it is necessary and sufficient that all the roots \( \lambda_j \) of the polynomial \( \chi_\lambda(\lambda) \) satisfy \( |\lambda_j|\leq 1 \), and that each root \( \lambda_j \) with \( |\lambda_j| = 1 \) have multiplicity \( n_j = 1 \) (i.e., that \( \lambda - \lambda_j \) divide \( \chi_\lambda(\lambda) \) but \( \lambda - \lambda_j)^2 \) not divide it). In order for all the solution sequences of the linear difference equation (1.13) to be sequences that tend to zero with increasing index, it is necessary and sufficient that all the roots \( \lambda_j \) of the polynomial \( \chi_\lambda(\lambda) \) satisfy \( |\lambda_j| < 1 \).

Indeed, by looking at the basic sequences we see that if a root \( \lambda_j \) of the polynomial \( \chi_\lambda(\lambda) \) has \( |\lambda_j| > 1 \), then the corresponding basic sequences become unbounded—exponentially fast—as the index increases to \( \infty \). Similarly, we see that if a root \( \lambda_j \) of the polynomial \( \chi_\lambda(\lambda) \) has \( |\lambda_j| = 1 \) but \( n_j \geq 2 \), then the absolute value of the corresponding basic sequence \( \{i \cdot \lambda_j^i\}_{i=0}^\infty \) equals the index and hence tends to \( \infty \). On the other hand, if the condition of the proposition is satisfied, then all the basic sequences corresponding to roots with \( |\lambda_j| < 1 \) have the form \( p(i) \cdot \lambda_j^i \), where \( p(i) \) is a polynomial, and it is well known and easy to verify (L’Hôpital’s rule) that such a sequence tends to zero as \( i \to \infty \), while the only basic sequence corresponding to a root with \( |\lambda_j| = 1 \) will be the sequence \( \{\lambda_j^i\}_{i=0}^\infty \) whose absolute value remains 1 for all values of the index. The condition for solution sequences to tend to zero is established with an analogous argument.

The method just employed can be adapted to handle inhomogeneous linear equations of the form

\[
X_{k+n} + a_{n-1}X_{k+n-1} + \cdots + a_0X_k = C_k
\]

provided that the r. h. sequence \( \{C_i\}_{i=0}^\infty \) “does not grow too fast.” Specifically, one wants it to grow no faster than geometrically—\( \{C_k\} \leq \rho^k \) for some \( \rho > 0 \)—since that will guarantee that its formal Z-transform series

\[
F_C(\lambda) = \sum_{k=0}^\infty \frac{C_k}{\lambda^k}
\]

converges in some neighborhood \( \{\lambda : |\lambda| > \rho\} \) of infinity. Repeating the arguments that led to the penultimate line of (1.16) above leads to the same result, but the next line is different because the r. h. s. of the difference equation (1.18) is not (in general) zero, but rather \( C_k \):

---

\(^7\) Maple has similar routines—indeed, the Matlab routines are probably borrowed from Maple—and people who prefer to use Maple for concrete problems might want to read the appropriate manuals or help-files. There is some very sketchy information in André Heck, *Introduction to Maple*, 2nd ed., Springer-Verlag (1996), p. 511 ff.

\(^8\) For fixed \( q(\lambda) \), this can be phrased in terms of line bundles over \( \mathbb{CP}^1 \). If that’s the sort of thing you want to do, you are definitely in the wrong course.
where exceptions) to the case in which the matrix solutions span the solution space of (1.20), and the uniqueness of the expansion is even more obvious.) For \( A X \) solution space. (To see this, expand any given \( \lambda \) e.g. for theoretical purposes, however, it is possible to use (1.19) to make estimates. In concrete cases, if its inverse Z-transform can be found—e.g., if it is a rational function which is zero at \( \lambda = 0 \) (its finiteness at \( \infty \) being automatic)—then one has solved (1.18) by a Z-transform technique. Even for theoretical purposes, however, it is possible to use (1.19) to make estimates.\(^{(9)}\)

The Matlab routines \texttt{ztrans} and \texttt{iztrans} handle the nonhomogeneous equation (1.11) perfectly well: with \( p \) a symbolic constant and with the initial value \( P_0 = 1 \) prescribed for the solution of (1.11), one first uses \texttt{ztrans} to return \( FC(z) = p z / (z - 1) \); then, setting \( FX(z) = (z + FC(z)) / (z - (1 - 2p)) \), one invokes \texttt{iztrans} to get the expression \( 1/2 + 1/2\cdot(1 - 2p)^k \) for \( P_k \), exactly what we found by elementary methods in (1.12) above.

1.5. Systems of Difference Equations. What was done with scalar sequences above can also be done with vectors. A \textbf{first-order linear homogeneous system of difference equations} (with constant coefficients) has the form

\[
X^{(i+1)} = AX^{(i)}
\]  

(1.20)

where \( A \) is an \( n \times n \) matrix over a field \( \mathbb{K} \) (typically \( \mathbb{R} \), \( \mathbb{C} \) or a finite field) and a \textbf{solution} of the system is (naturally) a sequence \( \{X^{(i)}\}_{i=k}^{\infty} \) in \( \mathbb{K}^n \) that satisfies the equation (or system) (1.20) for \( i = k, \ldots \). With no significant loss of generality one can restrict one’s attention to the case in which the initial index \( k \) equals zero. It is obvious—the consequence of a trivial\(^{(10)}\) induction on the index \( i \)—that any recursion of the form \( X^{(i+1)} = f(X^{(i)}) \), where the function \( f : \mathbb{K}^n \rightarrow \mathbb{K}^n \) is prescribed, determines a solution (sequence) \( \{X^{(i)}\}_{i=0}^{\infty} \) for any prescribed initial value \( X^{(0)} \in \mathbb{K}^n \) and that the solution sequence is uniquely determined by the initial value. In particular, the sequence \( X^{(i)} = A^j X^{(0)} \) is a solution of the system (1.20) and is the only solution that has the given value at \( i = 0 \). The set of all solution sequences of (1.20) forms a linear space—the \textbf{solution space} of (1.20)—over \( \mathbb{K} \) under index-by-index addition and scalar multiplication, and if \( \{F_j\}_{j=1}^{n} \subseteq \mathbb{K}^n \) is a(ny) basis of \( \mathbb{K}^n \), then the solutions \( \{A^j F_j\}_{j=1}^{n} \) of (1.20) form a basis for the solution space. (To see this, expand any given \( X^{(0)} \) into \( \sum_{j=1}^{n} \alpha_j F_j \); then \( A^j X^{(0)} = \sum_{j=1}^{n} \alpha_j A^j F_j \), so those solutions span the solution space of (1.20), and the uniqueness of the expansion is even more obvious.) For reasons that will be clearer below, we shall restrict our considerations (with a very few, explicitly labeled exceptions) to the case in which the matrix \( A \) is invertible.

If \( A \) is diagonalizable, so that there is a basis \( \{F_j\}_{j=1}^{n} \subseteq \mathbb{K}^n \) of \( \mathbb{K}^n \) for which \( AF_j = \lambda_j F_j \), then clearly \( A^j F_j = \lambda_j^j F_j \), the sequence \( \{\lambda_j^j F_j\}_{j=0}^{\infty} \) is a solution of the difference equation, and these sequences (for \( j = 1, \ldots, n \)) form a basis of the solution space of (1.20). This is true even if \( A \) fails to be invertible (so long as it is diagonalizable) since if \( AF_j = 0 \) then the sequence \( \{F_j, 0, \ldots, 0, \ldots\} \) is a solution of the difference equation. These solutions of (1.20) should be compared with the solutions of (1.2) discussed at the beginning of §1.3 above. The problem of finding explicit “closed-form formulas” for a basis of solutions of (1.20) when \( A \) is diagonalizable is thus equivalent in difficulty to the problem of finding a complete list of eigenvalues of \( A \) and a basis consisting of eigenvectors.\(^{(11)}\)

\(^{(9)}\) One approach to finding the (irrational) exponent of convergence in the secant and Müller methods of root-finding employs the Z-transform technique.

\(^{(10)}\) Purists may protest that one has to know something about the set-theoretic underpinnings of constructing sequences by induction. These purists generally know exactly what those underpinnings are, and they are competent to see to the details.

\(^{(11)}\) Again, this should be reminiscent of the corresponding problem for first-order linear systems of ordinary differential equations.
A first-order linear inhomogeneous system of difference equations (with constant coefficients) then has the expected form

\[ X^{(i+1)} = AX^{(i)} + G^{(i)} \]  \hspace{1cm} (1.21)

where \( A \) is an \( n \times n \) matrix over a field \( \mathbb{K} \) (typically \( \mathbb{R}, \mathbb{C} \) or a finite field) and \( \{G^{(i)}\}_{i=0}^{\infty} \) is a prescribed sequence (again, we are restricting consideration to the case in which the “initial index” \( k = 0 \)). Again, a solution of the system is a sequence \( \{X^{(i)}\}_{i=0}^{\infty} \) in \( \mathbb{K}^n \) that satisfies the equation (or system) (1.21) for \( i = 0, \ldots \). The associated homogeneous equation corresponding to (1.21) is, of course, just (1.20) (assuming it’s the same matrix of coefficients \( A \)). It is routine to verify that the (index-wise) difference of two solutions \( \{X^{(i)}\}_{i=0}^{\infty} \) and \( \{Z^{(i)}\}_{i=0}^{\infty} \) of (1.21) satisfies (1.20), and therefore—just as in the familiar case of differential equations—we can reduce the problem of solving (1.21) to two simpler problems: finding some particular solution of (1.21), and finding the general solution of (1.20). Thus if \( \{Z^{(i)}\}_{i=0}^{\infty} \) is a particular solution of (1.21), then the “general solution of (1.21)” is the set of sequences of the form

\[ X^{(i)} = A^iY^{(0)} + Z^{(i)} \]  \hspace{1cm} (1.22)

since by choosing \( Y^{(0)} = X^{(0)} - Z^{(0)} \) we obtain a solution of (1.21) that takes the prescribed value \( X^{(0)} \) at the zero index. The set of solutions is not a linear space, but rather is a translate in the space of all \( \mathbb{K}^n \)-valued sequences (by one, or by any, solution of (1.21)) of the solution space of (1.20).

It is possible to find a particular solution of (1.21) by a technique similar to the “integrating factor” technique familiar in solving ordinary differential equations. If one seeks a solution in the form \( Z^{(i)} = A^iW^{(i)} \), then (1.21) becomes (formally assuming the invertibility of \( A \) and setting \( W^{(0)} = 0 \)) the equation:

\[
\begin{align*}
A^{i+1}W^{(i+1)} &= AA^iW^{(i)} + G^{(i)} \quad \text{(1.23a)} \\
W^{(i+1)} &= W^{(i)} + A^{-(i+1)}G^{(i)} \quad \text{(1.23b)} \\
W^{(i+1)} - W^{(i)} &= A^{-(i+1)}G^{(i)} \quad \text{(1.23c)} \\
W^{(i)} &= \sum_{j=0}^{i-1}[W^{(j+1)} - W^{(j)}] = \sum_{j=0}^{i-1}A^{-(j+1)}G^{(j)} \quad \text{(1.23d)} \\
Z^{(i)} &= A^iW^{(i)} = \sum_{j=0}^{i-1}A^{i-j-1}G^{(j)} \quad \text{for } i \geq 1, \quad \text{(1.23e)}
\end{align*}
\]

where of course the “empty sum” \( Z^{(0)} = 0 \); and \textit{a posteriori} checking will show that even if \( A \) is not invertible, the sequence given by (1.23e) satisfies (1.21).\(^{(12)}\)

An important case of (1.21) is the one in which the “driving function” \( \{G^{(i)}\}_{i=0}^{\infty} \) is a constant sequence with value \( B \). In that case, (1.23e) above takes the form

\[
Z^{(i)} = \sum_{j=0}^{i-1}A^{i-j-1}B = \left[ \sum_{j=0}^{i-1}A^{i-j-1} \right] B \quad \text{for } i \geq 1 \quad \text{(1.24a)}
\]

and the formal finite geometric series inside the square brackets can be summed to yield

\[
Z^{(i)} = (I - A^i)(I - A)^{-1}B \quad \text{for } i \geq 1 \quad \text{(1.24b)}
\]

as a particular solution of (1.21),\(^{(13)}\) under the assumption that the “resolvent” \( (I - A)^{-1} \) exists. Indeed, if the equation \( (I - A)C = B \) admits a solution \( C \) (for whose existence the invertibility of \( I - A \) is sufficient but not necessary), it is easy to check that the sequence defined by \( Z^{(i)} = (I - A^i)C \) is a solution of (1.21).

\(^{(12)}\) This is the discrete analogue of the “convolution” or “impulse-function” solution of a linear ODE with constant coefficients.

\(^{(13)}\) The interested reader might try to compute the corresponding solution of an ODE of the form \( y' = ay + b \), where \( b \) is constant.
The Z-transform approach works even more smoothly in this context than it did in the setting of §1.4 above. In the homogeneous-equation case, the equation (1.20) gives \( \|X^{(i)}\| \leq \|A\| \cdot \|X^{(0)}\| \) in any norm \(^{(14)}\) on \( \mathbb{C}^n \) and its associated matrix norm (respectively), so an easy induction gives \( \|X^{(k)}\| \leq \|A\|^k \cdot \|X^{(0)}\| \) and insures the convergence of the \( \mathbb{C}^n \)-valued Z-transform series \( F_X(\lambda) = \sum_{k=0}^{\infty} \frac{1}{\lambda^k} X^{(k)} \) for \( |\lambda| > \|A\| \). Now since we have already found out that \( X^{(i)} = A^i X^{(0)} \), we have in fact

\[
F_X(\lambda) = \sum_{k=0}^{\infty} \frac{1}{\lambda^k} A^k X^{(0)} = \lambda(\lambda I - A)^{-1} X^{(0)} \tag{1.25}
\]

since the geometric series \( (\lambda I - A)^{-1} = \sum_{k=0}^{\infty} \frac{1}{\lambda^{k+1}} A^k \) is just the well-known C. Neumann series for the matrix-valued “resolvent” function \( (\lambda I - A)^{-1} \) near \( \infty \). The classical determinant formula for the inverse of a matrix shows that the matrix-valued function \( \lambda(\lambda I - A)^{-1} = \frac{\lambda}{\det(\lambda I - A)} \text{adj}(\lambda I - A) \), where “\( \text{adj}(\cdot) \)” denotes the classical adjoint,\(^{(16)}\) is a matrix whose entries are rational functions of \( \lambda \) with numerators of degree at most \( n \) and value zero at \( \lambda = 0 \), and with the denominators of all entries equal to \( \det(\lambda I - A) = \chi_A(\lambda) \) (which has no zero at \( \lambda = 0 \) since \( A \) was assumed invertible). It follows that each of the coordinate functions of the vector-valued function on the r. h. s. of (1.25)—which is, after all, just a way of writing \( F_X(\lambda) \)—is exactly the kind of function that we worked with in the case of a single linear homogeneous equation in §1.4 above. For concrete equations, this fact implies that Z-transforming a system coordinatewise using a symbolic manipulation program (that is capable of inverting matrices with symbolic entries) will extract the Z-transforms of the concrete solution sequences corresponding to concrete initial-condition vectors. Some coordinate-by-coordinate hand work may have to be done or subroutines written to get back from the Z-transforms of vector- or matrix-valued sequences to the sequences themselves: e. g., the \texttt{Matlab} routine \texttt{iztrans} seems to get indigestion when it is asked to process matrix-valued functions. However, the choice

\[
A = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}
\]

gave the resolvent kernel

\[
\frac{1}{(\lambda - 1)^2} \begin{bmatrix} \lambda(\lambda - 2) & \lambda \\ -\lambda & \chi^2 \end{bmatrix}
\]

by symbolic matrix inversion, and then entry-by-entry application of \texttt{iztrans} resulted in the matrix sequence \( \left\{ \begin{bmatrix} 1 - k & k \\ -k & 1 + k \end{bmatrix} \right\}_{k=0}^{\infty} \). As the reader can easily verify after the fact, the first and second columns of this matrix sequence give vector sequence solutions of the system of difference equations (1.20) (with this choice of \( A \)) whose values for \( k = 0 \) are the first and second standard basis vectors of \( \mathbb{C}^2 \) respectively, and this is exactly how one would expect things to turn out.

If the “driving sequence” \( \{G^{(k)}\}_{k=0}^{\infty} \) of (1.21) above satisfies an estimate of the form \( \|G^{(k)}\| \leq \rho^k \), so that the formal Z-transform series \( F_G(\lambda) = \sum_{k=0}^{\infty} \frac{G^{(k)}}{\lambda^k} \) converges for \( |\lambda| > \rho \), then one can Z-transform both sides of (1.21) to give

\[
\sum_{k=0}^{\infty} \frac{X^{(k+1)}}{\lambda^k} = \sum_{k=0}^{\infty} \frac{AX^{(k)}}{\lambda^k} + \sum_{k=0}^{\infty} \frac{G^{(k)}}{\lambda^k}
\]

\[
\lambda(F_X(\lambda) - X^{(0)}) = AF_X(\lambda) + F_G(\lambda)
\]

\[
(\lambda I - A)F_X(\lambda) = \lambda X^{(0)} + F_G(\lambda)
\]

\[
F_X(\lambda) = \lambda(\lambda I - A)^{-1} X^{(0)} + (\lambda I - A)^{-1} F_G(\lambda).
\tag{1.26}
\]

\(^{(14)}\) For brief introductions to norms on the usual coordinate spaces, see Atkinson, pp. 9–11 or (in greater detail) pp. 480–490; or see the relevant part (beginning on p. 8) of the notes on linear algebra for this course.

\(^{(15)}\) This matrix \(\lambda(\lambda I - A)^{-1} \) is sometimes called the resolvent kernel to distinguish it from the usual resolvent familiar to functional analysts. The origin of this name comes from probabilistic potential theory and can be disregarded in this course.

\(^{(16)}\) That is, the transposed matrix of cofactors of the given matrix.
This follows the pattern set in (1.19): the first term on the r. h. s. is the Z-transform of a solution of the homogeneous equation (1.20), and the second (which represents the case \(X^{(0)} = 0\)) is a particular solution of (1.21). If the coördinate functions of \(F_G(\lambda)\) are rational functions finite at \(\infty\) and zero at zero, then in concrete cases one can extract concrete solutions by taking the inverse Z-transform of each side of (1.26). The formal series computation in a neighborhood of \(\infty\)

\[
(\lambda I - A)^{-1}F_G(\lambda) = \left[\sum_{k=1}^{\infty} \frac{A^{k-1}}{\lambda^k}\right] \left[\sum_{j=0}^{\infty} \frac{G^{(j)}}{\lambda^j}\right] = \sum_{i=1}^{\infty} \left[\sum_{j=0}^{i-1} A^{-j-1}G^{(j)}\right] \frac{1}{\lambda^i}
\]

(1.27)

then shows that the corresponding sequence \(Z^{(i)} = \sum_{j=0}^{i-1} A^{-j-1}G^{(j)}\) is a particular solution of (1.21), in agreement with (1.23e); but (1.23e) is more general, in that it holds without assuming a growth condition on \(\|G^{(i)}\|\).

The remainder of this § requires a knowledge of the algebra of a single linear transformation in one of the following forms: (a) the algebra of modules over the polynomial ring \(\mathbb{C}[\lambda]\); (b) the Jordan canonical form; (c) the F. Riesz theory of the Laurent series of an operator at poles of the resolvent. Its reasoning may be omitted by anyone who hasn’t seen that material; however, those persons must take its conclusions on faith. (c) the F. Riesz theory of the Laurent series of an operator at poles of the resolvent. Its reasoning may be

The remainder of this § requires a knowledge of the algebra of a single linear transformation in one of the following forms: (a) the algebra of modules over the polynomial ring \(\mathbb{C}[\lambda]\); (b) the Jordan canonical form; (c) the F. Riesz theory of the Laurent series of an operator at poles of the resolvent. Its reasoning may be

The remainder of this § requires a knowledge of the algebra of a single linear transformation in one of the following forms: (a) the algebra of modules over the polynomial ring \(\mathbb{C}[\lambda]\); (b) the Jordan canonical form; (c) the F. Riesz theory of the Laurent series of an operator at poles of the resolvent. Its reasoning may be

The remainder of this § requires a knowledge of the algebra of a single linear transformation in one of the following forms: (a) the algebra of modules over the polynomial ring \(\mathbb{C}[\lambda]\); (b) the Jordan canonical form; (c) the F. Riesz theory of the Laurent series of an operator at poles of the resolvent. Its reasoning may be

The remainder of this § requires a knowledge of the algebra of a single linear transformation in one of the following forms: (a) the algebra of modules over the polynomial ring \(\mathbb{C}[\lambda]\); (b) the Jordan canonical form; (c) the F. Riesz theory of the Laurent series of an operator at poles of the resolvent. Its reasoning may be

The remainder of this § requires a knowledge of the algebra of a single linear transformation in one of the following forms: (a) the algebra of modules over the polynomial ring \(\mathbb{C}[\lambda]\); (b) the Jordan canonical form; (c) the F. Riesz theory of the Laurent series of an operator at poles of the resolvent. Its reasoning may be
corresponding to a column of \(\lambda (\lambda I - A)^{-1}\) with an entry having a pole of order index \(j\) at \(\lambda_j\), then the Z-transform \(F_X(\lambda) = \lambda (\lambda I - A)^{-1}X^{(0)}\) of the solution sequence with that initial value will have a pole of that order, and therefore one of the entries in the solution sequence \(\{X^{(i)}\}_{i=0}^{\infty}\) for that choice of \(X^{(0)}\) will contain a term of the form 
\[
\prod_{\ell=1}^{\text{index}_j-1} (i - \ell + 1) \cdot \lambda_j^{i-\text{index}_j+1}
\]
(the empty product that occurs when \(\text{index}_j = 1\) having the value 1). This argument is reversible: if some solution sequence has a term of the form \(p(i) \cdot \lambda_j^i\) where \(p(i)\) is a polynomial in the index, then some entry in the resolvent kernel must have a pole of order \(\deg(p) - 1\) at \(\lambda_j\), and a polynomial \(p(i)\) of maximum degree will give a pole of maximum order.

From these considerations follow boundedness and stability conditions very much like (indeed, as we shall see, generalizing) those for single equations of possibly-higher order:

**Proposition [Boundedness and Stability Conditions]:** In order for all the vector solution sequences of the homogeneous linear system of difference equation (1.20) to be (uniformly) bounded, it is necessary and sufficient that all the eigenvalues \(\lambda_j \in \sigma(A)\) of \(A\) satisfy \(|\lambda_j| \leq 1\), and that each root \(\lambda_j\) with \(|\lambda_j| = 1\) have index \(j = 1\) (i.e., that \((\lambda - \lambda_j)\) divide the minimal polynomial \(\mu_A(\lambda)\) but \((\lambda - \lambda_j)^2\) not divide it). In order for all the solution sequences of (1.20) to be sequences that tend to zero with increasing index, it is necessary and sufficient that all the eigenvalues \(\lambda_j \in \sigma(A)\) of \(A\) satisfy \(|\lambda_j| < 1\).

It suffices to look at these sequences coördinatewise, and there we know the way that the poles of the Z-transform (which are just the elements of \(\sigma(A)\) and their orders determine and are determined by what terms of the form 
\[
\prod_{\ell=1}^{\text{index}_j-1} (i - \ell + 1) \cdot \lambda_j^{i-\text{index}_j+1}
\]
enter in the solution.

**Corollary:** If all the eigenvalues \(\lambda_j \in \sigma(A)\) of \(A\) satisfy \(|\lambda_j| < 1\) and \(B \in \mathbb{C}^n\) is any constant vector, then all the solutions of
\[
X^{(i+1)} = AX^{(i)} + B
\]
(1.28)
converge (geometrically fast) as \(i \to \infty\) to the constant vector \((I - A)^{-1}B\).

In fact, it follows from the considerations of pp. 8–9 above that every solution of (1.28) has the form
\[
X^{(i)} = A^iY^{(0)} + (I - A^i)(I - A)^{-1}B
\]
(1.29)
and the matrix sequence \(\{A^i\}_{i=0}^{\infty}\) tends to the zero matrix (in whatever norm one wishes) faster than \(\rho^i\) for any \(\rho\) satisfying \(\max\{|\lambda_j|: \lambda_j \in \sigma(A)\} < \rho < 1\).

It would be tedious and probably not very useful to catalogue all the cases that one could encounter in solving a system of the form (1.21) under the additional hypothesis that \(\{G^{(i)}\}_{i=0}^{\infty}\) is a sequence such that \(F_G(\lambda)\) is a rational function finite at \(\infty\) and zero at zero. If (under that hypothesis) we denote\(^{(17)}\) by \(\sigma(G)\) the set of poles of \(F_G(\lambda)\), then the following broad outlines should be clear. Recall (1.26) above: the Z-transforms of solutions of (1.21) obey
\[
F_X(\lambda) = \lambda (\lambda I - A)^{-1}X^{(0)} + (\lambda I - A)^{-1}F_G(\lambda).
\]

If \(\sigma(A) \cap \sigma(G) = \emptyset\), then the (coördinate functions of the) second term on the r. h. s. will have (possible) poles at the points of \(\sigma(G)\) and (possible) poles at the points of \(\sigma(A)\), of order no greater than those of \(F_G(\lambda)\) and of \(\lambda (\lambda I - A)^{-1}\) respectively. The r. h. s. thus is a sum of finitely many vector-valued functions of two kinds: those that have (possible) poles at the points of \(\sigma(G)\), of order no greater than those of \(F_G(\lambda)\), and those that have (possible) poles at the points of \(\sigma(A)\), of order no greater than those of \(\lambda (\lambda I - A)^{-1}\).

The sum of all the terms of the latter class simply gives the Z-transform of a solution of the associated homogeneous equation. The sum of all the terms of the former class will be a particular solution of (1.21), with coördinate functions that are linear combinations of the inverse Z-transforms of whatever occurs in

\(^{(17)}\) This notation is not entirely standard but will certainly seem reasonable.
the partial-fraction expansion of $F_G(\lambda)$. However, if $\lambda_j \in \sigma(A) \cap \sigma(G) \neq \emptyset$, then a pole at $\lambda_j$ of the term $(\lambda - A)^{-1}F_G(\lambda)$ could have order as high as the sum of its orders in $F_A(\lambda)$ and $F_G(\lambda)$. To make a very simple illustration: for the matrix $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, we have $\sigma(A) = \{1\}$ but $\lambda_1 = 1$ is a pole of order 2 of the resolvent: $(\lambda - A)^{-1} = \frac{1}{(\lambda - 1)^2} \begin{bmatrix} \lambda & -1 \\ 0 & \lambda - 1 \end{bmatrix}$. If $G^{(i)} = i \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, then $F_G(\lambda) = \frac{\lambda}{(\lambda - 1)^2} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and thus $(\lambda - A)^{-1}F_G(\lambda) = \frac{\lambda}{(\lambda - 1)^2} \begin{bmatrix} 1 \\ \lambda - 1 \end{bmatrix}$. Its inverse Z-transform is the vector sequence

$$\{X^{(i)} = \begin{bmatrix} \frac{1}{3}i - \frac{1}{2}i^2 + \frac{1}{6}i^3 \\ -\frac{1}{2}i + \frac{1}{2}i^2 \end{bmatrix}\}_{i=0}^\infty.$$ 

It is a mechanical matter\(^{(18)}\) to check that this sequence is a solution of (1.21) for this choice of $A$ and $\{G^{(i)}\}_{i=0}^\infty$. Thus the increase in rate of growth can be “worst possible” for a “driving sequence” $\{G^{(i)}\}_{i=0}^\infty$ chosen in such a way as to reinforce the polynomial-growth behavior already present in the fine spectral structure of $A$: even though solutions of the associated homogeneous equation would only grow like $i$ and $\{G^{(i)}\}_{i=0}^\infty$ only grows like $i^2$, one can exhibit a solution of the inhomogeneous equation that grows, not like $i^2$ (as one might at first expect) but like $i^3$.

### 1.6. Single Equations, Revisited.

The results we got for a single equation of order $n$ can also be viewed from the standpoint of an $n$-dimensional first-order system; this can be done purely algebraically, and here is a sketch of the way to do it. Some of this construction, though not all of the details of proof, are relevant to the construction of numerical ODE solvers. Recall the homogeneous equation (1.13) above, which is repeated here to save the reader from having to look back:

$$X_{n+i} + a_{n-1}X_{n+i-1} + \cdots + a_0X_i = 0 \quad (a_n \neq 0);$$  
(1.30)

(1.2) can be put in this form by “reindexing” the $a_i$’s and $X_i$’s appropriately, dividing also by the “leading coefficient” if appropriate. We can now make (1.30) give rise to a first-order system of the form (1.20), namely

$$X^{(i+1)} = AX^{(i)}$$  
(1.31a)

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix},$$  
(1.31b)

in the following way: if $X^{(i)} = (X_{i-n+1}, X_{i-n+2}, \ldots, X_{i-1}, X_i)^T$ is the vector whose components are the current value of $X_i$ and the last previous $(n-1)$ values of $X_i$ (with the components indexed in increasing order of their indices), then $A$ “shifts the previous values up,” dropping $X_{i-n+1}$, and replaces the last component by the computed value of the next term $X_{i+1} = -a_{n-1}X_i - \cdots - a_1X_{i-n+2} - a_0X_{i-n+1}$ of the

---

\(^{(18)}\) And, like a sensible person, I checked it mechanically.
sequence, given by solving (1.30) for it. It is easy to check that the correspondence

$$\{X_i\}_{i=0}^{\infty} \rightarrow \left\{ \begin{bmatrix} X_i & X_{i+n-1} & \cdots & X_{i+2} & X_{i+1} \\ X_{i+1} & X_{i+n} & \cdots & X_{i+3} & X_{i+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ X_{i+n-2} & X_{i+n-3} & \cdots & X_{i+3} & X_{i+2} \\ X_{i+n-1} & X_{i+n-2} & \cdots & X_{i+3} & X_{i+2} \end{bmatrix} \right\} \bigg|_{i=0}^{\infty}$$

(1.32)

is a (linear) isomorphism between the space of solution sequences of (1.30) and the space of solution sequences of the system (1.31). In particular, the solution space of (1.30) has dimension $n$. The remainder of this § requires a knowledge of the algebra of a single linear transformation in the form of the algebra of modules over the polynomial ring $\mathbb{K}[\lambda]$, and may be omitted by anyone who hasn’t seen that material. It provides the outlines of a purely algebraic approach to these recursions, however, and \textit{mutatis mutandis} indicates how these questions might be considered if the ground field were finite, \textit{e.g.}, if it were $\mathbb{GF}(2^p)$. Not that the changes are all so direct: not all the prime polynomials are linear, and $\alpha \rightarrow \alpha^p$ is an automorphism in characteristic $p$.

Begin by observing that the matrix $A$ of (1.31) is the \textit{companion matrix} of the polynomial $\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 \in \mathbb{C}[\lambda]$, the polynomial obtained by substituting $\lambda^j$ for $X_j$, $j = 0, \ldots , n-1$, in (1.30) (and making the obvious cancellation). It is well known and easy to check—an exercise in expanding $\det(\lambda I - A)$ by minors—that the characteristic polynomial $\chi_A(\lambda)$ of $A$ is exactly the polynomial $\chi_A(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = \chi_A(\lambda)$ of §1.3 above. The set $\sigma(A)$ of eigenvalues of $A$ is exactly the set of roots of this polynomial—the “$\lambda_j$’s” used in forming the basic solution sequences described in §1.3. General abstract nonsense says that the prime (necessarily first-degree, if $\mathbb{K} = \mathbb{C}$) divisors of the minimal and characteristic polynomials of $A$ are the same, and the corresponding primary direct-sum decomposition of $\mathbb{C}^n$ (into $A$-invariant subspaces, or equivalently into submodules over $\mathbb{C}[\lambda]$) thus has the form

$$\mathbb{C}^n = \bigoplus_{\lambda_j \in \sigma(A)} V_{\lambda_j}.$$  (1.33)

(Let $E_{\lambda_j} : \mathbb{C}^n \rightarrow V_{\lambda_j}$ denote the $A$-invariant projection [in fact, $E_{\lambda_j}$ is a polynomial in $A$] of $n$-dimensional space onto $V_{\lambda_j}$ in what follows.] The matrix $A$ has a cyclic vector, namely $e_n = (0,0,\ldots,1)^T$, the $n$-th standard basis vector. The components of $e_n$ on the primary submodules $\{V_{\lambda_j} : \lambda_j \in \sigma(A)\}$ are thus cyclic vectors for these submodules. It follows by general abstract nonsense that for all $\lambda_j \in \sigma(A)$, the index of nilpotency of $(\lambda_j I - A)_{V_{\lambda_j}}$ cannot be smaller than $\dim(V_{\lambda_j})$, thus must equal $n_{\lambda_j} = \dim(V_{\lambda_j})$ and that $\chi_A(\lambda)$ is also the minimal polynomial of $A$. There is consequently just one Jordan block corresponding to $\lambda_j$, an $n_j \times n_j$ matrix of the form

$$\begin{bmatrix}
\lambda_j & 1 & 0 & \cdots & 0 \\
0 & \lambda_j & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
0 & 0 & 0 & \cdots & \lambda_j
\end{bmatrix}.$$  In the present case in which $A$ has the special companion-matrix form seen in (1.31b), however, it is possible to realize the basis of $V_{\lambda_j}$ explicitly, as follows. For any $i \geq n$ (and indeterminate $\lambda$), a simple matrix computation shows that the relation

$$\begin{bmatrix}
\lambda_i \\
\lambda_i^2 \\
\vdots \\
\lambda_i^{n-1} \\
\lambda_i^n
\end{bmatrix} = \begin{bmatrix}
0 \\
\lambda_i \\
\vdots \\
\lambda_i^{n-1} \\
\lambda_i^n
\end{bmatrix}$$

holds. Plugging a root $\lambda_j \in \sigma(A)$ of $\chi_A(\lambda) = 0$ into (1.34) gives
exhibiting the fact that \((\lambda^{i-n+1}, \lambda^{i-n+2}, \ldots, \lambda^{i-1}, \lambda^i)^T\) is an eigenvector of \(A\) belonging to \(\lambda_j\). However, if \(\lambda_j\) is a root of multiplicity \(n_j > 1\) of \(\chi_A(\lambda)\), so that we may factor \(\chi_A(\lambda) = (\lambda - \lambda_j)^{n_j} q(\lambda)\) (where \(q(\lambda_j) \neq 0\)), then all derivatives of \(\chi_A(\lambda)\) through the \((n_j - 1)\)-st also have a root at \(\lambda = \lambda_j\). Differentiating\(^{19}\) both sides of (1.34) with respect to \(\lambda\) and plugging in \(\lambda = \lambda_j\) gives

\[
\begin{align*}
\begin{bmatrix}
\lambda^{i-n+1} \\
\lambda^{i-n+2} \\
\vdots \\
\lambda^i \\
\lambda_j^{i-n+1} \\
\vdots \\
(i-n+1)\lambda_j^{i-n} \\
(i-n+2)\lambda_j^{i-n+1} \\
\vdots \\
(i-1)\lambda_j^{i-2} \\
\vdots \\
\vdots \\
\end{bmatrix} + (\lambda j - A)
\end{align*}
\begin{align*}
\begin{bmatrix}
(i-n+1)\lambda^{i-n} \\
(i-n+2)\lambda^{i-n+1} \\
\vdots \\
(i-1)\lambda_j^{i-2} \\
\vdots \\
\vdots \\
\end{bmatrix} = 0
\end{align*}
\]

(1.36a)

\[
\begin{align*}
\begin{bmatrix}
\lambda^{i-n+1} \\
\lambda^{i-n+2} \\
\vdots \\
\lambda^i \\
\lambda_j^{i-n+1} \\
\vdots \\
(i-n+1)\lambda_j^{i-n} \\
(i-n+2)\lambda_j^{i-n+1} \\
\vdots \\
(i-1)\lambda_j^{i-2} \\
\vdots \\
\vdots \\
\end{bmatrix} + (\lambda j - A)
\end{align*}
\begin{align*}
\begin{bmatrix}
(i-n+1)\lambda_j^{i-n} \\
(i-n+2)\lambda_j^{i-n+1} \\
\vdots \\
(i-1)\lambda_j^{i-2} \\
\vdots \\
\vdots \\
\end{bmatrix} = 0
\end{align*}
\]

(1.36b)

\[
\begin{align*}
\begin{bmatrix}
\lambda^{i-n+1} \\
\lambda^{i-n+2} \\
\vdots \\
\lambda^i \\
\lambda_j^{i-n+1} \\
\vdots \\
(i-n+1)\lambda_j^{i-n} \\
(i-n+2)\lambda_j^{i-n+1} \\
\vdots \\
(i-1)\lambda_j^{i-2} \\
\vdots \\
\vdots \\
\end{bmatrix} = \lambda_j
\end{align*}
\begin{align*}
\begin{bmatrix}
\lambda_j^{i-n+1} \\
\lambda_j^{i-n+2} \\
\vdots \\
\lambda_j^{i-1} \\
\vdots \\
\vdots \\
\end{bmatrix}
\end{align*}
\]

(1.36c)

exhibiting the fact that applying \(A\) to \(((i-n+1)\lambda_j^{i-n}, (i-n+2)\lambda_j^{i-n+1}, \ldots, (i-1)\lambda_j^{i-2}, i\lambda_j^{i-1})^T\) “shifts its index \(i\) to \(i+1\),” which is equivalent to showing that the scalar sequence \(\{\lambda_j^i\}_{i=0}^\infty\) satisfies the recursion (1.30). By differentiating the relation (1.34) repeatedly—up to \(n_j - 1\) times for an eigenvalue of multiplicity \(n_j\)—one can generate the remaining scalar solution sequences of the forms listed on p. 4 above, whose validity was previously established by the Z-transform technique.\(^{20}\) Incidentally, from the standpoint of the Jordan form of \(A\) (or companion matrices in general), this process (in which one may take \(i = 0\) if one wishes) successively generates an eigenvector \(F_1 = (1, \lambda_j, \ldots, \lambda_j^{n_j-1})^T\) and then linearly independent vectors \(F_2, \ldots, F_n\) satisfying \((\lambda j - A) F_k = F_{k-1}\) for \(k = 2, \ldots, n\). This is the basis of \(V_{\lambda_j}\) one needs to produce the corresponding Jordan block in the Jordan canonical form that is similar to \(A\).

\(^{19}\) Because the product of a matrix and a vector depends bilinearly on those two arguments, the familiar differentiation formulas are valid provided that the order of multiplication is observed in the two terms of the differentiated product.

\(^{20}\) The details of this are left to the interested reader, who should use the Leibniz formula for higher derivatives of a product in order to see that only two terms survive on the l. h. s.
2. Floating-Point Representation and Attendant Errors.

2.0. Preliminary Considerations. Atkinson’s §1.2 offers, in general, a perfectly adequate introduction to the topic of floating-point numbers as they appear inside a computer. In fact, his treatment is a little more than adequate: while we don’t intend to fight Atkinson’s book, the 2000 AD reader of a 1989 textbook’s treatment of a (soft- and) hardware-dependent subject has to disregard some of the care that one used to have to take. Much of the floating-point arithmetic that is now done on PC’s is routinely done in “double precision,” because the floating-point processors—whether outboard and named n87 or inboard as in the current Pentium processors—report their results in “double precision” and work internally in an 80-bit format, so that one can expect the double-precision results of floating-point operations to have been rounded from internal results of much greater precision. For this reason—particularly in an introductory course—there is no reason to think about any floating-point base except \( \beta = 2 \), and very little reason to think about “chopped” rather than rounded floating-point operations. The “single-precision” unit roundoff, as discussed by Atkinson pp. 15–16, will thus be \( \delta = 2^{-24} \); but it is more realistic to think in terms of the “double-precision” unit roundoff \( \delta = 2^{-53} \) of the currently mass-produced crop of floating-point processors. If you use Matlab as your numerical-analysis laboratory, then \( 2^{-53} \) is the unit roundoff you should expect (unless you perform experiments in which it is natural to take advantage of its “infinite precision” symbolic-computation capability). Moreover, with \( \beta = 2 \) a “normalized floating-point representation” as discussed on Atkinson’s p. 12 always has leading digit \( a_1 = 1 \) in the mantissa (and its storage can be suppressed). Thus from the mathematical standpoint, we will be working with normalized (binary) floating-point number having the form

\[
x = r \cdot 2^n \quad \text{where } \frac{1}{2} \leq r < 1
\]

and the (signed, if you wish) rational number \( r \) is a \( t \)-bit dyadic rational with (maximum) denominator \( 2^t \). The unit roundoff \( \delta \) will thus be \( 2^{-t} \), where \( t = 24 \) for single and \( t = 53 \) for double precision. Numbers of this form are frequently called machine numbers: the form in which they occur in computer memory is exact.

There are some conventions about representing errors to which most texts and papers on approximation theory adhere, and we’ll try to do that also. The relation between a real (or complex) number \( x_{\text{true}} \) and its approximation \( x_{\text{approx}} \) is conventionally treated as having been written in the form

\[
x_{\text{true}} = x_{\text{approx}} + \text{error}.
\]

This is essentially a sign convention for error terms; however, an equally conventional notation which we shall introduce below will stand that convention on its head. This convention agrees, e.g., with the customary way

\[
f(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k + \frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1}
\]

in which one writes “finite Taylor series with remainder term.”

The relative error in an approximation is the ratio of the error to the true value, \( \frac{\text{error}}{x_{\text{true}}} \) (of course this is meaningless if the true value is zero!). If we denote the relative error by \( \epsilon \), so \( \epsilon = \frac{\text{error}}{x_{\text{true}}} \) (note that, unlike most “epsilon,” this number can be negative) then an equivalent form for (2.1a) is

\[
x_{\text{true}} = x_{\text{approx}} + \epsilon \cdot x_{\text{true}}
\]

but in fact one will usually see that equation written with what amounts to the negative of the usual sign convention,

\[
x_{\text{approx}} = (1 + \epsilon) \cdot x_{\text{true}}.
\]

(21) That statement may have become obsolete as it was written.
One should try not to be annoyed at this inconsistency: in most cases one is interested in the magnitude rather than the sign of the relative error, and so one will usually see (2.1c) written in such a form as

$$x_{\text{approx}} = (1 + \epsilon) \cdot x_{\text{true}} \quad (|\epsilon| \leq \eta). \quad (2.1d)$$

The usefulness of (2.1d) as a form in which to express relative-error estimates will become increasingly clear as we progress.\(^{(22)}\) Indeed, we shall much prefer this notation to Atkinson’s Rel($x_{\text{approx}}$) = ($x_{\text{true}} - x_{\text{approx}}$)/$x_{\text{true}}$; it’s a lot easier form in which to calculate error estimates.

For examples of error generation and a more detailed discussion, see Atkinson’s §1.3, p. 17 ff.

One of the tacit agreements between computer/compiler sellers and buyers is that any real number $x$ that may not be a machine number will be approximated inside the machine by a number $\hat{x}(x)$ within a controlled relative error. Assuming base 2 and rounding rather than chopping, the agreement reads

$$\hat{x}(x) = x(1 + \epsilon) \quad |\epsilon| \leq \delta \quad (2.2)$$

where $\delta = 2^{-t}$ is the unit roundoff. This amounts to saying that $\hat{x}(x)$ is one of the (at least one and at most two) machine numbers closest to $x$: Atkinson gives the details on p. 14. A similar tacit agreement is that a rational operation ($+, -, \times$ or $\div$), for which Atkinson’s §1.4 uses the generic symbol $\omega$) applied to two 

**machine numbers** $x$ and $y$ will result in a number whose relative error has magnitude $\leq \delta$. That is, for all the rational operations on machine numbers $x$ and $y$, the computed (machine-number) value of $x \omega y$ should equal $(x \omega y)(1 + \epsilon)$, where $|\epsilon| \leq \delta$. For the operations $\times$ and $\div$, things are not too bad even when $x$ and $y$ have already been floated. We have $\hat{x}(x) = x(1 + \epsilon_1)$, $\hat{y}(y) = y(1 + \epsilon_2)$, and the computed value of $x \omega y$ when $\omega$ is multiplication or division is thus $\hat{x}(\hat{y}(x)) = (x \omega y)(1 + \epsilon_1)(1 + \epsilon_2)(1 + \epsilon_3)$, where all $|\epsilon_i| \leq \delta$. At this point we may as well prove the following general

**Lemma:** If in a finite product of real numbers $(1 + \epsilon_1)(1 + \epsilon_2) \cdots (1 + \epsilon_n)$ each of the numbers $\epsilon_i$ satisfies $|\epsilon_i| \leq \delta < 1$, then there exists a single $\epsilon$ with $|\epsilon| \leq \delta$ for which

$$(1 + \epsilon_1)(1 + \epsilon_2) \cdots (1 + \epsilon_n) = (1 + \epsilon)^n. \quad (2.3)$$

**Proof.** Call the value of the product $P$; then, since it is a monotone-increasing function of each $\epsilon_i$ for fixed values of the others, we have $(1 - \delta)^n \leq P \leq (1 + \delta)^n$. Evidently the minimum and maximum values of the continuous function $\epsilon \rightarrow (1 + \epsilon)^n$ on the real interval $[-\delta, \delta]$ are $(1 - \delta)^n$. By the intermediate-value theorem, there exists an $\epsilon \in [-\delta, \delta]$ for which $(1 + \epsilon)^n = P$.

It follows that the number actually computed for $x \times y$ or $x \div y$ is “off” by a factor of $(1 + \epsilon)^3$ for $|\epsilon| \leq \delta$, and that may be fairly acceptable; in fact, if $x$ and $y$ were both machine numbers, the factor is only $(1 + \epsilon)$. And—although we shall not be able to do much with this fact now—since the relation $x \hat{\omega} y = \hat{x}(\hat{y}(x))$ holds for all the rational operations, where $\hat{\omega}$ denotes the internal machine value computed for $x \omega y$, we can interpret machine operations on floating numbers as being exact operations performed on numbers slightly perturbed from their exact values. For example, if the operation is addition or subtraction, then the trivial equation

$$x \hat{\pm} y = [x(1 + \epsilon_1) \pm y(1 + \epsilon_2)](1 + \epsilon_3) = x(1 + \epsilon_1)(1 + \epsilon_3) \pm y(1 + \epsilon_2)(1 + \epsilon_3) \quad (2.4)$$

can be interpreted as saying that the computed value of $x \pm y$ is $\hat{x} \pm \hat{y}$, where $\hat{x}$ and $\hat{y}$ are slightly perturbed values of $x$ and $y$ respectively. This is the point of view of *backward error analysis*\(^{(23)}\) a method of conceptualization that allows one to think of the errors committed by computer arithmetic as having arisen from exact arithmetic on perturbed data. Relations similar to (2.4) are easily written for $\times$ and $\div$.

---

\(^{(22)}\) This notation has built into it the notion of *inverse error analysis*: think of doing exact operations with the number $x(1 + \epsilon)$ rather than approximate operations with an approximate number.

Unfortunately, relations like (2.4) do not tell the whole story. The two principal sources of (relative) error are loss of significance (more accurately, loss of significant digits or bits) and error growth due to iterated addition, and the study of these is no longer a preliminary consideration.

2.1. Loss of Significant Digits. This is something that everyone has seen happen with decimal numbers and hand subtraction (or addition of numbers of opposite sign). Each of the numbers 0.346184461 and 0.346027741 may be known to 9-decimal-digit accuracy, but if their difference is computed on an old pocket calculator that only knows 5-digit accuracy, then the minuend will be rounded to 0.34618 and the subtrahend to 0.34603, and their computed difference will be the 5-digit number 0.15000 \times 10^{-3}, while their true difference is the 5-digit number 0.15672 \times 10^{-3}. The relative error is about 0.043, very much larger than 3× unit roundoff, and it is enough to call the “5” in the computed value into question (the true value would round to 0.16 \times 10^{-3} as a 2-digit number). Thus over 3 decimal digits’ worth of accuracy have been lost from the 5 that one might uncritically have expected from the calculator.

The general phenomenon corresponding to this for subtraction of numbers in normalized binary floating-point form is easy to analyze precisely, and the analysis is instructive. Suppose \( x \) and \( y \) are normalized binary floating-point numbers, and for definiteness suppose \( x > y > 0 \). The numbers can be written as \( x = r \times 2^a \) and \( y = s \times 2^b \) respectively, with \( 1/2 \leq r < 1 \) and \( 1/2 \leq s < 1 \). Machine subtraction is essentially an integer operation on floating-point numbers with equal characteristics, but if \( x > y \) then \( y \) must be rewritten as \( y = (s \cdot 2^{-(a-b)}) \times 2^a \) before the subtraction takes place. The result of the subtraction is thus

\[
x - y = (r - s \cdot 2^{-(a-b)}) \times 2^a \tag{2.5}
\]

in which the characteristic (before normalization) is still that of \( x \) but the mantissa is

\[
r - s \cdot 2^{-(a-b)} = r \left( 1 - s \cdot \frac{2^b}{r \cdot 2^a} \right) = r \cdot \frac{x - y}{x}. \tag{2.6}
\]

Even after the result of the subtraction is normalized, there will only be as many “good”—i.e., valid—bits in the mantissa as are present in the number on both sides of (2.6). Since \( r < 1 \), there must be at least \( n \) “bad bits” in this mantissa, where \( n \) is the largest natural number for which \( \frac{x - y}{x} \leq 2^{-n} \): to normalize a number that is strictly \( < 2^{-n} \) it must be shifted at least \( n \) binary places to the left, and the new bits required to fill out the floating-point format are probably machine-supplied zeros. On the other hand, there is some good news: if \( m \) is the smallest natural number for which \( 2^{-m} \leq \frac{x - y}{x} \), then since \( 1/2 \leq r \) one has \( 2^{-(m+1)} \leq r \cdot \frac{x - y}{x} \) and therefore—since a “legal” mantissa must be \( < 1 \)—this number cannot be shifted more than \( m \) places to the left to normalize it. Thus at most \( m \) bits of the computed \( x - y \) can be “bad.”

This rather concrete computation helps to explain why alternating-series approximations are the pure analyst’s friend but the numerical analyst’s enemy. A subtraction that results in almost complete cancellation can result in numerical garbage. Thus, for example, while one might want to approximate \( e^x \) by a partial sum of the exponential series for all small \(|x|\) irrespective of sign—after all, the formula doesn’t care, and the alternating series for \( x < 0 \) converges even more rapidly than the series of positive terms—it might be better to divide into two cases:

\[
\exp_{\text{approx}}(x) = \begin{cases} 
1 + x + \frac{x^2}{2} + \cdots + \frac{x^n}{n!} & \text{for } x \geq 0 \\
1 + |x| + \frac{|x|^2}{2} + \cdots + \frac{|x|^n}{n!} & \text{for } x < 0
\end{cases}
\tag{2.7}
\]

Of course in the real world, programming in a high-level language, one assumes (hopes?) that the builder of the built-in exponential function did his/her homework. The reader might find it amusing to think about ways to approximate such functions as \( e^x - 1 - x \) for small \(|x|\) without asking for loss-of-significance errors: obviously the use of such formulas involving the built-in function \( \exp() \) as \( \exp(x) - 1.0 - x \) is a request for trouble in the less-significant bits.
2.2. Linear Difference Inequalities Majorized by Linear Difference Equations. The next thing on the agenda is studying the growth of floating-point error as summands are repeatedly added. This will present us with the first of many situations of the following kind: we shall be able to write an error term $\epsilon_i$ at the $i$-th step of some sequential process, and the successive errors will be governed by a sequence of inequalities of the form

$$|\epsilon_{i+1}| \leq a |\epsilon_i| + b \quad (i = 1, 2, \ldots)$$

(2.8)

where $a$ and $b$ are nonnegative constants (not depending on $i$). We shall not be able to “solve” the inequalities and write a closed-form formula for the sequence $\{\epsilon_i\}$, but we shall be able to get a sequence of estimates $E_i \geq 0$ in closed form, such that $|\epsilon_i| \leq E_i$ for all $i$ and such that the growth of the $E_i$ is easy to estimate. We do this by considering the difference equation

$$E_{i+1} = a E_i + b,$$

(2.9)

solving it explicitly, and observing that if $|\epsilon_1| \leq E_1$, then the simple inductive step

$$|\epsilon_k| \leq E_k \implies |\epsilon_{k+1}| \leq a |\epsilon_k| + b \leq a E_k + b = E_{k+1}$$

(2.10)

shows that the inequality $|\epsilon_i| \leq E_i$ persists for all indices $i$.

So the first task is to solve the difference equation (2.9) explicitly: but this is just a case of our simplest illustrative example (first order, scalar, inhomogeneous) in §1.3 above. We solve the equation in two steps: first we find a constant sequence $\{B_i \equiv B\}$ that satisfies it identically, and then we find the general solution of its related homogeneous linear difference equation $E_{i+1} = aE_i$, which will contain an undetermined constant. We add the two solutions to get the general solution of (2.9), and then adjust the constant to get the condition $E_1 = |\epsilon_1|$ to hold. The first step is easy: if we want $B$ to be a constant solution of (2.9) then we are asking for

$$B = aB + b \quad \text{and therefore} \quad B = \frac{b}{1-a}$$

(2.11)

(We shall assume $a \neq 1$ for the time being, and consider the case $a = 1$ separately.) The second step can be carried out by inspection: evidently $E_i = Ca^i$ satisfies $E_{i+1} = aE_i$ for any choice of the constant $C$. The general solution of (2.9) thus has the form

$$E_i = Ca^i + \frac{b}{1-a}.$$

(2.12)

(Cf. the discussion of formula (1.24) above.) If we want $E_1 = |\epsilon_1|$ then evidently we are asking for

$$|\epsilon_1| = C \cdot a + \frac{b}{1-a}$$

$$C = \frac{1}{a} \left[ |\epsilon_1| - \frac{b}{1-a} \right]$$

$$E_i = \left[ |\epsilon_1| - \frac{b}{1-a} \right] a^{i-1} + \frac{b}{1-a}$$

$$E_i = |\epsilon_1| a^i + \frac{(1-a^i-1)b}{a-1} \quad \text{or} \quad E_i = |\epsilon_1| a^i + \frac{(a^{i-1} - 1)b}{a-1}$$

(2.13)

where the first form is more intuitive if $a < 1$—the estimate decreases to $\frac{b}{1-a}$ as $i \to \infty$—while the second form, showing exponential growth as $i \to \infty$, is more intuitive if $a > 1$. 
We should still look at the case of (2.9) in which $a = 1$, but that is just a telescoping sum: $E_{i+1} = E_i + b$ is equivalent to $E_{i+1} - E_i = b$, and the successive equations
\begin{align*}
E_2 - E_1 &= b \\
E_3 - E_2 &= b \\
&\vdots \\
E_n - E_{n-1} &= b \quad \text{add up to give} \\
E_n - E_1 &= (n-1)b \\
E_n &= E_1 + (n-1)b \\
\end{align*}
(2.14)
as the solution in this case.(24)

2.3. [General] Sums with Summands of the Same Sign. Unfortunately, the attempted summation of (partial sums of) series of positive terms by computer can lead to difficulties similar to those we encountered when we looked at the difference of two nearly-equal numbers in §2.1 above. There is an implicit contract between the manufacturers of floating-point processors(25) and the computing public to the effect that if $x$ and $y$ are two nonnegative machine numbers and overflow does not occur, then the machine version of $x + y$ will be a float of the true value of $x + y$, and thus the machine version of $S = x + y$ will actually be a number $S^* = (1 + \epsilon)(x + y) = (1 + \epsilon)S$ where $|\epsilon| \leq \delta$, the unit roundoff error. The logical equivalence of the statements
\begin{align*}
S^* &= (1 + \epsilon)S = S + \epsilon S \\
\frac{S^* - S}{S} &= \epsilon \quad \text{with } |\epsilon| \leq \delta \\
\left| \frac{S^* - S}{S} \right| &\leq \delta \\
\end{align*}
(2.15a/b/c)
tells us that this is equivalent to saying that the relative error in $x + y$ will then be estimated by $\delta$. Unfortunately, if one adds $x_1, x_2$ and $x_3$ one gets
\begin{align*}
S_1^* &= x_1 \\
S_2^* &= (S_1^* + x_2)^* = (1 + \epsilon_2)(x_1 + x_2) \\
S_3^* &= (S_2^* + x_3)^* = (1 + \epsilon_3)[(1 + \epsilon_2)(x_1 + x_2) + x_3] \\
\end{align*}
and it becomes clear that as more terms are added additional factors of the form $(1 + \epsilon_i)$ will pile up on the early terms of the sum, tending to destroy whatever accuracy the earlier sums possessed.

Using some systematic notation to deal with this problem will help us in analyzing it. Let us assume we have an ordered sequence of nonnegative numbers $\{x_i\}_{i=1}^n$ to be added; for example, in Fortran one would be looking at the activity carried out by the do-loop
\begin{verbatim}
SUM = 0.
DO 20 I = 1,N
  20 SUM = SUM + X(I)
\end{verbatim}
and one might reasonably want to know how accurate the number in SUM is. Assume that each $x_i$ is a “machine number,” so that we don’t have to deal with the errors intrinsic to the floating point representation of the

(24) As we saw in a more general setting, the general equation (2.9) could be handled in a similar manner by multiplying the whole equation by $a^{-i}$ and considering the resulting equation for $F_i = a^{-i} E_i$.
(25) A spectacular violation of this contract occurred in the case of the early Pentium processors, which simply made mistakes. It was amusing to watch the manufacturer’s PR people try to simply blow off this “minor issue.” We are all assured that this issue no longer exists. Uh-huh.
$x_i$’s themselves. (For example, if $x_i = 3^{-i}$ then $x_i$ cannot be represented as a binary expansion in finitely many digits, so its machine representation must be wrong; we don’t want to have to deal with that.) Let $S_i$ denote the “true value” of the sum of the first $i$ terms: $S_i = x_1 + \cdots + x_i$. Let $S^*_i$ denote the “computed value” of the sum of the first $i$ terms: that is, $S^*_i$ is determined recursively by

\begin{align}
S^*_1 &= x_1 \\
S^*_{i+1} &= (S^*_i + x_{i+1})^* = (1 + \epsilon_i)(S^*_i + x_{i+1}) .
\end{align}

(2.16)

Let $\epsilon_i$ denote the error present in $S^*_i$, so that\(^{(26)}\)

$$\epsilon_i = S^*_i - S_i .$$

Then $\epsilon_1 = 0$, and there is a simple recursive estimate for each $\epsilon_i$ in terms of its predecessor:

\begin{align}
S^*_{i+1} &= (1 + \epsilon_i)[S^*_i + x_{i+1}] = (1 + \epsilon_i)[S^*_i + (S_{i+1} - S_i)] \\
&= (1 + \epsilon_i)[S^*_i - S_i + S_{i+1}] = (1 + \epsilon_i)(\epsilon_i + S_{i+1}) \\
S^*_{i+1} &= (1 + \epsilon_i)\epsilon_i + S_{i+1} + \epsilon_iS_{i+1} \\
\epsilon_{i+1} &= S^*_{i+1} - S_{i+1} = (1 + \epsilon_i)\epsilon_i + \epsilon_iS_{i+1} .
\end{align}

(2.18)

Equation (2.19) is exact but not too useful as it stands, since we do not know the values of most of its terms. However, we can estimate each $|\epsilon_i|$ by $\delta$, and because all the terms are nonnegative we can estimate $S_{i+1} \leq S_n$. Thus (2.19) leads to the recursive inequality

$$|\epsilon_{i+1}| \leq (1 + \delta)|\epsilon_i| + \delta S_n$$

(2.20)

which we treated completely in §0 above; with $\epsilon_1 = 0$, $a = (1 + \delta)$ and $b = \delta S_n$ in (2.13) we get

$$|\epsilon_i| \leq \frac{(1 + \delta)^{i-1} - 1}{2} \delta S_n = [(1 + \delta)^{i-1} - 1] S_n$$

(2.21)

and so for $i = n$

$$|\epsilon_n| \leq \frac{(1 + \delta)^{n-1} - 1}{2} \delta S_n = [(1 + \delta)^{n-1} - 1] S_n$$

$$|S^*_n - S_n| \leq [(1 + \delta)^{n-1} - 1] S_n$$

$$\left|\frac{S^*_n - S_n}{S_n}\right| \leq [(1 + \delta)^{n-1} - 1] .$$

(2.22)

The quantity $[(1 + \delta)^{n-1} - 1]$ is kind of like $(n - 1)\delta$, but not quite. Binomial expansion of the power of $(1 + \delta)$ gives

$$(1 + \delta)^{n-1} - 1 = 1 + (n - 1)\delta + \frac{(n-1)(n-2)}{2} \delta^2 + \cdots - 1 = (n - 1)\delta + \frac{(n-1)(n-2)}{2} \delta^2 + \cdots$$

and because $\delta^2 = 2^{-48}$ for single precision and $2^{-106}$ for double, there is a temptation to ignore all terms of higher degree in $\delta$ and say that the relative error in summing $n$ nonnegative terms is about $(n - 1)\delta$. But this cannot be right: $(1 + \delta)^{n-1}$ grows exponentially with $n$ and therefore faster than any power of $n$—certainly faster than the first power of $n$. Nonetheless, it is possible to give an estimate linear in $n$ provided one places a restriction $(n - 1)\delta \leq a$ on the number of summands. We change the “$n - 1$” to “$p$” just to make the computation seem more natural.

\(^{(26)}\) Note that, technically, this violates the sign convention for errors; we should say that it is the negative of the error, I suppose.
Lemma: If the natural number \( r \) satisfies \( r \delta < \ln k \) for some \( k > 1 \), then

\[
|(1 + \epsilon)^r - 1| \leq r \cdot |\epsilon| \quad \text{for } |\epsilon| \leq \delta .
\]  

(2.23)

Proof. One estimates the absolute values of the terms given by the binomial theorem by the terms of the exponential series for \( e^{r|\epsilon|} \), and throws in the remaining terms of the exponential series; then one factors out a single \( r|\epsilon| \) and replaces the remaining \( |\epsilon|'s \) by \( \delta \)'s. This gives

\[
|(1 + \epsilon)^r - 1| = \left| 1 + \frac{r \epsilon}{1!} + \frac{r(r-1) \epsilon^2}{2!} + \cdots + \frac{r(r-1)\cdots(r-k+1) \epsilon^k}{k!} + \cdots \right| - 1
\]

\[
\leq r|\epsilon| \cdot \left[ 1 + \frac{r \delta}{1!} + \frac{(r \delta)^2}{2!} + \cdots \right] = r|\epsilon| \cdot e^{r \delta} \leq r|\epsilon| \cdot k .
\]

The expression \( r|\epsilon| \cdot e^{r \delta} \) grows faster than exponentially in \( r \), but the factor \( r|\epsilon| \) is the most important part of the expression for moderate values of \( r \). For example, if we take \( k = 1.01 \) then \( \ln k \approx 0.00995033 \) and one can have the linear-in-\( r \) estimate \( |(1 + \epsilon)^r - 1| \leq r \cdot (1.01 \epsilon) \) for the rather mild restriction \( r \delta \leq 0.00995033 \), or even for single precision approximately \( r \leq 166939 \), at which value the function \( |(1 + \epsilon)^r - 1| \) would be estimated by \( 1.01 \cdot \ln 1.01 \approx 0.01 \) and less than two decimal digits’ accuracy would be guaranteed in any event. The choice of 1.01 was traditional but not otherwise necessary; the point is simply that if one “fudges \( \delta \) a bit and keeps \( n \) ‘sufficiently’ small,” then the relative error of a sum of \( n \) nonnegative terms grows approximately like \( (n-1)\delta \).

2.4. Long Sums with Large Summands. In contrast to the general situation discussed in the preceding \( \S \), it turns out that if each successive term is “not tiny in comparison with the current partial sum,” then the relative-error estimate \( (n-1)\delta \) is correct without fudging anything. This fact is the most convincing argument I know in favor of the practice of sorting sums so that they are added smallest-summand-first, since that order gives one the best chance of adding a new term that does not look small in comparison with the sum of those already accumulated. In any event, the exponentially-growing estimate of (2.22) is replaced by \( (n-1)\delta \) in the following situation, a fact which we want to state formally:

Proposition: If \( \{x_i\}_{i=1}^n \) is a sequence of machine numbers, the \( S_i = \sum_{j=1}^i x_j \) are their sequence of partial sums in order, and if for \( 1 \leq i < n \) each \( x_{i+1} \) satisfies the inequality

\[
\frac{x_{i+1}}{S_i} \geq \delta ,
\]

(\*)

then for \( 1 \leq i \leq n \) the relative error in the computed sums \( (S_i^* = x_1 \text{ and } S_{i+1}^* = fl(S_i^* + x_{i+1}) \text{ for } 1 \leq i < n) \) is bounded by

\[
\left| \frac{S_i^* - S_i}{S_i} \right| \leq (i-1)\delta
\]

(2.24a)

or, equivalently, the computed sums have the form

\[
S_i^* = [1 + (i-1)\eta_i]S_i \text{ where } |\eta_i| \leq \delta .
\]

(2.24b)

Remark: The inequality (\*) can be rephrased as \( S_{i+1} = S_i + x_{i+1} \geq (1 + \delta)S_i \), i.e., that \( S_{i+1} \) is at least as much relatively larger than \( S_i \) as \( 1 + \delta \) is than 1, so that “the machine knows that a term has been added.”

Proof of the proposition. We shall prove the equivalent form (2.24b) by induction on the index \( i \). The case \( i = 1 \) is clear: \( S_1^* = S_1 = x_1 \) since \( x_1 \) is a machine number. Suppose for a particular index \( 1 \leq i < n \)}
we know that (2.24b) holds. Then floating-point addition of \( x_{i+1} \) to the accumulated sum gives (for some \(|\epsilon_i| \leq \delta\))

\[
S_{i+1}^* = \text{fl}(S_i^* + x_{i+1}) = (1 + \epsilon_i)(S_i^* + x_{i+1}) = (1 + \epsilon_i)[(1 + (i - 1)\eta_i)S_i + x_{i+1}]
\]

\[
= (1 + \epsilon_i)(S_i + x_{i+1}) + (i - 1)[\eta_iS_i + \epsilon_i\eta_iS_i] .
\] (2.25)

Since (*) is equivalent to \( \delta S_i \leq x_{i+1} \) and we know \(|\eta_i| \leq \delta\), it follows that \( \eta_iS_i \) can be written in the form \( \eta_iS_i = \theta x_{i+1} \), where \(|\theta| \leq 1\). Equation (2.25) can therefore be rewritten as

\[
S_{i+1}^* = (1 + \epsilon_i)(S_i + x_{i+1}) + (i - 1)[\eta_iS_i + \epsilon_i\theta x_{i+1}] .
\] (2.26)

Consider the expression in square brackets in (2.26). The largest value that \( \eta_i \) and/or \( \epsilon_i\theta \) could have is \( \delta \), and the (algebraically) smallest is \(-\delta\), so the value of the expression in square brackets lies in the interval between \(-\delta \cdot S_{i+1} = -\delta \cdot (S_i + x_{i+1}) \) and \( \delta \cdot (S_i + x_{i+1}) = \delta \cdot S_{i+1} \). It thus has the form \( \zeta \cdot S_{i+1} \) for some \(-\delta \leq \zeta \leq \delta\). We can thus rewrite (2.26) in the form

\[
S_{i+1}^* = (1 + \epsilon_i)S_{i+1} + (i - 1)\zeta S_{i+1} = \{1 + \zeta + (i - 1)\zeta\}S_{i+1} .
\] (2.27)

Again, the largest value that the expression in square brackets in (2.27) could have is \( \delta + (i - 1)\delta = i\delta \) and the smallest is \(-\delta - (i - 1)\delta = -i\delta\), so there is an \( \eta_{i+1} \) with \(-i\delta \leq i\eta_{i+1} \leq i\delta\), in other words with \(|\eta_{i+1}| \leq \delta\), for which

\[
S_{i+1}^* = (1 + i\eta_{i+1})S_{i+1} \quad \text{where} \quad |\eta_{i+1}| \leq \delta
\] (2.28)

and that is just (2.24b) with \( i \) replaced by \( i + 1 \) throughout. This concludes the induction and proves the proposition.

This result suggests that (2.7) above might be better thought of as

\[
\exp_{\text{approx}}(x) = \begin{cases} 
\frac{x^n}{n!} + \cdots + \frac{x^2}{2} + x + 1 & \text{for } x \geq 0 \\
\frac{|x|^n}{n!} + \cdots + \frac{|x|^2}{2} + |x| + 1 & \text{for } x < 0
\end{cases}
\] (2.29)

for small \(|x|\), and this is true, if somewhat unrealistic. (In something more like real life, the \( x > 0 \) case would be evaluated by “nested multiplication,” i.e., in a do-loop corresponding to the form

\[
x \cdot \left( \cdots \left( \frac{x}{n - 1} \cdot \left( \frac{x}{n} + 1 \right) \right) + 1 \right)
\] (2.30)

and that probably only for \( 1/2 \leq x < 1 \), since \( \epsilon^\text{integer} \) would probably be handled by a combination of table look-ups, squaring and division, etc.)