The Automatic Solution of Recurrence Relations

I. Linear Recurrences of Finite Order with Constant Coefficients*

Roberto Bagnara (bagnara@cs.unipr.it)
Alessandro Zaccagnini (alessandro.zaccagnini@unipr.it)
Tatiana Zolo (zolo@cs.unipr.it)

Department of Mathematics
University of Parma
Italy

*This work has been partly supported by the Italian-Spanish *Azione Integrata* IT 229 "Advanced Development Environments for Logic Programs", by the University of Parma's FIL scientific research project (ex 60%) "Pure and Applied Mathematics", and by the MURST projects "Aggregate- and number-reasoning for computing: from decision algorithms to constraint programming with multisets, sets, and maps", "Constraint Based Verification of Reactive Systems", and "Zeta and L Functions and Diophantine Problems in Number Theory."

Series Foreword

Complexity analysis aims at the derivation of upper and lower bounds to the complexity of algorithms, processes and data structures. The results of such analyses, which may be part or wholly automated, can be used, say, to decide whether mobile agents should be allowed to run in a given context, assist programmers in reasoning about the behavior of programs, guide applications of optimized program transformations, and discover efficiency bugs that are otherwise very difficult to detect.

Recurrence relations play an important role in the field of complexity analysis since complexity measures of, e.g., programs, can usually be very elegantly expressed by means of such relations. Therefore there is significant demand for efficient software systems capable of solving, with a high degree of precision, systems of recurrence relations. Moreover, to be really useful, the solvers need to be fully automatic, obtaining such solutions without human intervention.

Although the mathematical and computing literature describes several techniques and software for solving recurrences, these do not fulfill all of the above requirements; on the one hand, many of the methods assume interaction with a human operator and only deal with a rather restricted range of cases, while, on the other hand, the fully automated tools that are available only provide quite crude approximations of the exact solutions.

The PURRS project (*Parma University's Recurrence Relation Solver*, see `http://www.cs.unipr.it/purrs/`) is working at improving the state of the art in this field. The aim of the project is to create a software library that provides the services needed for efficiently computing the solution or an approximation of the solution of a system of recurrence relations that arise in performing fully automated complexity analysis.

Finding exact solutions and/or tight approximations in closed form in an acceptable timescale for a large class of recurrence relations is a challenging task; such an objective requires solutions to a host of unresolved theoretical and practical problems. The current series of papers is devoted to the presentation of *all* the mathematics behind the PURRS project. As our aim is to provide both rigor and thoroughness, the series will not only include original work but also accounts of known results, describing any useful extensions or modifications.

Abstract

We describe algorithmic techniques for the efficient solution of a wide class of linear recurrences of finite order with constant coefficients. We give an outline of the underlying theory both from an abstract and a more concrete point of view, in an attempt to convey the general principles as clearly as possible yet providing a well marked path for the implementation. In particular, the presentation is thorough and reasonably self-contained, covering topics such as the automatic solution of polynomial equations and efficient, exact symbolic summation of some special functions.

Contents

1 Introduction

In this part of the series we deal only with linear recurrences of finite order with constant coefficients, in the cases when it is possible to give an exact closed formula for the solution without resorting to complex methods for summation in closed form.[1]

Linear recurrences of finite order with constant coefficients (LRFOCCs, for short) arise frequently in complexity analysis. For instance, several common algorithms on lists and other kinds of sequences work by manipulating a few elements on, say, the front of the sequence, and invoke themselves recursively to get the job done on the entire list. In the analysis of functional and logic programs this is most often the case, and linear recurrences with constant coefficients of order 1 or 2 cover the vast majority of the recurrences an automatic complexity analysis tool has to solve. LRFOCCs of higher order are also important since, as we will see, they arise in the solution of systems of LRFOCCs of order 1.

Luckily enough, LRFOCCs are relatively easy to solve efficiently, especially if their order is small. In this paper we give an outline of the theoretical results underlying the automatic solution of LRFOCCs. This is presented both from an abstract and a more concrete point of view, in an attempt to convey the general principles as clearly as possible yet providing a well marked path for the implementation. In fact, the illustrated techniques (or variants of them) are used in the `PURRS` system (*Parma University's Recurrence Relation Solver*, see `http://www.cs.unipr.it/purrs/`). Care is taken to provide a presentation that is thorough, reasonably self-contained, and can be followed both by mathematicians and computer scientists.

The plan of the paper is as follows: Section 2 introduces preliminary concepts and notations; Section 3 is the main one and presents solution techniques for linear recurrences with constant coefficients of order 1, 2 and higher; Section 4 shows how to solve systems of linear recurrences of order 1 with constant coefficients by reducing them to a set of autonomous recurrences (that is, independent of one another) of higher order; Section 5 discusses related work in this field while Section 6 concludes the main body of the paper. The appendix contains the tools required to turn the solution methods into actual algorithms. Procedures for the solution of polynomial equations of degree up to 4 are described in Appendix A. The exact symbolic summation of an important class of functions (linear combinations of products of polynomials and exponential functions) is treated in detail in Appendix B. Finally, several examples are collected in Appendix C so as to show the solution algorithms at work.

[1] Such methods are best viewed within the context of linear recurrences with variable coefficients [5]. Approximate summation will be the topic of another paper of this series [4]. We will discuss generalized recurrences like those arising from the analysis of "divide et impera" algorithms in [3].

2 Preliminaries

Throughout the paper we will use the following conventions: we denote by $\mathbb{N}$ the set of *nonnegative* integers, so that $0 \in \mathbb{N}$; we denote by $\mathbf{C}^{(k)}$ the k-th iterate of the operator $\mathbf{C}$; the value of an empty sum is 0.

We assume that the reader is familiar with the basic properties of polynomials (roots and their multiplicity, division with quotient and remainder, greatest common divisor), and has a working knowledge of calculus and of linear algebra (eigenvalues, eigenvectors, characteristic equation of a matrix, vector spaces, dimension, kernel of a linear operator).

We let $\mathbb{C}[n]$ denote the *ring of polynomials* in the *indeterminate* n with coefficients in $\mathbb{C}$. This is the smallest set (with respect to inclusion) that contains the set of complex numbers $\mathbb{C}$, the indeterminate $n \notin \mathbb{C}$, and is closed with respect to addition and multiplication. The rings $\mathbb{Z}[n]$, $\mathbb{Q}[n]$ and $\mathbb{R}[n]$ are defined similarly. Let $p \in \mathbb{C}[n] \setminus \{0\}$: we denote by $\deg(p)$ the *degree* of the polynomial p and by $\operatorname{lead}(p)$ its *leading coefficient*, that is, the coefficient of the monomial $n^{\deg(p)}$. We also let $\operatorname{coeff}_j(p)$ denote the coefficient of the monomial of degree j of the polynomial p. With this notation, $\operatorname{lead}(p) = \operatorname{coeff}_{\deg(p)}(p)$. A more abstract definition of polynomial is given in Mignotte and Ştefănescu [19].

We are particularly interested in deriving, when possible, mathematical formulas that are *in closed form*, a concept that we define informally as follows: a mathematical expression denoting a function $f \colon \mathbb{N} \to \mathbb{C}$ is said to be in *elementary closed form* if it is, *syntactically*, a linear combination of a fixed number, r, say, of products of polynomials in $\mathbb{C}[n]$ and (complex) exponentials. The number r must be an absolute constant, that is, it must be independent of all variables and parameters of the problem. As an example, the left hand side of the identity

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n$$

is not in closed form, while the right hand side is.

Generally speaking, we focus on effectively computable methods that can constitute the basis of practical, efficient algorithms. As a consequence, very often we refer to *syntax* instead of *semantics*. Moreover, in an attempt to keep the discussion more fluid, we do that implicitly. Thus, when we say "p is a polynomial" we actually mean "p is an expression that, syntactically, is a polynomial" or "p is, syntactically, of the form $a_0\lambda^k + a_1\lambda^{k-1} + \cdots + a_{k-1}\lambda + a_k$ where ..." Of course, sticking to pure syntactic equality would be too restricting. Hence, mathematical formulas are evaluated modulo a computable equality theory EQ. While it is reasonable to assume that this theory captures associativity and commutativity of '+' and '·' and other simple properties of elementary operations, EQ is otherwise left unspecified.

We will write $a \doteq b$ when we want to emphasize that the expressions a and b are equal under EQ. It must be stressed that, while '$\doteq$' is decidable, ordinary mathematical equality is not. For example, it is well-known that it is undecidable whether an expression involving polynomials and the sine function is equal to zero [7, 21, 22].

We prefer to use the term "recurrence relations," but in the literature it is also possible to encounter the term *difference equations*. The latter is mainly used in a mathematical context, in order to give prominence to the tie with the differential equations. In fact, difference equations can be considered the discrete analogue of differential equations, and there exists a collection of mathematical tools, called *difference calculus*, which is quite similar to *differential calculus*. Throughout the paper we will use both terms.

3 Linear Recurrences of Finite Order with Constant Coefficients

A *linear recurrence of finite order with constant coefficients* is a recurrence of the form

$$x_n = a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_k x_{n-k} + p(n), \tag{3.1}$$

for $n \geq k$, where k is a fixed positive integer that is called *order of the recurrence relation* (3.1). Here and throughout we implicitly assume that $a_k \neq 0$, for otherwise the order would be smaller. In Equation (3.1) the *coefficients* a_j are real or complex numbers. The function p is defined on $\mathbb{N} \cap [k, +\infty)$, and is the *non-homogeneous part* of the recurrence (3.1). If $p \doteq 0$ (that is, p vanishes identically from a syntactical point of view), we say that the recurrence (3.1) is *homogeneous*. Every recurrence of the form (3.1) is given with a set of k *initial values* x_0, x_1, ..., x_{k-1}, which we assume to be known.

In the next section we present the general solution technique for this class of recurrences, whereas Section 3.2 presents the problem and its solution in a more abstract setting. The problem is then reconsidered under a more concrete point of view. Sections 3.3 and 3.4 are devoted to the "easy" cases of linear recurrences with constant coefficients of order 1 and 2, respectively; the description of tailor-made, more efficient solution methods for these subclasses is justified by the observation that these cases appear to be the most frequent ones in the context of complexity analysis. Section 3.5 deals with the general case of linear recurrences with constant coefficients of any finite order, while in Section 3.6 we show how to solve special recurrences of higher order in a more efficient way.

3.1 The Basic Solution Technique

We associate to any recurrence like (3.1) the homogeneous recurrence

$$g_n = a_1 g_{n-1} + a_2 g_{n-2} + \cdots + a_k g_{n-k}. \tag{3.2}$$

We first solve this recurrence by means of the *characteristic equation* that, by definition, is the polynomial equation

$$\lambda^k = a_1 \lambda^{k-1} + \cdots + a_{k-1}\lambda + a_k. \tag{3.3}$$

We use throughout the notation g for the homogeneous recurrence associated to a general recurrence of finite order, and λ for the complex variable in the associated characteristic equation. In general, the characteristic equation (3.3) has k complex roots $\lambda_1, \ldots, \lambda_k$. If these roots are all distinct (we also say that they are *simple*), the general solution of the homogeneous recurrence (3.2) is given by

$$g_n = \alpha_1 \lambda_1^n + \cdots + \alpha_k \lambda_k^n, \tag{3.4}$$

where $\alpha_1, \ldots, \alpha_k$ are complex numbers. It is quite easy to see that $\alpha_i \lambda_i^n$ is a solution of (3.2), for $i = 1, \ldots, k$. This is an immediate consequence of the fact that each λ_i is a root of (3.3). Furthermore, it is also easy to see that the sum of any two solutions of (3.2) is again a solution, so that the right hand side of (3.4) gives indeed solutions of the homogeneous recurrence (3.2).

If the characteristic equation (3.3) has multiple (or *repeated*) roots, we collect them: say they are λ_1 with multiplicity $\mu_1, \ldots, \lambda_r$ with multiplicity μ_r, where the λ_j's are different complex numbers, and the μ_j's are positive integers. In this case the general solution is

$$g_n = \sum_{j=1}^{r} (\alpha_{j,0} + \alpha_{j,1} n + \cdots + \alpha_{j,\mu_j-1} n^{\mu_j - 1}) \lambda_j^n. \tag{3.5}$$

Of course, (3.5) contains (3.4) as the special case where $r = k$ and $\mu_j = 1$ for $j = 1, \ldots, r$. The proof of this result can be found in Kelley & Peterson [14, Theorem 3.7]. Note that assuming that $a_k \neq 0$ is tantamount to saying that $\lambda = 0$ is *not* a solution of (3.3), but it is clear from either (3.4) or (3.5) that this does not really matter.

The results above mean that *any* recurrence relation of the type (3.2) has a solution of the type (3.4) or (3.5), for suitable values of the coefficients α, which are determined by means of the initial values $g_0, \ldots, g_{k-1}$.

Once the general solution of the homogeneous equation (3.2) has been found, we have to solve the complete equation (3.1): by linearity, it is sufficient to find *any* solution of the complete equation (we call this a *particular solution*), sum the general solution of the homogeneous equation and impose that the initial conditions hold. This is the so-called *superposition principle* (see Kelley & Peterson [14, Theorem 3.3]): we show why it is true by means of an analogy. In order to know a straight line in the usual plane, it is sufficient to know which straight line through the origin it is parallel to (and this is uniquely determined) and *any* point on the original line, it does not matter which one. In our case we need to know the complete solution of the homogeneous equation (3.2) (which plays the role of the line through the origin), and any solution of the complete, non-homogeneous equation (3.1).

3.2 Abstract Setting of the Problem

We now restate the problem we are interested in in a more abstract manner, and fully characterize the set of non-homogeneous parts of a recurrence of the type (3.1) that possess a solution whose elementary closed form can be computed without transcendental summation methods that will be the topic of [5]. Some remarks on how to actually and efficiently compute these closed forms can be found in Appendix B.

In the language of linear algebra, we are trying to solve the equation $Ax = p$, where A is a *linear operator*, so we first compute the *kernel* of A which we denote by $\ker(A)$ (that is, we solve the equation $Ax = 0$). The theory of linear operators guarantees that the general solution of $Ax = p$ is given by the sum of an element of $\ker(A)$ and any solution of the complete equation. Furthermore, the dimension of $\ker(A)$ is precisely the same as the order of the recurrence, so that (3.5) gives *all* solutions of the homogeneous recurrence (3.2): in fact, there are exactly k unknown coefficients, and the functions $n^i \cdot \lambda_j^n$, for $j = 1, \ldots, r$ and $i = 0, \ldots, \mu_j - 1$, are linearly independent over $\mathbb{C}$ provided that the values of the λ_j's are different. This means that the dimension of the vector space generated by these functions is precisely k. More concretely, we are talking about the space $\mathcal{X}$ of complex-valued sequences defined over $\mathbb{N}$, which, endowed with the obvious operations, can be considered a vector space over $\mathbb{C}$: if $(x_n)_{n\in\mathbb{N}}$, $(y_n)_{n\in\mathbb{N}} \in \mathcal{X}$, and λ, $\mu \in \mathbb{C}$, then

$$\lambda \cdot (x_n)_{n\in\mathbb{N}} + \mu \cdot (y_n)_{n\in\mathbb{N}} \stackrel{\text{def}}{=} (\lambda x_n + \mu y_n)_{n\in\mathbb{N}} \in \mathcal{X}.$$

For a recurrence of order k as in (3.1), the operator A is defined by

$$A\bigl((x_n)_{n\in\mathbb{N}}\bigr) \stackrel{\text{def}}{=} \bigl(x_n - (a_1 x_{n-1} + a_2 x_{n-2} + \cdots + a_k x_{n-k})\bigr)_{n\in\mathbb{N}},$$

with the convention that, if $0 \le n < k$, the above definition has to be replaced by 0. It is easily seen that A is *linear*, that is

$$A\bigl(\lambda \cdot (x_n)_{n\in\mathbb{N}} + \mu \cdot (y_n)_{n\in\mathbb{N}}\bigr) = \lambda \cdot A\bigl((x_n)_{n\in\mathbb{N}}\bigr) + \mu \cdot A\bigl((y_n)_{n\in\mathbb{N}}\bigr),$$

for any $(x_n)_{n\in\mathbb{N}}$, $(y_n)_{n\in\mathbb{N}} \in \mathcal{X}$, and λ, $\mu \in \mathbb{C}$. Thus $\ker(A)$ is exactly the set of solutions of (3.2). The fact that A is linear is the property that ensures that the superposition principle holds.

Summing up, the solution of a homogeneous linear recurrence of finite order with constant coefficient is completely straightforward from a theoretical point of view, the main difficulty being in the solution of the characteristic equation (3.3). On the contrary, finding the particular solution of the general equation (3.1) is non-trivial, and an elementary closed form can only be obtained in comparatively few cases: in general we have recourse to summation formulæ and other approximations that will be explained in [4].

The ideas expressed above are now rigorously formalized.

Definition 3.1 (Shift and identity operators.) *Let $\mathcal{X}$ denote the complex vector space of all sequences defined over the set $\mathbb{N}$ of natural numbers. We denote by $(x_n)_{n\in\mathbb{N}}$ its generic element (sometimes dropping the subscript for brevity). We consider the* shift operator $\mathbf{E}\colon \mathcal{X} \to \mathcal{X}$ *defined by*

$$\mathbf{E}\big((x_n)_{n\in\mathbb{N}}\big) \stackrel{\text{def}}{=} (x_{n+1})_{n\in\mathbb{N}}.$$

We also consider the identity operator $\mathbf{I}\colon \mathcal{X} \to \mathcal{X}$ *(which does nothing).*

We need linear combinations of iterates of these two operators: we define $\Delta(\alpha) \stackrel{\text{def}}{=} \mathbf{E} - \alpha\mathbf{I}$, and introduce the family

$$\mathcal{F} \stackrel{\text{def}}{=} \big\{\, \Delta(\alpha_1) \circ \cdots \circ \Delta(\alpha_k) \;\big|\; k \in \mathbb{N} \setminus \{0\},\, \alpha_1, \dots, \alpha_k \in \mathbb{C} \,\big\}.$$

We remark that the operators Δ commute. With this notation, we first rewrite (3.1) in the form

$$x_{n+k} - \big(a_1 x_{n+k-1} + a_2 x_{n+k-2} + \cdots + a_k x_n\big) = p(n+k),$$

and then

$$\big(\mathbf{E}^{(k)} - a_1\mathbf{E}^{(k-1)} - \cdots - a_{k-1}\mathbf{E} - a_k\mathbf{I}\big)(x_n) = p(n+k).$$

If p vanishes identically, the formal analogy with the characteristic equation (3.3) is clear. It is also clear that we would like to write the linear operator on the left as a composition of $\Delta(\alpha)$ for suitable complex numbers α. This corresponds to solving the characteristic equation, for once we know that its roots are λ_1 with multiplicity μ_1, ..., λ_r with multiplicity μ_r, we have

$$\mathbf{E}^{(k)} - a_1\mathbf{E}^{(k-1)} - \cdots - a_{k-1}\mathbf{E} - a_k\mathbf{I} = \Delta(\lambda_1)^{(\mu_1)} \circ \cdots \circ \Delta(\lambda_r)^{(\mu_r)}. \tag{3.6}$$

Here we exploit the commutativity of these operators. In order to solve *homogeneous recurrences* we need to understand the *kernel* of these operators.

Lemma 3.2 *For any $\alpha \in \mathbb{C}$ and any integer $d \in \mathbb{N}$ we have the set identity*

$$\ker\big(\Delta(\alpha)^{(d+1)}\big) = \Big\{\, \big(q(n)\alpha^n\big)_{n\in\mathbb{N}} \;\Big|\; q \in \mathbb{C}[n],\, q = 0 \text{ or } \deg(q) \le d \,\Big\}.$$

Proof. First, assume that $q \in \mathbb{C}[n]$ is a polynomial of degree $\leq d$, and note that

$$\Delta(\alpha)\big(q(n)\alpha^n\big) = \alpha\big(q(n+1) - q(n)\big)\alpha^n.$$

Since $\deg\big(q(n+1) - q(n)\big) = d - 1$, we see that $\Delta(\alpha)^{(d)}\big(q(n)\alpha^n\big) = k\alpha^n$ for some constant k, and then it is easy to see that $\Delta(\alpha)(k\alpha^n)$ vanishes identically.

In the other direction, we assume that $\alpha \neq 0$, for otherwise the result is trivial. Now let $(x_n)_{n\in\mathbb{N}} \in \ker\big(\Delta(\alpha)^{(d+1)}\big)$. If $d = 0$, this means that x_n satisfies the recurrence $x_{n+1} = \alpha x_n$, whose solution is $x_n = \alpha^n x_0$. For $d > 0$ we write

$$0 = \big(\Delta(\alpha)^{(d+1)}\big)(x_n) = \big(\Delta(\alpha)^{(d)}\big)\big[\big(\Delta(\alpha)\big)(x_n)\big].$$

By the inductive hypothesis, there is a polynomial $p \in \mathbb{C}[n]$ of degree $\leq d-1$ such that

$$\big(\Delta(\alpha)\big)(x_n) = p(n)\alpha^n.$$

The proof is therefore complete if we show that there exists a polynomial $q \in \mathbb{C}[n]$ of degree $\deg(p) + 1$ such that

$$\alpha\big(q(n+1) - q(n)\big) = p(n)$$

identically. This is readily done by invoking Lemma B.1 on page 37 and observing that[2]

$$n_{(d)} = \frac{1}{d+1}\big[(n+1)_{(d+1)} - n_{(d+1)}\big],$$

since this ensures that we can write every monomial in the polynomial p in the desired form. □

The last displayed formula is the discrete analogue of the identity $x^d = (d+1)^{-1}\frac{\mathrm{d}}{\mathrm{d}x}x^{d+1}$. Indeed, as we already mentioned above, there is an analogy between recurrence relations and differential equations: in our case the role of the derivative is played by the *forward difference operator* $\Delta\colon \mathcal{X} \to \mathcal{X}$, which, by definition, has the following property:

$$\Delta\big((x_n)_{n\in\mathbb{N}}\big) \stackrel{\text{def}}{=} \big(x_{n+1} - x_n\big)_{n\in\mathbb{N}}.$$

In other words, $\Delta = \Delta(1)$. As in the case of differential equations, it can be proved (see Kelley & Peterson [14, §2.2]) that if the non-homogeneous part of the linear recurrence p belongs to the rather special class of functions $\mathcal{A}$ defined below, then a particular solution of (3.1) can be found in a suitable subclass of $\mathcal{A}$. The analogy with differential equations, though, can not be pushed too far: in the fairly elementary cases treated in this paper most things have an exact counterpart in the field of differential equations, an

[2] $n_{(k)}$ is the *falling factorial function* defined by $n_{(k)} \stackrel{\text{def}}{=} k!\binom{n}{k} = n(n-1)\cdots(n-k+1)$.

important exception being the order reduction method of Section 3.6, but in general (for recurrences with variable coefficients, to mention one instance) results and techniques can both be very different.

Theorem 3.3 *For all elements p of the set $\mathcal{A}$ defined by*

$$\mathcal{A} \stackrel{\text{def}}{=} \bigcup_{\mathbf{C} \in \mathcal{F}} \ker(\mathbf{C}),$$

it is possible to find an elementary closed formula for

$$P(n) \stackrel{\text{def}}{=} \sum_{k=0}^{n} p(k).$$

Proof. By Lemma 3.2 every element $p \in \mathcal{A}$ has the form $p(n) = r_1(n)\alpha_1^n + \cdots + r_k(n)\alpha_k^n$, where $r_i \in \mathbb{C}[n]$ for $i = 1, \ldots, k$, and the α_i are distinct complex numbers. It is sufficient to prove the theorem for an expression of the form $p(n) = r(n)\alpha^n$. Now $P(n)$ satisfies the first order recurrence $x_n = x_{n-1} + p(n)$ and $p \in \ker\bigl(\Delta(\alpha)^{(\deg(r)+1)}\bigr)$ by Lemma 3.2. Therefore $P \in \ker\bigl(\Delta(\alpha)^{(\deg(r)+1)}\bigr) \circ \Delta(1)$, so that $P \in \mathcal{A}$ by Lemma 3.2 again. $\square$

For ease of future reference, we state the following immediate Corollary.

Corollary 3.4 *Let $p \in \mathcal{A}$ have the form $p(n) = r(n)\alpha^n$, where $r \in \mathbb{C}[n]$ has degree d, and $\alpha \in \mathbb{C} \setminus \{0\}$, and let $\lambda_1, \ldots, \lambda_r$ be the* distinct *solutions of the characteristic equation* (3.3) *associated to the recurrence* (3.1), *with respective multiplicities $\mu_1, \ldots, \mu_r$. Then any solution of the recurrence* (3.1) *belongs to*

$$\begin{cases} \ker\bigl(\Delta(\lambda_1)^{(\mu_1)} \circ \cdots \circ \Delta(\lambda_r)^{(\mu_r)} \circ \Delta(\alpha)^{(d+1)}\bigr), & \textit{if } \alpha \notin \{\lambda_1, \ldots, \lambda_r\}; \\ \ker\bigl(\Delta(\lambda_1)^{(\mu_1+d+1)} \circ \cdots \circ \Delta(\lambda_r)^{(\mu_r)}\bigr), & \textit{if } \alpha = \lambda_1. \end{cases}$$

We give some examples in Appendix C.

We just saw that the elements of the set $\mathcal{A}$ satisfy some *homogeneous* recurrence relation of finite order with constant coefficients, and that polynomials, exponentials and their products all belong to $\mathcal{A}$. But, since we allow *complex values* of α, we see that the functions $\sin(n\theta)$, $\cos(n\theta)$ and their products also belong to $\mathcal{A}$, and in general, any linear combination of terms of the form $n^k\rho^n\cos(n\theta)$ and $n^k\rho^n\sin(n\theta)$, for $k \in \mathbb{N}$, ρ, $\theta \in \mathbb{R}$. Indeed, $n^k\rho^n\cos(n\theta)$ and $n^k\rho^n\sin(n\theta)$ both belong to $\ker\bigl(\Delta(\rho e^{i\theta})^{(k+1)}\bigr)$, and also to the kernel of the real operator

$$\mathbf{C} = \bigl(\mathbf{E}^{(2)} - 2(\rho\cos\theta)\mathbf{E} + \rho^2\mathbf{I}\bigr)^{(k+1)}.$$

See Appendix B for the description of algorithms for finding the particular solution of a non-homogeneous recurrence in the cases just stated, and Appendix A.8 for the computation of the coefficients α in the relation (3.5).

3.3 Special Case: Recurrences of Order 1

The next few subsections are devoted to the special but important case of recurrences of order 1, because we want to emphasize again some of the features of the solution of recurrences without having to deal with irrelevant details.

It is quite easy to prove by induction that the solution of the first-order recurrence

$$x_n = \alpha x_{n-1} + p(n),$$

is given by the formula

$$x_n = \alpha^n x_0 + \sum_{k=1}^{n} \alpha^{n-k} p(k). \tag{3.7}$$

We consider the special case $x_0 = 0$, and set

$$\begin{cases} y_n = \alpha y_{n-1} + p(n), & \text{for } n \geq 1; \\ y_0 = 0, \end{cases}$$

whence, by (3.7),

$$y_n = \sum_{k=1}^{n} \alpha^{n-k} p(k). \tag{3.8}$$

Clearly, we just have to solve the last recurrence since then $x_n = \alpha^n x_0 + y_n$. We also remark that, by linearity, if p is the sum of several functions, $p_1 + p_2$, say, by the superposition principle (or, more simply, by (3.8)) it is enough to solve two distinct recurrences, one with p_1 in place of p, the other with p_2, and then add the results. See Appendix C.1 for some concrete examples.

3.3.1 Polynomials and Exponentials

We insist on an important theoretical point: we often need the closed formula for the sum of the product of a polynomial and an exponential. In some cases, the answer is the product of the same exponential function and another polynomial of the same degree, and in some cases the degree increases: see the abstract discussion in Section 3.2 and in particular Corollary 3.4; for concrete cases, see Examples C.2, C.3, C.4, C.6, C.9, C.10.

Concretely, we are looking for a closed formula for the sum in (3.8) for some $\alpha \in \mathbb{C}$, where p is $q(n)\alpha^n$ and q is a polynomial. In this case we see that the closed formula has again the shape of the product of a polynomial and α^n, but the degree of the polynomial has increased by 1. We see why it has to be so by considering, more generally, the recurrence

$$x_n = \alpha x_{n-1} + q(n)\beta^n,$$

where α and β are fixed, non-zero complex numbers, and q is a polynomial. Using (3.8), we look for a closed formula for the sum

$$\sum_{k=1}^{n} \alpha^{n-k}\beta^k q(k) = \alpha^n \sum_{k=1}^{n} \Big(\frac{\beta}{\alpha}\Big)^k q(k) = \alpha^n \sum_{k=1}^{n} \gamma^k q(k), \tag{3.9}$$

say. We claim that the case $\gamma = 1$ (that is, $\alpha = \beta$) is quite special. In fact, if r is a polynomial of degree d, say, then

$$\gamma^k r(k) - \gamma^{k-1} r(k-1) = \gamma^k \big(r(k) - \gamma^{-1} r(k-1)\big).$$

The degree of the polynomial on the right is d if and only if $\gamma \neq 1$, and is $d-1$ otherwise. To see why this fact is relevant, assume that we can find a polynomial r such that $q(k) = r(k) - \gamma^{-1} r(k-1)$ identically. Now we can compute a closed formula for the sum in (3.9), far right:

$$\begin{aligned}\sum_{k=1}^{n} \gamma^k q(k) &= \sum_{k=1}^{n} \gamma^k \big(r(k) - \gamma^{-1} r(k-1)\big) = \sum_{k=1}^{n} \gamma^k r(k) - \sum_{k=1}^{n} \gamma^{k-1} r(k-1) \\ &= \sum_{k=1}^{n} \gamma^k r(k) - \sum_{k=0}^{n-1} \gamma^k r(k) = \gamma^n r(n) - r(0).\end{aligned}$$

This implies that the closed formula for the right hand side of (3.9) contains a polynomial of the same degree as q if $\alpha \neq \beta$, and of higher degree if $\alpha = \beta$. The facts just stated are analogous to the computation of derivatives and primitives. We say more on this topic in the Examples quoted at the beginning of this section.

Lemma 3.5 *Let $q \in \mathbb{C}[x]$ be any polynomial, and $\gamma \in \mathbb{C} \setminus \{0\}$. There exists a polynomial $r \in \mathbb{C}[x]$ such that*

$$q(x) = r(x) - \gamma^{-1} r(x-1)$$

for all $x \in \mathbb{C}$. Furthermore, if $\gamma \neq 1$ then $\deg(r) = \deg(q)$, while if $\gamma = 1$ then $\deg(r) = \deg(q) + 1$.

Proof. If q is constant, $q(x) = a$, say, then we may take $r(x) = a\gamma/(\gamma - 1)$ if $\gamma \neq 1$, and $r(x) = ax$ if $\gamma = 1$. If $d = \deg(q) \geq 1$ we proceed by induction: let $a_d = \mathrm{lead}(q)$, and set $s(x) = a_d \gamma x^d/(\gamma-1)$ if $\gamma \neq 1$, and $s(x) = a_d x^{d+1}/d$ if $\gamma = 1$. Let $t(x) = q(x) - (s(x) - \gamma^{-1} s(x-1))$. A simple check reveals that $\deg(t) < \deg(q)$, and therefore, by the induction hypothesis, there exists a polynomial $r_1 \in \mathbb{C}[x]$ such that $t(x) = r_1(x) - \gamma^{-1} r_1(x-1)$ for all $x \in \mathbb{C}$. Hence $r(x) = s(x) + r_1(x)$ is the desired polynomial. □

Notice that, if $q \in \mathbb{R}[x]$, then $r \in \mathbb{R}[x]$ and, similarly, $q \in \mathbb{Q}[x]$ implies $r \in \mathbb{Q}[x]$.

3.4 Special Case: Recurrences of Order 2

The case of recurrences of order 2 is also interesting because it is now possible to have characteristic equations with multiple roots, and again we can put emphasis on quite important facts in a fairly simple situation.

We now want to solve general recurrences of order 2, like

$$x_n = \alpha x_{n-1} + \beta x_{n-2} + p(n) \tag{3.10}$$

where p is a function defined over the natural numbers. Here we assume that $\beta \neq 0$, for otherwise the recurrence would have order 1. We set

$$\begin{cases} g_n = \alpha g_{n-1} + \beta g_{n-2}, & \text{for } n \geq 2; \\ g_0 = 1, \\ g_1 = \alpha, \end{cases} \tag{3.11}$$

for the *fundamental solution*. This is readily solved by means of the technique described in Section 3. It is not difficult to prove by induction that, for $n \geq 2$, the solution of (3.10) is

$$x_n = g_{n-1}x_1 + \beta g_{n-2}x_0 + \sum_{k=2}^{n} g_{n-k}p(k). \tag{3.12}$$

Here and below, the solutions, as they stand, are only valid for $n \geq 2$. We observe that if $\lambda_1 \neq \lambda_2$, then g_n has the explicit expression

$$g_n = \frac{\lambda_1^{n+1} - \lambda_2^{n+1}}{\lambda_1 - \lambda_2}.$$

If $\lambda_1 = \lambda_2$, that is $\alpha^2 + 4\beta = 0$, the formula is different, though the idea is similar; see Example C.6 on page 45. In fact (as in the case of the Fibonacci numbers, see Example C.5 on page 44) we have

$$g_n = a\lambda_1^n + b\lambda_2^n,$$

for suitable complex numbers a and b, and for $n = 0$ and $n = 1$ we find

$$\begin{cases} a + b = 1, \\ a\lambda_1 + b\lambda_2 = \alpha, \end{cases} \qquad \text{whence} \qquad \begin{cases} a = \lambda_1(\lambda_1 - \lambda_2)^{-1}, \\ b = -\lambda_2(\lambda_1 - \lambda_2)^{-1}. \end{cases}$$

Thus, by (3.12), for $n \geq 2$ we have

$$\begin{aligned} x_n &= g_{n-1}x_1 + \beta g_{n-2}x_0 + \sum_{k=2}^{n} \frac{\lambda_1^{n+1-k} - \lambda_2^{n+1-k}}{\lambda_1 - \lambda_2} p(k) \qquad (3.13) \\ &= g_{n-1}x_1 + \beta g_{n-2}x_0 + \frac{\lambda_1^{n+1}}{\lambda_1 - \lambda_2} \sum_{k=2}^{n} \lambda_1^{-k} p(k) - \frac{\lambda_2^{n+1}}{\lambda_1 - \lambda_2} \sum_{k=2}^{n} \lambda_2^{-k} p(k). \end{aligned}$$

If p is in the set $\mathcal{A}$ defined in Section 3.2, the sums above also belong to $\mathcal{A}$, and there is an algorithm to compute the closed formula (see Appendix B). A similar result also holds for linear recurrences of any finite order with constant coefficients: see Section 3.5 for the details.

3.4.1 "Guessing" the Solution in Special Cases

We now show how to solve directly equation (3.10) when p is a polynomial of degree d, say: we first compute the homogeneous part $x_1 g_{n-1} + \beta x_0 g_{n-2}$ and seek a particular solution of the complete equation (a different approach is based on the techniques of Appendix B). Corollary 3.4 implies that if p is a polynomial and $\lambda = 1$ is not a solution of the characteristic equation, then the particular solution itself is a polynomial of the same degree as p, whose coefficients we have to find. This leads to the solution of a set of linear equations with $d+1$ unknowns. If $\lambda = 1$ is a solution of the characteristic equation of multiplicity μ, we look for a polynomial of degree $d + \mu$.

Example 3.6 *Solve*

$$\begin{cases} x_n = \alpha x_{n-1} + \beta x_{n-2} + p(n), & \textit{for } n \geq 2; \\ x_0 = a, \\ x_1 = b. \end{cases} \tag{3.14}$$

The homogeneous fundamental equation is

$$\begin{cases} g_n = \alpha g_{n-1} + \beta g_{n-2}, & \text{for } n \geq 2; \\ g_0 = 1, \\ g_1 = \alpha. \end{cases}$$

For the time being, we assume that $\lambda = 1$ is *not* a solution of the characteristic equation, which amounts to assuming that $\alpha + \beta \neq 1$. By Corollary 3.4 we know that there is a polynomial q such that

$$x_n = \big(b + q(1)\big) g_{n-1} + \beta\big(a + q(0)\big) g_{n-2} - q(n),$$

where q has to be computed as a function of p, which is the non-homogeneous part of the recurrence. If $\lambda = 1$ is not a solution of the characteristic equation, q can be determined by imposing that it has the same degree as p and satisfies the functional equation

$$p(n) = \alpha q(n-1) + \beta q(n-2) - q(n). \tag{3.15}$$

This can be verified by induction. In order to find q it is necessary to solve a set of linear equations with $d+1$ unknowns, d being the degree of p.

Example 3.7 *Solve* (3.14) *for* $\alpha = \beta = 1$ *and* $p(n) = n^2$.

We have to find real numbers a, b and c such that

$$n^2 = \big(a(n-1)^2 + b(n-1) + c\big) + \big(a(n-2)^2 + b(n-2) + c\big) - \big(an^2 + bn + c\big)$$

holds identically, that is, collecting and simplifying,

$$(a-1)n^2 + (b-6a)n + (5a-3b+c) = 0.$$

By the identity principle for polynomials, this implies that $a = 1$, $b = 6$, $c = 13$. Note (in agreement with Lemma 3.5) that if $\lambda = 1$ were a solution of the characteristic equation of multiplicity μ, the right hand side of (3.15) would be a polynomial of degree $\deg(q) - \mu$: that is why, if $\mu > 0$, we have to look for a polynomial q of degree $\deg(p) + \mu$. We give more examples in Appendix C.2: see, in particular, Examples C.11–C.14.

3.5 Higher-Order Linear Recurrences with Constant Coefficients

The theory for the higher-order cases is analogous, but the computations are obviously heavier. Notice though that it is fairly easy to recognize special recurrences of order $k > 2$ such as $x_n = a(n)x_{n-k} + p(n)$, which can be treated as a (much easier and faster to solve) first order recurrence. The point is that in this case x_n does not depend on x_{n-1}, x_{n-2}, …, x_{n-k+1}, so that it is possible to use a simpler method. A similar remark holds for $x_n = x_{n-2} + x_{n-4}$, which should be treated as an order 2 recurrence. For more on this topic, see Section 3.6.

We now give the general solution of (3.1); we introduce the *fundamental solution* of the associated homogeneous equation, which is

$$\begin{cases} g_n = a_1 g_{n-1} + a_2 g_{n-2} + \cdots + a_k g_{n-k}, & \text{for } n \geq k; \\ g_0 = 1, & \\ g_n = a_1 g_{n-1} + a_2 g_{n-2} + \cdots + a_{n-1} g_1 + a_n g_0, & \text{for } 1 \leq n < k. \end{cases}$$

The general solution of (3.1) is then given by the formula

$$x_n = \sum_{i=k}^{n} g_{n-i} p(i) + \sum_{i=0}^{k-1} g_{n-i} \Big(x_i - \sum_{j=1}^{i} a_j x_{i-j} \Big), \tag{3.16}$$

which is valid for $n \geq k$ and can be proved by induction. This is essentially equation (5) of Cohen & Katcoff [10], and it contains as special cases both (3.7) and (3.12). We remark that the two sums in (3.16) correspond naturally to the non-homogeneous part p and to the initial conditions, respectively. Furthermore, the quantities in brackets on the right depend only on the initial values and on the coefficients of the recurrence.

3.6 Order Reduction

Suppose that we want to solve the recurrence

$$x_n = \alpha x_{n-k} + p(n),$$

where $k \geq 2$ is a fixed integer. As we said above, this is essentially a recurrence of order 1. We remark that the variable n runs over integers that are congruent mod k. Hence, setting $r \stackrel{\text{def}}{=} n \bmod k$, $n \stackrel{\text{def}}{=} km + r$ and $y_m^{(r)} \stackrel{\text{def}}{=} x_{km+r}$, we consider the k recurrences

$$\begin{cases} y_m^{(r)} \stackrel{\text{def}}{=} \alpha y_{m-1}^{(r)} + p(km + r), & \text{for } m \geq 1; \\ y_0^{(r)} \stackrel{\text{def}}{=} x_r, \end{cases} \tag{3.17}$$

which are actually of order 1 and can be solved by means of the techniques described in Section 3.3. It is not really necessary to solve k recurrences (one for each possible value of r) since, in a sense, the recurrences (3.17), which may actually be distinct as Example 3.8 below shows, can be considered as a single, parametric recurrence. Here we use the term "parametric" in a quite narrow sense, since the parameter r is only allowed to lie in the finite set $\{0, 1, \ldots, k-1\}$. Elsewhere in this document, a parameter is usually allowed to take any real (or indeed complex) value, without restrictions.

The advantage over the general method is best seen in the light of a few examples. Similar remarks, of course, apply for recurrences of the form $x_n = \alpha x_{n-k} + \beta x_{n-2k} + p(n)$.

Example 3.8 *Solve the recurrence*

$$x_n = x_{n-2} + n.$$

The general method gives the solution

$$x_n = \frac{1}{2}\bigl(1+(-1)^n\bigr)x_0 + \frac{1}{2}\bigl(1-(-1)^n\bigr)x_1 + \frac{1}{4}n^2 + \frac{1}{2}n + \frac{3}{8}\bigl((-1)^n - 1\bigr). \tag{3.18}$$

On the other hand, the substitution above leads to the recurrence

$$\begin{cases} y_m^{(r)} \stackrel{\text{def}}{=} y_{m-1}^{(r)} + 2m + r, & \text{for } m \geq 1; \\ y_0^{(r)} \stackrel{\text{def}}{=} x_r, \end{cases}$$

which is parametric in the parameter r in the above sense. Its solution is

$$y_m^{(r)} = m^2 + (1+r)m + y_0^{(r)},$$

so that, substituting again $m = \frac{1}{2}(n-r)$, we obtain

$$x_n = x_r + \frac{1}{4}n^2 + \frac{1}{2}n - \frac{1}{4}r(r+2), \tag{3.19}$$

where $r = n \bmod 2 \in \{0, 1\}$. It is immediate to check that this is the same as (3.18). It is also clear that (3.19) is more readable than (3.18), where there is a complex part depending on the initial conditions.

The following example is perhaps even more striking.

Example 3.9 *Solve*

$$x_n = x_{n-3}.$$

The reduction method yields at once $x_n = x_{n \bmod 3}$. The standard method entails the solution of the characteristic equation $\lambda^3 - 1 = 0$, whose roots are $\lambda_1 = 1$, $\lambda_2 = e^{2\pi i/3}$ and $\lambda_3 = e^{-2\pi i/3}$. For simplicity, we write $\omega = e^{2\pi i/3}$, so that the roots are 1, ω, $\overline{\omega} = \omega^2$. We also need to solve a system of three linear equations in order to find suitable constants a_1, a_2 and a_3 so that

$$a_1 + a_2\omega^n + a_3\omega^{2n} = x_n \qquad \text{for } n = 0,\, 1,\, 2.$$

The solution is therefore

$$\begin{aligned} x_n &= \frac{1}{3}(x_0 + x_1 + x_2) + \frac{1}{3}(x_0 + \omega^2 x_1 + \omega x_2)\omega^n + \frac{1}{3}(x_0 + \omega x_1 + \omega^2 x_2)\omega^{2n} \\ &= (1 + \omega^n + \omega^{2n})\frac{x_0}{3} + (1 + \omega^{n+2} + \omega^{2n+1})\frac{x_1}{3} + (1 + \omega^{n+1} + \omega^{2n+2})\frac{x_2}{3}. \end{aligned}$$

Though both forms of the solution are equally valid, there is no doubt that the latter is much more complicated than the former, and that it is more difficult to find. It is also plain that the reduction of the order allows one to solve $x_n = x_{n-k}$ for fixed integral, positive k, with the same amount of computation, regardless of k, whereas the standard method, even in this almost trivial case, requires computations whose cost increases with k. Other examples that stress the convenience of this method are in Appendix C.4.

The remainder of this section is devoted to the transformation of the solution obtained by means of the order reduction into the general solution that would be obtained using the standard method: in other words, we show in general how to deduce the exact form of the solution (3.18) from (3.19).

There are essentially two problems: expressing the terms x_r where $r = n \bmod k$, the initial conditions that appear in the "reduced" solution, and expressing r itself. The examples above suggest that the first problem is solved introducing suitable linear combinations of k-th roots of unity (the complex solutions of the polynomial equation $z^k = 1$), as the following Lemma proves. The same result solves also the second problem.

Lemma 3.10 *For any positive integer k and any $r \in \{0, \ldots, k-1\}$ there is a linear combination of the k-th roots of unity $\omega_1, \ldots, \omega_k$, with coefficients $\alpha_1(r), \ldots, \alpha_k(r)$ such that for any $n \in \mathbb{Z}$ we have*

$$\alpha_1(r)\omega_1^n + \cdots + \alpha_k(r)\omega_k^n = \begin{cases} 1 & \text{if } n \equiv r \bmod k, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. It is sufficient to take $\alpha_j(r) = \omega_j^{-r}/k$, since $\omega_1^n + \cdots + \omega_k^n = k$ if k divides n and is 0 otherwise. $\square$

It is now possible to express the part depending on the initial conditions. Let

$$\eta_{r,k}(n) \stackrel{\text{def}}{=} \frac{1}{k}\sum_{h=1}^{k} \omega_h^{n-r},$$

so that $\eta_{r,k}(n) = 1$ if $n \equiv r \bmod k$, and $\eta_{r,k}(n) = 0$ otherwise, by the observation in the proof of Lemma 3.10. Therefore we have

$$x_r = \sum_{s=0}^{k-1} \eta_{s,k}(n) x_s. \tag{3.20}$$

It might seem at first sight that the right hand side does *not* depend on r, but on second thoughts it will be seen that $\eta_{r,k}$ is periodic with period k, since the ω's are k-th roots of unity, so that the expression on the right depends only on $n \bmod k = r$, and the remark after the definition of the functions $\eta_{r,k}$ shows that (3.20) is actually correct.

Another important consequence of Lemma 3.10 is the fact that

$$n \bmod k = \sum_{s=0}^{k-1} s\eta_{s,k}(n) = \frac{1}{k}\sum_{s=0}^{k-1} s \sum_{h=1}^{k} \omega_h^{n-s} = \frac{1}{k}\sum_{h=1}^{k} \omega_h^{n} \sum_{s=0}^{k-1} s\omega_h^{-s}. \tag{3.21}$$

This expression is really too complicated to be of any practical use, but a neat trick using some identities of Appendix B.2 will help us in finding an equivalent expression which is much easier to handle, providing a feasible solution to our second problem.

Lemma 3.11 *Let k be a positive integer, and let $1 = \omega_1, \omega_2, \ldots, \omega_k$ be the complex k-th roots of unity. For any integer $n \in \mathbb{Z}$ we have*

$$n \bmod k = \frac{1}{2}(k-1) + \sum_{h=2}^{k} \frac{\omega_h}{1-\omega_h}\,\omega_h^n.$$

Proof. Let $z = \omega_h^{-1}$ for $h \neq 1$ and recall that $\omega_h^k = 1$ by definition. Since $z \neq 1$, by (B.5) with $j = 1$ we have

$$\sum_{s=0}^{k-1} sz^s = \frac{z}{(1-z)^2}\bigl(1 - kz^{k-1} + (k-1)z^k\bigr) = \frac{k\omega_h}{1-\omega_h}.$$

Therefore, by (3.21)

$$n \bmod k = \frac{1}{k}\sum_{h=1}^{k} \omega_h^n \sum_{s=0}^{k-1} s\omega_h^{-s} = \frac{1}{2}(k-1) + \sum_{h=2}^{k} \frac{\omega_h}{1-\omega_h}\,\omega_h^n,$$

as stated. $\square$

3.7 Summary of the Method

Let us summarize the overall solution methodology for linear recurrences of finite order with constant coefficients: The very first step is to recognize whether we can apply the order reduction method, which is always beneficial as it allows to work with polynomials of lower degree. After order reduction, the homogeneous part of the recurrence is solved by the characteristic equation method, which involves finding the roots of a polynomial along with their multiplicities. Now, to obtain the general solution of the complete equation, it is sufficient to find any one of its solutions (which is then called the *particular solution*), add it to the general solution of the homogeneous equation found with the previous step, and impose that the initial conditions hold. For the recurrences of the first or the second order when the characteristic equation has simple roots, a *non-closed* formula for the solution is simply given by (3.7) and (3.13), respectively. For higher-order recurrences (including order 2 when the characteristic equation has a multiple root), the general solution is given by (3.16), which requires the symbolic solution of a system of equations. Whatever the order is, finding a *closed-form* solution requires the computation of symbolic sums that, depending on the inhomogeneous term of the recurrence, can be arbitrarily difficult when not plainly impossible (see Appendix B for the symbolic summation of an important class of functions). In some special cases, i.e., when the non homogeneous term p belongs to the class $\mathcal{A}$ defined in Section 3.2, we have shown an alternative way for "guessing" the particular solution. Indeed, when $p \in \mathcal{A}$, Lemma 3.2 and Corollary 3.4 characterize the shape of the solution: in other words we know that it is the (sum of several terms of the form) product of a polynomial and an exponential function, where we know the degree of the polynomial and the base of the exponential function. The actual values of the coefficients in the polynomial can be determined by imposing that the solution really satisfies the initial recurrence. A similar strategy is used in Examples 3.7 and C.11–C.14.

In conclusion, the above procedure allows to reduce the problem of solving LRFOCCs to (1) the solution of polynomial equations and (2) the computation of closed forms for symbolic sums. In the appendix we illustrate some useful techniques for dealing with (1) and (2); other techniques exist and may be worth exploring, since any advancement in tackling problems (1) and (2) will extend the class of recurrences that can be solved by this method.

4 Systems of Linear Recurrences

In this section, we show how to solve systems of linear recurrences of finite order with constant coefficients. We first deal with systems of recurrences where each recurrence is of order 1. These have been studied, e.g., by Cohen

& Katcoff [10] and by Kelley & Peterson [14]. We then turn our attention to the solution of systems of recurrences of any order and show that they can all be reconducted to systems of order-1 recurrences.

4.1 Systems of Recurrences of Order 1

Sets of k recurrence relations with constant coefficients of order 1 may be tackled in two different, but essentially equivalent, ways, that are quite similar to the solution of sets of linear equations: one can either solve for one of the "variables" (a recurrence in our case) and then plug the result in the remaining equations, iterating if necessary in order to get only one equation, or see the problem from a more abstract point of view, the system being a linear operator from the set of k-tuples of recurrences to itself, so that the general theory developed in Section 3.2 applies. The first approach yields k recurrences that contain only one "variable," each recurrence being of order k, all with the same characteristic equation of degree k; the second one needs the solution of the characteristic equation of the linear operator referred to above, which has degree k and is exactly the same as the characteristic equation of the linear operator: that is why the two approaches are essentially equivalent. We give examples of both approaches below. We begin with a set of two first-order recurrences with constant coefficients in two variables, and we show how to transform it into two second-order recurrences, each with only one variable. Suppose we want to solve

$$\begin{cases} x_n = ax_{n-1} + by_{n-1} + p(n), \\ y_n = cx_{n-1} + dy_{n-1} + q(n). \end{cases} \tag{4.1}$$

The second equation with n replaced by $n-1$ yields $cx_{n-2} = y_{n-1} - dy_{n-2} - q(n-1)$. We now replace n by $n-1$ in the first equation and multiply throughout by c. Plugging these two relations into the second equation above, after some simplifications we obtain

$$y_n = (a+d)y_{n-1} - (ad-bc)y_{n-2} + q(n) - aq(n-1) + cp(n-1).$$

A similar argument gives

$$x_n = (a+d)x_{n-1} - (ad-bc)x_{n-2} + p(n) - dp(n-1) + bq(n-1).$$

Observe that the equations (4.1) imply that $x_1 = ax_0 + by_0 + p(1)$ and that $y_1 = cx_0 + dy_0 + q(1)$. Notice also that the coefficients of the new recurrences are the *trace* and the opposite of the *determinant* of the matrix of the coefficients in the system (4.1). This discussion means that we can always transform a system of two linear recurrences of the first order with constant coefficients into a pair of linear recurrences of the second order, with the variables decoupled.

The procedure is quite general, and, as a more difficult example, we show how to reduce a system of 5 equations of order 1 to a single equation of order 5 (this is Example 12 in Table II of the paper by Cohen & Katcoff [10]). Actually, one should repeat the same procedure in order to get 5 equations, one involving a single variable, but it is also possible to find the solution to one and then substitute in the original problem, repeating until all equations have been solved.

Example 4.1 *Solve*

$$\begin{cases} a_n = a_{n-1} + b_{n-1} + 2^n, \\ b_n = 5a_{n-1} + c_{n-1}, \\ c_n = -5a_{n-1} + d_{n-1} - 3^n, \\ d_n = -4a_{n-1} + e_{n-1}, \\ e_n = 4a_{n-1}. \end{cases} \tag{4.2}$$

The last equation implies that $e_{n-1} = 4a_{n-2}$. We substitute this into (4.2) and, omitting from now on the last equation, we find

$$\begin{cases} a_n = a_{n-1} + b_{n-1} + 2^n, \\ b_n = 5a_{n-1} + c_{n-1}, \\ c_n = -5a_{n-1} + d_{n-1} - 3^n, \\ d_n = -4a_{n-1} + 4a_{n-2}. \end{cases}$$

The last equation shows that $d_{n-1} = -4a_{n-2} + 4a_{n-3}$, and we substitute again:

$$\begin{cases} a_n = a_{n-1} + b_{n-1} + 2^n, \\ b_n = 5a_{n-1} + c_{n-1}, \\ c_n = -5a_{n-1} - 4a_{n-2} + 4a_{n-3} - 3^n. \end{cases}$$

We obtain $c_{n-1} = -5a_{n-2} - 4a_{n-3} + 4a_{n-4} - 3^{n-1}$, whence

$$\begin{cases} a_n = a_{n-1} + b_{n-1} + 2^n, \\ b_n = 5a_{n-1} - 5a_{n-2} - 4a_{n-3} + 4a_{n-4} - 3^{n-1}. \end{cases}$$

Finally $b_{n-1} = 5a_{n-2} - 5a_{n-3} - 4a_{n-4} + 4a_{n-5} - 3^{n-2}$, so that we are left with

$$a_n = a_{n-1} + 5a_{n-2} - 5a_{n-3} - 4a_{n-4} + 4a_{n-5} - 3^{n-2} + 2^n.$$

The characteristic equation factors as $(\lambda - 1)^2(\lambda + 1)(\lambda - 2)(\lambda + 2) = 0$, so that this recurrence is easily solved with the technique described in Section 3. Once a closed formula for a_n is found, it can be used to find the recurrences satisfied by the other variables, until all the original equations are solved.

This mechanical procedure shows quite clearly that we can eliminate one variable at the cost of increasing the order of the other equations by 1.

Kelley & Peterson [14, §4.1] describe a different method. Given the set of k linear recurrences with constant coefficients

$$\begin{cases} x_n^{(1)} = a_{11}x_{n-1}^{(1)} + \cdots + a_{1k}x_{n-1}^{(k)} + p_1(n), \\ \dots \\ x_n^{(k)} = a_{k1}x_{n-1}^{(1)} + \cdots + a_{kk}x_{n-1}^{(k)} + p_k(n), \end{cases}$$

we introduce the *vector equation*

$$\mathbf{x}_n = A\mathbf{x}_{n-1} + \mathbf{p}(n), \tag{4.3}$$

where

$$\mathbf{x}_n \stackrel{\text{def}}{=} \begin{bmatrix} x_n^{(1)} \\ \dots \\ x_n^{(k)} \end{bmatrix}, \quad A \stackrel{\text{def}}{=} \begin{bmatrix} a_{11} & \dots & a_{1k} \\ \dots & \dots & \dots \\ a_{k1} & \dots & a_{kk} \end{bmatrix}, \quad \mathbf{p}(n) \stackrel{\text{def}}{=} \begin{bmatrix} p_1(n) \\ \dots \\ p_k(n) \end{bmatrix}. \tag{4.4}$$

As above, we first study the *homogeneous* equation $\mathbf{x}_n = A\mathbf{x}_{n-1}$, whose solution is, obviously,

$$\mathbf{x}_n = A^n\mathbf{x}_0. \tag{4.5}$$

We recall that, by definition, an *eigenvector* of A relative to the *eigenvalue* λ is a non-zero vector $\mathbf{v}$ with the property that $A\mathbf{v} = \lambda\mathbf{v}$. Hence it is immediate that $\mathbf{x}_n = \lambda^n\mathbf{v}$ satisfies (4.5) with the initial condition $\mathbf{x}_0 = \mathbf{v}$. If A has k linearly independent eigenvectors (which certainly is the case if it has k distinct eigenvalues) then there is an invertible matrix M such that $A = M^{-1}BM$, where B is *diagonal*, with the eigenvalues of A on the main diagonal. Let's denote by $\lambda_1, \dots, \lambda_k$ these eigenvalues, so that $B_{ij} = \lambda_i$ if $i = j$ and 0 otherwise. We remark that B^n is easily computed since it is diagonal in its own turn, with the n-th powers of the λ's on the main diagonal. Since $A^n = M^{-1}B^nM$, we have that the solution of the homogeneous equation is

$$\mathbf{x}_n = M^{-1}B^nM\mathbf{x}_0.$$

Actually, there is no need to find the matrix M explicitly: indeed, if A has k linearly independent eigenvectors, they necessarily form a *basis* of $\mathbb{R}^k$ (with some care if there are complex eigenvalues): since by definition the eigenvector $\mathbf{v}_j$ relative to the eigenvalue λ_j has the property $A\mathbf{v}_j = \lambda_j\mathbf{v}_j$, and every vector $\mathbf{x} \in \mathbb{R}^k$ is a linear combination of eigenvectors, we can write the solution of (4.5) in the form

$$\mathbf{x}_n = \mu_1\lambda_1^n\mathbf{v}_1 + \cdots + \mu_k\lambda_k^n\mathbf{v}_k,$$

where $\mathbf{x}_0 = \mu_1 \mathbf{v}_1 + \cdots + \mu_k \mathbf{v}_k$. Anyway, computing this decomposition is equivalent to the inversion of the matrix M above. We remark that the situation is quite similar to the case of a single recurrence of order k: indeed, it can be proved that they are essentially equivalent.

Now we turn our attention to the complete equation (4.3). Formally, the solution of (4.3) is given by the formula (3.7):

$$\mathbf{x}_n = A^n \mathbf{x}_0 + \sum_{k=1}^{n} A^{n-k} \mathbf{p}(k).$$

The point here is that, since A satisfies its own *characteristic equation* $\det(A - \lambda I_k) = 0$ (where I_k is the identity matrix of order k), A^n can be written as a linear combination of $A^0 = I_k$, A, $\ldots$, A^{k-1}. For instance, in the case of the system (4.2), we have

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 5 & 0 & 1 & 0 & 0 \\ -5 & 0 & 0 & 1 & 0 \\ -4 & 0 & 0 & 0 & 1 \\ 4 & 0 & 0 & 0 & 0 \end{bmatrix}$$

whose characteristic equation is the same as above, namely

$$\det(A - \lambda I_k) = -(\lambda - 1)^2(\lambda + 1)(\lambda - 2)(\lambda + 2) = 0.$$

This means that we can avoid the explicit substitution, as we did above.

4.2 Systems of Recurrences of Higher Order

In the previous section we only considered systems of equations of order 1, although the first method developed above entails the transformation of the equations into higher-order ones. From a purely theoretical point of view, it is quite interesting to know that it is possible to transform *any* system of linear recurrences with constant coefficients of any finite order into an equivalent system with a larger number of variables, with the property that every equation has order exactly 1. From a computational point of view, such a transformation is probably inefficient, so that the first method explained above should usually be the preferred one.

The idea is quite simple: it is possible to reduce by 1 a suitable measure of the complexity of an equation of order at least 2 (not necessarily the order; we give precise definitions below), at the cost of introducing exactly one more variable and one more equation of order 1: this is, essentially, the inverse process of the first method above where we reduced the number of variables by increasing the order of some equations. The total complexity of the system, which is the sum of the complexities of the equations, does not change, while the complexity of an equation of order at least 2 decreases. Iterating the process, it is possible to eliminate all equations of order 2 or more, until we are left with a (larger) system with only order 1 equations.

In order to avoid exceedingly cumbersome notations, we stick to the vector notation introduced in (4.4), with a slight generalization.

Notation 4.2 *Let k and K be fixed, positive integers. For $r = 1, \ldots, K$ let*

$$A^{(r)} \stackrel{\text{def}}{=} \begin{bmatrix} a_{11}^{(r)} & \ldots & a_{1k}^{(r)} \\ \multicolumn{3}{c}{\ldots\ldots\ldots\ldots\ldots} \\ a_{k1}^{(r)} & \ldots & a_{kk}^{(r)} \end{bmatrix}.$$

We are interested in the system

$$\mathbf{x}_n = A^{(1)}\mathbf{x}_{n-1} + \cdots + A^{(K)}\mathbf{x}_{n-K} + \mathbf{p}(n), \tag{4.6}$$

for fixed positive integers k (the number of variables) and K, where, in complete analogy with the discussion of linear recurrences of finite order with constant coefficients, we tacitly assume that $A^{(K)}$ is not the zero matrix. Here it does not quite make sense to say that the *order* of the system is K, and we give a more suitable definition.

Definition 4.3 (Weight.) *For each $s = 1, \ldots, K$ let the* weight w_s *of the s-th equation of the system* (4.6) *be defined by*

$$w_s \stackrel{\text{def}}{=} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ a_{sj}^{(i)} \neq 0}}^{k} i.$$

We let the weight *of the system* (4.6) *be $w_1 + \cdots + w_K$.*

We are weighing each equation not only with the number of variables that actually occur in it (so that $a_{sj}^{(i)} \neq 0$), but also taking into account the order of each occurring variable. The weight, as defined here, is not the same thing as the order, so the above is not necessarily coherent with the case $k = 1$ (a single equation) nor with the case of k equations of order 1. We only need the concept of weight to prove that the strategy outlined above actually works.

We now formally describe a strategy for transforming a system like (4.6) with at least an equation of order 2 or more (if all equations have order 1 there is nothing to do) into an equivalent one with the same weight, one more variable and one more equation, and such that the weight of one equation has decreased by 1. We may assume that the first equation, say, has order $q > 1$ and weight w_1. The strategy is as follows:

- let i be an index such that $a_{1i}^{(q)} \neq 0$;
- add the new equation $x_n^{(k+1)} = x_{n-1}^{(i)}$ to the system;
- replace the instance of $x_{n-q}^{(i)}$ in the first equation by $x_{n-q+1}^{(k+1)}$, which is equivalent by definition.

We have plainly added one equation of order 1 and one variable, and also changed the first equation of the system: we have to prove that

- the weight of the new system is the same as the weight of the old one;
- the weight of the first equation has decreased by 1.

The first thing can be checked easily: in fact, the new equation in the first row has weight $w_1 - 1$ (we left every term untouched except for the i-th in the q-th matrix, which has been replaced by one of order one unit less), and the equation we have added has weight 1. Therefore the new system has the same weight as the original one. The second item above has been verified incidentally.

Example 4.4 *Using the strategy above, rewrite the system*

$$\begin{cases} x_n = x_{n-2}, \\ y_n = x_{n-1} + y_{n-1}. \end{cases}$$

Both equations have weight 2, and the system has weight 4. We add a new variable z_n satisfying $z_n = x_{n-1}$. The system becomes

$$\begin{cases} x_n = z_{n-1}, \\ y_n = x_{n-1} + y_{n-1}, \\ z_n = x_{n-1}. \end{cases}$$

The equations have weight 1, 2, 1 respectively, and each equation now has order 1, while the system has still weight 4. In terms of matrices, for the initial system we have

$$A^{(1)} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \qquad A^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},$$

and for the transformed system we have

$$B^{(1)} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad \text{so that} \qquad \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = B^{(1)} \begin{bmatrix} x_{n-1} \\ y_{n-1} \\ z_{n-1} \end{bmatrix}.$$

Adding the new equation $z_n = x_{n-1}$ corresponds to adding a new row to the matrix $A^{(1)}$ (with a zero entry in all columns, except for the first), and a new column (again, with a zero entry in all rows, except for the first).

Example 4.5 *Using the strategy above, rewrite the system*

$$\begin{cases} x_n = x_{n-2} + y_{n-3}, \\ y_n = x_{n-1} + y_{n-2}, \\ z_n = y_{n-1}. \end{cases}$$

Here the weights of the single equations are 5, 3 and 1, for a total weight of 9. Setting $t_n = y_{n-1}$, we obtain the new system

$$\begin{cases} x_n = x_{n-2} + t_{n-2}, \\ y_n = x_{n-1} + y_{n-2}, \\ z_n = y_{n-1}, \\ t_n = y_{n-1}, \end{cases}$$

where the equations have weights 4, 3, 1, 1 respectively, for a total weight of 9 again. It is plain that we can repeat the same procedure on the rewritten system until all equations have order 1.

5 Related Work

In this section we briefly outline the differences and similarities between our approach (and implementation) and the work by Lueker [18] and Cohen & Katcoff [10]. We use the notation of (3.1), page 3: in particular, we always denote the non-homogeneous part of the recurrence by $p(n)$. Recall that we denote by $\mathbf{C}^{(k)}$ the k-th iterate of the operator $\mathbf{C}$.

5.1 Lueker, 1980

Lueker [18] describes some useful techniques for solving a wide range of recurrences, presumably intended for manual computation.

As far as linear, first-order recurrences (both with constant and variable coefficients) are concerned, our approach is essentially the one of Lueker. The main differences arise when we consider non-homogeneous recurrences: we exploit the closed formula (3.7), (3.12) or (3.16) thereby transforming the problem of "guessing" a particular solution of the non-homogeneous recurrence into the problem of computing a finite sum, by means of the tools explained in Appendix B. Lueker, instead, introduces the *shift operator* $\mathbf{E}$ as we did in Section 3.2 and also (without saying it explicitly) the *identity operator* $\mathbf{I}$.

The analysis is quite similar to our Section 3.2. Indeed, the point here is that for some forms of $p(n)$ it is possible to find a suitable combination $\mathbf{C}$ of the operators $\mathbf{E}$ and $\mathbf{I}$ that *annihilates* p, that is $\mathbf{C}(p) \equiv 0$, the constant sequence zero. In the customary language of linear algebra, $p \in \ker(\mathbf{C})$: see our Corollary 3.4. Then we apply the operator to *both* sides of the recurrence (3.1), and we are left with a new, *homogeneous* recurrence, which can be solved by means of the characteristic equation. We know that if $p \in \mathcal{A}$, then there is such an operator $\mathbf{C}$ by Lemma 3.2.

The main drawback, however, is that the *order* of the new recurrence is higher than that of the original relation, since, for instance, the annihilator of the function $p(n) = n \cdot 2^n + 1$ is the operator $(\mathbf{E} - 2\mathbf{I})^{(2)}(\mathbf{E} - \mathbf{I})$, so that the order of the recurrence with p as a non-homogeneous part would increase by 3. It is true that it is quite easy to detect the roots of the characteristic polynomial introduced in this way, but one is left in the end with the problem of determining the unknown coefficients in formula (3.4) or (3.5), that is, with a set of linear equations, whose number of variables is precisely the order of the new recurrence relation. Actually, the determination of the operator $\mathbf{C}$ that annihilates p is equivalent to the classification of the function p (polynomial, exponential, a product of a polynomial and an exponential, ...). Thus, there is no particular advantage in this method, since we show in Appendix B how to find explicitly the particular solution in all of these cases.

Even though no experimental comparison of our technique and that proposed by Lueker was conducted (we are not aware of any available implementation of the latter), we believe our approach is more efficient, especially for the case of recurrences of low order that most frequently arises in automatic complexity analysis.

We now work out an example to illustrate the method described by Lueker. The same recurrence is tackled in Example C.2 on page 42 using the techniques of Section 3.

Example 5.1 *Solve the recurrence*

$$x_n = 2x_{n-1} + 2^{n-1} + n + 1.$$

We start by rewriting it as

$$x_{n+1} - 2x_n = 2^n + n + 2,$$

and then in the equivalent form

$$(\mathbf{E} - 2\mathbf{I})(x_n) = 2^n + n + 2.$$

We already know that the operator $\mathbf{E} - 2\mathbf{I}$ annihilates the sequence 2^n, so that

$$\begin{aligned}(\mathbf{E} - 2\mathbf{I})(\mathbf{E} - 2\mathbf{I})(x_n) &= (\mathbf{E} - 2\mathbf{I})(2^n + n + 2) \\ &= (2^{n+1} + n + 3) - 2(2^n + n + 2) \\ &= -n - 1.\end{aligned}$$

We are left with a polynomial of degree 1: we know that the operator $\mathbf{E} - \mathbf{I}$ reduces the degree of any polynomial by 1, and we need to apply it *twice* to get

$$\begin{aligned}
(\mathbf{E} - \mathbf{I})^{(2)}(\mathbf{E} - 2\mathbf{I})^{(2)}(x_n) &= (\mathbf{E} - \mathbf{I})^{(2)}(-n - 1) \\
&= (\mathbf{E} - \mathbf{I})\big(-(n+1) - 1 - (-n - 1)\big) \\
&= (\mathbf{E} - \mathbf{I})(-1) \\
&= 0.
\end{aligned}$$

We have therefore proved that the operator $\mathbf{C} \stackrel{\text{def}}{=} (\mathbf{E} - \mathbf{I})^{(2)}(\mathbf{E} - 2\mathbf{I})^{(2)}$ annihilates the sequence (x_n). In other words, we just found an order 4 *homogeneous* recurrence satisfied by (x_n), with the degree 4 characteristic equation

$$(\lambda - 1)^2(\lambda - 2)^2 = 0.$$

Hence the general solution is

$$x_n = (\alpha n + \beta) \cdot 2^n + \gamma n + \delta,$$

where α, β, γ and δ are suitable real numbers that can be found by taking into account the fact that $x_1 = 2x_0 + 3$, $x_2 = 4x_0 + 11$, $x_3 = 8x_0 + 30$. These values can be computed directly from the recurrence we started with. Finally, we solve the corresponding system of equations (in the parameter x_0), that is, we solve

$$\begin{cases}
\phantom{24\alpha + {}} \beta \phantom{{} + 3\gamma} + \delta = x_0 \\
2\alpha + 2\beta + \gamma + \delta = 2x_0 + 3 \\
8\alpha + 4\beta + 2\gamma + \delta = 4x_0 + 11 \\
24\alpha + 8\beta + 3\gamma + \delta = 8x_0 + 30
\end{cases}$$

and find the coefficients

$$\alpha = \frac{1}{2}, \qquad \beta = x_0 + 3, \qquad \gamma = -1, \qquad \delta = -3.$$

5.2 Cohen & Katcoff, 1977

Cohen & Katcoff [10] describe an *interactive* computer program for solving difference equations; in this respect, our focus is different as we target completely algorithmic solution techniques. They present two methods for solving recurrences, namely, (1) "guessing" the solution of the non-homogeneous part,[3] and (2) using the generating function. Since they "feel that neither of the preceding methods is suitable for direct use in the automatic solution

[3]That is, using the result contained in our Corollary 3.4 that suggests the shape of the solution.

of difference equations", they use a modified form of the generating function method for solving (systems of) linear recurrences. In contrast, in this paper we describe several ways for "guessing" the particular solution of a non-homogeneous linear recurrence with constant coefficients by means of a simple formula, at least in the rather common cases where p belongs to the class $\mathcal{A}$ defined in Section 3.2: see Appendix B on page 36 for the details. In particular, there are at least 4 different ways to find the closed form for the sum of values of a polynomial, which are described in Section B.3, each probably with a different range of optimal performance. The main reason for our choice is that we want to solve or approximate a wider set of recurrences: when solving divide-et-impera recurrences [3], for example, we often need to compute symbolic sums, like the ones that appear in (3.16), that do not necessarily arise as solutions of linear recurrences. Therefore, we felt the need to tackle such problems directly.

The overall solution algorithm we use in the `PURRS` system is quite similar to the one by Cohen & Katcoff, in that we first classify the recurrence (order, variable or constant coefficients, system), but then we use rather more direct ways for solving the simpler recurrences (that, as already noted, occur much more frequently in automatic complexity analysis).

Cohen & Katcoff give several examples of the actual computations carried out by their program [10, Table II]. `PURRS` easily solves their examples 1–7 (note that the last two contain symbolic coefficients) and should be soon able to solve also the systems 11–13. The more difficult recurrences 8–10 have variable coefficients, and belong to the class considered in [5]. Notice that `PURRS` gives *exact* answers whenever possible (including exact representation of rational numbers), whereas Cohen & Katcoff only give approximated coefficients rounded to the third decimal digit. We also point out that the solution they give for their example 3 is incorrect.[4]

In their final remarks, Cohen & Katcoff discuss possible improvements of their work: apart from the problem of factoring polynomials of degree 5 or more which we discuss in Appendix A (item 1 in their list), we share their feelings about the simplification routines (item 4). We incorporated into `PURRS` many routines that simplify expressions, in particular those that contain irrational numbers that may arise when solving the characteristic equation associated to a linear recurrence. Finally, we devote a future paper of this series [4] to approximate solution, an interesting problem that they suggest in item 8.

[4]`PURRS` computes it correctly as $a_n = \frac{4}{9} \cdot 2^n + (-1)^n\bigl(\frac{4}{3}n - \frac{4}{9}\bigr) + (-1)^n(1-n)a_0 + (-1)^{n+1}na_1$. The correct solution with the given initial conditions $a_0 = 1$, $a_1 = 2$ is therefore $a_n = \frac{4}{9} \cdot 2^n + (-1)^n\bigl(\frac{5}{9} - \frac{5}{3}n\bigr)$.

6 Conclusion

We have described some algorithmic techniques for the efficient solution of a wide class of (systems of) linear recurrences of finite order with constant coefficients. The basic theory has been exposed both from an abstract and a more concrete point of view, so as to convey the general principles and to provide, at the same time, a rather precise specification for the implementation. In fact, variants of these techniques, along with the associated overall solution methodology and the tools illustrated in the appendix, have been implemented in `PURRS`, a system for the automatic solution and approximation of a much wider class of recurrences (see `http://www.cs.unipr.it/purrs/`).

In this paper, we only tackled the problem of "elementary" summation in closed form. In many interesting applications, though, the non-homogeneous part of the LRFOCC does not belong to the class $\mathcal{A}$, and therefore the results described here can not be used. There is a wide class of non-homogeneous parts (the simplest element of this class being the function $p(n) = n^{\alpha}$, where α is not necessarily integral) that can be summed approximately, with rather sharp upper and lower bounds for the solution, and we deal with this problem in general in [4]. Another class (that contains for instance some, but not all, rational functions) that admits closed-form solution will be introduced in the paper on "transcendental" summation methods [5] that deals with the recent holonomy theory of hypergeometric summation [20], and contains results on recurrences with variable coefficients.

References

[1] L. M. Adelman and A. M. Odlyzko. Irreducibility testing and factorization of polynomials. *Mathematics of Computation*, 41:699–709, 1983.

[2] T. M. Apostol. *Introduction to Analytic Number Theory.* Springer-Verlag, Berlin, 1975.

[3] R. Bagnara, A. Zaccagnini, and T. Zolo. The automatic solution of recurrence relations. II. "Divide et Impera" recurrences. Quaderno, Dipartimento di Matematica, Università di Parma, Italy. To appear.

[4] R. Bagnara, A. Zaccagnini, and T. Zolo. The automatic solution of recurrence relations. III. Approximate solutions. Quaderno, Dipartimento di Matematica, Università di Parma, Italy. To appear.

[5] R. Bagnara, A. Zaccagnini, and T. Zolo. The automatic solution of recurrence relations. IV. Linear recurrences of finite order with variable coefficients. Quaderno, Dipartimento di Matematica, Università di Parma, Italy. To appear.

[6] B. L. Burrows and R. F. Talbot. Sums of powers of integers. *American Mathematical Monthly*, 91:394–403, 1984.

[7] B. F. Caviness. On canonical forms and simplification. *Journal of the ACM*, 17(2):385–396, 1970.

[8] P. Celis. Remark: Corrections and errors in John Ivie's some MACSYMA programs for solving recurrence relations. *ACM Transactions on Mathematical Software*, 10(4):477–478, 1984. See [13].

[9] L. Childs. *A Concrete Introduction to Higher Algebra*. Springer-Verlag, Berlin, 1979.

[10] J. Cohen and J. Katcoff. Symbolic solution of finite-difference equations. *ACM Transactions on Mathematical Software*, 3(3):261–271, 1977.

[11] A. W. F. Edwards. A quick route to sums of powers. *American Mathematical Monthly*, 93:451–455, 1986.

[12] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. Oxford Science Publications, 1979. Fifth Edition.

[13] J. Ivie. Some MACSYMA programs for solving recurrence relations. *ACM Transactions on Mathematical Software*, 4(1):24–33, 1978. See also [8].

[14] W. G. Kelley and A. C. Peterson. *Difference Equations. An Introduction with Applications*. Academic Press, Inc. Harcourt Brace Jovanovich, 1991.

[15] A. Lenstra, H. Lenstra Jr., and L. Lovasz. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[16] L. S. Levy. Summation of the series $1^n + 2^n + \cdots + x^n$ using elementary calculus. *American Mathematical Monthly*, 77:840–847, 1970. See also [17].

[17] L. S. Levy. Corrigendum. *American Mathematical Monthly*, 78:987, 1971. See also [16].

[18] G. S. Lueker. Some techniques for solving recurrences. *ACM Computing Surveys*, 12(4):419–436, 1980.

[19] M. Mignotte and D. Ştefănescu. *Polynomials. An Algorithmic Approach*. Springer-Verlag, Berlin, 1999.

[20] M. Petkovšek, H. S. Wilf, and D. Zeilberger. $A = B$. A. K. Peters, 1996.

[21] D. Richardson. Some undecidable problems involving elementary functions of a real variable. *Journal of Symbolic Logic*, 33(4):514–520, 1968.

[22] P. S. Wang. The undecidability of the existence of zeros of real elementary functions. *Journal of the ACM*, 21(4):586–589, 1974.

[23] E. Weisstein. The world of mathematics. A web site full of informations on Mathematics. See `http://mathworld.wolfram.com/`.

A Solving Polynomial Equations

We recall that the general formula for the polynomial equation of degree 5 or more does not exist. A possibility is the following: first look for repeated (multiple) roots (see Appendix A.1), for the simple root $x = 0$ and for rational roots with "small" numerator and denominator (see Appendix A.2). Failing this, one can give control to an equation solver that detects polynomials of a special shape (see Appendix A.3). As a last resort, one can try to solve the general polynomial equation of degree up to 4. Polynomials of degree 1 or 2 are easy to treat, whereas polynomials of degree 3 (see Appendix A.4) and 4 (see Appendix A.5) need special care.

A.1 Detecting Multiple Roots

In order to detect multiple roots of a polynomial equation $p(x) = 0$ it is possible to use the *square-free decomposition*, which yields a partially factored form of p, in the case that p has repeated factors. This can be accomplished by computing $\gcd(p, p')$, the *greatest common divisor* of the polynomial p and its derivative, since the rule for the derivative of a product immediately yields that if $p(x) = g(x)^k h(x)$, say, where g and h are polynomials and $k \geq 2$ is an integer, then $p'(x) = kg'(x)g(x)^{k-1}h(x) + g(x)^k h'(x)$, so that $g(x)^{k-1}$ is a non-trivial factor of $\gcd(p, p')$. We recall that it is extremely important to detect repeated roots (if any) of the characteristic equation (see Section 3) and their multiplicity, and we do so at a very early stage of the computation. We now give precise definitions.

Definition A.1 (Square-free polynomial.) *A polynomial $p \in \mathbb{Q}[x]$ is* square-free *if it is not divisible by the square of any polynomial $q \in \mathbb{Q}[x]$ of degree larger or equal to* 1*.*

Definition A.2 (Square-free decomposition.) *Let $p \in \mathbb{Z}[x]$ be a polynomial. We say that the decomposition*

$$p(x) = a \prod_{i=1}^{r} p_i(x)^{\alpha_i}$$

is a square-free decomposition *of p if the following conditions hold:*

- *$a \in \mathbb{Z} \setminus \{0\}$;*
- *$\alpha_i \in \mathbb{N} \setminus \{0\}$ for $i = 1, \ldots, r$ and $1 \le \alpha_1 < \alpha_2 < \cdots < \alpha_r$;*
- *$p_i \in \mathbb{Z}[x]$ and $\deg(p_i) \ge 1$ for $i = 1, \ldots, r$;*
- *$\gcd(p_i, p_j) = 1$ if $i \ne j$;*
- *p_i is square-free for $i = 1, \ldots, r$.*

We might express the last two conditions more concisely by saying that the polynomial

$$\prod_{i=1}^{r} p_i(x)$$

is square-free.

A.2 Detecting Rational Roots

It is also possible to detect rational roots (if any) of any polynomial with integer coefficients. This depends on the following Theorem of Gauss (see Hardy & Wright [12, Theorem 45] for a variant).

Theorem A.3 *If $a_0, a_1, \ldots, a_n$ are integers without common factors larger than 1, where a_0 and a_n do not vanish, and $x = p/q$ is a rational root of the polynomial equation*

$$a_n x^n + \cdots + a_1 x + a_0 = 0$$

(where p and $q \ne 0$ are mutually prime integers) then p divides a_0 and q divides a_n.

Looking for *all* rational roots would involve factoring the integers a_0 and a_n, a notoriously difficult problem: in fact, at the moment, the best factoring algorithms are non-polynomial. So one can be content with detecting rational roots p/q, where $|p|$ and $|q|$ do not exceed some threshold $M = m^2$, with $m \in \mathbb{N}$ small. The first step is to consider the divisors of a_0 and a_n. Let $x = a_0$ or $x = a_n$, $|x| < M$, be the integer whose divisors we are looking for. With a loop we check if the integer $i \in \{1, \ldots, m-1\}$ divides x, and in this case we push onto a suitable structure both i and its "conjugate" divisor $|x|/i$, if they are different, of course. The process stops when i exceeds the square root of $|x|$. We remark that although most recurrence relations have a characteristic equation whose leading coefficient is 1, we have to take into account the fact that the coefficients may be rational numbers.

Much more ingenuity would be needed if it were desired to have a general routine for finding *all* rational roots. In this case, it would probably be better to use a general algorithm for splitting polynomials with integer coefficients such as the one described in Lenstra, Lenstra & Lovasz [15], rather than trying to factor completely the integers a_0 and a_n. A simpler method is described in Childs [9, Part II, Chapters 11 & 13] and another in Adelman & Odlyzko [1]. Another feature we plan to add is an irreducibility test for polynomials with integral coefficients over $\mathbb{Q}$: see Childs [9, Part II, Chapter 8] for some simple such tests.

A.3 Equations of Special Form

There is one last important remark: the previous computations might destroy any "structure" of the equation: for example, if $p(x) = x^7 - 1$, we detect the rational root $x = 1$, but after this we are left with an equation of degree 6 which we cannot solve (actually p is a cyclotomic polynomial and it is known that it has exactly two irreducible factors over $\mathbb{Q}$, one being $x-1$). In this case, and more generally when $q(x) = r(x^n)$ for some $r \in \mathbb{Q}[x]$ and integral $n \geq 2$, probably an alternative approach would work better: before giving up, go back to the original equation, and check whether it has any special shape. It is quite easy to recognize polynomials of this shape. In particular, in our example above it is possible to transform the original equation into another (actually $y - 1 = 0$) in the variable $y = x^7$. After solving this equation, we compute the complex 7-th roots of all the values y. We perform this task by taking just one 7-th root, and then multiplying it by the 7-th roots of unity, which are given by the Euler-de Moivre formula $\cos\bigl(\frac{2k\pi}{7}\bigr) + i\sin\bigl(\frac{2k\pi}{7}\bigr)$ for $k = 0,\ldots, 6$. It might even be more efficient to check this at once, before embarking on any computations at all[5].

A.4 Solving the General Polynomial Equation of Degree 3

Suppose that the polynomial $p(x) = ax^3 + bx^2 + cx + d$ has no repeated roots, and that it has neither the root 0 (in other words, $d \neq 0$) nor "small" rational roots. The formula for solving the equation of degree 3 needs the computation of the *discriminant* Δ of the polynomial, and we take different routes according to the sign of Δ. Notice that $\Delta = 0$ if and only if p has repeated roots, which is not our case. It is well known that the roots are all real if and only if $\Delta \leq 0$, and that there is a pair of complex-conjugate roots otherwise (see Appendix A.6 for the general definition of the discriminant).

In order to solve

$$x^3 + a_1x^2 + a_2x + a_3 = 0, \tag{A.1}$$

we set

$$R \stackrel{\text{def}}{=} \frac{1}{54}\bigl(9a_1a_2 - 27a_3 - 2a_1^3\bigr), \qquad Q \stackrel{\text{def}}{=} \frac{1}{9}\bigl(3a_2 - a_1^2\bigr).$$

[5]The observation in this paragraph is in general valid, but we observe that, in the context of the computation of roots of characteristic equations in order to solve recurrences, this could become useless when the order reduction method is applied.

The discriminant is $\Delta \stackrel{\text{def}}{=} Q^3 + R^2$. The solutions of (A.1) are of the form

$$-\frac{1}{3}a_1 + \left(R + \Delta^{1/2}\right)^{1/3} + \left(R - \Delta^{1/2}\right)^{1/3}.$$

This yields 9 values, 3 for each possible choice of the *complex* cube root, but only three actually give roots of (A.1). Here and below we denote by $x^{1/3}$ the set of all possible solutions of the equation $z^3 = x$, that is all complex cube roots of the complex number x, and the same applies to $x^{1/2}$. On the other hand, we denote by $\sqrt{x}$, $\sqrt[3]{x}$ the "arithmetical" square and cube roots, that is roots of positive quantity that do not give rise to any ambiguities.

If $\Delta < 0$, then $Q < 0$ and a direct computation aided by some trigonometrical identities yields

$$\begin{aligned}
x_1 &\stackrel{\text{def}}{=} -\frac{1}{3}a_1 + 2\sqrt{-Q}\cos\Big(\frac{1}{3}\theta\Big),\\
x_2 &\stackrel{\text{def}}{=} -\frac{1}{3}a_1 + 2\sqrt{-Q}\cos\Big(\frac{1}{3}(\theta + 2\pi)\Big),\\
x_3 &\stackrel{\text{def}}{=} -\frac{1}{3}a_1 + 2\sqrt{-Q}\cos\Big(\frac{1}{3}(\theta + 4\pi)\Big),
\end{aligned}$$

where $\theta \stackrel{\text{def}}{=} \arccos\big(-R/(Q\sqrt{-Q})\big)$, so that all roots are real. If $\Delta > 0$ we set

$$\begin{cases} S \stackrel{\text{def}}{=} \sqrt[3]{R + \sqrt{\Delta}} \\ T \stackrel{\text{def}}{=} \sqrt[3]{R - \sqrt{\Delta}} \end{cases} \qquad \text{and} \qquad \begin{cases} t_1 \stackrel{\text{def}}{=} -\frac{1}{3}a_1 - \frac{1}{2}(S + T) \\ t_2 \stackrel{\text{def}}{=} \frac{1}{2}(S - T)i\sqrt{3} \end{cases}$$

(here we take the *arithmetical* cube root so that S and T are real numbers), where $i^2 = -1$. Finally, we compute the roots of equation (A.1) by means of

$$\begin{aligned}
x_1 &\stackrel{\text{def}}{=} -\frac{1}{3}a_1 + \frac{1}{2}(S + T),\\
x_2 &\stackrel{\text{def}}{=} t_1 + t_2,\\
x_3 &\stackrel{\text{def}}{=} t_1 - t_2.
\end{aligned}$$

Notice that x_1 is real. Notice also that, in the procedure outlined above, we only actually take arithmetical square and cube roots. For the details, we refer to Weisstein [23], or Childs [9, Part II, Chapter 3].

A.5 Solving the General Polynomial Equation of Degree 4

We briefly sketch an algorithm in order to solve the equation $x^4 + a_1x^3 + a_2x^2 + a_3x + a_4 = 0$, exposed in `http://www.1728.com/quartic2.htm`. We set

$$\begin{aligned}
f &\stackrel{\text{def}}{=} a_2 - (3a_1^2/8),\\
g &\stackrel{\text{def}}{=} a_3 + (a_1^3/8) - (a_1a_2/2),\\
h &\stackrel{\text{def}}{=} a_4 - (3a_1^4/256) + (a_1^2a_2/16) - (a_1a_3/4).
\end{aligned}$$

We first find the solutions y_1, y_2, y_3 of the third-degree polynomial equation $y^3 + (f/2)y^2 + \big((f^2 - 4h)/16\big)\, y - g^2/64 = 0$ by means of the algorithm described in the previous section. We observe that if $g = 0$ then 0 is a root of the cubic equation and in this case we can obtain the solution more efficiently calling the formula for equation of the second degree on $y^2 + (f/2)y + \big((f^2 - 4h)/16\big) = 0$.

Let p and q be two non-zero roots of the cubic equation; the solutions of the original equation are

$$
\begin{aligned}
x_1 &\stackrel{\text{def}}{=} p + q + r - s, \\
x_2 &\stackrel{\text{def}}{=} p - q - r - s, \\
x_3 &\stackrel{\text{def}}{=} -p + q - r - s, \\
x_4 &\stackrel{\text{def}}{=} -p - q + r - s.
\end{aligned}
$$

A.6 The Discriminant and the Resultant

We adapt some definitions from Childs [9, Part III, Chapter 15]. See also Mignotte and Ştefănescu [19, §1.5.2].

Definition A.4 (Resultant.) *Let f, $g \in \mathbb{C}[x]$ be polynomials of degrees m and n respectively, and roots α_1, …, α_m, and β_1, …, β_n respectively (with the convention that multiple roots are repeated according to multiplicities). Let a and b be their leading coefficients. The* resultant *of f and g is defined by*

$$R(f, g) \stackrel{\text{def}}{=} a^n b^m \prod_{i=1}^{m} \prod_{j=1}^{n} (\alpha_i - \beta_j).$$

A few remarks: it can be proved that if f and g have integer coefficients, then $R(f, g)$ is an integer. It is also clear that $R(f, g) = 0$ if, and only if, f and g have a common root. There are several ways to compute $R(f, g)$ without knowing the roots of the polynomials f and g. One of the possibilities is to consider an algorithm based on the following properties:

- $R(g, f) = (-1)^{\deg(f)\deg(g)} R(f, g)$;
- $R(f, g) = a^{\deg(g)-\deg(r)} R(f, r)$ if $g = fq + r$ for polynomials q and r. Here a is the leading coefficient of the polynomial f;
- $R(f, b) = b^{\deg(f)}$ if b is a scalar.

The second property shows that we can use Euclid's algorithm to compute the resultant. Another possibility is to compute the resultant as the determinant of the Sylvester matrix (see Weisstein [23]): this is useful when we deal with parametric polynomials, since the method based on Euclid's algorithm needs the computation of division with quotient and remainder, and therefore the partial results may be *rational functions* of the parameter. We need parametric polynomials in the actual practice when using Gosper's algorithm (see [5] and Petkovšek, Wilf and Zeilberger [20, Chapter 5]).

Definition A.5 (Discriminant.) *Let $f \in \mathbb{C}[x]$ be a polynomial of degree n. We define the* discriminant *of f by means of*

$$D(f) \stackrel{\text{def}}{=} (-1)^{n(n-1)/2} R(f, f').$$

We remark that $D(f) = 0$ if, and only if, f and f' have a common factor, and this happens if, and only if, f has a repeated root.

A.7 Parametric Equations

In the discussion in the last few sections we assumed that the coefficients of the polynomial equations under consideration are integers or rational numbers, but it would be very interesting to allow these coefficients to contain one or more unspecified parameters. Solving parametric polynomial equations of degree 2 or more, though, can be very difficult since we may have to take into account many possible cases, as we now explain.

We saw in Section A.6 that we can associate to every polynomial equation the *discriminant* d, which is an integer (resp. real or complex number) if the coefficients are integers (resp. real or complex). It can be proved that a polynomial equation has a multiple root if and only if $d = 0$: the discussion in Section 3 shows that we have to treat differently the case when the characteristic equation has only simple roots from the case of multiple roots. Not only this, we also want to handle possible complex roots with some care. When the degree of the characteristic equation (that is, the order of the recurrence) is 1, there can be no multiple roots; when it is 2, the sign of the discriminant can be used to distinguish among the various cases, and when the degree is 3 we can still meet both requirements (see Appendix A.4).

But when the degree exceeds 3, things become rapidly more difficult: if we were to allow parametric coefficients, we should be able to identify all possible situations for the multiplicities of the roots. As an example, if the degree is 5 and we know that $d = 0$, the possible multiplicities for the roots are:

$$5; \qquad 4,1; \qquad 3,2; \qquad 3,1,1; \qquad 2,2,1; \qquad 2,1,1,1.$$

If the degree is k, the number is exactly $p(k) - 1$, where $p(k)$ denotes the number of the *unrestricted partitions* of the integer k. There is a clash of notation with the remainder of this document, but the notation p for the number of partitions is well-established, and will not occur elsewhere. It is known that p grows very rapidly: indeed

$$p(n) \sim \frac{e^{K\sqrt{n}}}{4n\sqrt{3}} \qquad \text{as } n \to \infty,$$

where $K = \pi\bigl(2/3\bigr)^{1/2}$ (see Apostol [2, §14.6]). This means that we should give conditions on the parameters for each one of these possible occurrences, and that would be too difficult.

A.8 Solving a Set of Linear Equations

In order to compute the coefficients α in the general solution (3.5) is necessary to solve systems with n linear equations and n unknowns. For this is possible to use the well-known method of inverse matrix, which computes the inverse of a square matrix, where the elements of the matrix are the coefficients of the recurrence relation.

B Exact Summation of Some Special Functions

The aim of this section is the solution of this problem: given a polynomial p with complex coefficients, and a complex number z, compute an elementary closed formula for the expression

$$\sum_{k=0}^{N} p(k) z^k. \tag{B.1}$$

In the last subsections, we also deal with a more general problem with the elementary trigonometric functions, so that we essentially solve the problem above for any element of the set $\mathcal{A}$ defined in Section 3.

B.1 Exponentials

This is by far the easiest case: it all depends on the well-known formula

$$\sum_{k=0}^{N} z^k = \frac{1 - z^{N+1}}{1 - z} \tag{B.2}$$

which is valid for any complex $z \neq 1$ and any integer $N \geq 0$.

B.2 Products of a Polynomial and an Exponential

Here we show how to give a closed formula for (B.1) where p is a polynomial and $z \neq 1$ is a complex number. It all hinges on the following important result.

Lemma B.1 *Given a polynomial $p \in \mathbb{C}[n]$ of degree d, there exist complex numbers $b_0, \ldots, b_d$ such that*

$$p(n) = \sum_{k=0}^{d} b_k n_{(k)} \tag{B.3}$$

where $n_{(k)}$ is the falling factorial function *defined by*

$$n_{(k)} \stackrel{\text{def}}{=} k!\binom{n}{k} = n(n-1)\cdots(n-k+1).$$

In the language of linear algebra, the set $\{ n_{(k)} \mid k \geq 0 \}$ is a *basis* of the ring $\mathbb{C}[n]$. If $p \in \mathbb{R}[n]$ (resp. $\mathbb{Z}[n]$), then the coefficients b_j belong to $\mathbb{R}$ (resp. $\mathbb{Z}$).

Proof. We give the proof in the form of an algorithm for computing the coefficients b_k, and we show that it needs at most d iterations: in fact, assume that

$$p(n) = a_d n^d + \cdots + a_0,$$

where $a_d \neq 0$. Now set $p_d(n) \stackrel{\text{def}}{=} p(n)$, $b_d \stackrel{\text{def}}{=} a_d$ and define inductively the polynomial p_j and the complex number b_j for $j = d-1, \ldots, 0$ by means of

$$\begin{aligned} p_j(n) &\stackrel{\text{def}}{=} p_{j+1}(n) - b_{j+1} n_{(j+1)}, \\ b_j &\stackrel{\text{def}}{=} \operatorname{coeff}_j(p_j), \end{aligned}$$

where $\operatorname{coeff}_j(q)$ denotes the coefficient of degree j of the polynomial q. This gives the desired proof because by construction the polynomial p_j has degree at most j. □

Actually, we use a quite different method, which appears to be faster: we note that the polynomial $P(z) = z_{(k)}$ vanishes for $z = 0, 1, \ldots, k-1$. This implies that $b_0 = p(0)$, and that $p(z) - b_0$ has the factor z. Iterating, we find that b_1 is $\bigl(p(z) - b_0\bigr)/z$ evaluated at $z = 1$, and so on. In practice, we construct a sequence of polynomials $q_k(z)$ as follows: we set

$$\begin{cases} q_0(z) = p(z), \\ b_0 = p_0(0), \end{cases} \qquad \text{and} \qquad \begin{cases} q_{k+1}(z) = \dfrac{q_k(z) - b_k}{z-k}, \\ b_{k+1} = q_{k+1}(k+1). \end{cases}$$

Once p has been written in the form (B.3), we see that (B.1) becomes

$$\sum_{k=0}^{N} p(k) z^k = \sum_{k=0}^{N} \sum_{j=0}^{d} b_j k_{(j)} z^k = \sum_{j=0}^{d} b_j \sum_{k=0}^{N} k_{(j)} z^k. \tag{B.4}$$

The only remaining problem is the summation in closed form of single summands in (B.4), far right. The following Lemma B.2, whose proof is a simple verification and is therefore omitted, is the tool that can be used.

Lemma B.2 *For $z \in \mathbb{C}$ with $z \neq 1$ we have the identity*

$$\sum_{k=0}^{N} k_{(j)} z^k = \sum_{k=0}^{N} k(k-1)\cdots(k-j+1) z^k = z^j \frac{\mathrm{d}^j}{\mathrm{d}z^j} \sum_{k=0}^{N} z^k = z^j \frac{\mathrm{d}^j}{\mathrm{d}z^j} \frac{1-z^{N+1}}{1-z}. \tag{B.5}$$

B.3 Polynomials

If $z = 1$ the above idea fails: we suggest several different solutions. The first, more straightforward one, depends upon the following result.

Theorem B.3 *For any $k \in \mathbb{N}$ there exists a polynomial $p_k \in \mathbb{Q}[n]$ of degree $k+1$ such that for any $N \in \mathbb{N}$*

$$\sum_{n=1}^{N} n^k = p_k(N).$$

This is Theorem 1 of Levy [16], and we give a sketch of the proof below. See also the papers by Burrows & Talbot [6], and Edwards [11], for related results, that also bear on the algorithms. The most familiar instances of this fact are the formulæ

$$\begin{aligned} \sum_{n=1}^{N} n &= \frac{1}{2} N(N+1), \\ \sum_{n=1}^{N} n^2 &= \frac{1}{6} N(N+1)(2N+1), \\ \sum_{n=1}^{N} n^3 &= \left(\frac{1}{2} N(N+1)\right)^2. \end{aligned} \tag{B.6}$$

Once we know that it exists, the polynomial p_k can be determined by imposing $k+2$ conditions on its $k+2$ unknown coefficients (actually, one can prove that the leading coefficient is $(k+1)^{-1}$, the next one is $\frac{1}{2}$, and all other coefficients of the monomials x^j where j has the same parity as k, as well as the constant term vanish), and solving the arising system of linear equations. For more details see Levy [16]. A rather different method follows from Appendix B.2, and it also yields a proof of Theorem B.3.

Proof of Theorem B.3. We first prove by induction on N that for any $k \in \mathbb{N}$ we have

$$\sum_{n=1}^{N} n_{(k)} = \frac{(N+1) \cdot N \cdot (N-1) \cdots (N-k+1)}{k+1}. \tag{B.7}$$

This is basically a simple property of binomial coefficients: indeed, it is equivalent to

$$\sum_{n=1}^{N} \binom{n}{k} = \binom{N+1}{k+1}.$$

Hence the decomposition of the polynomial p provided by (B.3) yields the desired result, since we can write the polynomial $p(n) = n^k$ as a linear combination with integral coefficients of the polynomials $n_{(d)}$ for $d = 0, \ldots,$ k, and compute $\sum_{n=1}^{N} n_{(d)}$ by means of (B.7), which gives a polynomial in N of degree $d+1$. □

We now describe two more approaches to this problem: the first one depends on the computation of the Bernoulli numbers (see for example Apostol [2, §12.12] or Hardy & Wright [12, §7.9]), and the second is an iterative method described in Levy [16].

Definition B.4 (Bernoulli numbers.) *The* Bernoulli numbers*, denoted by B_n for $n \in \mathbb{N}$, are the coefficients in the Taylor series development*

$$\frac{z}{e^z - 1} = 1 - \frac{1}{2}z + \frac{B_1}{2!}z^2 - \frac{B_2}{4!}z^4 + \frac{B_3}{6!}z^6 - \cdots + (-1)^{k-1}\frac{B_k}{(2k)!}z^{2k} + \cdots$$

which is valid in the circle $\{ z \in \mathbb{C} \mid |z| < 2\pi \}$. In particular, $B_1 = \frac{1}{6}$, $B_2 = \frac{1}{30}$, $B_3 = \frac{1}{42}$.

Theorem B.5 *Setting $\beta_0 \stackrel{\text{def}}{=} 1$, $\beta_1 \stackrel{\text{def}}{=} -\frac{1}{2}$, $\beta_{2k} \stackrel{\text{def}}{=} (-1)^{k-1}B_k$, $\beta_{2k+1} \stackrel{\text{def}}{=} 0$ for $k \in \mathbb{N} \setminus \{0\}$, where the B_k are the Bernoulli numbers, we have*

$$\sum_{n=1}^{N-1} n^k = \sum_{r=0}^{k} \frac{1}{k+1-r}\binom{k}{r} N^{k+1-r}\beta_r. \tag{B.8}$$

This is formula (7.9.1) of Hardy & Wright [12]. The proof is given comparing coefficients of suitable Taylor series expansions. We remark that the left hand side of (B.8) contains $N-1$ summands, while the right hand side is a polynomial in N of degree $k+1$ and therefore contains at most $k+2$ summands. The number β_n can be computed by means of the following iterative formula: for $n \geq 2$

$$\beta_n = \sum_{k=0}^{n} \binom{n}{k} \beta_k.$$

This is Apostol [2, Theorem 12.15]. We remark that there is no established notation for the Bernoulli numbers: in particular, Apostol calls B_n the quantities that we call β_n. We also remark that it is quite important, for our purposes, that the β_n are rational numbers.

The iterative method referred to above depends on another result in Levy [16, Theorem 2].

Theorem B.6 *For each $k \in \mathbb{N} \setminus \{0\}$ there is a rational number C_k such that, in the notation of* Theorem B.3*, we have*

$$p_k(N) = \sum_{n=1}^{N} n^k = k \int_0^N p_{k-1}(t)\, \mathrm{d}t + C_k N.$$

The value of C_k can be computed putting $N = 1$, and it is known that $C_k = 0$ whenever $k > 1$ is odd (see Levy [16, Corollary 3]). We remark that the integral can be easily computed using a Computer Algebra System, since we are only dealing with polynomials.

B.4 Some More Exact Formulæ

For completeness's sake, we insert some more closed formulæ, which can be useful to a general recurrence relation solver. Their theoretical interest is increased by the fact that they can be used to avoid dealing with complex numbers when studying recurrences whose characteristic equation has a pair of complex conjugate roots. The first couple of formulæ is relevant in the theory of Fourier series. For $\theta \notin 2\pi\mathbb{Z}$ we have

$$\sum_{k=0}^{n} \sin(k\theta) = \frac{\cos((n+1/2)\theta) - \cos(\theta/2)}{2\sin(\theta/2)},$$
$$\sum_{k=0}^{n} \cos(k\theta) = \frac{1}{2} + \frac{\sin((n+1/2)\theta)}{2\sin(\theta/2)}.$$

The proof by induction is straightforward. Another (trickier) proof can be given by separating the real and imaginary parts of

$$\begin{aligned}\sum_{k=0}^{n} e^{ik\theta} &= \frac{1 - e^{i(n+1)\theta}}{1 - e^{i\theta}} = \frac{e^{i(n+1)\theta/2}}{e^{i\theta/2}} \cdot \frac{e^{i(n+1)\theta/2} - e^{-i(n+1)\theta/2}}{e^{i\theta/2} - e^{-i\theta/2}} \\ &= e^{in\theta/2} \cdot \frac{\sin\bigl((n+1)\theta/2\bigr)}{\sin\bigl(\theta/2\bigr)},\end{aligned}$$

where the first equality is just a special case of (B.2). These formulæ imply that both sums are, in absolute value,

$$\le \min\bigl(n+1, |\sin(\theta/2)|^{-1}\bigr) = \mathcal{O}\bigl(\min(n+1, \|\theta/\pi\|^{-1})\bigr)$$

where $\|x\|$ denotes the distance of the real number x from the nearest integer, so that $\|x\| \stackrel{\text{def}}{=} \min(\{x\}, 1 - \{x\})$. We remark that a similar trick enables one to find the closed formula for sums of the type $\sum_{k=0}^{n} \lambda^k \cos(k\theta)$, where λ is any real number. For brevity, we simply state the results: For $\theta \notin 2\pi\mathbb{Z}$

and real $\lambda \neq 0$ we have

$$\sum_{k=0}^{n} \lambda^k \sin(k\theta) = \frac{\lambda^{n+2}\sin(n\theta) - \lambda^{n+1}\sin((n+1)\theta) - \lambda\sin\theta}{\lambda^2 - 2\lambda\cos\theta + 1},$$
$$\sum_{k=0}^{n} \lambda^k \cos(k\theta) = \frac{\lambda^{n+2}\cos(n\theta) - \lambda^{n+1}\cos((n+1)\theta) + 1 - \lambda\cos\theta}{\lambda^2 - 2\lambda\cos\theta + 1}.$$

Combining these identities with the technique developed in Appendix B.2, one can obtain exact formulæ for sums of the type $\sum p(k)\lambda^k \cos(k\theta)$, where p is the product of a polynomial and an exponential. We remark that, because of the decomposition (B.3), we need only consider polynomials of the form $p(k) = k_{(j)}$ for some fixed non negative integer j. Next, we note that an identity similar to (B.5) with x replaced by $\lambda e^{i\theta}$, implies that

$$\sum_{k=0}^{n} k_{(j)}\lambda^k e^{ik\theta} = \lambda^j \frac{\partial^j}{\partial\lambda^j} \sum_{k=0}^{n} \lambda^k e^{ik\theta}.$$

Separating the real and imaginary parts by means of the above formulæ, we obtain

$$\sum_{k=0}^{n} k_{(j)}\lambda^k \sin(k\theta) = \lambda^j \frac{\partial^j}{\partial\lambda^j} \frac{\lambda^{n+2}\sin(n\theta) - \lambda^{n+1}\sin((n+1)\theta) - \lambda\sin\theta}{\lambda^2 - 2\lambda\cos\theta + 1}, \tag{B.9}$$

$$\sum_{k=0}^{n} k_{(j)}\lambda^k \cos(k\theta) = \lambda^j \frac{\partial^j}{\partial\lambda^j} \frac{\lambda^{n+2}\cos(n\theta) - \lambda^{n+1}\cos((n+1)\theta) + 1 - \lambda\cos\theta}{\lambda^2 - 2\lambda\cos\theta + 1}. \tag{B.10}$$

B.5 How To Avoid the Use of Complex Numbers

Here we describe how to use the results of the previous section to avoid the use of complex numbers altogether in some interesting cases. In particular, we refer to (3.13) when λ_1 and λ_2 are a pair of complex conjugate numbers, with $\Im(\lambda_1) > 0$. In this case we write $\rho \stackrel{\text{def}}{=} |\lambda_1| = |\lambda_2|$ and choose $\theta \in (0, \pi)$ such that $\lambda_1 = \rho e^{i\theta}$; since $\lambda_2 = \overline{\lambda_1}$, we also have $\lambda_2 = \rho e^{-i\theta}$. We remark that if $\lambda_1 = \alpha + i\beta$, say, then $\rho = \sqrt{\alpha^2 + \beta^2}$ and $\cos\theta = \alpha\rho^{-1}$, so that both ρ and θ can be computed staying safely within the set of real numbers. Our definitions imply that

$$\begin{aligned}\sum_{k=0}^{n} \frac{\lambda_1^{n+1-k} - \lambda_2^{n+1-k}}{\lambda_1 - \lambda_2}\, p(k) &= \sum_{k=0}^{n} \rho^{n+1-k}\, \frac{e^{i(n+1-k)\theta} - e^{-i(n+1-k)\theta}}{\rho e^{i\theta} - \rho e^{-i\theta}}\, p(k) \\ &= \sum_{k=0}^{n} \rho^{n-k}\, \frac{2i \sin((n+1-k)\theta)}{2i \sin\theta}\, p(k) \\ &= \frac{\rho^n}{\sin\theta} \sum_{k=0}^{n} \rho^{-k} \sin((n+1-k)\theta)\, p(k).\end{aligned}$$

If p is the product of a polynomial and an exponential, the trigonometrical identity for $\sin(\alpha-\beta)$ transforms the above formula into a linear combination of (B.9) and (B.10), for a suitable value of λ.

C Examples

Here we discuss some concrete examples: we assume familiarity with the topics in Appendix B.2. In particular, here we omit details pertaining to the computation of a closed formula for expressions of the form (B.1). We omit the initial conditions, unless we need them explicitly.

C.1 Recurrences of Order 1

Example C.1 *Solve*

$$x_n = x_{n-1} + p(n). \tag{C.1}$$

The procedure outlined in Section 3 yields $g_n = \alpha_1 \cdot 1^n = \alpha_1$, since the characteristic equation is $\lambda = 1$. We remark that the general solution of (C.1) is indeed

$$x_n = x_0 + p(1) + p(2) + \cdots + p(n) = x_0 + \sum_{k=1}^{n} p(k).$$

With the notation of (3.4), $g_n = \alpha_1$ corresponds to the constant x_0. The possibility of finding an exact, closed formula for the right hand side depends heavily on the function p. If p belongs to the set $\mathcal{A}$ defined in Section 3.2, we can find this closed expression (see Appendix B for details). In most cases, though, even if p is a rather "simple" function such as $p(n) = 1/n$, $p(n) = \sqrt{n}$ or $p(n) = \log n$, this is not possible and we can only find approximate solutions by means of suitable summation formulæ. We will devote a future paper of this series to the problem of approximate summation: see [4].

Example C.2 *Solve the first-order recurrence*[6]

$$x_n = 2x_{n-1} + 2^{n-1} + n + 1. \tag{C.2}$$

[6]This is taken from the complexity analysis of a computer program.

Here we have $\alpha \stackrel{\text{def}}{=} 2$ and $p(n) \stackrel{\text{def}}{=} 2^{n-1}+n+1$, which we split as $p_1(n) \stackrel{\text{def}}{=} 2^{n-1}$ and $p_2(n) \stackrel{\text{def}}{=} n+1$. We call the resulting recurrences y_n and z_n respectively. We compute the solution for p_1 by means of (3.8), which yields

$$y_n = \sum_{k=1}^{n} 2^{n-k} \cdot 2^{k-1} = \sum_{k=1}^{n} 2^{n-1} = n2^{n-1}. \tag{C.3}$$

The trick explained in Appendix B.2 shows that it is convenient to write

$$z_n = \sum_{k=1}^{n} 2^{n-k}(k+1) = 2^n \sum_{k=1}^{n} k2^{-k} + 2^n \sum_{k=1}^{n} 2^{-k}$$

and we find (simplifying somewhat and omitting details)

$$\begin{aligned} z_n &= 2^n \left(-n\left(\frac{1}{2}\right)^n - 2\left(\frac{1}{2}\right)^n + 2 \right) + 2^n \left(2\left(1 - \left(\frac{1}{2}\right)^{n+1}\right) - 1 \right) \\ &= \left(-n - 2 + 2^{n+1}\right) + \left(2^n - 1\right) = 3 \cdot 2^n - n - 3. \end{aligned}$$

Summing the three solutions, we finally get

$$x_n = x_0 2^n + n2^{n-1} + 3 \cdot 2^n - n - 3.$$

An easy induction shows that this is, indeed, the solution of our problem.

Example C.3 *The same procedure solves the simpler recurrence*[7]

$$x_n = 2x_{n-1} + 2^n.$$

We just sketch the argument: the above discussion shows that the solution has the form $x_n = x_0 2^n + y_n$, where y_n is the solution of the same recurrence with $y_0 = 0$. A computation similar to that in (C.3) gives $y_n = n2^n$, so that $x_n = (x_0 + n)2^n$, as an induction proves immediately.

Example C.4 *Solve*

$$x_n = \alpha x_{n-1} + n^2.$$

Corollary 3.4 implies that the case $\alpha = 1$ is rather special. By (3.8) we need to find a closed formula for

$$\sum_{k=1}^{n} \alpha^{n-k} k^2 = \alpha^n \sum_{k=1}^{n} \alpha^{-k} k^2.$$

If $\alpha = 1$, a direct application of (B.6) from Appendix B.3 yields the answer

$$x_n = x_0 + \frac{1}{6} n(n+1)(2n+1).$$

[7] This is also taken from the complexity analysis of a computer program.

If $\alpha \neq 1$, the technique described in Appendix B.2 suggests to write

$$\sum_{k=1}^{n} \alpha^{-k} k^2 = \sum_{k=1}^{n} k(k-1)\alpha^{-k} + \sum_{k=1}^{n} k\alpha^{-k},$$

so that the answer is

$$\begin{aligned}
\sum_{k=1}^{n} \alpha^{-k} k^2 &= \left[z^2 \frac{\mathrm{d}^2}{\mathrm{d}z^2} \frac{1-z^{n+1}}{1-z} \right]_{z=1/\alpha} + \left[z \frac{\mathrm{d}}{\mathrm{d}z} \frac{1-z^{n+1}}{1-z} \right]_{z=1/\alpha} \\
&= \left[\frac{z + z^2 - (n+1)^2 z^{n+1} + (2n^2+2n-1) z^{n+2} - n^2 z^{n+3}}{(1-z)^3} \right]_{z=1/\alpha} \\
&= \frac{\alpha^2 + \alpha + \alpha^{-n}\big(-n^2 + (2n^2+2n-1)\alpha - (n+1)^2\alpha^2\big)}{(\alpha-1)^3}.
\end{aligned}$$

The closed formula follows at once:

$$x_n = x_0 \cdot \alpha^n + \frac{\big(\alpha^{n+2} + \alpha^{n+1} - n^2 + (2n^2+2n-1)\alpha - (n+1)^2\alpha^2\big)}{(\alpha-1)^3}.$$

C.2 Recurrences of Order 2

Example C.5 *Another interesting case is the famous sequence of the Fibonacci numbers: here and henceforward we write f_n for the solution of the second order recurrence*

$$\begin{cases} f_n = f_{n-1} + f_{n-2}, & \text{for } n \geq 2; \\ f_0 = 0, \\ f_1 = 1. \end{cases}$$

The characteristic equation for this recurrence is

$$\lambda^2 = \lambda + 1$$

with roots $\lambda_1 = \frac{1}{2}(1+\sqrt{5})$ and $\lambda_2 = \frac{1}{2}(1-\sqrt{5}) = -\lambda_1^{-1}$. The general solution of any such recurrence is thus

$$\alpha\lambda_1^n + \beta\lambda_2^n,$$

but, since $f_0 = 0$ and $f_1 = 1$, we have

$$\begin{cases} \alpha + \beta = 0, \\ \alpha\lambda_1 + \beta\lambda_2 = 1, \end{cases} \qquad \text{whence} \qquad \begin{cases} \alpha = 5^{-1/2} \\ \beta = -5^{-1/2} \end{cases}$$

and the n-th Fibonacci number is

$$f_n = \frac{1}{\sqrt{5}} \left(\Big(\frac{1+\sqrt{5}}{2}\Big)^n - \Big(\frac{1-\sqrt{5}}{2}\Big)^n \right)$$

(this is known as Binet's formula). We observe a rather general phenomenon: here $|\lambda_1| > |\lambda_2|$ and hence the value of f_n is quite close to

$$\frac{1}{\sqrt{5}}\Big(\frac{1+\sqrt{5}}{2}\Big)^n.$$

Indeed, but this is rather special, since $|\lambda_2| < 1$, f_n is the integer nearest the above expression. More generally, we remark that even if we can not find *all* roots of the characteristic equation, for the asymptotic analysis of the behavior of the solution of the recurrence it is sufficient to find the ones with the largest modulus. We will develop this topic in a future paper dealing with approximate solutions: see [4].

It is convenient to note another general phenomenon: if x_n satisfies $x_n = x_{n-1} + x_{n-2}$, then $x_n = f_n x_1 + f_{n-1} x_0$, as can be seen by induction. In other words, the solution with $x_0 = 0$ and $x_1 = 1$ is quite important, and indeed it is closely allied to the fundamental one as defined in (3.11), which is $g_n = f_{n+1}$.

We want to see how things change if the characteristic equation has a double root.

Example C.6 *Solve*

$$\begin{cases} x_n = \alpha x_{n-1} + \beta x_{n-2}, & \textit{for } n \geq 2; \\ x_0 = 0, \\ x_1 = 1, \end{cases}$$

where $\alpha^2 + 4\beta = 0$*, with* $\beta \neq 0$*.*

We now look for a solution of the form

$$x_n = (a + bn)\lambda^n,$$

where λ satisfies $\lambda^2 = \alpha\lambda + \beta = \alpha\lambda - \frac{1}{4}\alpha^2$. Since $x_0 = 0$ and $x_1 = 1$ we have

$$\begin{cases} a = 0, \\ (a+b)\lambda = 1, \end{cases} \qquad \text{whence} \qquad \begin{cases} a = 0, \\ b = \lambda^{-1}. \end{cases}$$

Thus the solution is

$$x_n = n\lambda^{n-1} = n\Big(\frac{\alpha}{2}\Big)^{n-1},$$

as can be checked easily.

Example C.7 *Solve*

$$\begin{cases} x_n = 2x_{n-1} - 2x_{n-2}, & \textit{for } n \geq 2; \\ x_0 = 0, \\ x_1 = 1. \end{cases}$$

This is a nice example of a recurrence of order 2 whose characteristic equation has complex roots. Its first few values are 0, 1, 2, 2, 0, −4, −8, −8, 0, 16, 32, 32, 0, ... The procedure described in Section 3 yields

$$x_n = \frac{1}{2i}\big((1+i)^n - (1-i)^n\big).$$

Since $1+i = \sqrt{2}\cdot e^{i\pi/4}$, and $1-i = \sqrt{2}\cdot e^{-i\pi/4}$, we can also write

$$x_n = 2^{n/2}\frac{e^{in\pi/4} - e^{-in\pi/4}}{2i} = 2^{n/2}\sin\frac{n\pi}{4}.$$

Both solution are correct, though the latter can probably be considered easier to understand, as it conveys more readily the relevant informations about x_n and avoids complex numbers. The oscillations in absolute value are due to the presence of two complex conjugate roots.

Example C.8 *This recurrence arises from a simple combinatorial problem.*

$$x_n = x_{n-1} + x_{n-2} + 1. \tag{C.4}$$

Here $g_n = f_{n+1}$ is the $(n+1)$-st Fibonacci number.

$$x_n = x_1 f_n + x_0 f_{n-1} + (f_0 + f_1 + f_2 + \cdots + f_{n-1}) = x_1 f_n + x_0 f_{n-1} + f_{n+1} - 1.$$

The last equality follows from a well-known property of the Fibonacci numbers.

Example C.9 *Solve*

$$\begin{cases} x_n = \frac{1}{2}x_{n-1} + \frac{1}{2}x_{n-2} + 1, & \text{for } n \geq 2; \\ x_0 = 0, \\ x_1 = 0. \end{cases}$$

In this case the characteristic equation has a simple root $\lambda = 1$. Here $g(n) = \frac{2}{3}\big(1 - (\frac{-1}{2})^{n+1}\big)$ (the initial conditions are $g_0 = 1$ and $g(1) = \frac{1}{2}$) and the solution is

$$x_n = \frac{2}{3}n - \frac{4}{9}\left(1 - \left(\frac{-1}{2}\right)^n\right).$$

Example C.10 *Solve*

$$\begin{cases} x_n = 2x_{n-1} - x_{n-2} + 1, & \text{for } n \geq 2; \\ x_0 = 0, \\ x_1 = 0. \end{cases}$$

In this case the characteristic equation has the double root $\lambda = 1$. According to (3.11) we set

$$\begin{cases} g_n = 2g_{n-1} - g_{n-2}, & \text{for } n \geq 2; \\ g_0 = 1, \\ g_1 = 2. \end{cases}$$

Since $\lambda = 1$ is a double root, we look for a solution of the type $g(n) = an + b$ for suitable a, $b \in \mathbb{R}$. The initial conditions imply $a = b = 1$ so that $g(n) = n + 1$. The presence of the root $\lambda = 1$ forces the general solution to a polynomial of degree 2 (see Corollary 3.4), and actually

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

In the next few examples we show how to use the "guessing" procedure described in Section 3.4.1 in several cases. We recall that f_n denotes the n-th Fibonacci number.

Example C.11 *We start with a linear polynomial. Solve the recurrence*

$$\begin{cases} x_n = x_{n-1} + x_{n-2} + n, & \textit{for } n \geq 2; \\ x_0 = 0, \\ x_1 = 0. \end{cases}$$

Solving (3.15) for q we find $q(n) = n + 3$ so that we have

$$x_n = q(1)f_n + q(0)f_{n-1} - q(n) = 4f_n + 3f_{n-1} - n - 3.$$

Example C.12 *Next, a quadratic polynomial. Solve*

$$\begin{cases} x_n = x_{n-1} + x_{n-2} + n^2, & \textit{for } n \geq 2; \\ x_0 = 0, \\ x_1 = 0. \end{cases}$$

Here we find that the solution of (3.15) is $q(n) = n^2 + 6n + 13$ and we have

$$x_n = q(1)f_n + q(0)f_{n-1} - q(n) = 20f_n + 13f_{n-1} - n^2 - 6n - 13.$$

Example C.13 *Finally, a cubic polynomial*

$$\begin{cases} x_n = x_{n-1} + x_{n-2} + n^3, & \textit{for } n \geq 2; \\ x_0 = 0, \\ x_1 = 0. \end{cases}$$

In this case $q(n) = n^3 + 9n^2 + 30n + 54$ so that

$$x_n = q(1)f_n + q(0)f_{n-1} - q(n) = 94f_n + 54f_{n-1} - n^3 - 9n^2 - 30n - 54.$$

Example C.14 *More generally, solve*

$$x_n = \alpha x_{n-1} + \beta x_{n-2} + 2^n.$$

We look for a function of the type $q(n) = \gamma \cdot 2^n$ that satisfies the functional equation (3.15). It is quite easy to see that we need

$$2^n = \alpha\gamma \cdot 2^{n-1} + \beta\gamma \cdot 2^{n-2} - \gamma \cdot 2^n,$$

and this implies

$$\gamma = \frac{4}{2\alpha + \beta - 4}.$$

Note that this fails if $2\alpha+\beta = 4$ (that is, if 2 is a solution of the characteristic equation), and in this case we look for a solution of the form $q(n) = \gamma n \cdot 2^n$. Indeed, if $2\alpha + \beta = 4$ we have

$$2^n = \alpha\gamma(n-1) \cdot 2^{n-1} + \beta\gamma(n-2) \cdot 2^{n-2} - \gamma n \cdot 2^n,$$

so that

$$\gamma = -\frac{2}{\alpha + \beta}.$$

Again, this fails if we also have $\alpha + \beta = 0$, but the two conditions together imply that $\alpha = -\beta = 4$ so that 2 is a *double* root of the characteristic equation, and in this case we look for a solution of the type $q(n) = \gamma n^2 \cdot 2^n$. We remark that now the recurrence is indeed

$$x_n = 4x_{n-1} - 4x_{n-2} + 2^n.$$

We want to find $\gamma \in \mathbb{R}$ so that

$$2^n = 4\gamma(n-1)^2 \cdot 2^{n-1} - 4\gamma(n-2)^2 \cdot 2^{n-2} - \gamma n^2 \cdot 2^n.$$

Simplifying, we find $\gamma = -\frac{1}{2}$, and the general solution is $x_n = (1-n)x_0 \cdot 2^n + nx_1 \cdot 2^{n-1} - n \cdot 2^{n-2} + \frac{1}{2}n^2$.

C.3 Recurrences of Higher Order

Example C.15 *Solve*

$$x_n = 2x_{n-1} - x_{n-2} + 2x_{n-3}.$$

This is more difficult example than the previous ones, since the recurrence has order 3. The characteristic equation has the roots $\lambda_1 = 2$, $\lambda_2 = i$, $\lambda_3 = -i$, and the general solution is therefore

$$x_n = \alpha \cdot 2^n + \beta \cdot i^n + \gamma \cdot (-i)^n,$$

for suitable complex numbers α, β, γ. The computation of these coefficients reveals that the solution has the form

$$x_n = \frac{1}{5}(x_0 + x_2) \cdot 2^n + \begin{cases} \frac{1}{5}(4x_0 - x_2) & \text{if } n \equiv 0 \bmod 4, \\ -\frac{1}{5}(2x_0 - 5x_1 + 2x_2) & \text{if } n \equiv 1 \bmod 4, \\ -\frac{1}{5}(4x_0 - x_2) & \text{if } n \equiv 2 \bmod 4, \\ \frac{1}{5}(2x_0 - 5x_1 + 2x_2) & \text{if } n \equiv 3 \bmod 4, \end{cases} \tag{C.5}$$

where, clearly, the rightmost expression is periodic with period 4 and is therefore $\mathcal{O}_{x_0,x_1,x_2}(1)$. As above in Example C.7 we can also transform i and $-i$ into $e^{i\pi/2}$ and $e^{-i\pi/2}$ respectively, so that $i^n = \cos(n\pi/2) + i\sin(n\pi/2)$ and $(-i)^n = \cos(n\pi/2) - i\sin(n\pi/2)$. This implies the following alternative formula for x_n:

$$x_n = \frac{1}{5}(x_0 + x_2) \cdot 2^n + \frac{1}{5}(4x_0 - x_2)\cos\Big(\frac{n\pi}{2}\Big) + \frac{1}{5}(5x_1 - 2x_0 - 2x_2)\sin\Big(\frac{n\pi}{2}\Big),$$

which, of course, is the same as (C.5). We remark that it is possible to avoid any computations with complex numbers (see Appendix B.5), since we consider only real recurrences, and therefore the characteristic equation has pairs of complex conjugate roots, if any. Here we do not have oscillations in absolute value as in Example C.7, since there is only one root of maximal modulus, and it is real and positive: therefore the term containing 2^n dominates the solution, unless $x_0 + x_2 = 0$. If this is the case, the solution is $x_n = x_0\cos(n\pi/2) + x_1\sin(n\pi/2)$, and is periodic with period 4, and therefore bounded.

C.4 Order Reduction

Example C.16 *Solve*

$$x_n = 2x_{n-2} + 1.$$

Applying the traditional method exposed in Section 3.4, the solution is

$$\begin{aligned} x_n = 2^{\frac{n}{2}-1}\big(1 + (-1)^n\big)x_0 + 2^{\frac{n}{2}-\frac{3}{2}}\big(1 - (-1)^n\big)x_1 \\ + \sqrt{2}^n\left(\frac{1}{2} + \frac{1}{4}\sqrt{2}\right) + (-\sqrt{2})^n\left(\frac{1}{2} - \frac{1}{4}\sqrt{2}\right) - 1. \end{aligned}$$

Applying the order reduction method in Section 3.6, instead, the solution has the following shape

$$x_n = x_r\, 2^{-\frac{1}{2}r + \frac{1}{2}n} + 2^{-\frac{1}{2}r + \frac{1}{2}n} - 1,$$

where $r = n \bmod 2 \in \{0,\, 1\}$, more readable and concise than the previous one.

Example C.17 *Solve*

$$x_n = 2x_{n-2} - x_{n-4} + n.$$

The traditional method yields the solution

$$\begin{aligned} x_n =& \frac{1}{4}\big(2 + 2(-1)^n - (-1)^n n - n\big)x_0 + \frac{1}{4}\big(3 - 3(-1)^n + (-1)^n n - n\big)x_1 \\ &+ \frac{1}{4}\big((-1)^n n + n\big)x_2 + \frac{1}{4}\big(-1 + (-1)^n - (-1)^n n + n\big)x_3 \\ &+ \frac{5}{8} - \frac{5}{8}(-1)^n + \frac{1}{24}n^3 + \frac{1}{4}n^2 + \frac{7}{16}(-1)^n n - \frac{53}{48}n. \end{aligned}$$

Applying the order reduction method we obtain the equivalent solution

$$\begin{aligned} x_n = \left(1 - \frac{1}{2}n + \frac{1}{2}r\right) x_r + \left(\frac{1}{2}n - \frac{1}{2}r\right) x_{r+2} \\ + \frac{1}{12}r^3 + \frac{1}{2}r^2 + \frac{1}{24}n^3 + \frac{1}{4}n^2 - \frac{1}{8}r^2 n - \frac{3}{4}rn + \frac{2}{3}r - \frac{2}{3}n, \end{aligned}$$

where $r = n \bmod 2 \in \{0,\, 1\}$, which is again much simpler.