Solving Linear and Transformable Non-linear Recursive Sequences using Generating Functions

Liu Jingyu

Contents

1 Introduction

The Fibonacci sequence is famous for its presence in nature [1] and aesthetic values [2]. It is defined by that every term equals the sum of the two preceding terms, starting with: 0, 1, 1, 2, 3, 5, ... However, it seems rather difficult to determine a general formula for the Fibonacci sequence, because people cannot directly reveal the solution from the recursive formula, unlike in simple arithmetic and geometric sequences. This leaves a problem while calculating any terms in the sequence when terms get large, where the tiniest error of addition will lead to completely wrong results. Due to several useful applications of this sequence in modern engineering [3] and architecture [4], finding a closed-form formula for the Fibonacci is imperative.

In the seventeenth century, a French mathematician named Jacques Philippe Marie Binet successfully solved the problem. He gave a general formula, called Binet's formula, associating the term number n and the nth term in the Fibonacci sequence. However there was something unexpected: Binet's formula consists of the sum of powers of irrational numbers, which the result is always an integer. [5] How could a sequence of integers have an irrational general formula? Why do the irrational parts always cancel out in the formula?

To explain this unexpected result, it is necessary to discuss the way where Binet found this formula. Actually, this formula can be found easily, using only elementary algebra and constructing a new sequence from the original recursive sequence.

The Fibonacci sequence has a closed-form general formula. As the simplest recursion, it only involves 2 backward terms and all coefficients of backward terms are 1. Does a recursive sequence involving 2 backward terms with some coefficients of other numbers have a calculable general formula? Or more generally, is it possible to find a general formula for a recursive sequence involving many backward terms with various sets of coefficients?

All sequences discussed above are linear and homogeneous, where the power indexes of terms are all 1 and there is no constant term. When it comes to non-homogeneous linear recursions which are a more general situation of linear recursions, the work to find the general formula is even more complicated. Linear relationships are the most common in research and applications, but a clear and complete proof with method showcase of the general situation using generating functions is still lacking among current mathematics studies. Also, many non-linear recursions can be translated into linear ones, which can be solved more easily.

This paper intends to discuss ways of finding a closed-form general formula of linear and transformable non-linear sequences from a recursively defined relationship. It starts with an original method to find the general formula given general homogeneous and non-homogeneous linear recursions using generating functions. Secondly, it summarises methods to solve some special non-linear recursions that can be translated into linear recursions.

2 Literature Review

There is a large amount of research on linear recursions at present. It is well-known that the Fibonacci sequence has the generating function [6]:

$$G(x) = 0 + 1x + 1x^2 + 2x^3 + 5x^4 + \cdots = \frac{x}{1 - x - x^2}.$$

The general solution to the Fibonacci sequence can be found using its generating function or pure algebra operations, and this general formula is [5]

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right].$$

Moreover, for a general situation where $a_n = m_1 a_{n-1} + m_2 a_{n-2}$, its solution can be found by first identifying its *characteristic equation*:

$$C(x) = x^2 - m_1 x - m_2 = 0.$$

if solving the equation above gives two distinct roots λ_1 and λ_2, then the general solution of this sequence is [7]

$$a_n = A_1 \lambda_1^n + A_2 \lambda_2^n \tag{2.1}$$

where A_1 and A_2 are constants that could be found by plugging in values of a_0 and a_1 and solving a set of simultaneous equations.

This method is not yet perfect, due to its limitation to order 2 homogeneous linear recursions which have characteristic equations with two distinct roots. This method cannot be applied for other order 2 homogeneous linear recursions with duplicate roots, simply because the sequence may not follow a geometric sequence. For example, the sequence $a_n = 2a_{n-1} - a_{n-2}$ where $a_0 = 1, a_1 = 3$ has clearly duplicate roots for the characteristic equation: $\lambda_1 =$

$\lambda_2 = 1$. Plugging in $a_0 = 1$ will give the general formula $a_n = 1 \cdot 1^n = 1$ which is apparently wrong.

A paper [8] mentioned that for a general homogeneous linear recursion of order t:

$$a_n = \sum_{r=1}^{t} m_r a_{n-r},$$

if the characteristic equation can be decomposed to

$$x^t - \sum_{r=1}^{t} m_r x^{t-r} = \prod_{s=1}^{j} (x - \lambda_r)^{p_s} = 0$$

where $\lambda_1, \lambda_2 \ldots \lambda_j$ are distinct roots, the general solution to the recursive sequence is

$$a_n = \sum_{r=1}^{j} P_r(n)\, \lambda_r^n, \tag{2.2}$$

where P_r are polynomials with degree $< p_r$. However, that paper did not give a method to find the general solution, nor proved the general solution is correct.

Another recent study [9] stated that general cases of linear recursive sequences of integers, both homogeneous and non-homogeneous ones, can be solved using *Bell polynomial*, which is far beyond the scope of elementary algebra. An alternative method can be given to reach a result more simply and more comprehensibly.

Non-linear recursive sequences cannot be generalised into fixed forms, so research on them usually focuses on specific cases. For example, there are professional studies on non-linear recursions such as $x_{n+1} = \frac{\alpha + \beta x_n}{B x_n + C x_{n-1}}$ [10] or $x_{n+1} = p + \frac{x_{n-k}}{x_n}$ [11].

With respect to formula (2.1), it is reasonable to make the hypothesis that, for any homogeneous order t recursive sequence that has t distinct roots for its characteristic equation, the general solution is

$$a_n = \sum_{r=1}^{t} A_r\, \lambda_r^n, \tag{2.3}$$

where A_r are constants to be found.

This paper confirms the hypothesis. Moreover, it discusses general linear recursion and some relative non-linear recursion using generating functions.

3 Methodology

This paper mainly discusses problems about recursive sequences within the field of algebra. It mainly uses generating functions of sequences together with intermediate algebraic operations to find the general formula. More specifically, a carefully selected *recurrence function* is multiplied by the generating function of the sequence, so that the generating function can be represented in another shorter form. Then the general formula of the coefficients of the generating function can be found using Taylor expansion, and thus the general formula for terms in the recursive sequence is found.

4 Preliminaries

To make the latter discussions more rigorous and meaningful, several definitions are given and assumptions are made as follows.

4.1 Definitions

The *order* of a recursion is the number of non-terminal terms of the sequence involved in the recursive equation. A general order t recursive equation may be in the form of

$$a_n = F(a_{n-1}, a_{n-2}, a_{n-3}, \dots a_{n-t}), \forall n \geq t, n \in \mathbb{N}. \tag{4.1.1}$$

where F is a function of t entities.

The *generating function* of a sequence is an infinite formal series where the coefficients are corresponding terms of an infinite series. For an infinite sequence (a_n): $a_0, a_1, a_2, \dots a_n, \dots$, the generating function of the sequence is

$$G(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + \cdots = \sum_{n=0}^{\infty} a_n x^n. \tag{4.1.2}$$

The variable x in the generating function has no actual meaning and acts only as an intermediate. It can belong to any domain as long as it can help solve the problem.

A *linear recursive sequence* or a *linear recursion* is that each term in the recursive sequence equals to a fixed linear combination of a finite number of its predecessors and a constant. A general case for a linear recursion of order t has the formula:

$$a_n = m_1 a_{n-1} + m_2 a_{n-2} + \cdots + m_t a_{n-t} + c = \sum_{r=1}^{t} m_r a_{n-r} + c, m_t \neq 0. \quad (4.1.3)$$

If the constant term $c = 0$, then the recursion is called *homogeneous* linear recursion because there is no constant term. Otherwise, the recursion is called *non-homogeneous* linear recursion.

The *characteristic equation* for homogeneous linear recursion is the equation obtained by replacing a_r in the recursive formula like (4.1.3) with $c = 0$ by x^r and then dividing the equation by x^{n-t}. In this paper, the notation $C(x) = 0$ is used to represent the characteristic equation. For recursive formula (4.1.3) with $c = 0$, its characteristic equation is

$$C(x) = x^t - m_1 x^{t-1} - m_2 x^{t-2} - \cdots - m_{t-1} x - m_t = x^t - \sum_{r=1}^{t} m_r x^{t-r} = 0. \quad (4.1.4)$$

As (4.1.3) defined, $m_t \neq 0$, so there is always a constant term $-m_t \neq 0$, which means that none of the roots of the characteristic equation is zero.

4.2 Assumptions

4.2.1 The recursive sequence (a_n) of order t must have the initial values $a_0, a_1, a_2 \dots a_{t-1}$ defined and given.

Consider if the initial values are not given, then the next value a_t cannot be derived from the recursive formula (4.1.1), and neither can the subsequent terms. The sequence may have a calculable general formula, but the method will depend on some specific recursive formulae given or other conditions provided, which should not be discussed in this paper for the general case.

4.2.2 The terms in the recursive sequence are not all zero, and the coefficients $m_1, m_2, \dots m_t$ in (4.1.3) are not all zero.

This assumption ensures that trivial cases are excluded from the discussion. For a sequence that the terms are all zero, the general formula is simply $a_n = 0$. For a linear recursion like (4.1.3) with all zero linear coefficients $m_r = 0$, the general formula is

$$a_n = \begin{cases} c & n \geq t \\ a_n & n < t \end{cases}.$$

where a_n for $n < t$ are given as initials (Assumption 4.2.1).

4.2.3 The recursive sequence is infinite, thus forming a generating function.

By Assumption 4.2.1 and definition 4.1.1, the recursive sequence must have a_t defined, and then a_{t+1} defined. By strong induction, the sequence must have a_n defined for $\forall n \in \mathbb{N}_0$.

Therefore, the recursive sequence is infinitely defined, which is sufficient for applying a generating function to the sequence.

5 Homogeneous Linear Recursions

5.1 The recurrence function $R(x)$

Consider the homogeneous linear recursion of order t:

$$a_n = m_1 a_{n-1} + m_2 a_{n-2} + \cdots + m_t a_{n-t} = \sum_{r=1}^{t} m_r a_{n-r}\, , m_t \neq 0, \tag{5.1.1}$$

and its generating function as (4.1.2). If a general formula for the coefficients of the infinite generating function can be found, then can the sequence. The two biggest problems are that the recursive formula is not related to the generating function, and the generating function is infinite.

Let the *recurrence function* $R(x)$ of a recursive sequence be a finite function that

$$R(x) \cdot G(x) = f(x) \Leftrightarrow G(x) = \frac{f(x)}{R(x)} \tag{5.1.2}$$

where $f(x)$ is another finite function. If this $R(x)$ could be found, then the generating function can be represented by the ratio of two finite functions, which can be expanded into infinite series using Taylor expansion. Then, the general formula for the coefficients of the generating function can be found.

5.2 Finding the recurrence function for homogeneous linear recursions

First, assume that $R(x)$ and $f(x)$ are finite polynomials. Notice that for a term x^{ε} for $\forall \varepsilon \geq t$ in $f(x)$, $R(x)$ and $G(x)$ multiply their terms of x^r and $x^{\varepsilon - r}$ for every $r = 0, 1, 2, \dots \varepsilon$ and sum every product for the result for $f(x)$. So, if let recurrence function be

$$R(x) = 1 - \sum_{r=1}^{t} m_r x^r \tag{5.2.1}$$

then $\forall \varepsilon \geq t$, the calculation of x^{ε} for $f(x)$ is

$$1 \cdot a_{\varepsilon} x^{\varepsilon} - \sum_{r=1}^{t} (m_r x^r \cdot a_{\varepsilon - r} x^{\varepsilon - r})$$

which is

$$\left(a_{\varepsilon} - \sum_{r=1}^{t} m_r a_{\varepsilon - r} \right) x^{\varepsilon} = 0.$$

From (5.1.1), it is clear that the term above is zero. Therefore, $f(x)$ is a finite polynomial with maximum degree $< t$. The construction of $R(x)$ ensures that all terms of $f(x)$ with degree $\geq t$ always cancel out due to the recursive formula; terms that have degree $< t$ can be found by manually multiplying $R(x)$ and the first t terms of $G(x)$, which are given as initial conditions (Assumption 4.2.1).

Hence it is proved that, if $R(x)$ is taken as (5.2.1), then $f(x)$ is a finite polynomial with maximum degree $< t$, and the generating function of the sequence is

$$G(x) = \frac{f(x)}{R(x)} = \frac{f(x)}{1 - \sum_{r=1}^{t} m_r x^r}. \tag{5.2.2}$$

Then, the general formula of coefficients of terms in $G(x)$ can be found by Taylor expansion on the right-side of equation (5.2.2).

5.3 Simplifying the generating function

By observing the recurrence function $R(x)$ in (5.2.1) and the characteristic equation $C(x) = 0$ as (4.1.4), the roots of these two polynomials are highly related. By fundamental theorem of algebra, $C(x)$ must have exactly t roots counting multiplicity. So, the factorization of $C(x)$ can be written as:

$$C(x) = x^t - \sum_{r=1}^{t} m_r x^{t-r} = \prod_{s=1}^{j} (x - \lambda_s)^{p_s}, \tag{5.3.1}$$

where λ_s are distinct roots of multiplicity p_s of the characteristic equation where $\sum p_s = t$.

By referencing (5.3.1), assume that $x \neq 0$, then consider the function $R(x^{-1})$:

$$R(x^{-1}) = 1 - \sum_{r=1}^{t} m_r x^{-r} = \frac{x^t - \sum_{r=1}^{t} m_r x^{t-r}}{x^t} = \frac{C(x)}{x^t} = \frac{\prod_{s=1}^{j} (x - \lambda_s)^{p_s}}{x^t}.$$

By the definition of generating functions (4.1.2), x can be in any domain, so $x \neq 0$ is fulfilled. Therefore, the root condition of $R(x^{-1})$ is completely identical to that of $C(x)$, then the roots of $R(x)$ should be reciprocals of corresponding roots of $C(x)$, i.e.,

$$R(x) = -m_t \prod_{s=1}^{j} (x - \lambda_s^{-1})^{p_s} = -m_t (-1)^t \prod_{s=1}^{j} (\lambda_s^{-1} - x)^{p_s}. \tag{5.3.2}$$

The coefficient $-m_t$ is multiplied because according to (5.2.1) the coefficient of x^t in $R(x)$ is $-m_t$.

Substituting (5.3.2) into (5.2.2) gives

$$G(x) = \frac{f(x)}{-m_t(-1)^t \prod_{s=1}^{j}(\lambda_s^{-1} - x)^{p_s}} = \frac{\frac{(-1)^{t+1}}{m_t} f(x)}{\prod_{s=1}^{j}(\lambda_s^{-1} - x)^{p_s}}.$$

Notice that $f(x)$ is a polynomial with maximum degree $< t$ (5.2.2), so is $\frac{(-1)^{t+1}}{m_t} f(x)$. Because the denominator polynomial has degree $= t$, $G(x)$ can be decomposed using partial fractions:

$$G(x) = \sum_{s=1}^{j} \sum_{q=1}^{p_s} \frac{A_{s,q}}{(\lambda_s^{-1} - x)^q}. \tag{5.3.3}$$

where $A_{s,q}$ are constants dependent on s and q. Then, the infinite series of $G(x)$ can be revealed by applying Taylor expansion on individual terms and summing up.

It is clearer to expand individual terms first. Consider

$$\frac{A_{s,q}}{(\lambda_s^{-1} - x)^q} = A_{s,q}(\lambda_s^{-1})^{-q}\left(1 - \frac{x}{\lambda_s^{-1}}\right)^{-q} = A_{s,q}\lambda_s^q(1 - \lambda_s x)^{-q}.$$

To be rigorous, the Taylor series of above converges if and only if, $|\lambda_s x| < 1$ for all s. By arbitrarily letting $x \in \left(0, \frac{1}{max\{|\lambda_s|\}}\right)$, this condition can be satisfied. Moreover, the results for $A_{s,q}\lambda_s^q$ are fixed values only dependent on s and q. To simplify the problem, let $B_{s,q} = A_{s,q}\lambda_s^q$. Then

$$\begin{aligned} &B_{s,q}(1 - \lambda_s x)^{-q} \\ &= B_{s,q}\left(1 + \frac{-q}{1!}(-\lambda_s x) + \frac{(-q)(-q-1)}{2!}(-\lambda_s x)^2 + \cdots \right. \\ &\left. + \frac{(-q)(-q-1)\ldots(-q-n+1)}{n!}(-\lambda_s x)^n + \cdots\right). \end{aligned}$$

Let the minus signs cancel each other, and further simplify the result using combinations:

$$\begin{aligned} &B_{s,q}(1 - \lambda_s x)^{-q} \\ &= B_{s,q}\left(1 + \frac{q}{1!}\lambda_s x + \frac{q(q+1)}{2!}(\lambda_s x)^2 + \cdots + \frac{q(q+1)\ldots(q+n-1)}{n!}(\lambda_s x)^n \right. \\ &\left. + \cdots\right) = B_{s,q}\left(1 + \binom{q}{1}\lambda_s x + \binom{q+1}{2}\lambda_s^2 x^2 + \cdots + \binom{q+n-1}{n}\lambda_s^n x^n + \cdots\right) \\ &= \sum_{n=0}^{\infty} B_{s,q}\binom{q+n-1}{n}\lambda_s^n x^n. \end{aligned}$$

Substitute the result from Taylor expansion to (5.3.3) gives

$$G(x) = \sum_{s=1}^{j} \sum_{q=1}^{p_s} \left[\sum_{n=0}^{\infty} B_{s,q} \binom{q+n-1}{n} \lambda_s^n x^n \right].$$

For any individual value of $n = 0, 1, 2 \ldots$, the coefficient of the corresponding x^n term is

$$a_n = \sum_{s=1}^{j} \sum_{q=1}^{p_s} B_{s,q} \binom{q+n-1}{n} \lambda_s^n. \tag{5.3.4}$$

5.4 Final solution to homogeneous linear recursions

To further simplify (5.3.4), notice that

$$\binom{q+n-1}{n} = \binom{n+q-1}{q-1} = \frac{(n+q-1)(n+q-2)\ldots(n+1)}{(q-1)!} = H_q(n), \tag{5.4.1}$$

so the binomial coefficient can be represented by a polynomial $H_q(n)$ with coefficients dependent on q and degree $= q-1$.

Now consider the whole summation about q. It is the sum of p_s polynomials with degree less than p_s. Collecting the terms $n^0, n^1, n^2 \ldots n^{p_s-1}$, and letting $C_{s,q}$ be the new coefficients of n^{q-1} gives

$$\sum_{q=1}^{p_s} B_{s,q} H_q(n) = \sum_{q=1}^{p_s} C_{s,q} n^{q-1} \tag{5.4.2}$$

which is a polynomial of n with degree $\leq q-1$. Substitute this result into (5.3.4), there is the general formula for the sequence:

$$a_n = \sum_{s=1}^{j} \sum_{q=1}^{p_s} C_{s,q} n^{q-1} \lambda_s^n. \tag{5.4.3}$$

For all $n = 0, 1, 2 \ldots$, and for $n = 0$ and $q = 1$ it is defined that $n^{q-1} = 1$. By representing whole (5.4.2) as $P_r(n)$, the result (5.4.3) can be turned into (2.2), which is given by another paper.

Because $\sum p_s = t$, the number of constants $C_{s,q}$ to be found is t. They can be found by plugging in the initial values $a_0 \sim a_{t-1}$ which is given by Assumption 4.2.1. A set of simultaneous linear equations can be solved and the constants can be found. If some of the linear equations are linearly dependent (which is uncommon), then the constants can still be found by plugging in $a_t, a_{t+1}, \ldots$ until a solution to $C_{s,q}$ is found.

Notice that the values of $C_{s,q}$ are independent of function $f(x)$, so it is unnecessary to solve for $f(x)$ to find the general solution.

5.5 A special case

If the roots of the characteristic equation of the recursive sequence are all distinct, which means $p_1 = p_2 = \cdots = p_t = 1$ and $j = t$, the general formula in (5.4.3) can be written as

$$a_n = \sum_{s=1}^{t}\sum_{q=1}^{1} C_{s,q}\, n^{q-1}\lambda_s^n = \sum_{r=1}^{t} C_r\, \lambda_r^n \tag{5.5.1}$$

which is exactly equation (2.3). Hence the hypothesis is true.

5.6 Applying the generating function in a homogeneous linear recursion

Consider the following order 3 recursive sequence: $a_0 = -1, a_1 = 1, a_2 = 3$, with the recursive formula $a_n = 5a_{n-1} - 3a_{n-2} - 9a_{n-3}$. To find the general solution, first identify the characteristic equation:

$$C(x) = x^3 - 5x^2 + 3x + 9 = (x+1)(x-3)^2 = 0.$$

So, clearly $\lambda_1 = \lambda_2 = 3, \lambda_3 = -1$. Then according to (5.4.3), the general formula must be

$$a_n = C_{1,1} \cdot 3^n + C_{1,2} \cdot n \cdot 3^n + C_{2,1} \cdot (-1)^n. \tag{5.6.1}$$

Substituting the initial values forms simultaneous equations:

$$\begin{cases} C_{1,1} + C_{2,1} = -1 \\ 3C_{1,1} + 3C_{1,2} - C_{2,1} = 1 \\ 9C_{1,1} + 18C_{1,2} + C_{2,1} = 3 \end{cases}.$$

Solving these equations gives

$$C_{1,1} = -\frac{1}{4}, C_{1,2} = \frac{1}{3}, C_{2,1} = -\frac{3}{4}.$$

Therefore, by substituting these values in (5.6.1), the general formula of the sequence is

$$a_n = -\frac{1}{4} \cdot 3^n + \frac{1}{3} \cdot n \cdot 3^n - \frac{3}{4}(-1)^n. \tag{5.6.2}$$

The following table show the first twenty terms of this sequence calculated by a computer using the recursive and general formula (5.6.2). The value a_n from two algorithms are the same for all $n = 0,1,2 \ldots 19$. Hence the correctness of formula (5.4.3) is confirmed.

Table 5.1 Computer verification of first twenty values calculated from two algorithms

n value	a_n from recursive formula	a_n from general formula	Are they equal?
0	-1	-1	YES
1	1	1	YES
2	3	3	YES
3	21	21	YES
4	87	87	YES
5	345	345	YES
6	1275	1275	YES
7	4557	4557	YES
8	15855	15855	YES
9	54129	54129	YES
10	182067	182067	YES
11	605253	605253	YES
12	1992903	1992903	YES
13	6510153	6510153	YES
14	21124779	21124779	YES
15	68157309	68157309	YES
16	218820831	218820831	YES
17	699509217	699509217	YES
18	2227667811	2227667811	YES
19	7070423925	7070423925	YES

6 Non-homogeneous Linear Recursions

Non-homogeneous linear recursions have the general recursive formula (4.1.3) where $c \neq 0$.

6.1 Recurrence function of non-homogeneous linear recursions

The non-homogeneous case is very similar to the homogeneous case. By trying the recurrence function (5.2.1) and considering $f(x) = G(x) \cdot R(x)$, the x^{ε} term of $f(x)$ for every $\varepsilon = t, t+1, t+2$ ... gives

$$1 \cdot a_{\varepsilon} x^{\varepsilon} - \sum_{r=1}^{t} m_r x^r \cdot a_{\varepsilon-r} x^{\varepsilon-r} = \left(1 \cdot a_{\varepsilon} - \sum_{r=1}^{t} m_r a_{\varepsilon-r}\right) x^{\varepsilon}$$

According to the recursive formula (4.1.3), simplifying the coefficient gives

$$1 \cdot a_{\varepsilon} - \sum_{r=1}^{t} m_r a_{\varepsilon-r} = c$$

which is the constant term. Therefore, it is proved that all coefficients of terms with degree $\geq t$ in $f(x)$ are c. Then, $f(x)$ can be expressed as

$$f(x) = q(x) + cx^t + cx^{t+1} + cx^{t+2} + \cdots = q(x) + c\sum_{n=t}^{\infty} x^n,$$

where $q(x)$ is a polynomial with maximum degree $< t$.

By arbitrarily letting $|x| < 1$, the infinite series in $f(x)$ can be regarded as a converging geometric sequence. So,

$$f(x) = q(x) + \frac{cx^t}{1-x} \tag{6.1.1}$$

which is a finite function. The polynomial $q(x)$ can be found by manually multiplying $R(x)$ and the first t terms in the generating function (Assumption 4.2.1).

6.2 Proving the existence of the solution by generating functions

By applying (6.1.1) to (5.1.2), the generating function of the non-homogeneous recursive equation is

$$G(x) = \frac{f(x)}{R(x)} = \frac{q(x) + \frac{cx^t}{1-x}}{R(x)} = \frac{(1-x)q(x) + cx^t}{(1-x)R(x)} \tag{6.2.1}$$

where $q(x)$ and $R(x)$ are finite functions. Because $q(x)$ is a polynomial with maximum degree $< t$, $R(x)$ is a polynomial with degree $= t$, the numerator polynomial has degree $= t$, and the denominator polynomial has degree $= t + 1$. Therefore, the generating function can be decomposed into a sum of partial fractions, which can be separately expanded by Taylor series.

Hence it is proved that a method (6.2.1) exists to solve non-homogeneous linear recursions. The Taylor expansion of the generating function can be very complicated, thus not discussed here.

6.3 An alternative method for non-homogeneous recursions

In most cases, this method can be implemented to solve non-homogeneous recursions. Notice that if a new sequence (b_n) is constructed and defined as $b_n = a_n - k$ for all n$\in \mathbb{N}_0$ where k is a constant, then the original recursion like (4.1.3) can be written in b_n as

$$(b_n + k) = \sum_{r=1}^{t} m_r(b_{n-r} + k) + c = \sum_{r=1}^{t} m_r b_{n-r} + k\sum_{r=1}^{t} m_r + c.$$

This is another linear recursion on (b_n). If the constant term is zero, then (b_n) is a homogeneous linear recursion which can be easily solved by (5.4.3). Collecting constant terms forms an equation:

$$0 = k\sum_{r=1}^{t} m_r - k + c = k\left(\sum_{r=1}^{t} m_r - 1\right) + c$$

which is a linear equation with one unknown k. The equivalent condition for this equation to have a solution to k is that the coefficient of k is not zero. Therefore, when

$$\sum_{r=1}^{t} m_r \neq 1, \tag{6.3.1}$$

the solution of k is

$$k = \frac{c}{1 - \sum_{r=1}^{t} m_r}.$$

Hence, by letting $b_n = a_n - \frac{c}{1-\sum_{r=1}^{t} m_r}$, the non-homogeneous linear recursion (a_n) can be translated into a homogeneous linear recursion (b_n) which can be solved using (5.4.3). The initial values of b_n can be directly calculated from initial values of (a_n) given by Assumption 4.2.1.

6.4 Comparison between two methods

The generating function method (6.2.1) can be applied to any non-homogeneous linear recursions. However, method 6.3 can only be applied to non-homogeneous linear recursions which satisfy (6.3.1). The generating function method involves a complicated Taylor expansion, while method 6.3 is much simpler.

Hence, it is suggested to use method 6.3 for recursions satisfying (6.3.1). Because the general solution to (b_n) is in the form of (5.4.3), the general formula of a_n can be written as

$$a_n = \frac{c}{1 - \sum_{r=1}^{t} m_r} + \sum_{s=1}^{j}\sum_{q=1}^{p_s} C_{s,q}\, n^{q-1}\lambda_s^n. \tag{6.4.1}$$

This can be directly solved by plugging the initial values and forming simultaneous equations, without first solving for (b_n).

For those recursions that do not satisfy (6.3.1), the generating function method can be used. In practice, a person is very unlikely to expand (6.2.1) correctly using Taylor series without the help of a computer. The following example is carefully chosen so that (6.2.1) can be manually expanded.

6.5 Applying the generating function method in a non-homogeneous linear recursion

Consider the non-homogeneous linear recursion $a_n = 2a_{n-1} - a_{n-2} + 1$ with initial values $a_0 = 2, a_1 = 3$. Obviously, the recursion does not satisfy (6.3.1), so the generating function method is used. First, identify the recurrence function according to (5.2.1) as

$$R(x) = x^2 - 2x + 1 = (x-1)^2.$$

Substituting initial values gives the first two terms of the generating function:

$$G(x) = 2 + 3x + \cdots$$

Multiplying $R(x)$ and $G(x)$ gives

$$f(x) = R(x) \cdot G(x) = 2 + (3-4)x + \cdots = 2 - x + \cdots$$

From the recursive formula, $t = 2$ and $c = 1$. Referencing (6.2.1), the product of $R(x)$ and $G(x)$ contributes to $q(x) + \frac{x^2}{1-x}$ where $q(x)$ is a polynomial with degree < 2. Therefore, the generating function can be written as a ratio of finite polynomials:

$$G(x) = \frac{f(x)}{R(x)} = \frac{2 - x + \frac{x^2}{1-x}}{(x-1)^2} = \frac{(2-x)(1-x) + x^2}{(1-x)^3} = \frac{2x^2 - 3x + 2}{(1-x)^3}.$$

To apply Taylor expansion, $G(x)$ can be decomposed using partial fractions:

$$G(x) = \frac{2}{1-x} - \frac{1}{(1-x)^2} + \frac{1}{(1-x)^3} = 2(1-\mathrm{x})^{-1} - (1-\mathrm{x})^{-2} + (1-\mathrm{x})^{-3}.$$

Then, by letting $|x| < 1$, Taylor expansion of the generating function is

$$\begin{aligned}
G(x) = {} & 2\left[1 + (-1)(-x) + \frac{(-1)(-2)}{2!}(-x)^2 + \cdots + \frac{(-1)(-2)\ldots(-n)\ldots}{n!}(-x)^n + \cdots\right] \\
& - \left[1 + (-2)(-x) + \frac{(-2)(-3)}{2!}(-x)^2 + \cdots + \frac{(-2)(-3)\ldots(-n-1)\ldots}{n!}(-x)^n + \cdots\right] \\
& + \left[1 + (-3)(-x) + \frac{(-3)(-4)}{2!}(-x)^2 + \cdots + \frac{(-3)(-4)\ldots(-n-2)}{n!}(-x)^n + \cdots\right] \\
= {} & 2\sum_{n=0}^{\infty} x^n - \sum_{n=0}^{\infty} (n+1)x^n + \sum_{n=0}^{\infty} \frac{(n+1)(n+2)}{2} x^n \\
= {} & \sum_{n=0}^{\infty} \left[2 - (n+1) + \frac{(n+1)(n+2)}{2}\right] x^n
\end{aligned}$$

$$= \sum_{n=0}^{\infty} \left(\frac{1}{2}n^2 + \frac{1}{2}n + 2\right) x^n.$$

Hence, the general formula for the sequence is

$$a_n = \frac{1}{2}n^2 + \frac{1}{2}n + 2. \quad (6.5.1)$$

Values of the first twenty terms of this recursive sequence are calculated by a computer using the recursive formula and formula (6.5.1). All corresponding values are the same for two algorithms. Hence, the correctness of formula (6.5.1) is confirmed.

Table 6.1 Computer verification of first twenty values calculated from two algorithms

n value	a_n from recursive formula	a_n from general formula	Are they equal?
0	2	2	YES
1	3	3	YES
2	5	5	YES
3	8	8	YES
4	12	12	YES
5	17	17	YES
6	23	23	YES
7	30	30	YES
8	38	38	YES
9	47	47	YES
10	57	57	YES
11	68	68	YES
12	80	80	YES
13	93	93	YES
14	107	107	YES
15	122	122	YES
16	138	138	YES
17	155	155	YES
18	173	173	YES
19	192	192	YES

7 Methods to translate special non-linear recursions into linear ones

It is impossible to formulate non-linear recursions to a clear fixed expression. Here discusses three methods that can translate certain non-linear recursions into linear recursions. According to sections 5 and 6, all linear recursions are solvable; therefore, these special non-linear recursions may have calculable general formulae.

7.1 Substitution method

This method often solves recursions that involve the term number n. If a non-linear order t recursive sequence (a_n) contains n in its recursive formula, and there is another linear order t recursive sequence (b_n) such that a relationship $h: \mathbb{C} \times \mathbb{N}_0 \to \mathbb{C}$ exists between a_n and b_n, then a 1-1 correspondence can be constructed between (a_n, n) and b_n. In other words, by letting

$$b_n = h(a_n, n), \tag{7.1.1}$$

b_n can be solved where the initial values can be found from initial values of a_n. Then, an equation like (7.1.1) can be formed and the general formula of a_n can be found.

The method above may seem messy and unclear, but the following application shows its simplicity and usage.

Example: for the recursive sequence with initial values $a_1 = 1$, $a_2 = 2$ ($n = 0$ is undefined) and recursive formula $a_n = \frac{2(n-1)}{n} a_{n-1} - \frac{n-2}{n} a_{n-2}$, find a general formula for a_n.

Solution: first simplify the recursive formula to

$$na_n = 2(n-1)a_{n-1} - (n-2)a_{n-2}.$$

By referencing (7.1.1), the new variable b_n should form a solvable linear recursion to apply the method. Therefore, by letting $b_n = na_n$, the formula can be translated into

$$b_n = 2b_{n-1} - b_{n-2}$$

which is a linear recursive formula for b_n. This recursion can be solved using formula (5.4.3), or more simply, by observing that $b_n - b_{n-1} = b_{n-1} - b_{n-2}$,so the sequence (b_n) is an arithmetic sequence. The initial values for b_n are $b_1 = 1 \cdot 1 = 1, b_2 = 2 \cdot 2 = 4$, so the arithmetic sequence has the general formula $b_n = 1 + (n-1) \cdot 3 = 3n - 2$. Hence by the relationship $b_n = na_n$, the general formula for a_n is

$$a_n = \frac{3n-2}{n}.$$

The biggest obstacle to applying this method is that a proper b_n defined as (7.1.1) cannot be easily found from the original recursive formula. It usually involves a lot of algebraic manipulation of the recursive formula and does not keep a fixed form. Nevertheless, this method is especially efficient when a suitable b_n is found. An important conclusion derived from this method is shown below.

Proposition: all recursions in the form of

$$a_n = \sum_{r=1}^{t} m_r a_{n-r} + dn + c \tag{7.1.2}$$

which satisfying (6.3.1), have a calculable general formula.

Proof: consider constructing

$$b_n = a_n - kn$$

where k is a constant to be found, so that (7.1.2) can be translated into a linear recursion like

$$b_n = \sum_{r=1}^{t} m_r b_{n-r} + c', \tag{7.1.3}$$

where c' is another constant term to be found. Simplifying equation (7.1.3) gives

$$a_n - kn = \sum_{r=1}^{t} \left[m_r\big(a_{n-r} - k(n-r)\big)\right] + c' = \sum_{r=1}^{t} m_r a_{n-r} - kn \sum_{r=1}^{t} m_r + k \sum_{r=1}^{t} m_r r + c',$$

which means

$$a_n = \sum_{r=1}^{t} m_r a_{n-r} + \left(1 - \sum_{r=1}^{t} m_r\right) kn + k \sum_{r=1}^{t} m_r r + c'.$$

comparing the result above with (7.1.2) and letting the corresponding terms involving n and constant term equal forms two equations:

$$\begin{cases} \left(1 - \sum_{r=1}^{t} m_r\right) k = d \\ k \sum_{r=1}^{t} m_r r + c' = c \end{cases}$$

which have the solutions

$$\begin{cases} k = \dfrac{d}{1 - \sum_{r=1}^{t} m_r} \\ c' = c - \dfrac{d \sum_{r=1}^{t} m_r r}{1 - \sum_{r=1}^{t} m_r} \end{cases}$$

given the recursion satisfies (6.3.1). Hence, by letting $b_n = a_n - kn$, a new linear recursion (7.1.3) can be formed from the original recursive formula. Because condition (6.3.1) is satisfied, b_n has a calculable general formula in the form of (6.4.1). Let the general formula be $b_n = F(n)$, then the general formula for a_n is

$$a_n = F(n) + kn.$$

This completes the proof. ■

7.2 Logarithmic method

This method usually solves recursions that involve multiplication in the recursive formula.

First, a definition of logarithms of complex numbers must be given. The natural logarithm of a non-zero complex number $z = re^{i\theta}$ is

$$\ln z = \ln re^{i\theta} = \ln r + i\theta$$

where $\ln r$ is defined as the natural logarithm of the positive real number r. Therefore, the domain of logarithms is any non-zero complex numbers.

Now consider the recursive formula

$$a_n = k \cdot a_{n-1}^{p_1} \cdot a_{n-2}^{p_2} \dots a_{n-t}^{p_t} = k \prod_{r=1}^{t} a_{n-r}^{p_r} \tag{7.2.1}$$

where k is a non-zero constant and p_r are power indexes for each term. If none of the initial values are zero, all terms in the sequence are not zero so the logarithm on both sides of (7.2.1) is defined. By taking natural logarithms on both sides, the continuous multiplication can be turned into continuous sums, which fits the linear recursive formula

$$\ln a_n = \sum_{r=1}^{t} p_r \ln a_{n-r} + \ln k.$$

Therefore, the sequence $(\ln a_n)$ is a linear recursion, and has a calculable general formula according to section 6. Let this general formula be $\ln a_n = F(n)$, then the general formula of a_n is $a_n = e^{F(n)}$.

More generally, if there is a bijection $h: \mathbb{C} \to \mathbb{C}$ that the original recursive formula of a sequence can be turned in the form of

$$h(a_n) = k \prod_{r=1}^{t} [h(a_{n-r})]^{p_r} \tag{7.2.2}$$

where $h(a_n) \neq 0$ for all n, then there must be a general for a_n using the logarithmic method.

Taking natural logarithms on both sides gives

$$\ln h(a_n) = \sum_{r=1}^{t} p_r \ln h(a_{n-r}) + \ln k$$

which forms a linear recursion for sequence $(\ln h(a_n))$. By using the initial values of a_n given by Assumption 4.2.1, the initial values of $\ln h(a_n)$ can be easily calculated. Therefore, there is a general formula for $\ln h(a_n)$, say $\ln h(a_n) = F(n)$. Given that h is a bijection, the inverse of h must exist. Hence, the general formula for a_n is

$$a_n = h^{-1}\left[e^{F(n)}\right].$$

The recursion (7.2.1) is a special case for recursion (7.2.2) where h is the identity function on $\mathbb{C}$ or $h(z) = z$. Although the logarithmic method is useful when solving recursions in the form of (7.2.2), it is sometimes hard to factorize the recursive formula and deduce a bijection h by observing.

Example: let the recursive formula be $a_n = a_{n-1}a_{n-2} - 2a_{n-1} - 2a_{n-2} + 6$ and the initial values be $a_0 = 6, a_1 = 7$. Find a general formula for a_n.

Solution: rewrite the recursive formula as

$$a_n - 2 = a_{n-1}a_{n-2} - 2a_{n-1} - 2a_{n-2} + 4 = (a_{n-1} - 2)(a_{n-2} - 2).$$

So, by letting $h(z) = z - 2$, the equation above can be written as

$$h(a_n) = h(a_{n-1}) \cdot h(a_{n-2})$$

which is in the form of (7.2.2). Because $h(a_0) = 4, h(a_1) = 5$, all terms in this sequence are not zero by induction. Therefore, logarithms are defined for all $h(a_n)$. Taking natural logarithms gives

$$\ln h(a_n) = \ln h(a_{n-1}) + \ln h(a_{n-2})$$

which forms a homogeneous linear recursion. Let

$$b_n = \ln h(a_n) = \ln(a_n - 2) \Leftrightarrow a_n = e^{b_n} + 2 \tag{7.2.3}$$

the linear recursion formula is

$$b_n = b_{n-1} + b_{n-2}$$

with initial values $b_0 = \ln 4\,, b_1 = \ln 5$. The characteristic equation of this recursion is

$$x^2 - x - 1 = 0$$

which have roots $\lambda_1 = \frac{1+\sqrt{5}}{2}, \lambda_2 = \frac{1-\sqrt{5}}{2}$. According to (5.5.1) the general formula for b_n must in the form of

$$b_n = C_1\left(\frac{1+\sqrt{5}}{2}\right)^n + C_2\left(\frac{1-\sqrt{5}}{2}\right)^n.$$

Substituting b_0 and b_1 gives two simultaneous equations:

$$\begin{cases} C_1 + C_2 = b_0 = \ln 4 \\ C_1 \cdot \dfrac{1+\sqrt{5}}{2} + C_2 \cdot \dfrac{1-\sqrt{5}}{2} = b_1 = \ln 5 \end{cases}.$$

Solving these equations gives

$$C_1 = \frac{\sqrt{5}\ln 2 - \ln 2 + \ln 5}{\sqrt{5}}, C_2 = \frac{\sqrt{5}\ln 2 + \ln 2 - \ln 5}{\sqrt{5}}.$$

So, the general formula of b_n is found. Hence by (7.2.3), the general formula of a_n is

$$a_n = \exp\left[C_1\left(\frac{1+\sqrt{5}}{2}\right)^n + C_2\left(\frac{1-\sqrt{5}}{2}\right)^n\right] + 2. \qquad (7.2.4)$$

The results calculated from the computer is not shown because the terms grow enormous very quickly due to composited exponential functions ($a_{10} \approx 3.2 \times 10^{22}$). The general formula (7.2.4) for a sequence of integers includes irrational numbers and even transcendental numbers (the exponential function and natural logarithms in C_1 and C_2); however, as long as the procedure to solve the recursion is correct, the general formula would always produce the correct integers.

7.3 Trigonometric method

This method can solve special recursions with recursive formula in the form of trigonometrical identities. If some terms and constants in a recursive formula can be regarded as trigonometric function values so that the whole recursive formula forms a trigonometrical identity, then the problem may be solved by substituting trigonometrical values in and simplifying the identity. Here shows an example.

Example: the recursive sequence (a_n) has the recursive formula $a_n = \frac{2a_{n-1}}{1-a_{n-1}^2}$ with initial value $a_0 = \sqrt{3}$. Find the general formula for a_n.

Solution: Observe that the right-side of the formula could form a double angle formula for tangents. So, by letting $a_n = \tan\theta_n$ the original recursive formula can be written as

$$\tan \theta_n = \frac{2 \tan \theta_{n-1}}{1 - \tan^2 \theta_{n-1}} = \tan 2\theta_{n-1}.$$

According to the properties of the tangent function,

$$\tan \theta_n = \tan 2\theta_{n-1} \Leftrightarrow \theta_n = 2\theta_{n-1} + k\pi, k \in \mathbb{Z}.$$

To simplify the solution, arbitrarily let $k = 0$, so

$$\theta_n = 2\theta_{n-1} \implies \tan \theta_n = \tan 2\theta_{n-1}.$$

Therefore, the original recursion can be translated into the geometric sequence $\theta_n = 2\theta_{n-1}$ where the initial value is $\theta_0 = \arctan\sqrt{3} = \frac{\pi}{3}$. The general formula for θ_n is

$$\theta_n = 2^n \frac{\pi}{3}.$$

Hence, the general formula for a_n is

$$a_n = \tan \frac{2^n \pi}{3}.$$

This method has many limitations although it is effective against special cases. Domains and codomains of the substituting trigonometric functions must be considered carefully because trigonometric functions are not bijective on $\mathbb{C}$.

8 Conclusion

This paper solved homogeneous and non-homogeneous linear recursions using generating functions, and provided three methods to translate specific types of non-linear recursions into linear ones.

The general formula of a homogeneous linear recursion can be found by first identifying its recurrence function from the coefficients of the recursive formula. The generating function of the recursion can be expressed as a ratio of the recurrence function and another finite function. Then the original generating function can be found using Taylor expansion on the rational function. Simplifying the result gives the form of the general formula, which is a sum of powers of the roots of the characteristic equation. The constants in the general formula can be found by plugging in some initial values and solving simultaneous equations. Most non-homogeneous linear recursions can be easily turned into corresponding linear ones, and thus has a general formula in the form of a sum of powers of the characteristic equation and a known constant. For other types of non-homogeneous linear recursions, its general formula can be found by manually finding the finite rational expression of its generating function and expanding the generating function using Taylor series. Three types of non-linear recursions can be translated into linear ones using substitution, logarithmic and trigonometric methods, which may involve ingenious identification of substituting parts. By finding the general formula from a recursive sequence, one can calculate any term in the sequence without calculating all its predecessors.

There were some limitations in this research. The methods provided to solve linear recursions have relatively low computational efficiency. In most cases, solving linear recursions of order t always involves inverting an order t matrix of complex numbers, which grows in complexity as t becomes large. For other non-linear recursions, the Taylor expansion of their generating functions can be insanely complicated. The methods to translate non-linear recursions into linear recursions can only solve those with special recursive formulae, and require intelligence or accidental hints to identify the correct method.

However, this paper indeed provided detailed methods to solve all linear recursions using the generating function and thus reduced the work needed to find a specific term in the recursive sequence. The method descriptions and proofs were fairly rigorous and comprehensible. It also summarised three useful methods to solve some special non-linear recursions.

To improve the computational efficiency, the algorithm involving Bell polynomials is suggested to solve linear recursions. The concept of generating functions and recurrence functions can be used to solve other non-linear recursions in further research and development. For example, the general formula of Catalan numbers can be found by letting the recurrence function be the same as its generating function. Further studies may also focus on the solution to other non-linear recursions which are not discussed in this paper.

9 References

[1] Boman, B. M., Dinh, T. N., Decker, K. E. I. T. H., Emerick, B., Raymond, C. H. R. I. S. T. O. P. H. E. R., & Schleiniger, G. (2017). Why do fibonacci numbers appear in patterns of growth in nature. *A model for tissue renewal based on asymmetric cell division. Fibonacci Q*, *55*(5), 30-41.

[2] Power, S. A., & Shannon, A. G. (2019). Natural Mathematics, the Fibonacci Numbers and Aesthetics in Art. *Journal of Advances in Mathematics*, *17*.

[3] Chavan, A., & Suryawanshi, C. (2020). Correlation of fibonacci sequence and golden ratio with its applications in engineering and science. *International Journal of Engineering and Management Research*.

[4] Fehér, K., Szilágyi, B., & Halmos, B. (2018). Golden Ratio and Fibonacci sequence in pentagonal constructions of medieval architecture. *YBL Journal of Built Environment*, *6*(1), 37-46.

[5] Spickerman, W. R., & Joyner, R. N. (1984). Binet's formula for the recursive sequence of order k. *Fibonacci Q*, *22*, 327-331.

[6] Glaister, P. (1995). Fibonacci power series. *The Mathematical Gazette*, *79*(486), 521-525.

[7] Patrick, D. (2022). In *Intermediate counting and probability* (pp. 201–202). essay, San Diego, CA: AoPS Incorporated.

[8] Mignotte, M. (1975). A note on linear recursive sequences. *Journal of the Australian Mathematical Society*, *20*(2), 242-244.

[9] Withers, C. S., & Nadarajah, S. (2015). Solutions of linear recurrence equations. *Applied Mathematics and Computation*, *271*, 768–776

[10] El-Afifi, M. (2004). On the recursive sequence $x_{n+1} = \frac{\alpha+\beta x_n}{Bx_n+Cx_{n-1}}$. *Applied mathematics and computation*, *147*(3), 617-628.

[11] DeVault, R., Kent, C., & Kosmala, W. (2003). On the Recursive Sequence $x_{n+1} = p + \frac{x_{n-k}}{x_n}$. *Journal of Difference Equations and Applications*, *9*(8), 721-730.