

Folding Difference and Differential Systems into Higher Order Equations

H. SEDAGHAT ¹

Abstract

A typical system of k difference (or differential) equations can be compressed, or folded into a difference (or ordinary differential) equation of order k . Such foldings appear in control theory as the canonical forms of the controllability matrices. They are also used in the classification of systems of three nonlinear differential equations with chaotic flows by examining the resulting jerk functions. The solutions of the higher order equation yield one of the components of the system's k -dimensional orbits and the remaining components are determined from a set of associated passive equations. The folding algorithm uses a sequence of substitutions and inversions along with index shifts (for difference equations) or higher derivatives (for differential equations). For systems of two difference or differential equations this compression process is short and in some cases yields second-order equations that are simpler than the original system. For all systems, the folding algorithm yields detailed amount of information about the structure of the system and the interdependence of its variables. As with two equations, some special cases where the derived higher order equation is simpler to analyze than the original system are considered.

1 Introduction

It is common knowledge that a difference or ordinary differential equation of order k may be “unfolded” in a standard way to a system of k first order difference or differential equations. A reverse process that compresses, or “folds” systems into higher order equations is also possible, not just in rare instances. Folding linear systems in both continuous and discrete time is seen in control theory; and folding nonlinear systems of differential equations appears in the study of conditions that lead to the occurrence of chaotic flows.

In control theory the “controllability canonical form” is precisely the folding, in the sense to be made precise here, of a controllability matrix into a linear higher order equation, whether in continuous or discrete time; see, e.g., [2], [5], [9]. Using standard algebraic methods, a completely controllable system is found to be equivalent to a linear equation (difference in discrete time, differential in continuous time) whose order equals the rank of the controllability matrix.

In an entirely different line of research, in [4] and [10] a variety of nonlinear differential systems displaying chaotic behavior are studied and classified by converting them to ordinary differential

¹Department of Mathematics, Virginia Commonwealth University Richmond, Virginia, 23284-2014, USA; Email: h.sedaghat@discretedynamics.net

equations of order 3 that are defined by jerk functions (time rates of change of acceleration). These systems include the well-known systems of Lorenz [11] and Rössler [12] and most of the 19 autonomous, minimally nonlinear systems introduced by Sprott in [15] each containing only a single quadratic term. These results define new categories for distinguishing among a broad range of differential systems; for instance, the systems of Lorenz and Rössler can both be converted to third-order ordinary differential equations using the same approach, but while Rössler's system folds globally with a jerk function having no singularities, known jerk functions for Lorenz's system are not defined globally.

In this paper, we find that the aforementioned ideas in control theory and in chaotic differential systems are special instances of the same concept, namely, folding systems to equations. We take three further steps beyond the aforementioned literature. First, we apply folding to systems generally, whether linear or nonlinear, autonomous or not, in discrete time or in continuous time. The basic idea behind our approach here is simple: starting with a given system, a higher order equation is derived through a sequence of *substitutions*, *inversions* and *index shifts* (for difference systems) or *higher derivatives* (for differential ones). These three actions constitute the main components of the *folding algorithm*. Although not the most elegant, an algorithmic approach is preferred in the general context because folding a typical system with many equations is an intricate and often tedious process.

For systems of two equations folding is a short, one-step process that is practically useful in some cases where the resulting second-order equation is simpler or more tractable than the original system. For a system of 3 or more difference or differential equations, the folding algorithm, namely, the iteration of the aforementioned 3-component folding process in principle converts the system to a higher order equation with the inversion component being the most technically uncertain part. Once a system is folded, results formulated for the higher order equation may be used to analyze the solutions of the system.

Taking a further step beyond the existing literature, we illustrate how folding may be used in a precise way to examine the interdependence of variables in a system. The aforementioned distinction between chaotic differential systems of Lorenz, Rössler and those in the Sprott family is a case in point. In addition, non-fully controllable systems can be folded by the same algorithm as the fully controllable ones and the folding process indicates some of the differences between the two types of systems. More generally, for parameter-dependent systems certain parameter values tend to uncouple the system, at least partially. In the folding process, such parameter values appear as singularity conditions and usually lead to equations with orders that are lower than the number of equations in the original system. Whether lower order foldings occur as a result of parameter changes or are due to other structural features of a system, folding a system provides new insights into the interdependence of variables.

Finally, in principle folding applies to non-autonomous systems in the same way that it does to autonomous ones. Consideration of non-autonomous systems is an important generalization on its own merit and it also leads to practical benefits in some cases. We discuss examples of non-autonomous systems that fold to higher order equations that are either autonomous or periodic,

thus making a greater range of methods applicable to those systems.

2 Folding difference systems with two equations

We begin with systems of two equations for which the folding process is relatively simple. A (recursive or explicit) system of two first-order difference equations is typically defined as

$$\begin{cases} x_{n+1} = f(n, x_n, y_n) \\ y_{n+1} = g(n, x_n, y_n) \end{cases} \quad n = 0, 1, 2, \dots \quad (1)$$

where $f, g : \mathbb{N} \times D \rightarrow S$ are given functions, \mathbb{N} is the set of non-negative integers, S a nonempty set and $D \subset S \times S$. An initial point $(x_0, y_0) \in D$ generates a (forward) orbit or solution $\{(x_n, y_n)\}$ of (1) in the state-space $S \times S$ through the iteration of the function

$$(n, x_n, y_n) \rightarrow (f(n, x_n, y_n), g(n, x_n, y_n)) : \mathbb{N} \times D \rightarrow S \times S$$

for as long as the points (x_n, y_n) remain in D . If (1) is *autonomous*, i.e., the functions f, g do not depend on the index n then $(x_n, y_n) = F^n(x_0, y_0)$ for every n where F^n denotes the composition of the map $F(u, v) = (f(u, v), g(u, v))$ of $S \times S$ with itself n times.

A second-order, scalar difference equation in S is defined as

$$s_{n+2} = \phi(n, s_n, s_{n+1}), \quad n = 0, 1, 2, \dots \quad (2)$$

where $\phi : \mathbb{N} \times D' \rightarrow S$ is a given function and $D' \subset S \times S$. A pair of initial values $s_0, s_1 \in S$ generates a (forward) solution $\{s_n\}$ of (2) in S if $(s_0, s_1) \in D'$. As in the case of systems, if $\phi(n, u, v) = \phi(u, v)$ is independent of n then (2) is autonomous.

An equation of type (2) may be “unfolded” to a system of type (1) in a standard way; e.g.,

$$\begin{cases} s_{n+1} = t_n \\ t_{n+1} = \phi(n, s_n, t_n) \end{cases} \quad (3)$$

In the system (3) the temporal delay in (2) is converted to an additional variable in the state space. All solutions of (2) are reproduced from the solutions of (3) in the form $(s_n, s_{n+1}) = (s_n, t_n)$ so in this sense, higher order equations may be considered to be special types of systems. In general, (2) has many possible unfoldings into systems of two equations and (3) is not unique.

2.1 An example

To highlight some issues of interest consider the system

$$\begin{cases} x_{n+1} = x_n y_n \\ y_{n+1} = (a + b x_n) / y_n \end{cases} \quad (4)$$

of difference equations with $a, b \in \mathbb{R}$ (see Section 5 below for how a system like (4) may be derived). The domain D of this system is the complement of the x-axis ($y_n \neq 0$) in the plane \mathbb{R}^2 . The first equation yields

$$y_n = \frac{x_{n+1}}{x_n} \quad (5)$$

on the complement of the y -axis ($x_n \neq 0$). Further, shifting the index of x_n in the first equation in (4) by 1 yields

$$x_{n+2} = x_{n+1}y_{n+1} = x_{n+1} \left(\frac{a + bx_n}{y_n} \right) = x_{n+1} \left(\frac{ax_n + bx_n^2}{x_{n+1}} \right) = x_n(a + bx_n) \quad (6)$$

This second-order equation is effectively first-order in the sense that its even and odd terms

$$\begin{aligned} x_{2k+2} &= x_{2k}(a + bx_{2k}), & x_0 \text{ given and } k \geq 0 \\ x_{2k+1} &= x_{2k-1}(a + bx_{2k-1}), & x_1 = x_0y_0 \text{ given and } k \geq 1 \end{aligned}$$

separately satisfy the same first-order recurrence $r_{n+1} = r_n(a + br_n)$. This recurrence is conjugate to the logistic equation when $a, b \neq 0$ since

$$r_{n+1} = ar_n \left(1 + \frac{b}{a}r_n \right) \quad \text{so if } s_n = -\frac{b}{a}r_n \text{ then } s_{n+1} = as_n(1 - s_n). \quad (7)$$

This equation exhibits well-known dynamics in a bounded interval, including chaotic behavior for a range of values of b .

The above equations determine the value of the x-component x_n in the pair (x_n, y_n) . If $x_n \neq 0$ for all n then y_n may be calculated from (5) without having to use the second equation in the system. Otherwise, it is necessary to use the second equation of (4) to calculate y_n recursively. This situation occurs when $x_0 = 0$ in which case, $x_1 = x_0y_0 = 0$ also and we obtain the trivial solution of (6) i.e., $x_n = 0$ for all n . In this case, the second equation of the system gives

$$y_{n+1} = \frac{a + bx_n}{y_n} = \frac{a}{y_n}$$

All solutions $\{y_0, a/y_0, y_0, a/y_0, \dots\}$ of this simple first-order autonomous equation have period 2. For the planar system, these translate into 2-cycles on the y -axis; i.e., the y -axis is invariant and all solutions that start on it are (stable but non-attracting) 2-cycles. It is worth mentioning that this situation does not occur in the exceptional case $a = 0$.

Next, we note that *certain solutions of (6) do not yield a solution for the system*, a situation that occurs because the domain of the system is properly contained in the domain of (6), namely, \mathbb{R}^2 . If $a, b \neq 0$, $x_0 = -a/b$ and $y_0 \neq 0$ then from (6) we find that $x_{2k} = 0$ for all k while x_{2k-1} is determined by the logistic equation (7). This solution of (6) does not generate a solution for (4) since $y_1 = 0$.

The preceding analysis determines the *global behavior* of the solutions of the system (4). In particular, it is easy to verify that the results are consistent with information obtained through the standard determination of fixed points and their local stability.

2.2 Semi-invertibility

The preceding discussion in particular raises the following issues:

- It is possible to solve at least one equation of the system for one of the variables. In the above example, the first equation is solved for y_n as (5). This not only eliminates y_n but also makes it possible to determine y_n from a solution of (6) or without having to solve the second equation of (4) as a (nonautonomous) difference equation;
- To take advantage of the preceding item, certain domain and range restrictions must be specified in order to pin down the precise relationship between the solutions of the system and those of the higher order equation.

The following concept addresses the above two issues.

Definition 1 *Let S, T be nonempty sets and consider a function $f : T \times D \rightarrow S$ where $D \subset S \times S$. Then f is **semi-invertible** or **partially invertible** if there are sets $M \subset D$, $M' \subset S \times S$ and a function $h : T \times M' \rightarrow S$ such that*

$$w = f(t, u, v) \Rightarrow v = h(t, u, w) \quad \text{for all } t \in T, (u, v) \in M \text{ and } (u, w) \in M'. \quad (8)$$

The function h may be called a semi-inversion, or partial inversion of f . If f is independent of t then t is dropped from the above notation (see the examples below).

Semi-inversion refers more accurately to the *solvability* of the equation $w - f(t, u, v) = 0$ for v . This recalls the implicit function theorem. A general version that is based on the contraction principle holds in Banach spaces [8] yields the following existence result when applied to the function $w - f(u, v)$ (the t -independent case). The reader who is not interested in this level of generality may simply think of the Banach space as the real line.

Proposition 2 *Let S be a Banach space and \mathcal{O} an open set in $S \times S$. Let $f : \mathcal{O} \rightarrow S$ be a C^k -differentiable function for a positive integer k and suppose that $f(u_0, v_0) = w_0$ for some $(u_0, v_0) \in \mathcal{O}$. If the partial derivative $\partial_2 f(u_0, v_0)$ is invertible (as a linear map) then there exists $r > 0$ and a C^k function $h_0 : B_r(w_0, u_0) \rightarrow S$ defined on the open ball of radius r centered at $(w_0, u_0) \in S \times S$ such that $f(u, h_0(w, u)) = w$ for all $(w, u) \in B_r(w_0, u_0)$. Further,*

$$\text{if } \|(w - w_0, u - u_0)\|, \|v - v_0\| < r \text{ and } w = f(u, v) \text{ then } v = h_0(w, u).$$

The function h_0 in Proposition 2 is thus a semi-inversion of f on the open ball $B_r(w_0, u_0)$ in the autonomous case. Often a function h that is defined on a larger set than the ball and equals h_0 on the ball satisfies the above conditions, although the preceding result does not imply that much in general.

Going in a different direction, a substantial class of semi-invertible functions is supplied (globally) by the following idea. This notion is defined in an algebraically general context although it is less generic analytically.

Definition 3 Let $(G, *)$ be a nontrivial group, T a nonempty set and let $f : T \times G \times G \rightarrow G$. If there are functions $f_1, f_2 : T \times G \rightarrow G$ such that

$$f(t, u, v) = f_1(t, u) * f_2(t, v)$$

for all $u, v \in G$ and every $t \in T$ then we say that f is separable on G and write $f = f_1 * f_2$ for short.

Every affine function $f(n, u, v) = a_n u + b_n v + c_n$ with a_n, b_n, c_n in a ring R with identity is separable on the additive group $(R, +)$ for all $n \geq 1$ with $T = \mathbb{N}$. Similarly, $f(t, u, v) = a(t)u + b(t)v + c(t)$ is separable on $(\mathbb{R}, +)$ for all $t \in \mathbb{R}$.

Now, suppose that $f_2(t, \cdot)$ is a bijection for every $t \in T$ and $f_2^{-1}(t, \cdot)$ is its inverse for each t ; i.e., $f_2(t, f_2^{-1}(t, v)) = v$ and $f_2^{-1}(t, f_2(t, v)) = v$ for all v . Evidently, a separable function f is semi-invertible if the component function $f_2(t, \cdot)$ is a bijection for each fixed t , since for every $u, v, w \in G$ and $t \in T$

$$w = f_1(t, u) * f_2(t, v) \Rightarrow v = f_2^{-1}(t, [f_1(t, u)]^{-1} * w)$$

where map inversion and group inversion, both denoted by -1 , are distinguishable from the context. In this case, an explicit expression for the semi-inversion h exists globally as

$$h(t, u, w) = f_2^{-1}(t, [f_1(t, u)]^{-1} * w) \tag{9}$$

with $M = M' = G \times G$. We summarize this observation as follows.

Proposition 4 Let $(G, *)$ be a nontrivial group and $f = f_1 * f_2$ be separable. If $f_2(t, \cdot)$ is a bijection for each t then f is semi-invertible on $G \times G$ with a semi-inversion uniquely defined by (9).

For the system (4) in the last section, the function $f(u, v) = a + uv$ is separable if $a = 0$ on G where G is the group of nonzero real numbers under ordinary multiplication. Here $f_1(u) = f_2(u) = u$ are both bijections, making f semi-invertible. If $a \neq 0$ then f is not separable but it is semi-invertible with $h(u, w) = (w - a)/u$ on the complement of the y-axis in the plane (the set M'). Note that f is semi-invertible for all values of a where $u \neq 0$.

The function $f(u, v) = (v \circ u)^{-1} = u^{-1} \circ v^{-1}$ on a permutation group G is separable with $f_2(v) = v^{-1}$ a bijection. Thus, f is semi-invertible with $h(u, w) = (u \circ w)^{-1}$. The function $f(u, v) = a_n u + b_n v + c_n$ with a_n, b_n, c_n in a ring R with identity is separable on the additive group $(R, +)$ for all $n \geq 1$ with $f_1(n, v) = a_n v + c_n$ and $f_2(n, v) = b_n v$. If b_n is a unit in R for all large n then $f_2(n, \cdot)$ is a bijection for all such n and thus, f is semi-invertible on R with $h(n, u, w) = b_n^{-1}(w - a_n u - c_n)$ for large n . If a_n, b_n are not units for large n then f is separable but not semi-invertible for either u or v .

Next, suppose that $\{(x_n, y_n)\}$ is a solution of (1) in D . Assume that one of the component functions in (1), say, f is semi-invertible. Then for $n \geq 0$ there is a set $M \subset D$, a set $M' \subset S \times S$ and a function $h : \mathbb{N} \times M' \rightarrow S$ such that for $(x_n, y_n) \in M$ and $(x_n, x_{n+1}) \in M'$

$$x_{n+1} = f(n, x_n, y_n) \Rightarrow y_n = h(n, x_n, x_{n+1}) \tag{10}$$

Therefore,

$$x_{n+2} = f(n+1, x_{n+1}, y_{n+1}) = f(n+1, x_{n+1}, g(n, x_n, y_n)) = f(n+1, x_{n+1}, g(n, x_n, h(n, x_n, x_{n+1}))) \quad (11)$$

For each $n \geq 0$ the function

$$\phi(n, u, w) = f(n+1, w, g(n, u, h(n, u, w))) \quad (12)$$

is defined on $\mathbb{N} \times M'$. If $\{s_n\}$ is the solution of (2) with initial values $s_0 = x_0$ and $s_1 = x_1 = f(0, x_0, y_0)$ where ϕ defined by (12) then

$$s_2 = f(1, s_1, g(0, s_0, h(0, s_0, s_1))) = f(1, x_1, g(0, x_0, h(0, x_0, x_1))) = f(1, x_1, g(0, x_0, y_0)) = x_2$$

By induction, $s_n = x_n$ and by (10) $h(n, s_n, s_{n+1}) = h(n, x_n, x_{n+1}) = y_n$. It follows that

$$(x_n, y_n) = (s_n, h(n, s_n, s_{n+1})) \quad (13)$$

i.e., the solution $\{(x_n, y_n)\}$ of (1) can be obtained from a solution $\{s_n\}$ of (2). These observations establish the following result.

Theorem 5 *Suppose that f in (1) is semi-invertible with M, M' and h as in (8). Then each orbit of (1) in D on which h is defined may be derived from a solution of (2) via (13) with ϕ given by (12).*

The following gives a name to the pair of equations that generate the solutions of (1) in the above theorem.

Definition 6 (*Folding*) *The pair of equations*

$$\begin{aligned} s_{n+2} &= \phi(n, s_n, s_{n+1}), & s_0 &= x_0, & s_1 &= f(0, x_0, y_0) \\ y_n &= h(n, x_n, x_{n+1}) \end{aligned}$$

where ϕ is defined by (12) is a folding of the system (1).

Note that the equation for y_n is *passive* in the sense that it simply evaluates a given function and no dynamics or iterations are involved. Also observe that (1) may be considered an unfolding of the second-order equation above. It is generally not equivalent to the standard unfolding (3) of that equation.

If one of the component functions in the system is separable then a global result is obtained using (9).

Corollary 7 Let $(G, *)$ be a nontrivial group and $f = f_1 * f_2$ be separable on $G \times G$. If $f_2(n, \cdot)$ is a bijection for every n then every solution $\{(x_n, y_n)\}$ of (1) in G is derived from a solution $\{s_n\}$ of

$$s_{n+2} = f_1(n+1, s_{n+1}, g(n, s_n, f_2^{-1}(n, [f_1(n, s_n)]^{-1} * s_{n+1}))) \quad (14)$$

that yields the x -component x_n with the initial values $s_0 = x_0$, $s_1 = f_1(0, x_0) * f_2(0, y_0)$. Further, the solution $\{s_n\}$ of (14) yields, passively via (9), the y -component

$$y_n = f_2^{-1}(n, [f_1(n, s_n)]^{-1} * s_{n+1}). \quad (15)$$

For example, if f, g are as defined in (4) then (14) yields

$$s_{n+2} = s_{n+1} \frac{b + cs_n}{s_{n+1}/s_n} = s_n(b + cs_n)$$

which matches the earlier results above. In this case, G is the group of nonzero real numbers under multiplication.

The next result is, strictly speaking, a special case of Corollary 7.

Corollary 8 Let a_n, b_n, c_n be sequences in a ring R with identity and let $g : \mathbb{N} \times R \times R \rightarrow R$. If b_n is a unit for all n then the semilinear system

$$\begin{cases} x_{n+1} = a_n x_n + b_n y_n + c_n \\ y_{n+1} = g(n, x_n, y_n) \end{cases} \quad (16)$$

folds into the second-order difference equation

$$\begin{aligned} s_{n+2} &= \phi(n, s_n, s_{n+1}), \quad \text{where: } s_0 = x_0, \quad s_1 = a_0 x_0 + b_0 y_0 + c_0, \\ \phi(n, u, w) &= c_{n+1} + a_{n+1} w + b_{n+1} g(n, u, b_n^{-1}(w - a_n u - c_n)) \end{aligned} \quad (17)$$

For each solution $\{s_n\}$ of (17) the y -components of orbits of (16) are given by the passive equation

$$y_n = b_n^{-1}(s_{n+1} - a_n s_n - c_n).$$

For instance, consider the nonautonomous semilinear system

$$\begin{cases} x_{n+1} = (-1)^n x_n + y_n \\ y_{n+1} = \psi(x_n) + (-1)^n y_n \end{cases} \quad (18)$$

on a ring R with identity where $\psi : \mathbb{R} \rightarrow \mathbb{R}$ is an arbitrary function (see Section 5 below for how a system like (18) is derived). This system folds via (17) into the (autonomous) equation

$$s_{n+2} = \psi(s_n) - s_n \quad (19)$$

as is readily verified. For each initial point (x_0, y_0) in \mathbb{R}^2 the numbers $s_0 = x_0$ and $s_1 = x_0 + y_0$ generate a solution $\{s_n\}$ of the difference equation (19), which is similar to (6) with respect to the odd- and even-numbered terms of its solutions. The y -components of an orbit of the system (18) are then calculated using the passive equation

$$y_n = s_{n+1} - (-1)^n s_n.$$

3 Folding differential systems with two equations

Consider a planar system of two first-order differential equations of type

$$\begin{cases} dx/dt = f(t, x(t), y(t)) \\ dy/dt = g(t, x(t), y(t)) \end{cases} \quad t \in \mathbb{R} \quad (20)$$

where $f, g : \mathbb{R} \times D \rightarrow \mathbb{R}$ are given functions and $D \subset \mathbb{R} \times \mathbb{R}$. Although not considered here, the functions f, g may actually be defined on more general spaces where differentiation is defined.

The folding process for the system (20) of differential equations is analogous to that discussed in the previous section for difference systems. Suppose that f is semi-invertible, so by (8) there are sets M, M' and $h : \mathbb{R} \times M' \rightarrow \mathbb{R}$ such that

$$x'(t) = f(t, x(t), y(t)) \Rightarrow y(t) = h(t, x(t), x'(t)) \quad \text{for all } t. \quad (21)$$

Under suitable differentiability hypotheses, using the subscript notation for partial derivatives we obtain

$$\begin{aligned} x'' &= \frac{d}{dt} f(t, x, y) = f_t(t, x, y) + x' f_x(t, x, y) + y' f_y(t, x, y) \\ &= f_t(t, x, y) + x' f_x(t, x, y) + g(t, x, y) f_y(t, x, y) \end{aligned}$$

Now using (21),

$$x'' = f_t(t, x, h(t, x, x')) + x' f_x(t, x, h(t, x, x')) + g(t, x, h(t, x, x')) f_y(t, x, h(t, x, x')). \quad (22)$$

Let $\phi(t, x, x^{prime})$ denote the right hand side of (22). From an initial point $(x(t_0), y(t_0)) \in D$ for a given $t_0 \in \mathbb{R}$, the ordinary differential equation

$$s'' = \phi(t, s, s') \quad (23)$$

with initial values $s(t_0) = x(t_0)$, $s'(t_0) = f(t_0, x(t_0), y(t_0))$ generates the x-component of the flow $(x(t), y(t))$ of (20). The y-component is calculated from the passive equation

$$y(t) = h(t, x(t), x'(t)) \quad (24)$$

As in the discrete case, the pair of equations (23) and (24) constitute a folding of the differential system (20).

In cases where (22) is simpler than (20) a practical advantage is gained. To illustrate, consider the non-autonomous system

$$\begin{cases} x' = tx^2 - y \\ y' = ax + x^2 + 2t^2x^3 - 2txy \end{cases}, \quad a \in \mathbb{R} \quad (25)$$

(see Section 5 below for how a system like (25) is derived). The first equation above may be solved for y to give

$$y = tx^2 - x' = h(x, x'). \quad (26)$$

Now (22) (or direct differentiation with respect to t and substitution) yields

$$x'' = x^2 + 2txx' - (2t^2x^3 + x^2 + ax - 2txy) = x^2 + 2txx' - 2t^2x^3 - x^2 - ax + 2tx(tx^2 - x') = -ax$$

The linear autonomous equation $x'' + ax = 0$ is elementary and its solution (depending on the parameter value a) determines $x(t)$. Then $y(t)$ is calculated from the passive equation (26).

The role of a as a bifurcation parameter is not obvious from (25) directly. Further, the important issue of the *existence of solutions* for a system of differential equations is sometimes more easily addressed when (22) is a simple equation, as in the above case.

We close this section by pointing out the following relevant facts.

1. The above discussion may be generalized to real-valued functions f, g on Banach spaces with x and y both from \mathbb{R} into a given Banach space. Differentiability in Banach spaces is discussed in [8] so it is not necessary to derive the technical details. However, we limit our work on differential systems here to the real case.
2. The proper discrete analog for the differential system (20) is not (1) but rather, the difference system

$$\begin{cases} \Delta x_n = f(n, x_n, y_n) \\ \Delta y_n = g(n, x_n, y_n) \end{cases} \quad (27)$$

where $\Delta x_n = x_{n+1} - x_n$ and similarly for Δy_n . Although (27) can be cast in the form (1) by slightly modifying f, g it is useful to keep the distinction between (1) and (27) in mind when comparing difference and differential systems; see Remark 21 below.

4 Folding linear systems into equations

The procedure described in the preceding sections applies, in particular, to linear systems. Where there is no loss of clarity, we use the term “linear” loosely to include nonhomogeneous or affine equations. Both of the coordinate functions in linear systems are separable over the additive group of an underlying ring, though the requirement that one of the constituent functions be a bijection is a multiplicative issue. Thus semi-inversion requires the full ring structure.

4.1 Linear difference systems and equations

Consider the linear (nonhomogeneous, nonautonomous) system

$$\begin{cases} x_{n+1} = a_n x_n + b_n y_n + \alpha_n \\ y_{n+1} = c_n x_n + d_n y_n + \beta_n \end{cases} \quad (28)$$

where $a_n, b_n, c_n, d_n, \alpha_n, \beta_n$ are given sequences in a ring R with identity. Then shifting the index by 1 in the first equation above gives

$$\begin{aligned} x_{n+2} &= a_{n+1}x_{n+1} + b_{n+1}y_{n+1} + \alpha_{n+1} \\ &= a_{n+1}x_{n+1} + b_{n+1}c_nx_n + b_{n+1}d_ny_n + b_{n+1}\beta_n + \alpha_{n+1} \end{aligned} \quad (29)$$

If b_n is a *unit* in R for each n then f is semi-invertible and

$$w = f(n, u, v) = a_nu + b_nv + \alpha_n \Rightarrow v = b_n^{-1}(w - a_nu - \alpha_n).$$

Thus the y -component y_n is evaluated as

$$y_n = b_n^{-1}(x_{n+1} - a_nx_n - \alpha_n) = h(n, x_n, x_{n+1}). \quad (30)$$

Substituting this into (29) and rearranging terms yields

$$x_{n+2} = A_nx_{n+1} + B_nx_n + C_n \quad (31)$$

where

$$\begin{aligned} A_n &= a_{n+1} + b_{n+1}d_nb_n^{-1}, \\ B_n &= b_{n+1}(c_n - d_nb_n^{-1}a_n), \\ C_n &= b_{n+1}(\beta_n - d_nb_n^{-1}\alpha_n) + \alpha_{n+1}. \end{aligned}$$

Note that the sequences α_n, β_n do not appear as bound coefficients in (31) but within the free term C_n . If $\{x_n\}$ is a solution of this difference equation in R then y_n is passively calculated from (30).

It is clear that if c_n is a unit for every n (rather than b_n) then we may switch the roles of x and y in the preceding discussion.

If neither b_n nor c_n are units for infinitely many n then (30), or its analog for c_n , are not available so we proceed as follows: the second equation of (28) yields

$$b_{n+1}y_{n+1} = b_{n+1}c_nx_n + b_{n+1}d_ny_n + b_{n+1}\beta_n \quad (32)$$

which may be used in (29) to obtain

$$x_{n+2} = a_{n+1}x_{n+1} + b_{n+1}c_nx_n + b_{n+1}d_ny_n + b_{n+1}\beta_n + \alpha_{n+1} \quad (33)$$

Next, multiply the first equation in (28) by d_n to obtain

$$d_nx_{n+1} = d_na_nx_n + d_nb_ny_n + d_n\alpha_n$$

Finally, if b_n is central (commutes with all elements of R) for every n or if R is a commutative ring then the above equality yields

$$b_n d_n y_n = d_n x_{n+1} - d_n a_n x_n - d_n \alpha_n$$

and this together with (33) produces

$$\begin{aligned} b_n x_{n+2} &= b_n a_{n+1} x_{n+1} + b_n b_{n+1} c_n x_n + b_{n+1} d_n x_{n+1} - b_{n+1} d_n a_n x_n - b_{n+1} d_n \alpha_n + b_n b_{n+1} \beta_n + b_n \alpha_{n+1} \\ &= (b_n a_{n+1} + b_{n+1} d_n) x_{n+1} + b_{n+1} (b_n c_n - d_n a_n) x_n - b_{n+1} (d_n \alpha_n - b_n \beta_n) + b_n \alpha_{n+1} \end{aligned} \quad (34)$$

If $\{x_n\}$ is a solution of (34) then the y -components are obtained from the second equation of (28) as

$$y_{n+1} = d_n y_n + (c_n x_n + \beta_n) \quad (35)$$

Though not as easy to use as (30) because (35) is not a passive equation, it is a linear and first-order difference equation that is not difficult to solve.

If $b_n = b$ is a constant (albeit not a unit) then (34) takes a simpler form in which the term x_{n+2} is free. From the first equation of (28) and (29) we obtain

$$\begin{aligned} x_{n+2} &= a_{n+1} x_{n+1} + b y_{n+1} + \alpha_{n+1} \\ &= a_{n+1} x_{n+1} + b c_n x_n + d_n b y_n + b \beta_n + \alpha_{n+1} \\ &= a_{n+1} x_{n+1} + b c_n x_n + d_n (x_{n+1} - a_n x_n - \alpha_n) + b \beta_n + \alpha_{n+1} \\ &= (a_{n+1} + d_n) x_{n+1} - (d_n a_n - b c_n) x_n - d_n \alpha_n + b \beta_n + \alpha_{n+1} \end{aligned} \quad (36)$$

Once a solution $\{x_n\}$ of the above equation is obtained it is necessary to solve the first-order equation (35) for y_n . Unlike (34), Eq. (36) is recursive and its solutions may be obtained by iteration.

The next result summarizes the preceding discussion.

Proposition 9 Consider the linear difference system (28) over a commutative ring with identity.

(a) If b_n is a unit for all n then the linear difference equation (31) together with the passive equation (30) constitute a folding of (28).

(b) If b_n is not a unit for infinitely many n then (28) folds into the (non-recursive) linear difference equation (34) that yields the x -component of its orbits while its y -component is determined as a solution of (35).

(c) If $b_n = b$ is a constant for all n then (28) folds into the recursive linear equation (36) that yields the x -component of its orbits while its y -component is determined either (i) passively from (30) if b is a unit or (ii) as a solution of (35) if b is not a unit.

4.2 Linear difference systems with periodic foldings

We now consider linear systems that fold into equations with periodic coefficients. *We do not assume that the parameters of the linear system are periodic*; it is only necessary that the derived second-order equation has periodic coefficients so the results in this section apply to many linear systems with non-periodic parameters as well. We consider only difference systems in this section. Differential systems that fold into a second-order ordinary differential equation with periodic coefficients may be studied with the aid of the continuous-time Floquet theory [6].

Suppose that the coefficients a_n, b_n, c_n, d_n in the system (28) are all sequences where b_n is a unit for each n (to limit the range of possible cases) and the coefficients A_n, B_n in (31) are periodic with period p i.e., $A_{n+p} = A_n$, $B_{n+p} = B_n$. In particular this is true if all the coefficients in the system are periodic with period p (not necessarily prime or minimal). However, non-periodic systems may also fold into periodic equations of type (31). In fact, if A_n, B_n are sequences with period p and b_n, d_n are arbitrary sequences then defining

$$a_n = A_{n-1} - b_n d_{n-1} b_{n-1}^{-1}, \quad c_n = b_{n+1}^{-1} B_n + d_n b_n^{-1} a_n$$

ensures that the homogeneous part of (31) has periodic coefficients. Other combinations of system parameters that yield periodic A_n and B_n are possible.

To study the solutions of (31) with periodic coefficients, one possible approach is to unfold it back to a system and then use the Floquet theory adapted to the discrete case; see, e.g., [7], [17]. Alternatively, we may use the eigensequence method in [14] which applies directly to (31) without the need for unfolding it and further, it works whether C_n is periodic or not. We need only find an eigensequence for the homogeneous part of (31) in the ring R (in particular, an eigensequence with period p). Any eigensequence, periodic or not, yields a semiconjugate factorization [13] of the second-order equation into a pair of first-order ones.

An eigensequence of period p exists in R if the characteristic equation of (31), i.e., the first-order quadratic difference equation

$$r_{n+1} r_n = A_{n-1} r_n + B_{n-1} \tag{37}$$

has a solution of period p in the ring R for some initial value $r_1 \in R$. The following result is from [14].

Theorem 10 *Let R be a ring with identity 1 and for $j = 1, 2, \dots, p$, let α_j, β_j be obtained by iteration from (31) subject to the initial values*

$$\alpha_0 = 0, \alpha_1 = 1; \quad \beta_0 = 1, \beta_1 = 0.$$

If a root r_1 of the quadratic polynomial

$$r \alpha_p r + r \beta_p - \alpha_{p+1} r - \beta_{p+1} = 0 \tag{38}$$

is a unit in R and the recurrence

$$r_{j+1} = A_{j-1} + B_{j-1}r_j^{-1} \quad (39)$$

also generates units r_2, \dots, r_p in R then $\{r_n\}_{n=1}^{\infty}$ is a unitary eigensequence of (31) with period p that yields the triangular system of first-order equations (a semiconjugate factorization)

$$\begin{aligned} t_{n+1} &= C_{n-1} - B_{n-1}r_n^{-1}t_n, & t_1 &= x_1 - r_1x_0 \\ x_{n+1} &= r_{n+1}x_n + t_{n+1}. \end{aligned}$$

The polynomial in (38) simplifies further if the coefficients A_j, B_j are in the center of R . Then α_j, β_j are also in the center of R so (38) reduces to

$$\alpha_p r^2 + (\beta_p - \alpha_{p+1})r - \beta_{p+1} = 0. \quad (40)$$

If $A_n = A$ and $B_n = B$ are constants then $p = 1$ and the quadratic polynomial (40) further reduces to $r^2 - Ar - B = 0$. This is recognizable as the characteristic polynomial of the autonomous linear equation of order 2.

For illustration suppose that the parameters of (28) are real with $b_n \neq 0$ for all n and the following equalities hold

$$\begin{aligned} a_{n+1}b_n + b_{n+1}d_n &= 2b_n \cos \frac{2\pi(n+1)}{3} \\ b_n c_n - a_n d_n &= \frac{b_n}{b_{n+1}} \\ \alpha_n d_n - \beta_n b_n &= \frac{\alpha_{n+1}b_n}{b_{n+1}}. \end{aligned}$$

These equalities imply that

$$A_n = 2 \cos \frac{2\pi(n+1)}{3}, \quad B_n = 1, \quad C_n = 0$$

In this case, (28) folds into the following:

$$x_{n+1} = 2 \cos \frac{2\pi n}{3} x_n + x_{n-1} \quad (41)$$

which has a coefficient of period 3. The numbers α_j, β_j are readily calculated as

$$\alpha_2 = -1, \alpha_3 = 2, \alpha_4 = 3, \beta_2 = 1, \beta_3 = -1, \beta_4 = -1.$$

The quadratic equation (40) $2r^2 - 4r + 1 = 0$ in this case has two roots $(2 \pm \sqrt{2})/2$. Let $r_1 = (2 - \sqrt{2})/2$ and use (39) to calculate $r_2 = 1 + \sqrt{2}$, $r_3 = -2 + \sqrt{2}$. Since these are units in \mathbb{R} ,

by Theorem 10 a unitary eigensequence with period 3 is obtained. The semiconjugate factorization of (41) is readily calculated and the solution of the factor equation is found to be

$$t_{3j+1} = \rho^j t_1, \quad t_{3j+2} = -\frac{\rho^j t_1}{r_1}, \quad t_{3j+3} = \frac{\rho^j t_1}{r_1 r_2}, \quad j \geq 0, \quad t_1 = x_1 - r_1 x_0, \quad \rho = -1/(r_1 r_2 r_3).$$

Since $\rho = 1 + \sqrt{2} > 1$ it follows that all solutions of (41) with $t_1 \neq 0$ are unbounded. However, for initial values satisfying $x_1 = r_1 x_0$ we have $t_1 = 0$; so $t_n = 0$ for all n . When inserted in the cofactor equation $x_{n+1} = r_{n+1} x_n + t_{n+1}$ this yields

$$x_{3n} = \frac{(-1)^n x_0}{\rho^n}, \quad x_{3n+1} = \frac{(-1)^n x_0 r_1}{\rho^n}, \quad x_{3n+2} = \frac{(-1)^n x_0 r_1 r_2}{\rho^n}, \quad n \geq 1. \quad (42)$$

These special solutions of (41) converge to 0 exponentially for all x_0 . If $\{x_n\}$ is any solution of (41) then the y-component of the corresponding orbit $\{(x_n, y_n)\}$ of the system is given by (30). For the special solutions (42), if the coefficient a_n is bounded then $y_n \approx -\alpha_n/b_n$ for large n .

4.3 Linear differential systems and equations

Many of the results stated above for difference systems and equations have differential analogs. The essentials are as follows: Consider the system of linear differential equations

$$\begin{cases} x'(t) = a(t)x(t) + b(t)y(t) + \alpha(t) \\ y'(t) = c(t)x(t) + d(t)y(t) + \beta(t) \end{cases} \quad (43)$$

with differentiable functions $a(t), b(t), c(t), d(t), \alpha(t), \beta(t)$ defined on \mathbb{R} or some open subset of it.

If $b(t) \neq 0$ for all t then $f(t, x, y) = a(t)x(t) + b(t)y(t) + \alpha(t)$ is a semi-invertible function that yields

$$y(t) = \frac{1}{b(t)}[x'(t) - a(t)x(t) - \alpha(t)] = h(t, x, x'). \quad (44)$$

Further, using (22) or by straightforward calculation under suitable differentiability hypotheses,

$$x'' = \left(a + d + \frac{b'}{b}\right)x' + \left(bc - ad + a' - \frac{b'a}{b}\right)x - d\alpha + b\beta - \frac{b'\alpha}{b} + \alpha' \quad (45)$$

where for brevity we have omitted explicit mention of the variable t . If $x(t)$ is a solution of (45) then the x-component of an orbit of (43) is determined and the y-component is readily found using (44).

If $b(t) = 0$ for some t but $c(t) \neq 0$ for all t then the above calculations may be repeated with the roles of x and y switched. If both $b(t)$ and $c(t)$ vanish for some values of t then (44) is not applicable and procedures discussed for difference equations can be implemented here too. These results yield a second-order differential equation similar to (45) that gives the x-component of an

orbit of the differential system (43); the y-component is then obtained using the second equation in (43).

If $b(t) = b$ is a constant then $b'(t) = 0$ and (45) reduces to

$$x'' = [a(t) + d(t)]x' + [bc(t) - a(t)d(t) + a'(t)]x - d(t)\alpha(t) + b\beta(t) + \alpha'(t) \quad (46)$$

which is analogous to (36) in the discrete case. In particular, if $a(t) = a$ is a constant then the trace and the determinant of the coefficients matrix of (43) can be identified in the coefficients of (46). The following summarizes the above results for differential systems.

Proposition 11 *Consider the linear differential system (43) in \mathbb{R}^2 .*

(a) *If $b(t) \neq 0$ for all t then the linear differential equation (45) together with the passive equation (44) constitute a folding of (43).*

(b) *If $b(t) = b$ is a constant for all t then (43) folds into the linear differential equation (46) which yields the x-component of its orbits while its y-component is determined either (i) passively from (44) if $b \neq 0$ or (ii) as a solution of the second equation of (43) if $b = 0$.*

5 An inverse problem

Folding a given nonlinear system into a higher order equation does not necessarily simplify the study of solutions. From a practical point of view, a significant gain in terms of simplifying the analysis of solutions is desirable. To address this issue *systematically* we chart a course backward, from a higher order equation with a desirable property to the system that yields it through the folding procedure.

In this section we determine and study classes of systems that fold into difference or differential equations of order 2 with known properties. We assume that one of the two equations of the system, say, the one specified by f , is given along with a known function ϕ that defines a second-order equation. Then a function g is determined with the property that the system with components f and g folds into an equation of order 2 defined by ϕ .

This inverse process in particular yields a (non-standard) unfolding of ϕ that is based on the given component f . In the special case $f(t, u, v) = v$ we obtain $g = \phi$ so the corresponding system is just a familiar, standard unfolding of the equation defined by ϕ .

5.1 Difference equations

Suppose that a function f satisfies condition (8). By (12) the following

$$f(n+1, w, g(n, u, h(n, u, w))) = \phi(n, u, w)$$

is a function of n, u, w . Since f is semi-invertible, once again using (8) we obtain

$$g(n, u, h(n, u, w)) = h(n+1, w, \phi(n, u, w)) \quad (47)$$

Now, suppose that $\phi(n, u, w)$ is prescribed on a set $\mathbb{N} \times M'$ where $M' \subset S \times S$ and we seek g that satisfies (47). Assume that a subset M of D exists with the property that $f(\mathbb{N} \times M) \times \phi(\mathbb{N} \times M') \subset M'$. For $(n, u, v) \in \mathbb{N} \times M$ define

$$g(n, u, v) = h(n + 1, f(n, u, v), \phi(n, u, f(n, u, v))) \quad (48)$$

In particular, if $v \in h(\mathbb{N} \times M')$ then g above satisfies (48). These observations establish the following result.

Theorem 12 *Let f be a semi-invertible function with h given by (8). Further, let ϕ be a given function on $\mathbb{N} \times M'$. If g is given by (48) then (1) folds into the difference equation $s_{n+2} = \phi(n, s_n, s_{n+1})$ plus a passive equation.*

As a check, consider $f(u, v) = uv$ and $\phi(u, w) = u(a + bu)$, both independent of n . Then $h(u, w) = w/u$ and (48) gives

$$g(u, v) = \frac{u(a + bu)}{uv} = \frac{a + bu}{v}$$

This g yields (4), as expected. The function f in this example is separable. In separable cases, explicit expressions are possible with the aid of (9). Note that semilinear systems are included in the next result.

Corollary 13 *Let $(G, *)$ be a nontrivial group and $f(n, u, v) = f_1(n, u) * f_2(n, v)$ be separable on $G \times G$ with f_2 a bijection. If ϕ is a given function on $\mathbb{N} \times G \times G$ and g is given by*

$$g(n, u, v) = f_2^{-1}(n + 1, [f_1(n + 1, f_1(n, u) * f_2(n, v))]^{-1} * \phi(n, u, f_1(n, u) * f_2(n, v)))$$

then (1) folds into the difference equation $s_{n+2} = \phi(n, s_n, s_{n+1})$ plus a passive equation.

The next result yields a class of systems that reduce (effectively) to first-order difference equations.

Corollary 14 *Assume that f, h satisfy the hypotheses of Theorem 12 and let $\phi(n, \cdot)$ be a function of one variable for each n . If*

$$g(n, u, v) = h(n + 1, f(n, u, v), \phi(n, u)) \quad (49)$$

then (1) folds into the difference equation $s_{n+2} = \phi(n, s_n)$ whose even terms and odd terms are solutions of the first-order equation

$$r_{n+1} = \phi(n, r_n).$$

For example, if $f(n, u, v) = (-1)^n u + v$ and $\phi(s)$ is a given function (independent of n) then $h(n, u, w) = w - (-1)^n u$ and (49) yields

$$g(n, u, v) = \phi(u) - (-1)^{n+1}[(-1)^n u + v] = \phi(u) + u + (-1)^n v$$

Thus the non-autonomous semilinear system

$$\begin{cases} x_{n+1} = (-1)^n x_n + y_n \\ y_{n+1} = \phi(x_n) + x_n + (-1)^n y_n \end{cases}$$

folds in the sense of Corollary (14) into the autonomous, first-order difference equation $r_{n+1} = \phi(r_n)$ plus a passive equation. Note that the above system is the same as (18) with $\psi(u) = \phi(u) + u$.

The next result yields a class of systems that actually reduce to first-order difference equations.

Corollary 15 *Assume that f, h satisfy the hypotheses of Theorem 12 and let $\phi(n, \cdot)$ be a function of one variable for each n . If*

$$g(n, u, v) = h(n + 1, f(n, u, v), \phi(n, f(n, u, v)))$$

then (1) folds into the difference equation $s_{n+2} = \phi(n, s_{n+1})$ with order 1 plus a passive equation.

For example, let $f(u, v) = u/v$ so that $h(u, w) = u/w$. If $\phi(s) = as + b$ then define

$$g(u, v) = h(f(u, v), af(u, v) + b) = \frac{u/v}{au/v + b} = \frac{u}{au + bv}.$$

Then the rational system

$$\begin{cases} x_{n+1} = x_n/y_n \\ y_{n+1} = x_n/(ax_n + by_n) \end{cases}$$

folds into the first-order linear equation $s_{n+2} = as_{n+1} + b$ plus the passive equation $y_n = s_n/s_{n+1}$ with $s_0 = x_0$, $s_1 = x_0/y_0$.

The next result concerns systems that fold into *autonomous* linear difference equations of order 2.

Corollary 16 *Assume that f, h satisfy the hypotheses of Theorem 12 and let $\phi(u, w) = \alpha u + \beta w$ be a linear function. If*

$$g(n, u, v) = h(n + 1, f(n, u, v), \alpha u + \beta f(n, u, v))$$

then (1) folds into the linear autonomous difference equation $s_{n+2} = \alpha s_n + \beta s_{n+1}$ plus a passive equation.

The procedure described in this section generates a system that may be considered an unfolding (non-standard) of the equation $s_{n+2} = \phi(n, s_n, s_{n+1})$. Indeed, if $f(n, u, v) = v$ then $h(n, u, w) = w$ and by (48) $g(n, u, v) = \phi(n, u, v)$, i.e., $g = \phi$ so the resulting system is the standard unfolding (3). In general, for each fixed ϕ there exists a distinct unfolding of $s_{n+2} = \phi(n, s_n, s_{n+1})$ for every semi-invertible function f .

5.2 Differential equations

The inverse problem above has a differential analog. Assume that a function f is semi-invertible with h given by (21). Then the right hand side of (22) may be written as follows as a function of t, u, w

$$f_t(t, u, h(t, u, w)) + f(t, u, h(t, u, w))f_u(t, u, h(t, u, w)) + g(t, u, h(t, u, w))f_v(t, u, h(t, u, w)) = \phi(t, u, w) \quad (50)$$

Now suppose, inversely, that the function $\phi(t, u, w)$ is prescribed and we wish to determine a function g such that (50) is satisfied. Define

$$g(t, u, v) = \frac{1}{f_v(t, u, v)} [\phi(t, u, f(t, u, v)) - f(t, u, v)f_u(t, u, v) - f_t(t, u, v)] \quad (51)$$

provided that $f_v(t, u, v) \neq 0$.

Note that unlike (48), using the chain rule made a second application of semi-invertibility unnecessary in deriving (51). These observations establish the following result.

Theorem 17 *Let f be a semi-invertible function and its semi-inversion h be given by (21) and let ϕ be a given function. If $f_v(t, u, v) \neq 0$ for all t, u, v and g is given by (51) then system (20) folds into the following second-order ordinary differential equation (plus a passive equation)*

$$s'' = \phi(t, s, s')$$

To illustrate, consider $f(u, v) = tu^2 - v$ as in (25) and let $\phi(u, w) = \alpha u$. Then $f_t(u, v) = u^2$, $f_u(u, v) = 2tu$ and $f_v(u, v) = -1$ so from (51)

$$g(u, v) = -[\alpha u - 2tu(tu^2 - v) - u^2] = -\alpha u + u^2 + 2t^2u^3 - 2tuv$$

If $a = -\alpha$ then $g(u, v) = au + u^2 + 2t^2u^3 - 2tuv$ and system (25) is obtained.

Theorem 17 can be used to categorize classes of differential systems that fold into second-order, nonlinear differential equations with known properties. For example, we may determine systems that fold into the equations of Lienard, van der Pol or Duffing; see, e.g., [3] or [16]. In particular, the periodically forced, special case of Duffing's equation

$$x'' + bx' + kx^3 = A \sin \omega t$$

which generates chaotic flows has a different (non-standard) unfolding for every choice of $f(t, u, v)$ with $f_v \neq 0$. For instance, if $f(t, u, v) = \alpha u + \beta v$ for constants α, β with $\beta \neq 0$ then with $\phi(t, u, w) = A \sin \omega t - ku^3 - bw$, (51) yields

$$\begin{aligned} g(t, u, v) &= \frac{1}{\beta} [A \sin \omega t - ku^3 - b(\alpha u + \beta v) - \alpha(\alpha u + \beta v)] \\ &= \frac{A}{\beta} \sin \omega t - \frac{k}{\beta} u^3 - \frac{\alpha}{\beta} (\alpha + b)u - (\alpha + b)v \end{aligned}$$

In particular, if we set $\alpha = -b$ and $\beta = 1$ then the system

$$\begin{cases} x' = -bx + y \\ y' = -kx^3 + A \sin \omega t \end{cases}$$

folds into Duffing's equation plus the passive equation $y = x' + bx$; other variations are clearly possible.

Remark 18 *Analogs of Corollaries (14), (15) and (16) for differential systems are also true but we do not list them explicitly.*

6 Folding larger systems into equations

Folding a system of k equations into an equation of order $k \geq 3$ using the method of the preceding sections is also possible if suitable inversions exist. The process of folding a system of k equations may in principle be viewed as iterating what was discussed above. The larger the value of k , the longer and more complex the k -th order equation needs to be in order to include all the information that is contained in the system.

Linear systems illustrate this situation: while a system of 2 linear equations is defined by 4 parameters (the coefficient matrix has 4 entries) a system of 3 linear equations requires 9 parameters. The increased amount of information, both algebraically (dealing with a 3×3 matrix) and analytically (orbits exist in 3 dimensions rather than 2) must be configured within a single equation. For *nonlinear* equations potential technical difficulties related to inversions need to be considered.

In certain cases where the higher order equation that is derived from the system is much simpler than the system itself, folding yields a practical advantage. We discuss examples of such systems in this section, including a special type of difference system with k equations that readily folds into an equation of order k without requiring any type of inversion.

6.1 Folding systems of 3 equations

In this section, we discuss folding systems of 3 difference or differential equations into higher order equations. Let S be a nonempty set and let $f_1, f_2, f_3 : \mathbb{N} \times D \rightarrow S$ be given functions where $D \subset S \times S \times S$. Consider the system

$$\begin{cases} x_{1,n+1} = f_1(n, x_{1,n}, x_{2,n}, x_{3,n}) \\ x_{2,n+1} = f_2(n, x_{1,n}, x_{2,n}, x_{3,n}) \\ x_{3,n+1} = f_3(n, x_{1,n}, x_{2,n}, x_{3,n}) \end{cases} \quad (52)$$

Starting from an initial point $(x_{1,0}, x_{2,0}, x_{3,0}) \in D$, iteration generates points $(x_{1,n}, x_{2,n}, x_{3,n})$ of an orbit of (52) for as long as the orbit stays within the domain D of the system. Assume that for every n the following *partial inversion* holds

$$r = f_1(n, u, v, w) \Rightarrow w = \phi_1(n, u, v, r) \quad (53)$$

Then the first equation of (52) yields

$$x_{3,n} = \phi_1(n, x_{1,n}, x_{2,n}, x_{1,n+1}), \quad n = 1, 2, \dots$$

Now

$$\begin{aligned} x_{1,n+2} &= f_1(n+1, x_{1,n+1}, x_{2,n+1}, x_{3,n+1}) \\ &= f_1(n+1, x_{1,n+1}, f_2(n, x_{1,n}, x_{2,n}, x_{3,n}), f_3(n, x_{1,n}, x_{2,n}, x_{3,n})) \\ &= f_1(n+1, x_{1,n+1}, f_2(n, x_{1,n}, x_{2,n}, \phi_1(n, x_{1,n}, x_{2,n}, x_{1,n+1})), f_3(n, x_{1,n}, x_{2,n}, \phi_1(n, x_{1,n}, x_{2,n}, x_{1,n+1}))) \end{aligned}$$

The last expression does not involve the variable $x_{3,n}$ so we use it to define a new function

$$f_1^{(1)}(n, u, v, r) = f_1(n+1, r, f_2(n, u, v, \phi_1(n, u, v, r)), f_3(n, u, v, \phi_1(n, u, v, r)))$$

Thus, $x_{1,n+2} = f_1^{(1)}(n, x_{1,n}, x_{2,n}, x_{1,n+1})$. Next, assume that \bar{f} has the following partial inversion

$$s = f_1^{(1)}(n, u, v, r) \Rightarrow v = \phi_2(n, u, r, s) \quad (54)$$

Then $x_{2,n} = \phi_2(n, x_{1,n}, x_{1,n+1}, x_{1,n+2})$ and the following is obtained:

$$x_{1,n+3} = f_1^{(1)}(n+1, x_{1,n+1}, x_{2,n+1}, x_{1,n+2}) = f_1^{(2)}(n, x_{1,n}, x_{1,n+1}, x_{1,n+2}), \quad (55)$$

where

$$f_1^{(2)}(n, u, r, s) = f_1^{(1)}(n+1, r, f_2(n, u, \phi_2(n, u, r, s), \phi_1(n, u, \phi_2(n, u, r, s), r)), s).$$

If $\{x_{1,n}\}$ is a solution of (55) with initial values

$$\begin{aligned} x_{1,0}, x_{1,1} &= f_1(0, x_{1,0}, x_{2,0}, x_{3,0}), \quad x_{1,2} = f_1(1, x_{1,1}, x_{2,1}, x_{3,1}) \\ \text{where } x_{2,1} &= f_2(0, x_{1,0}, x_{2,0}, x_{3,0}), \quad x_{3,1} = f_3(0, x_{1,0}, x_{2,0}, x_{3,0}) \end{aligned}$$

then the second and third components of an orbit $\{(x_{1,n}, x_{2,n}, x_{3,n})\}$ are obtained passively via the partial inversion equations:

$$x_{2,n} = \phi_2(n, x_{1,n}, x_{1,n+1}, x_{1,n+2}), \quad x_{3,n} = \phi_1(n, x_{1,n}, x_{2,n}, x_{1,n+1}) \quad (56)$$

without needing to solve additional difference equations.

As in Definition 6, we say that the third-order equation (55) together with the two passive equations (56) constitute a folding of (52).

To illustrate the above process, we apply it to the system

$$\begin{cases} x_{1,n+1} = ax_{2,n} + x_{3,n} \\ x_{2,n+1} = bx_{1,n} + cx_{3,n} \\ x_{3,n+1} = \rho(x_{1,n}) + dx_{2,n} \end{cases} \quad (57)$$

where $a, b, c, d \in \mathbb{R}$ and $\rho : \mathbb{R} \rightarrow \mathbb{R}$ is a given function. From the first equation of (57),

$$x_{3,n} = x_{1,n+1} - ax_{2,n} \quad (58)$$

This is the function ϕ_1 in the above algorithm. Next, shifting the index by 1 in the first equation of (57) and using substitutions

$$x_{1,n+2} = ax_{2,n+1} + x_{3,n+1} = abx_{1,n} + acx_{3,n} + x_{3,n+1} = acx_{1,n+1} + abx_{1,n} + \rho(x_{1,n}) + (d - a^2c)x_{2,n} \quad (59)$$

This equation is easily solved for $x_{2,n}$ to obtain the function ϕ_2 in the above algorithm. Next, shift the index once more by 1 and use substitutions to obtain

$$\begin{aligned} x_{1,n+3} &= ax_{2,n+2} + x_{3,n+2} = abx_{1,n+1} + acx_{3,n+1} + \rho(x_{1,n+1}) + dx_{2,n+1} \\ &= abx_{1,n+1} + \rho(x_{1,n+1}) + ac\rho(x_{1,n}) + acdx_{2,n} + d(bx_{1,n} + cx_{3,n}) \\ &= abx_{1,n+1} + \rho(x_{1,n+1}) + ac\rho(x_{1,n}) + acdx_{2,n} + bdx_{1,n} + cd(x_{n,n+1} - ax_{2,n}) \\ &= (ab + cd)x_{1,n+1} + \rho(x_{1,n+1}) + bdx_{1,n} + ac\rho(x_{1,n}) \end{aligned} \quad (60)$$

Since the terms containing $x_{2,n}$ cancel out in the substitution process, it is not necessary to substitute ϕ_2 for $x_{2,n}$ in this case to obtain (60). But deriving ϕ_2 is still necessary to calculate the orbits of (57) once a solution $\{x_{1,n}\}$ of (60) is determined. We distinguish two cases: If $d \neq a^2c$ then the passive equation

$$x_{2,n} = \frac{1}{d - a^2c}(x_{1,n+2} - acx_{1,n+1} - abx_{1,n} - \rho(x_{1,n}))$$

yields the component $x_{2,n}$ and then (58) gives the third component $x_{3,n}$. If $d = a^2c$ then the second-order difference equation (59) reduces to

$$x_{1,n+2} = acx_{1,n+1} + abx_{1,n} + \rho(x_{1,n}) \quad (61)$$

indicating that (57) is not strictly 3-dimensional in this case. The second component $x_{2,n}$ may be found from the system by eliminating $x_{3,n}$ from the first two equations in (57) to obtain the difference equation

$$x_{2,n+1} = bx_{1,n} + cx_{3,n} = -acx_{2,n} + bx_{1,n} + cx_{1,n+1}$$

With the sequence $\{x_{1,n}\}$ being given as a solution of (61), the above equation is a linear, nonhomogeneous first-order equation that yields the component $x_{2,n}$ as a solution. Then (58) yields $x_{3,n}$. Alternatively, we may find $x_{3,n}$ first by eliminating $x_{2,n}$ from the first and the third equations in (57) to obtain a linear difference equation that yields $x_{3,n}$ as a solution. Then $x_{2,n}$ maybe found using any equation of the system.

In principle, the algorithm described above can be extended to systems containing any number k of equations; we discuss a general folding algorithm for difference systems in the next section.

Systems of 3 differential equations also fold into ordinary differential equations using the analog of the above algorithm in which higher order derivatives replace index shifts. We do not enter the details for the differential case here but illustrate the process using the following analog of system (57)

$$\begin{cases} x_1'(t) = ax_2(t) + x_3(t) \\ x_2'(t) = bx_1(t) + cx_3(t) \\ x_3'(t) = \rho(x_1(t)) + dx_2(t) \end{cases} \quad (62)$$

The first equation of (62) yields

$$x_3(t) = x_1'(t) - ax_2(t).$$

Now taking the derivative of the first equation of (62) and using substitutions

$$\begin{aligned} x_1''(t) &= ax_2'(t) + x_3'(t) = abx_1(t) + acx_3(t) + \rho(x_1(t)) + dx_2(t) \\ &= abx_1(t) + \rho(x_1(t)) + acx_1'(t) + (d - a^2c)x_2(t) \end{aligned}$$

The above equation may be solved for $x_2(t)$ if $d \neq a^2c$. Next, take derivatives once more and use substitutions to obtain

$$\begin{aligned} x_1'''(t) &= ax_2''(t) + x_3''(t) = abx_1'(t) + acx_3'(t) + \rho'(x_1(t))x_1'(t) + dx_2'(t) \\ &= abx_1'(t) + \rho'(x_1(t))x_1'(t) + ac\rho(x_1(t)) + acdx_2(t) + dbx_1(t) + dcx_3(t) \\ &= abx_1'(t) + \rho'(x_1(t))x_1'(t) + ac\rho(x_1(t)) + acdx_2(t) + bdx_1(t) + dcx_1'(t) - dcax_2(t) \\ &= (\rho'(x_1(t)) + ab + cd)x_1'(t) + ac\rho(x_1(t)) + bdx_1(t) \end{aligned}$$

The last ordinary differential equation above is what system (62) folds into. If $x_1(t)$ is a solution of this third-order differential equation then $x_2(t)$ and $x_3(t)$ are determined in a passive way that is analogous to the determination of $x_{2,n}$ and $x_{3,n}$ for the difference system (57).

Of interest is a comparison (in the autonomous case) of the above algorithmic or iterative method with the method that is discussed in [4]. The latter method applies to a general class of differential systems of three equations and yields a jerk function provided certain conditions hold. Jacobian inversion (see [1]) is used (possibly over-used) to derive a type of system that is amenable to being folded in such a way that the flows of the system are represented by the scalar equation. A system of the following type is derived

$$\begin{cases} x_1'(t) = \eta_1(x_1(t)) + b_{11}x_2(t) + b_{13}x_3(t) + c_1 \\ x_2'(t) = \eta_2(x_1(t), x_2(t), x_3(t)) + b_{21}x_1(t) + b_{22}x_2(t) + b_{23}x_3(t) + c_2 \\ x_3'(t) = \eta_3(x_1(t), x_2(t), x_3(t)) + b_{31}x_1(t) + b_{32}x_2(t) + b_{33}x_3(t) + c_3 \end{cases} \quad (63)$$

where constants b_{ij} , c_i and functions η_i are arbitrary except for the following conditions

$$b_{12}\eta_2(x_1(t), x_2(t), x_3(t)) + b_{13}\eta_2(x_1(t), x_2(t), x_3(t)) = \xi(x_1(t), b_{12}x_2(t) + b_{13}x_3(t)) \quad (64)$$

$$b_{12}^2b_{23} - b_{13}^2b_{32} + b_{12}b_{13}(b_{33} - b_{22}) \neq 0 \quad (65)$$

with ξ an arbitrary function. Condition (65) is needed to ensure that (63) folds to a third-order (jerk) equation and not to a second-order one; see [4]. Condition (64) ensures the elimination of variables x_2 and x_3 in a relatively general setting.

The linear nature of the first equation in the variables x_2 and x_3 in (63) is due to the derivation in [4]; it makes partial inversion and solving globally for x_3 (or x_2) easy. However, this restriction is not strictly necessary; for example, a separable form with invertible functions would also work in the algorithmic case as in the following system

$$\begin{cases} x'_1(t) = \eta(x_1(t)) + \zeta(x_3(t)) \\ x'_2(t) = g(x_1(t), x_2(t), x_3(t)) \\ x'_3(t) = \rho(x_1(t)) + ax_2(t) \end{cases} \quad (66)$$

where $a \neq 0$, η, g, ρ are arbitrary functions and ζ is a bijection of \mathbb{R} . If these maps are nonlinear then the above system is not of type (63) but may be folded using the folding algorithm. The first equation of (66) yields $x_3(t) = \zeta^{-1}(x'_1(t) - \eta(x_1(t)))$ so

$$x''_1(t) = \eta'(x_1(t))x'_1(t) + \zeta'(x_3(t))x'_3(t) = \eta'(x_1(t))x'_1(t) + \zeta'(\zeta^{-1}(x'_1(t) - \eta(x_1(t))))[\rho(x_1(t)) + ax_2(t)]$$

The above equation is readily solved for $x_2(t)$ in terms of the variables $x_1(t), x'_1(t)$ and $x''_1(t)$ so $x'''_1(t)$ can then be also expressed in terms of $x_1(t), x'_1(t)$ and $x''_1(t)$. These observations suggest that the algorithmic approach is more general, if not the most succinct.

6.2 A folding algorithm for difference systems

In this section we extend the folding process for a system of 3 equations discussed the preceding section to a general folding algorithm for systems of k difference equations. The process is similar, in essence, for differential systems but we leave the differential case for future considerations. In each step of the iterative process a “space” variable $x_{j,n}$ is converted to a “temporal extension” of $x_{1,n}$. Iteration stops when the $k - 1$ space variables $x_{2,n}, \dots, x_{k,n}$ have been converted to the temporal extensions $x_{1,n+1}, \dots, x_{1,n+k}$ (or possibly earlier). This algorithm arbitrarily centers on the variable $x_{1,n}$ for convenience, although in principle any one of $x_{2,n}, \dots, x_{k,n}$ can be used instead if it simplifies calculations.

We begin with the recursive system

$$x_{i,n+1} = f_i(n, x_{1,n}, x_{2,n}, \dots, x_{k,n}), \quad i = 1, 2, \dots, k \quad (67)$$

of k first-order difference equations in k variables $x_{1,n}, x_{2,n}, \dots, x_{k,n}$.

1. *Shift* the indices of the first equation ($i = 1$) in (67) by 1 to obtain

$$x_{1,n+2} = f_1(n + 1, x_{1,n+1}, x_{2,n+1}, \dots, x_{k,n+1})$$

(a) Use (67) to *substitute*

$$x_{j,n+1} = f_j(n, x_{1,n}, x_{2,n}, \dots, x_{k,n}), \quad j = 2, \dots, k.$$

(b) Assume that f_1 has the following *partial inversion*

$$s_1 = f_1(n, u_1, u_2, \dots, u_k) \Rightarrow u_k = \phi_1(n, s_1, u_1, u_2, \dots, u_{k-1})$$

(c) *Substitute* $\phi_1(n, s_1, u_1, \dots, u_{k-1})$ for u_k in $f_j(n, u_1, u_2, \dots, u_k)$ for $j \geq 2$ to define

$$\begin{aligned} f_1^{(1)}(n, s_1, u_1, \dots, u_{k-1}) &= f_1(n+1, s_1, u_2^{(1)}, \dots, u_k^{(1)}) \\ \text{where: } u_j^{(1)} &= f_j(n, u_1, \dots, u_{k-1}, \phi_1(n, s_1, u_1, \dots, u_{k-1})) \end{aligned}$$

Since $u_j^{(1)}$ stands for $x_{j,n+1}$ for $j \geq 2$ we may write

$$x_{1,n+2} = f_1^{(1)}(n, x_{1,n+1}, x_{2,n}, \dots, x_{k-1,n}). \quad (68)$$

Notice that the variable $x_{k,n}$ has been replaced with the new variable $x_{1,n+1}$.

2. *Shift* the indices in (68) by 1 to obtain

$$x_{1,n+3} = f_1^{(1)}(n+1, x_{1,n+2}, x_{2,n+1}, \dots, x_{k-1,n+1})$$

(a) Use (67) to *substitute* $f_j(n, x_{1,n}, x_{2,n}, \dots, x_{k,n})$ for $x_{j,n+1}$ for $j = 2, \dots, k-1$, as in Step 1a above.

(b) Assume that $f_1^{(1)}$ has the following *partial inversion*

$$s_2 = f_1^{(1)}(n, s_1, u_1, \dots, u_{k-1}) \Rightarrow u_{k-1} = \phi_2(n, s_2, s_1, u_1, \dots, u_{k-2})$$

(c) *Substitute* $\phi_1(n, s_1, u_1, \dots, u_{k-1})$ for u_k in $f_j(n, u_1, \dots, u_{k-1}, u_k)$ and then *substitute* the quantity $\phi_2(n, s_2, s_1, u_1, \dots, u_{k-2})$ for u_{k-1} in both of the preceding quantities to obtain

$$f_1^{(2)}(n, s_2, s_1, u_1, \dots, u_{k-2}) = f_1^{(1)}(n+1, s_2, u_2^{(2)}, \dots, u_{k-1}^{(2)})$$

where, abbreviating $\phi_2(n, s_2, s_1, u_1, \dots, u_{k-2})$ by ϕ_2 to save space, we have

$$u_j^{(2)} = f_j(n, u_1, \dots, u_{k-2}, \phi_2, \phi_1(n, s_1, u_1, \dots, u_{k-2}, \phi_2)) \quad j = 2, \dots, k-1.$$

Since $u_j^{(2)}$ stands for $x_{j,n+2}$ for $j \geq 2$ we may write

$$x_{1,n+3} = f_1^{(2)}(n, x_{1,n+2}, x_{1,n+1}, x_{2,n}, \dots, x_{k-2,n}).$$

In this step, the variables $x_{k,n}$ and $x_{k-1,n}$ have been replaced with the new variables $x_{1,n+1}, x_{1,n+2}$.

The above procedure repeats as long as inversions are possible. The i -th step of the algorithm is as follows:

i. With $x_{1,n+i} = f_1^{(i-1)}(n, x_{1,n+i-1}, x_{2,n}, \dots, x_{k-i+1,n})$ obtained in Step $i - 1$, shift indices by 1 to get

$$x_{1,n+i+1} = f_1^{(i-1)}(n + 1, x_{1,n+i}, x_{2,n+1}, \dots, x_{k-i,n+1})$$

(a) Use (67) to *substitute* $f_j(n, x_{1,n}, x_{2,n}, \dots, x_{k,n})$ for $x_{j,n+1}$ for $j = 2, \dots, k - 1$, as in Steps 1a, 2a above.

(b) Assume that $f_1^{(i-1)}$ has the following *partial inversion*

$$s_i = f_1^{(i-1)}(n, s_{i-1}, \dots, s_1, u_1, \dots, u_{k-i+1}) \Rightarrow u_{k-i+1} = \phi_i(n, s_i, \dots, s_1, u_1, \dots, u_{k-i})$$

(c) A series of *substitutions* now eliminates the variables u_{k+j} for $j = 0, 1, \dots, k - i + 1$ as in 2c. Substitute $\phi_1(n, s_1, u_1, \dots, u_{k-1})$ for u_k in $f_j(n, u_1, \dots, u_{k-1}, u_k)$ to eliminate u_k . Then substitute $\phi_2(n, s_2, s_1, u_1, \dots, u_{k-2})$ for u_{k-1} in the expression just obtained to remove u_k, u_{k-1} . Continuing, we obtain a function $f_1^{(i)}(n, s_i, \dots, s_1, u_1, \dots, u_{k-i})$ that is independent of $u_k, u_{k-1}, \dots, u_{k-i+1}$.

Remark 19 Execution of the above algorithm may stop before $k - 1$ steps if $f_1^{(i)}$ is independent of u_2, \dots, u_{k-i} for some $i < k - 1$, i.e.,

$$f_1^{(i)}(n, s_i, \dots, s_1, u_1, \dots, u_{k-i}) = f_1^{(i)}(n, s_i, \dots, s_1, u_1).$$

An extreme example is the following system

$$\begin{aligned} x_{1,n+1} &= f_1(n, x_{1,n}, x_{2,n}, \dots, x_{k,n}), \\ x_{i,n+1} &= f_i(n, x_{1,n}), \quad i = 2, \dots, k \end{aligned} \tag{69}$$

where

$$x_{1,n+2} = f_1(n + 1, x_{1,n+1}, x_{2,n+1}, \dots, x_{k,n+1}) = f_1(n + 1, x_{1,n+1}, f_2(n, x_{1,n}), \dots, f_k(n, x_{1,n}))$$

is already a difference equation that does not involve any of the variables $x_{2,n}, \dots, x_{k,n}$ (this system is a special case of the system discussed in the next section).

In general, partial inversions may be needed in some steps of the above algorithm and not in other steps. The folding algorithm above may also stop if $f_1^{(i)}$ has no known partial inversions for some $i < k$. In this case using a different variable $x_{j,k}$ for $j > 1$ may result in a successful execution of the folding algorithm.

The preceding discussion suggests a relevant notion that may be associated with systems to gauge the degree of interdependence of variables.

Definition 20 Assume that a system is foldable into a higher order difference (or differential) equation. Then the order κ of the higher order equation is the **interdependence degree (i.d.)** of the system. If $\kappa = k$ then the foldable system is **fully interdependent** (or fully coupled).

In the above language, a completely controllable system in control theory is fully interdependent. We note that $1 \leq \kappa \leq k$ for a foldable system. The i.d. of a system can often be determined through the use of the folding algorithm even when explicit formulas for the folding equation (or some of the partial inversions needed to calculate it) are not known or do not exist; existence proofs may suffice to carry the algorithm forward and non-existence arguments may be used to terminate it before k steps.

For a first-order equation $x_{n+1} = f(n, x_n)$ we define $\kappa = 1$ and for an *uncoupled system* of first-order equations

$$x_{i,n+1} = f_i(n, x_{i,n}), \quad i = 1, \dots, k$$

with $k \geq 2$ we define $\kappa = 0$. Both of these are examples of *unfoldable* systems since the folding process cannot initiate but their i.d. values are intuitively obvious.

For some foldable systems, $1 \leq \kappa < k$; e.g., for system (69) $\kappa = 2$ but k may be any integer greater than 1. In general, the value of κ depends not only on the component functions f_i but also on the values of the parameters involved. For example, for (57) $\kappa = 2$ if $d = a^2c$ but $\kappa = 3$ otherwise. Such a system has a *variable i.d.* depending on the parameters. The same conclusions hold for the differential system (62). Similarly, for a linear system of two equations, $\kappa = 1$ if the coefficients matrix has determinant 0 and $\kappa = 2$ otherwise; see, e.g., (36). On the other hand, a linear system whose coefficients matrix is diagonal with nonzero entries has nonzero determinant and yet, it is an uncoupled system with $\kappa = 0$.

Finally, if a particular system splinters into two or more blocks or subsystems with disjoint sets of variables for a range of parameter values then such a system has an undefined i.d. for certain parameter ranges. Of course, each of the subsystems containing more than one variable may be separately folded in such cases.

6.3 Folding without inversions

There are systems that fold into equations without requiring any partial inversions. This feature reduces the difficulty of calculations considerably in nonlinear systems. Consider the following difference system

$$\begin{cases} x_{1,n+1} = f_1(n, x_{1,n}, x_{2,n}, x_{3,n}, \dots, x_{k,n}) \\ x_{2,n+1} = f_2(n, x_{1,n}, x_{3,n}, \dots, x_{k,n}) \\ \vdots \\ x_{k-1,n+1} = f_{k-1}(n, x_{1,n}, x_{k,n}) \\ x_{k,n+1} = f_k(n, x_{1,n}) \end{cases} \quad (70)$$

or more concisely,

$$\begin{aligned} x_{i,n+1} &= f_i(n, x_{1,n}, x_{i+1,n}, x_{i+2,n}, \dots, x_{k,n}), \quad i = 1, 2, \dots, k-1 \\ x_{k,n+1} &= f_k(n, x_{1,n}). \end{aligned}$$

Here we *do not assume* that the functions f_i have partial inversions that yield any of the variables $x_{1,n}, x_{2,n}, \dots, x_{k,n}$. In this case, rather than using the folding algorithm of the last section, it is more expedient to proceed as follows. Shift the indices of the first equation in (70) by $k-1$ to obtain

$$x_{1,n+k} = f_1(n+k-1, x_{1,n+k-1}, x_{2,n+k-1}, x_{3,n+k-1}, \dots, x_{k,n+k-1}) \quad (71)$$

Each component in f_1 above is determined from the system itself as

$$\begin{aligned} x_{i,n+k-1} &= f_i(n+k-2, x_{1,n+k-2}, x_{i+1,n+k-2}, \dots, x_{k,n+k-2}), \quad i = 1, 2, \dots, k-1 \\ x_{k,n+k-1} &= f_k(n+k-2, x_{1,n+k-2}) \end{aligned} \quad (72)$$

This step gives the last component $x_{k,n+k-1}$ in (71) exclusively in terms of $x_{1,n+k-2}$. The remaining components $x_{i,n+k-1}$ for $i = 2, 3, \dots, k-1$ are found recursively in terms of $x_{1,n+k-j}$ for suitable values of j . If $i = k-1$ then

$$\begin{aligned} x_{k-1,n+k-1} &= f_{k-1}(n+k-2, x_{1,n+k-2}, x_{k,n+k-2}) \\ &= f_{k-1}(n+k-2, x_{1,n+k-2}, f_k(n+k-3, x_{1,n+k-3})) \end{aligned} \quad (73)$$

Although this expression is more complicated, it is built up from (72) since the term $x_{k,n+k-2}$ is essentially that in (72) but with index reduced by 1. Similarly,

$$\begin{aligned} x_{k-2,n+k-1} &= f_{k-2}(n+k-2, x_{1,n+k-2}, x_{k-1,n+k-2}, x_{k,n+k-2}) \\ &= f_{k-2}(n+k-2, x_{1,n+k-2}, f_{k-1}(n+k-3, x_{1,n+k-3}, x_{k,n+k-3}), f_k(n+k-3, x_{1,n+k-3})) \\ &= f_{k-2}(n+k-2, x_{1,n+k-2}, f_{k-1}(n+k-3, x_{1,n+k-3}, f_k(n+k-4, x_{1,n+k-4})), \\ &\quad f_k(n+k-3, x_{1,n+k-3})) \end{aligned}$$

in which the terms $x_{k-1,n+k-2}$ and $x_{k,n+k-2}$ are again given by (73) and (72), respectively but with indices reduced by 1 in each case. In this way, we determine all of the components in (71) recursively by going backward. The expressions become more involved as the value of i decreases to 2 but the algorithm is well-defined.

Let $\{x_{1,n}\}$ be a solution of (71) in which the terms $x_{2,n+k-1}, x_{3,n+k-1}, \dots, x_{k,n+k-1}$ are replaced by appropriate expressions in terms of $x_{1,n+k-j}$ as outlined above so that (71) is a difference equation of order k . Then we may calculate $x_{i,n}$ for $i = 2, \dots, k$ using the system. Given the initial values $x_{i,0}$ for $i = 1, \dots, k$ we first calculate the initial values $x_{1,j}$, $j = 1, \dots, k$ for (71) as follows:

$$x_{1,j} = f_1(j-1, x_{1,j-1}, x_{2,j-1}, \dots, x_{k,j-1}), \quad x_{i,j-1} = f_i(j-2, x_{1,j-2}, x_{i+1,j-2}, \dots, x_{k,j-2})$$

for $j = 2, \dots, k$ and of course, $x_{1,1} = f_1(0, x_{1,0}, x_{2,0}, \dots, x_{k,0})$. Next, once $\{x_{1,n}\}$ is determined for all n from (71) we calculate $x_{i,n}$ for $i = 2, \dots, k$ in a backward fashion as follows:

$$\begin{aligned} x_{k,n} &= f_k(n-1, x_{1,n-1}), \quad n \geq 1 \\ x_{k-1,n} &= f_{k-1}(n-1, x_{1,n-1}, x_{k,n-1}), \\ &\vdots \\ x_{2,n} &= f_2(n-1, x_{1,n-1}, x_{3,n-1}, \dots, x_{k,n-1}) \end{aligned}$$

Thus, an orbit of (70) may be determined from a solution of (71) without having to solve additional difference equations or to find partial inversions.

To illustrate the above procedure, consider the autonomous system

$$\begin{cases} x_{1,n+1} = f(ax_{2,n} + bx_{3,n}) \\ x_{2,n+1} = cx_{1,n} + g(x_{3,n}) \\ x_{3,n+1} = \alpha x_{1,n} + \beta \end{cases} \quad (74)$$

where $f, g : \mathbb{R} \rightarrow \mathbb{R}$ are any non-constant functions and $a, b, c, \alpha, \beta \in \mathbb{R}$ with $a, \alpha \neq 0$. We do not assume the existence of partial inversions to solve either the first or the second equation of the above system for $x_{2,n}$ or $x_{3,n}$. We calculate

$$\begin{aligned} x_{1,n+3} &= f(ax_{2,n+2} + bx_{3,n+2}) \\ &= f(acx_{1,n+1} + ag(x_{3,n+1}) + b\alpha x_{1,n+1} + b\beta) \\ &= f((ac + b\alpha)x_{1,n+1} + ag(\alpha x_{1,n} + \beta) + b\beta) \end{aligned} \quad (75)$$

Equation (75) is derived from the system (74) without any partial inversions. In particular, if

$$ac + b\alpha = 0 \quad (76)$$

then (75) reduces to

$$x_{1,n+3} = f(ag(\alpha x_{1,n} + \beta) + b\beta) \quad (77)$$

Note that for $i = 0, 1, 2$

$$x_{1,3(k+1)+i} = f(ag(\alpha x_{1,3k+i} + \beta) + b\beta)$$

so if $\{t_k\}$ is a solution of the first-order equation

$$t_{k+1} = f(ag(\alpha t_k + \beta) + b\beta) \quad (78)$$

with some initial value t_0 then

$$t_0 = x_{1,i} \Rightarrow t_k = x_{1,3k+i} \quad i = 0, 1, 2$$

Therefore, every solution of (77) is obtained from 3 solutions of (78). In this sense, system (74) may be reduced to a first-order equation if (76) holds. If $\{x_{1,n}\}$ is a solution of (77) then for $n = 1, 2, 3, \dots$ the remaining components are found passively as

$$x_{3,n} = \alpha x_{1,n-1} + \beta, \quad x_{2,n} = c x_{1,n-1} + g(x_{3,n-1}).$$

Remark 21 *The ideas discussed in this section do NOT apply to systems of differential equations of type*

$$\begin{aligned} x'_i(t) &= f_i(t, x_1(t), x_{i+1}(t), x_{i+2}(t), \dots, x_k(t)), \quad i = 1, 2, \dots, k-1 \\ x'_k(t) &= f_k(t, x_1(t)) \end{aligned} \quad (79)$$

because the interdependence of variables in such systems is not the same as that in (70). A more suitable analog of (79) is the system of difference equations

$$\begin{aligned} \Delta x_{i,n} &= f_i(n, x_{1,n}, x_{i+1,n}, x_{i+2,n}, \dots, x_{k,n}), \quad i = 1, 2, \dots, k-1 \\ \Delta x_{k,n} &= f_k(n, x_{1,n}) \end{aligned} \quad (80)$$

where $\Delta x_{i,n} = x_{i,n+1} - x_{i,n}$. In this system the value of $x_{i,n+1}$ depends on $x_{i,n}$ which is expressly not the case in (70). Both of the systems (79) and (80) require partial inversions and are more difficult to fold into equations than (70).

In general, difference and differential systems do not benefit equally from folding because if a singularity occurs in the folding of a differential system then it is less avoidable than the analogous situation in the discrete case. Simply put, flows of differential systems are continuous and may thus cross over or into singularity manifolds of the folding rather than jumping over them. For instance, when Lorenz's system is folded (see [10]) the third order jerk function has a singularity in the terms x'/x that the familiar, butterfly-shaped flows cannot seem to avoid. On the other hand, the occurrence of singularities in the folding is not always fatal if the main flows are not affected. For instance, the Volterra predator-prey model

$$\begin{cases} x' = x(a - by) \\ y' = y(c - dx) \end{cases}$$

where $a, b, c, d > 0$ folds as follows: $by = a - x'/x$ so

$$x'' = ax' - bx'y - bxy' = \frac{(x')^2}{x} + (x' + ax)(c - dx)$$

Although a singularity exists in the folding on the y -axis this does not involve the positive solutions that are important in modelling. Existence and behavior of solutions for the above second-order equation may be tied to similar issues about the flows of the Volterra system; such issues require closer scrutiny in the future.

References

- [1] Abhyankar, S.S., *Algebraic Geometry for Scientists and Engineers*, Math. Surv. and Monographs No.35, AMS, 1990.
- [2] Barnett, S., *Introduction to Mathematical Control Theory*, Clarendon Press, Oxford, 1990.
- [3] Darzin, P.G., *Nonlinear Systems*, Cambridge, 1992.
- [4] Eichhorn, R., S.J. Linz and P. Hänggi, “Transformations of nonlinear dynamical systems to jerky motion and its application to minimal chaotic flows,” *Phys. Rev. E* **58** (1998) 7151-7164.
- [5] Elaydi, S., *An Introduction to Difference Equations*, 3rd ed., Springer, New York, 2005.
- [6] Halanay, A., *Differential Equations: Stability, Oscillations, Time lags*, Academic Press, New York, 1966.
- [7] Halanay, A. and V. Rasvan, *Stability and Stable Oscillations in Discrete Time Systems*, Gordon and Breach, Singapore, 2000.
- [8] Lang, Serge, *Analysis I*, Addison-Wesley, Reading, 1968.
- [9] LaSalle, J.P., *The Stability and Control of Discrete Processes*, Appl. Math. Sci. 62, Springer-Verlag, New York, 1986.
- [10] Linz, S.J., “Nonlinear dynamical models and jerky motion,” *Am. J. Phys.* **65** (1997) 523-526.
- [11] Lorenz, E.N., “Deterministic nonperiodic flows,” *J. Atmos. Sci.* **20** (1963) 130-141.
- [12] Rössler, O.E., “An equation for continuous chaos,” *Phys. Lett. A* **57** (1976) 397-398.
- [13] Sedaghat, H., *Form Symmetries and Reduction of Order in Difference Equations*, Chapman & Hall/CRC Press, Boca Raton, 2011.
- [14] Sedaghat, H., Semiconjugate factorizations of higher order linear difference equations in rings, *J. Difference Eq. and Appl.* **20** (2014) 251-270; preprint available on *arXiv.org*, arXiv:1301.2804 [math.CA]
- [15] Sprott, J.C., “Some simple chaotic flows,” *Phys. Rev. E* **50** (1994) 647-650.
- [16] Strogatz, S.H., *Nonlinear Dynamical Systems and Chaos*, Addison-Wesley, Reading, 1994.
- [17] Van Dooren, P. and J. Sreedhar, When is a periodic discrete-time system equivalent to a time-invariant one? *Proceedings of the Conference on Information Sciences & Systems*, (1993) 194-195.