Acta Numerica (2008), pp. 1–86
doi: 10.1017/S0962492906340019

Linear algebra algorithms as dynamical systems

Moody T. Chu*
Department of Mathematics,
North Carolina State University,
Raleigh, North Carolina, 27695-8205, USA
E-mail: chu@math.ncsu.edu

In memory of Gene Golub, a good friend and mentor

Any logical procedure that is used to reason or to infer either deductively or inductively, so as to draw conclusions or make decisions, can be called, in a broad sense, a realization process. A realization process usually assumes the recursive form that one state develops into another state by following a certain specific rule. Such an action is generally formalized as a dynamical system. In mathematics, especially for existence questions, a realization process often appears in the form of an iterative procedure or a differential equation. For years researchers have taken great effort to describe, analyse, and modify realization processes for various applications.

The thrust in this exposition is to exploit the notion of dynamical systems as a special realization process for problems arising from the field of linear algebra. Several differential equations whose solutions evolve in submanifolds of matrices are cast in fairly general frameworks, of which special cases have been found to afford unified and fundamental insights into the structure and behaviour of existing discrete methods and, now and then, suggest new and improved numerical methods. In some cases, there are remarkable connections between smooth flows and discrete numerical algorithms. In other cases, the flow approach seems advantageous in tackling very difficult open problems. Various aspects of the recent development and application in this direction are discussed in this paper.

* This research was supported in part by the National Science Foundation under grants DMS-0505880 and CCF-0732299.

CONTENTS

1. Introduction

At the risk of oversimplifying an extremely complex mechanism of thinking, we begin with a large and loose metaphor to delineate the characteristics of a realization process. A realization process usually comprises three components. First, we have two abstract problems, of which one is an artificial problem whose solution is easy to find, while the other is the real problem whose solution is hard to attain. Secondly, we need to design a bridge or a path that connects the easy problem to the difficult problem. The basic idea is to utilize the bridge to set the rule for a certain dynamical system that evolves from the solution of the easy problem to the solution of the difficult problem. Once the blueprint for the bridge construction is in place, we finally need a practical method allowing us to move along the path so that the desirable solution is reached at the end of the process.

The steps taken for the realization, that is, the changes from one state to the next state along the bridge, can be discrete or continuous. Given the limitations of current computing technology, however, it is generally accepted that the most common and effective way to execute a computation is by means of floating-point arithmetic (Goldberg 1991). As such, it is almost a mandate that a continuous realization process must be discretized first before it can be put into operation numerically (Allgower and Georg 2003). For this reason, and perhaps more so for convenience, we have observed that a majority of numerical algorithms in practice are iterative in nature. It could very well be the case that an iterative scheme was initially devised without the notion of a 'connecting bridge' in mind. Its convergence and hence the appearance of a bridge connecting the starting point to the limit point are often not immediately evident, but are rather the result of hard analysis. In hindsight, we now recognize that most iterative methods can be categorically classified as realization processes.

Our principal goal in this exposition is to characterize the relationship between the dynamics of classical iterative methods and that of certain differential systems. We note that in certain cases the continuous model 'interpolates' exactly the iterates of the corresponding discrete method, or that the discrete model 'samples' the solution flow of the corresponding differential equation at integer times, while in other case we can only suggest a straightforward continuous extension or an obvious discretization. In all cases, we think that the interplay between dynamical systems and computational methods is not only of theoretical interest but also has important consequences, as will be made manifest in the subsequent discussion.

Needless to say, the success of a realization process depends on how the bridge is extended from the trivial solution to the desirable solution. Sometimes we have specific guidelines in building the bridge. Bridges underlying the projected gradient method (Chu and Driessel 1990), the interior point method (Karmarkar 1984, Wright 1997, Potra and Wright 2000, Wright

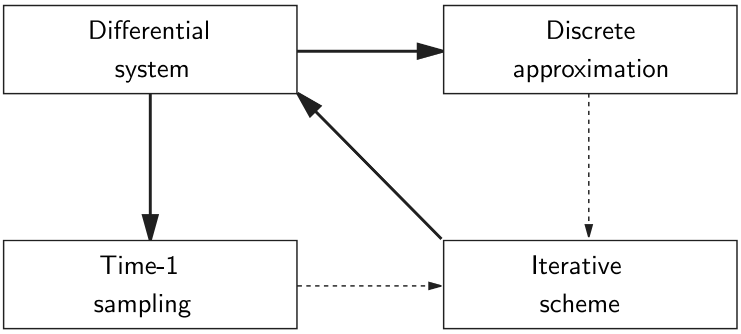

Figure 1.1. Possible links between continuous and discrete dynamical systems.

2005) or the conjugate gradient method (Hestenes and Stiefel 1952, Greenbaum 1997, Meurant 2006), for example, are based on the principle of systematically optimizing the values of certain objective functions. Sometimes the bridge is developed more or less on the basis of 'innate inclination', where we can only hope that the bridge will connect to the other end. The continuous Newton method (Smale 1977), or the homotopy method (Allgower and Georg 1980, García and Gould 1980, Morgan 1987), for example, requires extra efforts to make sure that the bridge actually makes the desirable connection. In other situations, such as for the QR algorithm (Francis 1961/1962, Watkins 1982) or the Rayleigh quotient iteration (Parlett 1974), it appears that the bridge comes into existence in an anomalous way. But the fact is that a much deeper mathematical or physical cause is often involved. When the theory is unveiled, we are often amazed to see that these seemingly aberrant bridges do exist by themselves naturally.

Figure 1.1 serves as a reminder of the possible links between continuous and discrete dynamical systems. The dotted lines indicate that an iterative scheme might be generated or regenerated from a differential system. Going from a continuous system to a discrete system is usually regarded as 'natural' since most numerical ODE techniques are doing precisely that task, but one major thrust of this paper is to illustrate some non-traditional ways of discretization that are not as straightforward as an ordinary ODE scheme, but could lead to new and effective algorithms. On the other hand, going from an iterative scheme to a differential system is not always as obvious as merely considering the discrete scheme as an Euler step of a differential system. Other mechanisms, such as control, acceleration, optimization, or structure preservation, can also induce continuous dynamical systems. Our presentation in this paper centres around describing, case by case, each direction in the flowchart of Figure 1.1, with applications arising from linear algebra algorithms.

2. Numerical analysis versus dynamical systems

Most of the iterative methods developed for practical purposes assume the format of an *m-step sequential process* (Ortega and Rheinboldt 2000),

$$\mathbf{x}_{k+1} = G_k(\mathbf{x}_k, \ldots, \mathbf{x}_{k-m+1}), \quad k = 0, 1, \ldots, \tag{2.1}$$

where

$$G_k : D_k \subset V^m \to V \tag{2.2}$$

are some predetermined maps, V is a designated vector space and m is a fixed integer. Obviously, to start up an m-step iteration, initial values $\mathbf{x}_0, \mathbf{x}_{-1}, \ldots, \mathbf{x}_{-m+1}$ must be specified first. An m-step process is said to be *stationary* if all iteration maps G_k together with the domains D_k are independent of k.

Conventional numerical integrators such as the Runge–Kutta methods and the Adams methods for an initial value problem,

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(t, \mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{2.3}$$

are typical one-step and multi-step sequential processes, respectively. The corresponding iterative maps G_k have evident definitions for explicit methods, but their construction is more devious for implicit methods. Discussions on issues of stability and convergence for discrete methods in this context are abundant in the literature. We shall not review any numerical ODE techniques in this paper, but would recommend the seminal books by Hairer, Nørsett and Wanner (1993) and Hairer and Wanner (1996) as general references on this subject. Our focus in this paper is concentrated primarily on a few very specific iterative processes that were developed originally for problems from fields other than ODEs. It will become apparent that the differential systems associated with the applications to be discussed are of a distinct character and that special numerical techniques might be needed. It is perhaps fitting to echo what Gear (1981) has suggested: that there are more things to do with ODE techniques.

It should be stressed that the subject of discrete dynamical systems has its own distinguished role in nonlinear analysis, providing models for many natural phenomena, and is itself a discipline of extensive and deep research activity. For example, there is Sarkovskii's theorem, remarkable for its lack of hypotheses and for its qualitative universality, asserting that if the discrete dynamical system formed by iterating a continuous function $f : \mathbb{R} \to \mathbb{R}$ has a point of period 3, then it has points of all periods. This topic is beyond the scope of our current discussion, but we find the introductory textbooks by Devaney (1992) and Elaydi (2005), as well as the extended article by Galor (2005), very accessible. The book by Kulenović and Merino (2002) is interesting in that it contains ready-to-use software for computer simulation. For more rigorous theoretical development and a rich collection of applications, we recommend the monograph by Sedaghat (2003). Of course, the fundamental textbook by Wimp (1984) remains the absolute reference for computational issues associated with finite difference equations.

2.1. Dynamics of iterative maps

A subtle line must be drawn in that the classical convergence analysis and stability theory of numerical analysis consider only systems with trivial asymptotic behaviour, namely convergence to a unique equilibrium point, whereas most dynamical systems show more complicated behaviour, with limit cycles or even strange attractors (Stuart and Humphries 1996). From a numerical analysis point of view, the discretization of a differential equation is primarily meant to trace the solution flow with reliable and reasonable accuracy. From a dynamical systems perspective, however, the analysis of a sequential process seeks to differentiate the intrinsic geometric structure. There is considerable overlap between these two disciplines, but there are also significant differences, as Guckenheimer (2002) explains:

> 'The tension between geometric and more traditional analysis of numerical integration algorithms can be caricatured as the interchange between two limits. The object of study is systems of ordinary differential equations and their flows. Numerical solution of initial value problems for systems of ordinary differential equations discretizes the equations in time and produces sequences of points that approximate solutions over time intervals. Dynamical systems theory concentrates on questions about long-time behavior of the solution trajectories, often investigating intricate geometry in structures formed by the trajectories. The two limits of (1) discretizing the equations with finer and finer resolution in time and (2) letting time tend to infinity do not commute. Classical theories of numerical analysis give little information about the limit behavior of numerical trajectories with increasing time. Extending these theories to do so is feasible only by making the analysis specific to classes of systems with restricted geometric properties. The blend of geometry and numerical analysis that is taking place in current research has begun to produce a subject with lots of detail and richness.'

Perhaps a simple example can best demonstrate the above points. Consider the task of solving the logistic equation,

$$\frac{\mathrm{d}x}{\mathrm{d}t} = x(1-x), \quad x(0) = x_0, \tag{2.4}$$

by the Euler method,

$$x_{k+1} = x_k + \epsilon x_k(1-x_k), \tag{2.5}$$

with step size ϵ. The exact solution of (2.4) is given by

$$x(t) = \frac{x_0}{x_0 + \mathrm{e}^{-t}(1-x_0)}, \tag{2.6}$$

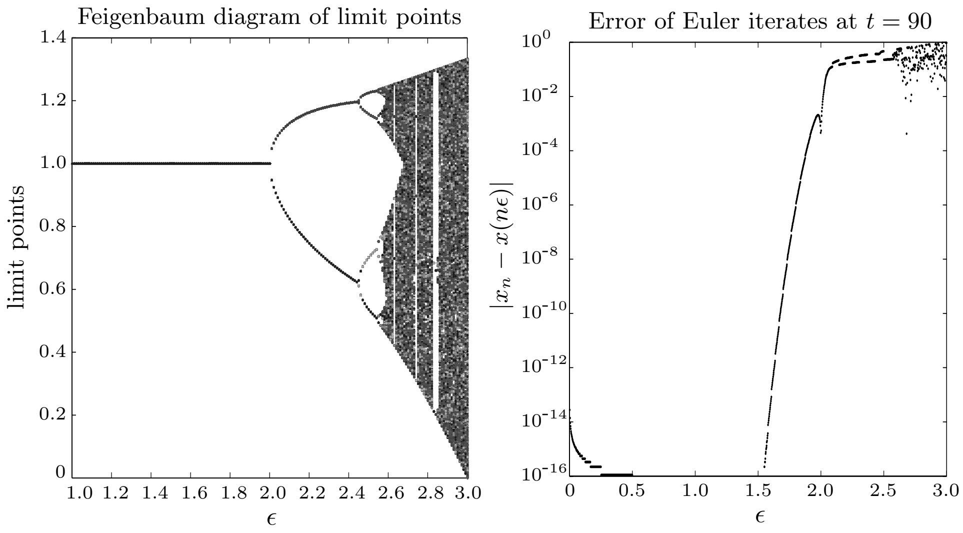

Figure 2.1. Euler iterations for the logistic equation.

which converges to the equilibrium $x(\infty) = 1$ for any initial value $x_0 \neq 0$. Traditional numerical analysis concerns and proves the convergence of x_n to $x(t)$ at each fixed t in the sense that $n \to \infty$ but $t = n\epsilon$. With $n = \lceil \frac{90}{\epsilon} \rceil$ and $0 < \epsilon \leq 3$, we plot the absolute error $|x_n - x(n\epsilon)|$ in the right-hand graph of Figure 2.1. The graph for ϵ between approximately 0.5 and 1.5 is omitted because of the logarithm at machine zero. Note that even at ridiculously large step sizes the errors follow the theoretic estimate $\mathcal{O}(\epsilon)$. On the other hand, with each fixed ϵ, if we iterate the Euler steps 5000 times, then the sequence $\{x_k\}$ exhibits period doubling when ϵ is larger than 2. The left-hand graph in Figure 2.1 shows the limit points, as a function of ϵ, of the corresponding sequence $\{x_k\}$. The so-called Feigenbaum diagram clearly indicates a cascade of period doubling as ϵ increases, which eventually leads to numerical chaos. Note in particular that the equilibrium $x(\infty) = 1$ for (2.4) is no longer an attractor to the discrete dynamical system (2.5) when ϵ is sufficiently large. This equilibrium of the original differential equation does not even appear in the Feigenbaum diagram for large ϵ values. In contrast, implicit schemes such as

$$x_{k+1} = x_k + \epsilon x_k(1 - x_{k+1}) \tag{2.7}$$

or

$$x_{k+1} = x_k + \epsilon x_{k+1}(1 - x_{k+1}) \tag{2.8}$$

converge to the equilibrium $x(\infty) = 1$ for any step size ϵ.

With this lesson in mind, we must be careful in distinguishing between the limiting behaviour of an iterative algorithm, which is designed originally by a numerical practitioner to solve a specific problem, and that of a discrete approximation of a differential system, which is formulated to mimic an existing iterative algorithm. Likewise, we must also distinguish the asymptotic behaviour of a differential system, which is developed originally from a specific realization process, and that of its discrete approximation, which becomes an iterative scheme.

2.2. Pseudo-transient continuation

It might be worthwhile to illustrate a general mechanism for advancing a specific continuous system. This idea is not the only way to discretize a continuous system and does not work for every kind of differential system, but it illustrates an interesting view of how the trajectory of a continuous system can be approximately tracked so as to find the equilibrium point, by using numerical ODE techniques in a somewhat non-traditional way.

We shall see in Section 7.3 that it is often the case in many applications that the solution $\mathbf{x}^*$ is realized as the limit point

$$\mathbf{x}^* = \lim_{t\to\infty} \mathbf{x}(t), \tag{2.9}$$

where $\mathbf{x}(t)$ is the solution to the gradient flow

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\nabla F(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{2.10}$$

with respect to a specified smooth objective function $F : \mathbb{R}^n \to \mathbb{R}$. At first glance, we should be able to find $\mathbf{x}^*$ by solving the first-order optimality condition

$$\nabla F(\mathbf{x}) = 0,$$

with some general-purpose Newton-like iterative methods. Such an approach, however, ignores the gradient property of ∇F and may locate a solution which is different from $\mathbf{x}^*$, and might even be dynamically unstable. Employing some existing ODE integrators to carefully trace the trajectory $\mathbf{x}(t)$ is another way of finding $\mathbf{x}^*$. As reliable as this approach might be, it requires expensive computation at the transient states which is not needed for computing $\mathbf{x}^*$.

One feasible discretization of (2.10) is as follows. Assuming that an approximate solution $\mathbf{x}_k$ has already been computed, one implicit Euler step with step size ϵ_k to (2.10) yields a nonlinear equation,

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \epsilon_k \nabla F(\mathbf{x}_{k+1}), \tag{2.11}$$

for the next step $\mathbf{x}_{k+1}$. Instead of solving (2.11) to high precision as an ODE integrator would normally do, we perform the correction using only one Newton iteration starting at $\mathbf{x}_k$ and accept the outcome as $\mathbf{x}_{k+1}$. The idea is to stay near the true trajectory, but not to strive for accuracy. It is not difficult to see that one Newton step for (2.11) leads to the iterative scheme

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left(\frac{1}{\epsilon_k} I_n + \nabla^2 F(\mathbf{x}_k)\right)^{-1} \nabla F(\mathbf{x}_k). \tag{2.12}$$

This scheme is a special implicit upwind method which has been applied successfully for computing steady-state solutions in the PDE community (Mulder and van Leer 1985). Note that for small values of ϵ_k the scheme (2.12) behaves like a steepest descent method, whereas for large values of ϵ_k it behaves like a Newton iteration. Taking into account the fact that $\nabla F(\mathbf{x})$ should have small norm near the optimal point $\mathbf{x}^*$, the so-called 'switched evolution relaxation' strategy for selecting the step sizes, namely,

$$\epsilon_{k+1} = \epsilon_k \frac{\|\nabla F(\mathbf{x}_k)\|}{\|\nabla F(\mathbf{x}_{k+1})\|}, \tag{2.13}$$

seems to be able to capture the characteristics of being relatively large in the initial phase, and small in the terminal phase of the iteration. The method described above is referred to as *pseudo-transient continuation* in Kelley and Keyes (1998), where convergence theory and implementation issues are also discussed. For a review of its applications, see the recent paper by Kelley, Liao, Qi, Chu, Reese and Winton (2007).

In the subsequent sections of this paper, we shall review various kinds of numerical algorithms, especially those related to linear algebra problems, and explore the possibility of recasting them as dynamical systems. Not only do we want to establish the relationship for theoretical interest, but we also wish to gain some insights via this interpretation, and to develop some new algorithms. A few of these ideas have already been reported in an earlier review by Chu (1988). It is hoped that this paper will bring up to date some more recent developments advanced in the past two decades and point out some new areas for research.

3. Dynamical systems for linear equations

Iterative methods for linear systems have a significant role in history and in applications. This class of methods has come a long way with a dazzling array of developments. See, for example, the various 'templates' discussed in the book by Barrett *et al.* (1994). Research is still evolving even now. Current techniques range from the ingenious acceleration of classical iterative schemes (Hageman and Young 1981) to effective Krylov subspace approximation (van der Vorst 2003), to the more geometrically motivated multi-grid (Briggs 1987, Bramble 1993) or domain decomposition approaches (Toselli and Widlund 2005). Some favourites of practitioners include the preconditioned conjugate gradient (PCG) method (Hestenes and Stiefel 1952), the generalized minimum residual (GMRES) method (Saad and Schultz 1986), the quasi-minimal residual (QMR) method (Freund and Nachtigal 1991), and so on. It is impossible to discuss the dynamics of these methods one by one in this presentation. We outline briefly only two principal ideas in this section.

3.1. Stationary iteration

Most classical iterative methods, such as the Jacobi, the Gauss–Seidel, or the SOR methods, for the linear system

$$A\mathbf{x} = \mathbf{b}, \tag{3.1}$$

where $A \in \mathbb{R}^{n\times n}$ is non-singular and $\mathbf{b} \in \mathbb{R}^n$, are one-step stationary sequential processes of the form

$$\mathbf{x}_{k+1} = G\mathbf{x}_k + \mathbf{c}, \quad k = 0, 1, 2, \ldots. \tag{3.2}$$

The *iteration matrix* $G \in \mathbb{R}^{n\times n}$ plays a crucial role in the convergence of $\{\mathbf{x}_k\}$ in this scheme. Indeed, a necessary and sufficient condition for the convergence of (3.2) from any given starting value $\mathbf{x}_0$ to the unique solution $\mathbf{x}^*$ of (3.1) is that the spectral radius $\rho(G)$ is strictly less than one (Varga 2000). Extensive efforts have been made to construct G to ensure convergence. This is usually done as follows. At the fixed point $\mathbf{x}^*$, we see the relationship

$$\begin{aligned} G &= I - K^{-1}A, \qquad (3.3)\\ \mathbf{c} &= K^{-1}\mathbf{b}, \end{aligned}$$

for some non-singular matrix K. Because A is 'split' by K in the sense that

$$A = K - KG,$$

K is called a *splitting matrix* of A. Thus, in designing an effective iterative method, attention turns to the selection of a splitting matrix K of A, such that $\rho(I - K^{-1}A) < 1$, for which K^{-1} is relatively easy to compute. The mathematical theory developed for this traditional approach can be found in the seminal book by Varga (2000).

It is trivially seen that the iterative scheme (3.2) is equivalent to an Euler step with unit step size applied to the differential system

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}; K) := -K^{-1}(A\mathbf{x} - \mathbf{b}), \tag{3.4}$$

whose analytic solution is given by

$$\mathbf{x}(t) = \mathrm{e}^{-K^{-1}At}(\mathbf{x}_0 - A^{-1}\mathbf{b}) + A^{-1}\mathbf{b}. \tag{3.5}$$

For convergence, however, there is a fundamental difference between the difference equation (3.2) and the differential equation (3.4) in the condition to be imposed on the splitting matrix K. The concern in (3.2) is to make $\rho(I - K^{-1}A)$ as small as possible. Indeed, an ideal K would be one for which the eigenvalues of $K^{-1}A$ are clustered around the real value $\lambda = 1$. Of course, the obvious choice $K = A$ is not practical, because computing A^{-1} is precisely the task we want to circumvent by doing iteration. In contrast, the concern in (3.4) is to make the real part of eigenvalues of $K^{-1}A$ positive and large for fast convergence to the limit point $\mathbf{x}^*$. It might also be desirable to keep the eigenvalues of $K^{-1}A$ clustered to avoid stiffness or high oscillation.

All of these requirements imposed on eigenvalues of $K^{-1}A$ in either case can be met by employing techniques for multiplicative inverse eigenvalue problems, which are discussed in the book by Chu and Golub (2005). For specific applications, finding the most suitable preconditioner has been a major research effort, since it can significantly improve the efficiency of an iterative method. In practice, however, preconditioning is an inexact science because different preconditioners work better for different kinds of problems. To stay within the theme of this article, we shall not elaborate on the choice of K, but assume that it has been constructed in some fashion.

The question now is how to integrate (3.4) so as to reach its equilibrium point quickly. Certainly there are various ways to discretize the differential system (3.4), including the pseudo-transient continuation method described earlier. There are also many different choices of the splitting matrix K, including an obvious choice $K^{-1} = A^\top$ which leads to a gradient flow

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -A^\top(A\mathbf{x} - \mathbf{b}), \tag{3.6}$$

for the objective function $f(\mathbf{x}) = \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|_2^2$, which works even when A is a rectangular matrix. Once a decision is made, what is the dynamics of the resulting iterative map?

We shall describe in the next section how the discretization of (3.4) can be related to the Krylov subspace method. At present, it might be appropriate to recall two scenarios already described in (Chu 1988) that demonstrate the 'tension' referred to by Guckenheimer (2002) between geometric and more traditional analysis of numerical integration algorithms.

First, suppose that the trapezoidal rule with step size ϵ is applied to (3.4). We obtain an iterative scheme,

$$\mathbf{x}_{k+1} = \left(I + \frac{\epsilon}{2}K^{-1}A\right)^{-1}\left(I - \frac{\epsilon}{2}K^{-1}A\right)\mathbf{x}_k + \epsilon\left(I + \frac{\epsilon}{2}K^{-1}A\right)^{-1}K^{-1}\mathbf{b}, \tag{3.7}$$

which makes an interesting comparison with the analytic solution,

$$\mathbf{x}(t+\epsilon) = \mathrm{e}^{-\epsilon K^{-1}A}\mathbf{x}(t) + \int_t^{t+\epsilon} \mathrm{e}^{(t+\epsilon-u)A}(K^{-1}\mathbf{b})\,\mathrm{d}u. \tag{3.8}$$

Specifically, the iteration matrix $\left(I + \frac{\epsilon}{2}K^{-1}A\right)^{-1}\left(I - \frac{\epsilon}{2}K^{-1}A\right)$, being the $(1,1)$-pair Padé approximation, agrees with the exponential matrix $\mathrm{e}^{-\epsilon K^{-1}A}$ up to the ϵ^2 term in the series expansion. Likewise, the second term in (3.7) agrees with the integral in (3.8) to the same order of accuracy. Though it might not be practical for real computation, the iterative scheme (3.7), using the trapezoidal rule, on one hand tracks the solution curve closely for small ϵ, and on the other hand converges to $\mathbf{x}^*$ for any step size ϵ.

Secondly, recall that the well-known polynomial acceleration methods applied to (3.2) usually assume a three-term recursive relationship,

$$\begin{aligned}\mathbf{x}_1 &= \epsilon_1(G\mathbf{x}_0 + \mathbf{c}) + (1-\epsilon_1)\mathbf{x}_0,\\ \mathbf{x}_{k+1} &= \alpha_{k+1}\left[\epsilon_{k+1}(G\mathbf{x}_k + \mathbf{c}) + (1-\epsilon_{k+1})\mathbf{x}_k\right] + (1-\alpha_{k+1})\mathbf{x}_{k-1},\end{aligned} \tag{3.9}$$

with some properly defined real numbers α_k and ϵ_k (Hageman and Young 1981, Chapters 4–6). Note that the scheme (3.9) amounts to a two-step sequential process. It is not difficult to rewrite the recursive relationship as

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{x}_0 + \epsilon_1\mathbf{f}_0,\\ \mathbf{x}_{k+1} &= \alpha_{k+1}\mathbf{x}_k + (1-\alpha_{k+1})\mathbf{x}_{k-1} + \epsilon_{k+1}\alpha_{k+1}\mathbf{f}_k,\end{aligned} \tag{3.10}$$

with $\mathbf{f}_k := \mathbf{f}(\mathbf{x}_k; K)$, which is the vector field in (3.4). This identification offers an interesting interpretation, that is, the polynomial acceleration procedure (3.9) can be regarded as the application of a sequence of explicit two-step methods (3.10) to the differential system (3.4) with step size ϵ_{k+1}. Beware, however, of the subtle distinction that the two-step method (3.10) has a low order of accuracy (of order one, indeed) if regarded as an ODE method, but has a faster rate of convergence (with appropriately selected step size ϵ_k) to the equilibrium $\mathbf{x}^*$ if regarded as an iterative scheme.

3.2. Krylov subspace methods

We have seen how a basic iterative system (3.2) motivates the continuous system (3.4), which we now rewrite as

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = K^{-1}\mathbf{r}, \tag{3.11}$$

with $\mathbf{r} := \mathbf{b} - A\mathbf{x}$ denoting the residual vector. Instead of considering the iterative scheme,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \epsilon_k K^{-1}\mathbf{r}_k, \tag{3.12}$$

as one Euler step with variable step size ϵ_k, we interpret (3.12) as a line search in the $K^{-1}\mathbf{r}_k$ direction for a given K^{-1}. In this context, we can even put aside the concern of requiring eigenvalues of $K^{-1}A$ to reside in the right half of the complex plane. If the search is intended to minimize the size of the residual vector, say, $\mathbf{r}_{k+1}^\top \mathbf{r}_{k+1}$, then the optimal step size is given by

$$\epsilon_k = \frac{\langle AK^{-1}\mathbf{r}_k, \mathbf{r}_k\rangle}{\langle AK^{-1}\mathbf{r}_k, AK^{-1}\mathbf{r}_k\rangle}, \tag{3.13}$$

where $\langle \mathbf{u}, \mathbf{v}\rangle := \mathbf{u}^\top \mathbf{v}$ stands for the inner product. If A is symmetric and positive definite and $\mathbf{r}_{k+1}A^{-1}\mathbf{r}_{k+1}$ is to be minimized, then the optimal step size is given by

$$\epsilon_k = \frac{\langle K^{-1}\mathbf{r}_k, \mathbf{r}_k\rangle}{\langle AK^{-1}\mathbf{r}_k, K^{-1}\mathbf{r}_k\rangle}. \tag{3.14}$$

In the special case $K = I$, the two step size selection strategies (3.13) and (3.14) correspond precisely to the ORTHOMIN(1) and steepest descent methods (Greenbaum 1997), respectively.

We can also adopt a two-step sequential process similar to the accelerator (3.10), except that conventionally we prefer to write the scheme as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \epsilon_k\big[K^{-1}\mathbf{r}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1})\big], \tag{3.15}$$

with step size ϵ_k. Such a scheme, if regarded as an ODE method for the differential system (3.11), would have low order of accuracy. However, by defining $\mathbf{p}_0 = K^{-1}\mathbf{r}_0$ and

$$\mathbf{p}_k := K^{-1}\mathbf{r}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1}) = K^{-1}\mathbf{r}_k + \beta_k\mathbf{p}_{k-1}, \tag{3.16}$$

with $\beta_k := \epsilon_{k-1}\gamma_k$, we see an interesting non-stationary iteration embedded in (3.15), that is,

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + \epsilon_k\mathbf{p}_k,\\ \mathbf{r}_{k+1} &= \mathbf{r}_k - \epsilon_k A\mathbf{p}_k,\end{aligned}$$

which has profound consequences. In particular, under the assumption that A is symmetric and positive definite and K is symmetric, it can be verified that the iterative scheme (3.15) with the specially selected scalars

$$\epsilon_k = \frac{\langle \mathbf{p}_k, \mathbf{r}_k\rangle}{\langle A\mathbf{p}_k, \mathbf{p}_k\rangle}, \tag{3.17}$$

$$\beta_{k+1} = -\frac{\langle K^{-1}\mathbf{r}_{k+1}, A\mathbf{p}_k\rangle}{\langle A\mathbf{p}_k, \mathbf{p}_k\rangle}, \quad k = 0, 1, \ldots, \tag{3.18}$$

corresponds precisely to the well-known preconditioned conjugate gradient method with K^{-1} as the preconditioner (Greenbaum 1997). Among the many nice properties of the conjugate gradient method, the most significant one is that the sequence $\{\mathbf{x}_k\}$ converges in exact arithmetic to the equilibrium point $\mathbf{x}_*$ in at most n iterations. Such a phenomenon of reaching convergence in only a finite number of steps (by a somewhat laughably inaccurate method as far as solving (3.11) is concerned) is perhaps unexpected from a numerical ODE point of view.

There is a variety of different formulations of the Krylov subspace methods (van der Vorst 2003). We remark that quite a few of them can be derived in a similar spirit, but space limitation prohibits us from giving the details here. Referring to the diagram in Figure 1.1, the lesson we have learned is that from a very basic discrete dynamical system such as (3.2) we can arrive at a very general continuous dynamical system such as (3.4). Instead of tracing the continuous dynamics by some very refined numerical ODE methods, we could use the system as a guide to draw up some general procedures such as (3.10) or (3.15). These discrete procedures roughly solve the continuous system, but not with great accuracy. However, upon aptly tuning the parameters which masquerade as the step sizes in the procedures, we can often achieve fast convergence to the equilibrium point of the continuous system, eventually accomplishing the goal of the original basic discrete dynamical system.

4. Control systems for nonlinear equations

The dynamical system (3.11) for linear equations $A\mathbf{x} = \mathbf{b}$, where K is interpreted as a splitting matrix or a preconditioner of A, is merely a special case of a much more general setting. The following approach sets forth a framework from which many new algorithms can be derived.

The notion that many important numerical algorithms can be interpreted via systems and control theory has long been in the minds of researchers. In the seminal book by Tsypkin (1971) and the follow-up volume (Tsypkin 1973), for example, it was advocated that the gradient dynamical systems 'cover many iterative formulas of numerical analysis'. Following the ideas suggested by Bhaya and Kaszkurewicz (2006), we cast the various numerical techniques for finding zero(s) of a given differentiable function

$$\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^n$$

in an input–output control framework with different control strategies. Our point is, again, a comparison of similarities between continuous and discrete dynamical systems.

4.1. Continuous control

Consider the basic model

$$\begin{aligned} \frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} &= \mathbf{u}(t), \\ \mathbf{y}(t) &= -\mathbf{r}(t), \end{aligned} \tag{4.1}$$

where the state variable $\mathbf{x}(t)$ is controlled by $\mathbf{u}(t)$ while the output variable $\mathbf{y}(t)$ is observed from the residue function

$$\mathbf{r}(t) = -\mathbf{g}(\mathbf{x}(t)).$$

Table 4.1. Control strategies and the associated dynamical systems.

$\phi(\mathbf{x},\mathbf{r})$	$\frac{\mathrm{d}V}{\mathrm{d}t}$	$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}$
$\mathbf{g}'(\mathbf{x})^{-1}\mathbf{r}$	$-\Vert\mathbf{r}\Vert_2^2$	$-\mathbf{g}'(\mathbf{x})^{-1}\mathbf{g}(\mathbf{x})$
$\mathbf{g}'(\mathbf{x})^\top\mathbf{r}$	$-\Vert\mathbf{g}'(\mathbf{x})^\top\mathbf{r}\Vert_2^2$	$-\mathbf{g}'(\mathbf{x})^\top\mathbf{g}(\mathbf{x})$
$\mathbf{g}'(\mathbf{x})^{-1}\mathrm{sgn}(\mathbf{r})$	$-\Vert\mathbf{r}\Vert_1$	$-\mathbf{g}'(\mathbf{x})^{-1}\mathrm{sgn}(\mathbf{g}(\mathbf{x}))$
$\mathrm{sgn}(\mathbf{g}'(\mathbf{x})^\top\mathbf{r})$	$-\Vert\mathbf{g}'(\mathbf{x})^\top\mathbf{r}\Vert_1$	$-\mathrm{sgn}(\mathbf{g}'(\mathbf{x})^\top\mathbf{g}(\mathbf{r}))$
$\mathbf{g}'(\mathbf{x})^\top\mathrm{sgn}(\mathbf{r})$	$-\Vert\mathbf{g}'(\mathbf{x})^\top\mathrm{sgn}(\mathbf{r})\Vert_2^2$	$-\mathbf{g}'(\mathbf{x})^\top\mathrm{sgn}(\mathbf{g}(\mathbf{x}))$

One obvious approach is to employ both the state and the output as a feedback to estimate the control strategy, that is,

$$\mathbf{u} = \phi(\mathbf{x},\mathbf{r}), \tag{4.2}$$

based on some properly selected ϕ. Different choices of ϕ can be used to design the control and, hence, lead to various algorithms. Of course, it is often that case that the choice of the control strategy ϕ depends on what cost function $V(\mathbf{x}(t),\mathbf{u}(t))$ is to be optimized. In turn, the cost function often plays the role as a Lyapunov function for the dynamical system. Table 4.1 summarizes just a few possible choices for the control $\mathbf{u}$ and the derivatives of the associated cost functions (Bhaya and Kaszkurewicz 2006). Notably, the first case in the table is the well-known continuous Newton method (Hirsch and Smale 1979, Smale 1977).

It is not difficult to verify that the cost functions are $V(t) = \frac{1}{2}\Vert\mathbf{r}(t)\Vert_2^2$ in the first four cases and $V(t) = \Vert\mathbf{r}(t)\Vert_1$ in the last case, respectively. Be aware of the fact that the vector fields for $\mathbf{x}(t)$ are only piecewise continuous in the last three cases. A discretization of the differential system may not be trivial, which we will draw a distinct line from the discrete control in the next section. trivial, which we will make a clear distinction from the discrete control in the next section. Regardless of the possible non-smoothness in the trajectory $\mathbf{x}(t)$, it is evident that the choice of the control $\mathbf{u}(t)$ always causes the cost function $V(t)$ to decrease in t and, if $\mathbf{g}'(\mathbf{x}(t))$ is always non-singular, the residual function $\mathbf{r}(t)$ converges to zero.

4.2. Discrete control

An Euler analogue of (4.1) is the discrete input-output control system,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u}_k, \tag{4.3}$$

where the control $\mathbf{u}_k$ follows the feedback law,

$$\mathbf{u}_k = \epsilon_k \phi(\mathbf{x}_k, \mathbf{r}_k), \tag{4.4}$$

with $\mathbf{r}_k = -\mathbf{g}(\mathbf{x}_k)$. To estimate the step size ϵ_k, observe the *informal* Taylor series expansion,

$$\mathbf{r}_{k+1} \approx \mathbf{r}_k - \epsilon_k \mathbf{g}'(\mathbf{x}_k)\phi(\mathbf{x}_k, \mathbf{r}_k). \tag{4.5}$$

The step size that best reduce the Euclidean norm of the vector on the right side of (4.5) is given by the expression,

$$\epsilon_k = \frac{\langle \mathbf{g}'(\mathbf{x}_k)\phi(\mathbf{x}_k, \mathbf{r}_k), \mathbf{r}_k \rangle}{\langle \mathbf{g}'(\mathbf{x}_k)\phi(\mathbf{x}_k, \mathbf{r}_k), \mathbf{g}'(\mathbf{x}_k)\phi(\mathbf{x}_k, \mathbf{r}_k) \rangle}. \tag{4.6}$$

We have already seen a special case of (4.6) in (3.13) when the equation $\mathbf{g}(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ is linear and the control $\phi(\mathbf{x}, \mathbf{r}) = K^{-1}\mathbf{r}$ is employed, which is the ORTHOMIN(1) method. Another special case corresponding to the choice of control $\phi(\mathbf{x}_,\mathbf{r}) = \mathbf{g}'(\mathbf{x})^{-1}\mathbf{r}$ leads to $\epsilon_k = 1$, which of course is the classical Newton iteration. Interestingly enough, the various choices of $\phi(\mathbf{x}, \mathbf{r})$ described in Table 4.1 together with the associated ϵ_k defined in (4.6) set forth different zero-finding iterative schemes, some of which are perhaps new. We do not think that all convergence properties of these schemes have been well understood.

Be aware that the approximation in (4.5) is not necessarily true in general. The increment $\mathbf{u}_k$ from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$, for instance, may not be small enough to warrant the expansion of $\mathbf{g}(\mathbf{x}_{k+1})$ at $\mathbf{x}_k$: the approximation is in jeopardy. The step size ϵ_k defined in (4.6) therefore does not necessarily decrease the magnitude of the residual function $\mathbf{r}(\mathbf{x})$. This is precisely the dividing line between a discrete dynamical system which often converges only locally and the continuous dynamical system which converges globally. The well-known convergence behaviour of the classical Newton iteration and the continuous Newton algorithm serves well to exemplify our points: the classical Newton iteration with $\epsilon_k = 1$ does not necessarily give rise to a descent step for the residual function $\mathbf{r}(\mathbf{x})$, whereas the continuous Newton flow always does. The relationship between the convergence rates of iterative and continous processes has recently been studied in Hauser and Nedić (2007).

It is certainly possible to adopt models more sophisticated than (4.1) or (4.3). For example, the two-step scheme

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \epsilon_k \big[\phi(\mathbf{x}_k, \mathbf{r}_k) + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1})\big] \tag{4.7}$$

is analogous to (3.15) and can be converted into a nonlinear conjugate gradient method (Daniel 1967, Savinov 1983, Yabe and Takano 2004). We shall not elaborate on zero-finding algorithms here, but we hope the above discussion has shed some light on how a realization process, either continuous or discrete, can be developed either from or for a dynamical system in the way suggested in Figure 1.1. There seems to be a rich interpretation of the analogy between a discrete scheme and its continuous counterpart. It would be interesting to see whether further consideration along these lines, such as higher-order or multiple-step processes, can develop into new numerical algorithms. Indeed, such a notion has been known as 'higher-order controllers' for given plants in the community of control systems. Some of the theory developed in that discipline might be useful in this regard, and *vice versa* (Bhaya and Kaszkurewicz 2006).

5. Lax dynamical systems and isospectrality

One classical problem of fundamental importance in many critical applications is to find the spectral decomposition,

$$A_0 = U_0 \Lambda_0 U_0^\top, \tag{5.1}$$

of a given real-valued symmetric matrix A_0. In the factorization, U_0 is an orthogonal matrix composed of eigenvectors of A_0 and Λ is the diagonal matrix of the corresponding eigenvalues. Currently, one of the most effective techniques for eigenvalue computation is by an iterative process called the QR algorithm (Golub and Van Loan 1996). The algorithm performs well due to the cooperation of several ingenious components, one of which is the employment of suitable shift strategies that greatly improve the convergence behaviour. Viewing the shifts as feedback control variables, some studies have been made by Helmke and Wirth (2000, 2001) to analyse the controllability of the inverse power method. As far as we know, however, modelling the shift strategies used in a practical QR algorithm by a dynamical system is still an open question. For simplicity, we demonstrate only the basic QR algorithm with no shift.

Recall the fact that any matrix A enjoys the QR decomposition

$$A = QR,$$

where Q is orthogonal and R is upper triangular. The basic QR scheme defines a sequence of matrices $\{A_k\}$ via the recursion (Francis 1961/1962)

$$\begin{cases} A_k &= Q_k R_k, \\ A_{k+1} &= R_k Q_k. \end{cases} \tag{5.2}$$

The iteration implies that

$$A_{k+1} = Q_k^T A_k Q_k, \tag{5.3}$$

showing not only the isospectrality of A_k to A_0, but also the mechanism of orthogonal congruence transformations applied to A_0. It can be proved that the sequence $\{A_k\}$ converges to a diagonal matrix and, hence, the decomposition (5.1) is realized through the iterative scheme (5.2). One is immediately curious why the swapping of Q_k and R_k works in (5.2). Indeed, there is a much deeper theory involved. Referring to the diagram in Figure 1.1, we now identify a differential system to which the QR algorithm corresponds, not as a discrete approximation but rather as a time-1 sampling.

5.1. Isospectral flow

Consider the initial value problem,

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} := [X(t), k_1(X(t))], \quad X(0) := X_0, \tag{5.4}$$

where $k_1 : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is some selected matrix-valued function to be specified later, and

$$[A, B] := AB - BA \tag{5.5}$$

denotes the Lie commutator (bracket) operation between matrices A and B. We shall refer to (5.4) as a general *Lax dynamical system* with the Lax pair (X, k_1). Associated with (5.4), we define two *parameter dynamical systems*:

$$\frac{\mathrm{d}g_1(t)}{\mathrm{d}t} := g_1(t)k_1(X(t)), \quad g_1(0) := I, \tag{5.6}$$

and

$$\frac{\mathrm{d}g_2(t)}{\mathrm{d}t} := k_2(X(t))g_2(t), \quad g_2(0) := I, \tag{5.7}$$

with the property that

$$k_1(X) + k_2(X) = X. \tag{5.8}$$

The following facts are useful but easy to prove, and have been established in an early paper by Chu and Norris (1988).

Theorem 5.1. For any t within the interval of existence, the solutions $X(t)$, $g_1(t)$, and $g_2(t)$ of the systems (5.4), (5.6), and (5.7), respectively, are related to each other by the following three properties.

(1) Similarity property:

$$X(t) = g_1(t)^{-1}X_0g_1(t) = g_2(t)X_0g_2(t)^{-1}. \tag{5.9}$$

(2) Decomposition property:

$$\exp(tX_0) = g_1(t)g_2(t). \tag{5.10}$$

(3) Reversal property:

$$\exp(tX(t)) = g_2(t)g_1(t). \tag{5.11}$$

The implication of Theorem 5.1 is quite remarkable. First, it shows that eigenvalues are invariant. For this reason, $X(t)$ is called an isospectral flow. Secondly, let the product $g_1(t)g_2(t)$ in (5.10) be called the *abstract* g_1g_2 *decomposition* of $\exp(tX_0)$ because at present we do not know the individual structure, if there is any, of the parameter matrices $g_1(t)$ or $g_2(t)$. By setting $t=1$ in both (5.10) and (5.11), we see the relationship

$$\begin{cases} \exp(X(0)) = g_1(1)g_2(1), \\ \exp(X(1)) = g_2(1)g_1(1). \end{cases} \tag{5.12}$$

Since the dynamical system for $X(t)$ is autonomous, it follows that the phenomenon characterized by (5.12) will occur at every integer time within the interval of existence for these initial value problems. Corresponding to the abstract g_1g_2 decomposition, the above iterative process (5.12) for all feasible integers will be called the *abstract* g_1g_2 *algorithm*. It is thus seen that the curious iteration in the QR algorithm is completely generalized and abstracted via the mere splitting (5.8) of the identity map. Choosing a different splitting leads to a different algorithm.

In particular, let any given matrix X be decomposed as

$$X = X^o + X^- + X^+,$$

where X^o, X^-, and X^+ denote the diagonal, the strictly lower triangular, and the strictly upper triangular parts of X, respectively. Define

$$k_1(X) = \Pi_0(X) := X^- - X^{-\top}. \tag{5.13}$$

The resulting Lax dynamical system,

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = [X(t), \Pi_0(X(t))], \quad X(0) = X_0, \tag{5.14}$$

is known as the *Toda lattice* (though initially the lattice is referred to only in the case when X_0 is symmetric and tridiagonal). It is important to note that the matrix $k_1(X(t))$ in the Toda lattice is skew-symmetric and thus $g_1(X(t))$ is orthogonal for all t. Furthermore, $k_2(X(t))$ is upper triangular and thus so is $g_2(X(t))$. In other words, the abstract g_1g_2 decomposition of $\exp(X)$ is precisely the QR decomposition of $\exp(X)$. It follows that the sequence $\{X(k)\}$ by sampling the solution of the Toda flow (5.14) at integer times gives rise to exactly the same iterates as the QR algorithm (5.2) applied to the matrix $A_0 = \exp(X_0)$.

The connection between the QR algorithm and the Toda lattice was first discovered by Symes (1981/82) when studying the asymptotic behaviour of momenta of particles in a non-periodic Toda lattice. The same relationship was found later to be also closely related to the quotient-difference algorithm developed much earlier by Rutishauser (1954).

In contrast to the association between a discrete system and a continuous system described earlier in Sections 3 and 4, which perhaps can be best characterized as 'mimicry', the correspondence between the QR algorithm and the Toda lattice exhibits a new type of involvement, namely, the result of an iterative scheme is entirely 'embedded' in the solution curve of a continuous dynamical system or, equivalently, the solution curve of a differential equation smoothly 'interpolates' all points generated by a discrete dynamical system. Because of this close relationship, the evolution of $X(t)$, which starts from a symmetric initial value X_0 and converges isospectrally to a limit point which is a diagonal matrix, can almost be expected without the need for any extra inculcation in the classical theory of the QR algorithm, and *vice versa* (Deift, Nanda and Tomei 1983).

It is important to point out that, strictly speaking, the QR algorithm applied to a non-symmetric matrix A_0 with complex eigenvalues does not converge to any fixed limit point at all in the conventional mathematical sense. The iterates from the QR algorithm only *pseudo-converge* to a block upper triangular form with at most 1×1 or 2×2 blocks along the main diagonal. Such a structure is a necessity when dealing with complex-conjugate eigenvalues of a real-valued matrix by real arithmetic. For later reference, we shall refer to any matrix with this kind of structure as an *upper quasi-triangular matrix.* We stress again that the QR algorithm (and many other algorithms) produces only this 'form', but not any fixed matrix, in its limiting behaviour.

Likewise, the Toda flow applied to a non-symmetric matrix X_0 does not have any asymptotically stable equilibrium point in general. Rather, the flow converges to an upper quasi-triangular form where each of the 2×2 blocks actually represents an ω-limit cycle. Now that we know the Toda flow interpolates the iterates of the QR algorithm, the limit cycle behaviour of the Toda flow offers a nice theoretical explanation of the pseudo-convergence behaviour of the QR algorithm. Without causing ambiguity, we shall henceforth refer to such limiting behaviour as 'convergence to an upper quasi-triangular matrix'.

5.2. Complete integrability

The Lax dynamical system (5.4) actually arises in a much broader area of applications. Consider the one-dimensional Korteweg–de Vries (KdV) equation,

$$\frac{\partial u}{\partial t} + 6u\frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0, \tag{5.15}$$

for $u = u(x,t)$. It is a classical result that the KdV equation is completely integrable in the sense there are infinitely many conserved quantities or constants of motion. Lax (1968) proved that the KdV equation is precisely the *compatibility condition*

$$\frac{\mathrm{d}L}{\mathrm{d}t} = [B, L], \tag{5.16}$$

for the pair of differential operators

$$L\psi := \frac{\partial^2\psi}{\partial x^2} + u\psi, \tag{5.17}$$

$$B\psi := -4\frac{\partial^3\psi}{\partial x^3} - 6u\frac{\partial\psi}{\partial x} - 3\frac{\partial u}{\partial x}\psi. \tag{5.18}$$

In other words, by recognizing the fact that

$$\left[\frac{\partial}{\partial x}, x\right]\psi = \frac{\partial(x\psi)}{\partial x} - x\frac{\partial\psi}{\partial x} = \psi$$

as the identity of differential operator,

$$\left[\frac{\partial}{\partial x}, x\right] = \mathrm{id},$$

the equation (5.16) holds if and only if u satisfies (5.15). The eigenvalues $\lambda \in \mathbb{R}$ of the one-dimensional Schrödinger equation,

$$L\psi = \lambda\psi, \tag{5.19}$$

$$\frac{\partial\psi}{\partial t} = B\psi, \tag{5.20}$$

for the wave function $\psi = \psi(x, t; \lambda)$ with $u(x, t)$ as the potential constitute precisely the integrals of the KdV equation. The second equation, (5.20), characterizes how the wave function evolves in time. The pair of operators (L, B) is referred to as a *Lax pair.*

Under the assumption that λ is invariant over t, note that the two equations (5.19) and (5.20) are sufficient to imply the compatibility condition (5.16) when acting on the eigenfunction ψ of the operator L. This is true regardless of how the operators L and B are defined. In terms of the notation adopted in our preceding section, we may interpret the Lax pair as $(X, k_2(X))$, where

$$\frac{\mathrm{d}X}{\mathrm{d}t} = [k_2(X), X], \tag{5.21}$$

$$\frac{\mathrm{d}\psi}{\mathrm{d}t} = k_2(X)\psi, \tag{5.22}$$

and $\psi(t)$ tells us how the eigenvector corresponding to the invariant eigenvalue λ varies in time.

We have seen that sampling the solution flow $X(t)$ at integer times gives rise to an iterative scheme, such as the *QR* algorithm. The question now is whether an effective discretization can be derived to handle the integration of equation (5.21) directly.

It has to be pointed out that a central theme in the game of engaging dynamical systems such as (5.21) is to maintain isospectrality. Nonetheless, Calvo, Iserles and Zanna (1997) proved that most of the conventional numerical ODE methods, in particular the multi-step and the Runge–Kutta schemes, simply cannot preserve isospectral flows. One remedy is to perform numerical integration over one of the parameter dynamical systems (5.6) or (5.7) and then employ the similarity property (5.9) to reclaim $X(t)$. Solving the parameter dynamical system still requires the preservation of some structures, but can be handled more easily. In the case of (5.14), for example, the flow $g_1(X(t))$ of orthogonal matrices can be tracked by orthogonal integrators developed by Dieci, Russell and Van Vleck (1994). Approaches such as this follow the paradigm of discretization from the numerical analysis perspective. We want to emphasize that there is more beyond this traditional way of thinking. The Toda lattice itself has more structure, so that a completely different perspective of discretization could be, and should be, taken into account.

Two separate but related approaches that suggest integrable discretization of the Toda lattice (for symmetric and tridiagonal matrices) are outlined in Sections 5.3 and 5.4. We shall present the theory in these two sections, but refrain from discussing the actual implementation, since eigenvalue computation is a well-developed subject. Even so, the facts we are about to introduce, namely, that the solution to the Toda lattice and, hence, the iterates generated by the QR algorithm can be represented in 'closed form', strongly suggest that an appropriate discretization can make the computation very effective. In Section 6, we will have a chance to exploit these ideas further, and describe in detail an integrable discretization for the more complicated singular value decomposition.

5.3. Orthogonal polynomials, moments and measure deformation

The first approach makes an interesting connection between orthogonal polynomials and the solution of (5.14) when X_0 is tridiagonal, which sheds light on the notion of integrable discretization. In particular, we shall represent the solution to the Toda lattice in terms of moments associated with a specific measure.

Recall that a set of orthogonal polynomials $\{p_k(x)\}$ defined by a positive measure $\mu(x)$ over $\mathbb{R}$, that is,

$$\int p_k(x)p_\ell(x)\,\mathrm{d}\mu(x) = \delta_{k,\ell}, \quad k,\ell = 0,1,\ldots,$$

always satisfies a three-term recurrence relationship,

$$xp_k(x) = a_k p_{k+1}(x) + b_k p_k(x) + a_{k-1}p_{k-1}(x), \quad k = 1,2,\ldots, \tag{5.23}$$

with $p_{-1}(x) \equiv 0$ and $p_0(x) \equiv 1$. This recurrence can be neatly written in a semi-infinite matrix form:

$$\underbrace{\begin{bmatrix} b_0 & a_0 & 0 & & & \\ a_0 & b_1 & a_1 & 0 & & \\ 0 & a_1 & b_2 & a_2 & 0 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}}_{J} \begin{bmatrix} p_0(x) \\ p_1(x) \\ p_2(x) \\ \vdots \end{bmatrix} = x \begin{bmatrix} p_0(x) \\ p_1(x) \\ p_2(x) \\ \vdots \end{bmatrix}. \qquad (5.24)$$

Indeed, there is a one-to-one correspondence between the measure μ and the coefficient matrix J defined above (Akhiezer 1965, Aptekarev, Branquinho and Marcellán 1997). This is closely related to the classical moment problem. Let the moments corresponding to μ be denoted by

$$s_j := \int x^j \, \mathrm{d}\mu(x), \quad j = 0, 1, \ldots. \qquad (5.25)$$

Define further the so-called *Hankel determinants*,

$$H_k := \det \begin{bmatrix} s_0 & s_1 & \ldots & s_{k-1} \\ s_1 & s_2 & & s_k \\ \vdots & & & \vdots \\ s_{k-1} & s_k & \ldots & s_{2k-2} \end{bmatrix}. \qquad (5.26)$$

It is known that the monic orthogonal polynomials $\{\tilde{p}_k(x)\}$ associated with $\{p_k(x)\}$ are given by (Akhiezer 1965, Szegő 1975)

$$\tilde{p}_k(x) = \frac{1}{H_k} \det \begin{bmatrix} s_0 & s_1 & \ldots & s_k \\ s_1 & s_2 & & s_{k+1} \\ \vdots & & & \vdots \\ s_{k-1} & s_k & \ldots & s_{2k-1} \\ 1 & x & \ldots & x^k \end{bmatrix}. \qquad (5.27)$$

If we write $\tilde{p}_k(x)$ as

$$\tilde{p}_k(x) = x^k + c_1^{(k)} x^{k-1} + \ldots + c_{k-1}^{(k)} x + c_k^{(k)},$$

then its coefficients are given by

$$c_j^{(k)} = \frac{(-1)^j}{H_k} \det \begin{bmatrix} s_0 & \ldots & s_{k-j-1} & s_{k-j+1} & \ldots & s_k \\ s_1 & & & & & s_{k+1} \\ \vdots & & \vdots & \vdots & & \vdots \\ s_{k-1} & \ldots & s_{2k-j-2} & s_{2k-j} & \ldots & s_{2k-1} \end{bmatrix}. \qquad (5.28)$$

Corresponding to (5.23), the recurrence relation for $\{\tilde{p}_k(x)\}$ becomes

$$x\tilde{p}_k(x) = \tilde{p}_{k+1} + b_k \tilde{p}_k(x) + a_{k-1}^2 \tilde{p}_{k-1}(x). \qquad (5.29)$$

By comparing the corresponding coefficients, we conclude that

$$a_k^2 = \frac{H_k H_{k+2}}{H_{k+1}^2}, \tag{5.30}$$

$$b_k = c_1^{(k)} - c_1^{(k+1)}. \tag{5.31}$$

This is a classical result connecting the measure $\mu(x)$, the moments $s_j(x)$ and the orthogonal polynomials $p_k(x)$.

Suppose now that the coefficients in J are time-dependent. Then the corresponding measure μ is also time-dependent. Finding the relationship

$$J(t) \leftrightarrow \mu(x;t)$$

allows us to write the coefficients of the orthogonal polynomials $\{p_k(x;t)\}$ in terms of the corresponding moments $s_j(t)$. In general, this is a fairly difficult task. Only very few cases are known to have exact solutions, among which one is the $J(t)$ associated with the Toda lattice (Aptekarev *et al.* 1997).

The relationship is most conspicuous in the semi-infinite Toda lattice. By identifying (the symmetric and tridiagonal matrix) $X(t)$ with the tridiagonal matrix J in (5.24), the entries in the differential system (5.14) can be expressed as the system

$$\frac{\mathrm{d}a_k}{\mathrm{d}t} = a_k(b_{k+1} - b_k), \tag{5.32}$$

$$\frac{\mathrm{d}b_k}{\mathrm{d}t} = 2(a_k^2 - a_{k-1}^2), \tag{5.33}$$

with $a_{-1} \equiv 0$. This differential system characterizes how $X(t)$ or, equivalently, the family of polynomials varies in time. We just need a measure that can ensure the orthogonality of these polynomials. It turns out that the corresponding one-parameter deformation of the measure that can introduce the desirable orthogonality has been shown by Moser (1975) to be

$$\mathrm{d}\mu(x;t) := \mathrm{e}^{tx}\,\mathrm{d}\mu(x;0). \tag{5.34}$$

Equipped with this measure, we can easily calculate the solution to the Toda lattice. That is, the entries $a_k(t)$ and $b_k(t)$ of $X(t)$ can be calculated via (5.30) and (5.31), once the moments given by the integrals (5.25) are computed. In fact, note that with this measure (5.34) we even enjoy the recursion

$$\frac{\mathrm{d}s_\ell}{\mathrm{d}t} = s_{\ell+1}, \quad \ell = 0, 1, \ldots. \tag{5.35}$$

Since these moments are computable in analytic form, we may say that the solution to the Toda lattice (of symmetric and tridiagonal matrices) and hence the iterates by the QR algorithm are now characterized in closed form.

In this sense, we have obtained a discretization while maintaining complete integrability.

It is informative to depict the relationship just described for the Toda lattice as solid lines in Figure 5.1. We stress that the commuting diagram composed of the top four boxes holds in general. That is, the coefficients of the orthogonal polynomials corresponding to a given measure can be

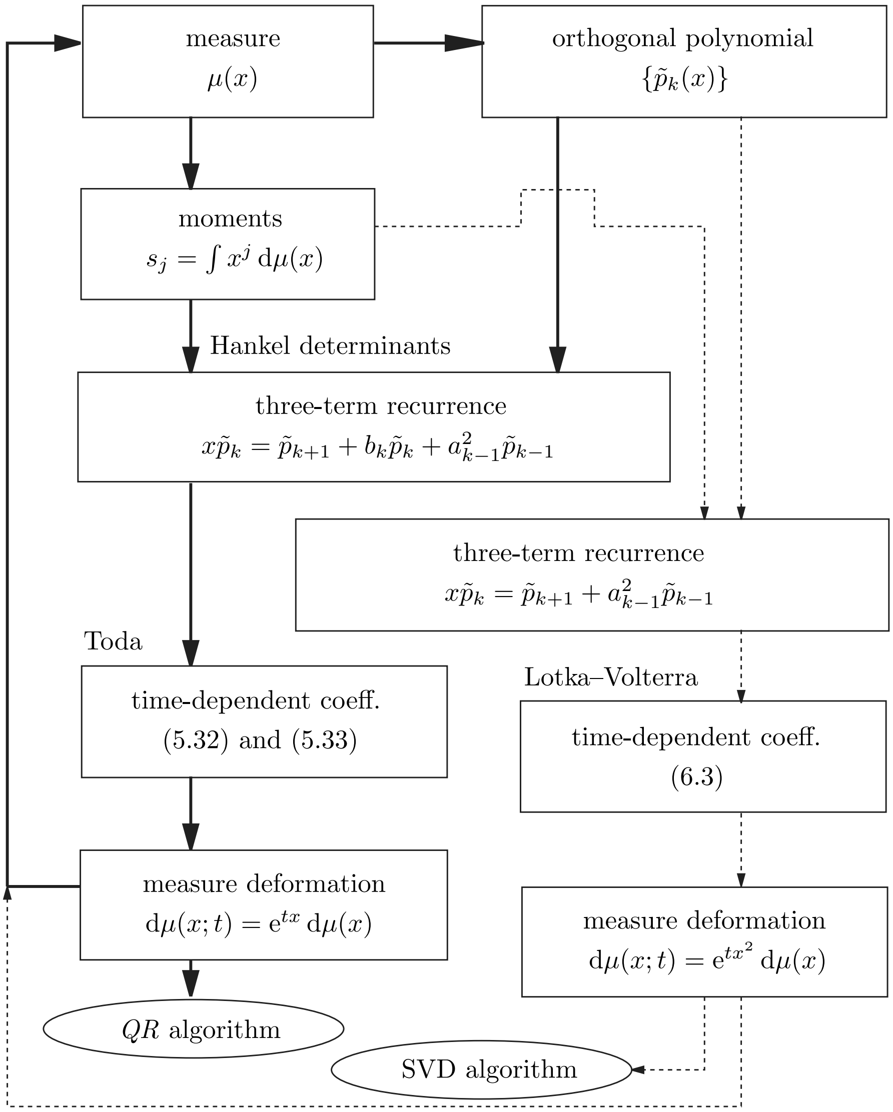
measure
$\mu(x)$
orthogonal polynomial
$\{\tilde{p}_k(x)\}$
moments
$s_j = \int x^j \, \mathrm{d}\mu(x)$
Hankel determinants
three-term recurrence
$x\tilde{p}_k = \tilde{p}_{k+1} + b_k\tilde{p}_k + a_{k-1}^2\tilde{p}_{k-1}$
three-term recurrence
$x\tilde{p}_k = \tilde{p}_{k+1} + a_{k-1}^2\tilde{p}_{k-1}$
Toda
time-dependent coeff.
(5.32) and (5.33)
Lotka–Volterra
time-dependent coeff.
(6.3)
measure deformation
$\mathrm{d}\mu(x;t) = \mathrm{e}^{tx}\,\mathrm{d}\mu(x)$
measure deformation
$\mathrm{d}\mu(x;t) = \mathrm{e}^{tx^2}\,\mathrm{d}\mu(x)$
QR algorithm
SVD algorithm

Figure 5.1. Integrable discretization of Toda lattice (solid line) and Lotka–Volterra equation (dashed line) via Hankel determinants.

expressed in terms of Hankel determinants of the corresponding moments. For the Toda lattice where the coefficients $\{a_k\}$ and $\{b_k\}$ are governed by the differential system (5.32) and (5.33), respectively, the commuting diagram comprised of the left five boxes indicates how the inverse problem is solved. An efficient calculation of the Hankel determinants is all we need for an effective eigenvalue computation. This *modus operandi* is very different from the orthogonal integrator approach mentioned earlier.

We mention in passing that a similar relationship also holds for the singular value decomposition. Specifically, there is a dynamical system whose solution is related to the singular value decomposition (for bidiagonal matrices) in the same way as the Toda lattice to the QR decomposition (for symmetric and tridiagonal matrices). This dynamical system, known as the Lotka–Volterra equation, will be defined in Section 6. In analogy to the Toda lattice, the solution to the Lotka–Volterra equation can be expressed in terms of moments and Hankel determinants associated with a special measure,

$$\mathrm{d}\mu(x;t) = \mathrm{e}^{tx^2}\,\mathrm{d}\mu(x). \tag{5.36}$$

For completion and comparison, such a relationship depicted as dotted lines is also included in Figure 5.1, but we shall omit the details here. Readers are referred to the paper by Nakamura (2004) for an overview of this subject. The book by Nakamura (2006) contains many more details and interesting historical notes, but is written (for now) in Japanese.

In most linear algebra applications, we are perhaps more interested in a finite-dimensional matrix. This can be done by truncating the infinite-dimensional coefficient matrix J into an $n \times n$ matrix L. Then (5.24) is reduced to the equation

$$\underbrace{\begin{bmatrix} b_0 & a_0 & 0 & & & \\ a_0 & b_1 & a_1 & 0 & & \\ 0 & a_1 & b_2 & a_2 & 0 & \\ & & \ddots & \ddots & \ddots & \\ & & & & & a_{n-2} \\ & & & 0 & a_{n-2} & b_{n-1} \end{bmatrix}}_{L} \begin{bmatrix} p_0(x) \\ p_1(x) \\ p_2(x) \\ \vdots \\ \\ p_{n-1}(x) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \\ a_{n-1}p_n(x) \end{bmatrix} = x \begin{bmatrix} p_0(x) \\ p_1(x) \\ p_2(x) \\ \vdots \\ \\ p_{n-1}(x) \end{bmatrix}.$$

Clearly, λ is a root of the polynomial $p_n(x)$ if and only if λ is an eigenvalue of the finite-dimensional tridiagonal matrix L. Other than this requirement of special values for λ, this finite-dimensional eigenvalue problem remains a segment of the semi-infinite system (5.24). As far as the evolution of the entries of L is concerned, it is the same as those of J as long as λ is time-invariant. This isospectrality is precisely what is entailed in the one-dimensional Schrödinger equation (5.19). Under the condition of isospectrality throughout the evolution, the theory developed above for the semi-infinite Toda lattice remains applicable to the finite-dimensional eigenvalue problem. In particular, the solution to the finite-dimensional Toda lattice can still be represented in terms of moments.

5.4. Tau functions and determinantal solution

The second approach utilizes the notion of τ functions originally introduced by the 'Kyoto school' as a central element in the description of the soliton theory for the Kadomtsev–Petviashvili or Hirota–Miwa hierarchies (Date, Kashiwara, Jimbo and Miwa 1983, Hirota, Tsujimoto and Imai 1993, Pöppe 1989). We limit our attention in this section to the basic idea applied to the Toda lattice only.

With the change of variable,

$$c_k(t) := a_k^2\left(\frac{t}{2}\right), \tag{5.37}$$

the off-diagonal entries in the Toda lattice (5.32) can be expressed as a second-order but self-contained equation,

$$\frac{\mathrm{d}^2 \ln c_k}{\mathrm{d}t^2} = c_{k+1} - 2c_k + c_{k-1}. \tag{5.38}$$

If we impose another sequence of new variables $\{\tau_k(t)\}$ implicitly via the relationship

$$c_k = \frac{\tau_{k+1}\tau_{k-1}}{\tau_k^2}, \tag{5.39}$$

then naturally we have

$$\ln c_k = \ln \tau_{k+1} - 2\ln \tau_k + \ln \tau_{k-1}. \tag{5.40}$$

Upon comparison of (5.38) and (5.40), a compatibility condition is that

$$c_k = \frac{\mathrm{d}^2 \ln \tau_k}{\mathrm{d}t^2} \tag{5.41}$$

or, equivalently, that $\{\tau_k\}$ must satisfy the Hirota bilinear form

$$\tau_k \frac{\mathrm{d}^2 \tau_k}{\mathrm{d}t^2} - \left(\frac{\mathrm{d}\tau_k}{\mathrm{d}t}\right)^2 = \tau_{k-1}\tau_{k+1}, \tag{5.42}$$

with $\tau_0 \equiv 1$. The bilinear form (5.42) is sufficient for generating a sequence $\{\tau_k(t)\}$ of solution recursively. For example, starting with an arbitrary initial value $\tau_1(t) = \phi(t)$ that is infinitely differentiable, we obtain

$$\begin{aligned}\tau_2(t) &= \phi\frac{\mathrm{d}^2\phi}{\mathrm{d}t^2} - \left(\frac{\mathrm{d}\phi}{\mathrm{d}t}\right)^2,\\ \tau_3(t) &= -\left(\frac{\mathrm{d}^2\phi}{\mathrm{d}t^2}\right)^3 + \phi\left(\frac{\mathrm{d}^2\phi}{\mathrm{d}t^2}\right)\frac{\mathrm{d}^4\phi}{\mathrm{d}t^4} - \left(\frac{\mathrm{d}\phi}{\mathrm{d}t}\right)^2\frac{\mathrm{d}^4\phi}{\mathrm{d}t^4}\\ &\quad + 2\left(\frac{\mathrm{d}\phi}{\mathrm{d}t}\right)\left(\frac{\mathrm{d}^2\phi}{\mathrm{d}t^2}\right)\frac{\mathrm{d}^3\phi}{\mathrm{d}t^3} - \phi\left(\frac{\mathrm{d}^3\phi}{\mathrm{d}t^3}\right)^2,\end{aligned}$$

and so on. Obviously, the expression for $\tau_k(t)$ becomes more and more involved when k gets higher. The beauty of the τ functions is that there is a much better representation for $\tau_k(t)$ in general.

From a given $\phi(t)$, define the Hankel determinant $\hat{H}_k(t)$ by

$$\hat{H}_k(t) := \det\begin{bmatrix} \phi & \phi^{(1)} & \dots & \phi^{(k-1)} \\ \phi^{(1)} & \phi^{(2)} & & \phi^{(k)} \\ \vdots & & & \vdots \\ \phi^{(k-1)} & \phi^{(k)} & \dots & \phi^{(2k-2)} \end{bmatrix}, \tag{5.43}$$

where for simplicity we adopt the abbreviation

$$\phi^{(\ell)} = \frac{\mathrm{d}^\ell\phi}{\mathrm{d}t^\ell}, \quad \ell = 1, 2, \dots.$$

Let $\hat{H}_k\left[\begin{smallmatrix} i \\ j \end{smallmatrix}\right]$ denote the determinant of the submatrix by deleting the ith row and the jth column from the matrix defining $\hat{H}_k$. Observe that

$$\frac{\mathrm{d}\hat{H}_k}{\mathrm{d}t} = \hat{H}_{k+1}\begin{bmatrix} k+1 \\ k \end{bmatrix}, \tag{5.44}$$

$$\frac{\mathrm{d}^2\hat{H}_k}{\mathrm{d}t^2} = \hat{H}_{k+1}\begin{bmatrix} k \\ k \end{bmatrix}. \tag{5.45}$$

On the other hand, recall the Sylvester determinant identity (Horn and Johnson 1990)

$$\hat{H}_{k+1}\hat{H}_{k-1} = \det\begin{bmatrix} \hat{H}_{k+1}\begin{bmatrix} k+1 \\ k+1 \end{bmatrix} & \hat{H}_{k+1}\begin{bmatrix} k+1 \\ k \end{bmatrix} \\ \hat{H}_{k+1}\begin{bmatrix} k \\ k+1 \end{bmatrix} & \hat{H}_{k+1}\begin{bmatrix} k \\ k \end{bmatrix} \end{bmatrix}. \tag{5.46}$$

In conclusion, we see that $\hat{H}_k(t)$ satisfies precisely the differential equation (5.42). As a consequence, we have obtained a closed form solution for $c_k(t)$

via (5.39), where $\tau_k(t)$ is given by

$$\tau_k(t) = \det \begin{bmatrix} \phi & \phi^{(1)} & \dots & \phi^{(k-1)} \\ \phi^{(1)} & \phi^{(2)} & & \phi^{(k)} \\ \vdots & & & \vdots \\ \phi^{(k-1)} & \phi^{(k)} & \dots & \phi^{(2k-2)} \end{bmatrix}. \tag{5.47}$$

The existence of a determinantal solution to the Toda lattice provides insightful information for the discretization of integrable systems (Iwasaki and Nakamura 2006). With appropriate discretization, for example, it can be shown that the above formula leads to the Rutishauser *qd* algorithm (Nakamura 2004, Rutishauser 1954). Instead of detailing here how this can be done for the eigenvalue computation, which is a well-studied subject, we shall demonstrate a similar application to the much more sophisticated singular value decomposition in the next section.

6. Lotka–Volterra equation and singular values

Given a rectangular matrix $A_0 \in \mathbb{R}^{m\times n}$ with $m \geq n$, the singular value decomposition (SVD) of A_0 is a factorization of the form

$$A_0 = U_0 \Sigma_0 V_0^{\top}, \tag{6.1}$$

where $U_0 \in \mathbb{R}^{m\times m}$ and $V_0 \in \mathbb{R}^{n\times n}$ are unitary matrices and $\Sigma_0 \in \mathbb{R}^{m\times n}$ is a diagonal matrix with non-negative diagonal entries. The notion of SVD has been a powerful tool for matrix analysis and has been a centrepiece in many areas of applications (Golub and Van Loan 1996, Horn and Johnson 1990).

The use of the SVD and associated ideas has a rich history. In the interesting treatise of Stewart (1993), the early history of the SVD was traced back to Beltrami in 1873 and Jordan in 1874. Before high-speed digital computers became available, the SVD could only be approximated (Chu and Funderlic 2002, Horst 1965). Today, there are a number of highly efficient ways to compute the SVD (Demmel, Gu, Eisenstat, Slapničar, Veselić and Drmač 1999). Some are perhaps more polished and possibly more accurate than others (Demmel and Kahan 1990). In this section, we consider only the basic and conventional approach proposed by Golub and Kahan (1965).

A standard practice in the SVD computation consists of two phases. First, two orthogonal matrices P_1 and Q_1 are found such that $B_0 = P_1^{\top} A_0 Q_1$ is in bidiagonal form. This step of reduction can be done directly. Then an iterative procedure is employed to compute the SVD of B_0. This main step of iteration is mathematically equivalent to the QR algorithm applied to the tridiagonal matrix $B_0^{\top} B_0$, except that the product $B_0^{\top} B_0$ is never formed explicitly. Needless to say, extra tactics, such as implicit-shift, could be added to the iterative process to increase efficiency in computation.

6.1. SVD flow

In view of how the Toda lattice is related to the QR algorithm, Chu (1986*b*) proposed a peculiar continuous dynamical system of the form

$$\frac{\mathrm{d}B}{\mathrm{d}t} = B\Pi_0(B^\top B) - \Pi_0(BB^\top)B, \quad B(0) = B_0, \tag{6.2}$$

where Π_0 is the operator defined in (5.13), and proved that the sequence $\{B(\ell)\}$ produced by $B(t)$ corresponds to the iterates produced by the Golub–Kahan SVD algorithm. One special feature of (6.2) is that $B(t)$ stays bidiagonal for all t. What other properties of this SVD flow can we exploit for applications?

Without loss of generality, we shall assume henceforth that B_0 is an $n \times n$ matrix. By denoting

$$B(t) := \operatorname{diag}\left\{ \begin{array}{ccccccccc} & b_2(t) & & \cdots & & b_{2n-2}(t) & \\ b_1(t) & & b_3(t) & & \cdots & & b_{2n-1}(t) \end{array} \right\},$$

and defining

$$u_{2k-1}(t) := b_{2k-1}^2\left(\frac{t}{2}\right),$$
$$u_{2k}(t) := b_{2k}^2\left(\frac{t}{2}\right),$$

the differential system (6.2) can be condensed into the expression

$$\frac{\mathrm{d}u_k}{\mathrm{d}t} = u_k(u_{k+1} - u_{k-1}), \quad k = 1, 2, \ldots, 2n-1, \tag{6.3}$$

with $u_0(t) \equiv 0$ and $u_{2n}(t) \equiv 0$, which is known as the *continuous-time finite Lotka–Volterra equation.*

The dynamical system (6.3) is Hamiltonian, that is, it can be written in the form of Hamilton's equations (Deift, Demmel, Li and Tomei 1991). The system is also integrable and enjoys a determinantal solution which can be derived from the theory of τ functions as follows.

Define a change of variable by

$$u_k = \frac{\tau_{k+2}\tau_{k-1}}{\tau_{k+1}\tau_k}. \tag{6.4}$$

Clearly, we have

$$\frac{\mathrm{d}\ln u_k}{\mathrm{d}t} = \frac{\mathrm{d}}{\mathrm{d}t}\ln\frac{\tau_{k+2}}{\tau_{k+1}} - \frac{\mathrm{d}}{\mathrm{d}t}\ln\frac{\tau_k}{\tau_{k-1}}. \tag{6.5}$$

A comparison between (6.3) and (6.5) suggests that a compatibility condition could be

$$\frac{\tau_{k+2}\tau_{k-1}}{\tau_{k+1}\tau_k} = \frac{\mathrm{d}}{\mathrm{d}t}\ln\frac{\tau_{k+1}}{\tau_k}, \tag{6.6}$$

which is equivalent to

$$\frac{\mathrm{d}\tau_k}{\mathrm{d}t}\tau_{k+1} - \tau_k \frac{\mathrm{d}\tau_{k+1}}{\mathrm{d}t} + \tau_{k-1}\tau_{k+2} = 0. \tag{6.7}$$

The differential equation (6.7) can be used to generate $\tau_k(t)$ recursively. Assuming starting values $\tau_{-1} \equiv 0$, $\tau_0 \equiv 1$, $\tau_1(t) = 1$ and $\tau_2(t) = \psi(t)$, we obtain from (6.7)

$$\begin{aligned} \tau_3 &= \frac{\mathrm{d}\psi}{\mathrm{d}t}, \\ \tau_4 &= \det \begin{bmatrix} \psi & \psi^{(1)} \\ \psi^{(1)} & \psi^{(2)} \end{bmatrix}, \end{aligned}$$

and in general it can be proved that (Tsujimoto 1995)

$$\tau_{2k-1} = \overline{H}_{k-1,1}, \tag{6.8}$$

$$\tau_{2k} = \overline{H}_{k,0}, \tag{6.9}$$

where

$$\overline{H}_{k,j}(t) := \det \begin{bmatrix} \psi^{(j)} & \psi^{(j+1)} & \dots & \psi^{(j+k-1)} \\ \psi^{(j+1)} & \psi^{(j+2)} & \dots & \psi^{(j+k)} \\ \vdots & \vdots & & \vdots \\ \psi^{(j+k-1)} & \psi^{(j+k)} & & \psi^{(j+2k-2)} \end{bmatrix}, \quad j = 0 \text{ or } 1, \tag{6.10}$$

is the determinant of a $k \times k$ Hankel matrix and

$$\overline{H}_{-1,j}(t) \equiv 0, \quad \overline{H}_{0,j}(t) \equiv 1, \quad \overline{H}_{n+1,j}(t) \equiv 0. \tag{6.11}$$

The general solution to the Lotka–Volterra equation, therefore, is given by the formula (Tsujimoto, Nakamura and Iwasaki 2001)

$$u_{2k-1}(t) = \frac{\overline{H}_{k,1}(t)\overline{H}_{k-1,0}(t)}{\overline{H}_{k,0}(t)\overline{H}_{k-1,1}(t)}, \tag{6.12}$$

$$u_{2k}(t) = \frac{\overline{H}_{k+1,0}(t)\overline{H}_{k-1,1}(t)}{\overline{H}_{k,1}(t)\overline{H}_{k,0}(t)}, \quad k = 1, 2, \dots, n, \tag{6.13}$$

By assuming that all the derivatives of ψ are obtainable from elementary calculus, it is true in principle that all these Hankel determinants can be calculated algebraically. Since all quantities involved in (6.12) and (6.13) are now in the analytic form, we may say that the SVD flow and, hence, the iterates from the SVD algorithm are representable in closed form.

This determinantal solution for the continuous Lotka–Volterra equation can be utilized to effectuate numerical computation. Indeed, it motivates the notion of integrable discretization of (6.3), which we consider in the next section.

6.2. Integrable discretization

A key step in the integrable discretization of the Lotka–Volterra equation (6.3) is a particular Euler-type scheme of the form (Hirota *et al.* 1993)

$$u_k^{[\ell+1]} = u_k^{[\ell]} + \delta\big(u_k^{[\ell]} u_{k+1}^{[\ell]} - u_k^{[\ell+1]} u_{k-1}^{[\ell+1]}\big), \tag{6.14}$$

where $u_k^{[\ell]}$ represents the approximation solution of $u_k(t)$ at $t = \ell\delta$ with boundary conditions $u_0^{[\ell]} \equiv 0$ and $u_{2n}^{[\ell]} \equiv 0$ for all ℓ. Be aware of the notation that the superscript ${}^{[\ell+1]}$ in brackets indicates the advance in time by a step of size δ whereas the subscript ${}_{k+1}$ indicates the $(k+1)$th bidiagonal entry of the matrix $B(t)$.

In hindsight, the scheme (6.14) appears to be simply a mixture of both explicit and implicit Euler methods. The fact of the matter is that it takes considerable insight to get the right combination so that, as in the continuous case, the discrete Lotka–Volterra equation (6.14) still enjoys a determinantal solution. Specifically, we claim without proof that the solution to the finite difference equation (6.14) is given by (Iwasaki and Nakamura 2002)

$$u_{2k-1}^{[\ell]} = \frac{\widetilde{H}_{k,1}^{[\ell]} \widetilde{H}_{k-1,0}^{[\ell+1]}}{\widetilde{H}_{k,0}^{[\ell]} \widetilde{H}_{k-1,1}^{[\ell+1]}}, \tag{6.15}$$

$$u_{2k}^{[\ell]} = \frac{\widetilde{H}_{k+1,0}^{[\ell]} \widetilde{H}_{k-1,1}^{[\ell+1]}}{\widetilde{H}_{k,1}^{[\ell]} \widetilde{H}_{k,0}^{[\ell+1]}}, \quad k = 1, 2, \ldots, n, \tag{6.16}$$

where $\widetilde{H}_{k,j}^{[\ell]}$ is the Hankel determinant defined by

$$\widetilde{H}_{k,j}^{[\ell]} = \det \begin{bmatrix} \widetilde{\psi}_j^{[\ell]} & \widetilde{\psi}_j^{[\ell+1]} & \cdots & \widetilde{\psi}_j^{[\ell+k-1]} \\ \widetilde{\psi}_j^{[\ell+1]} & \widetilde{\psi}_j^{[\ell+2]} & \cdots & \widetilde{\psi}_j^{[\ell+k]} \\ \vdots & \vdots & & \vdots \\ \widetilde{\psi}_j^{[\ell+k-1]} & \widetilde{\psi}_j^{[\ell+k]} & & \widetilde{\psi}_j^{[\ell+2k-2]} \end{bmatrix}, \quad j = 0 \text{ or } 1, \tag{6.17}$$

with boundary conditions

$$\widetilde{H}_{-1,j}^{[\ell]} \equiv 0, \quad \widetilde{H}_{0,j}^{[\ell]} \equiv 1, \quad \widetilde{H}_{n+1,j}^{[\ell]} \equiv 0, \tag{6.18}$$

in which $\{\widetilde{\psi}_0^{[\ell]}\}$ is a given initial sequence and $\widetilde{\psi}_1^{[\ell]}$ is the quotient difference defined by

$$\widetilde{\psi}_1^{[\ell]} := \frac{\widetilde{\psi}_0^{[\ell+1]} - \widetilde{\psi}_0^{[\ell]}}{\delta}. \tag{6.19}$$

The knowledge of a solution $u_k^{[\ell]}$ in the form of (6.15) and (6.16) enables us to gain considerable insight into its asymptotic behaviour as ℓ goes to infinity. We shall skip that part of discussion in this paper, but rather pay more attention to a possible numerical implementation for the remainder of this section.

We modify (6.14) to the more general variable-step scheme

$$u_k^{[\ell+1]}\big(1+\delta^{[\ell+1]}u_{k-1}^{[\ell+1]}\big) = u_k^{[\ell]}\big(1+\delta^{[\ell]}u_{k+1}^{[\ell]}\big), \tag{6.20}$$

referred to hereafter as the *vdLV scheme*. In a series of extensive studies (Tsujimoto *et al.* 2001, Iwasaki and Nakamura 2002, 2004, 2006), the *vdLV* scheme has been implemented as an alternative means for the SVD computation. Numerical experiments show its strong competitiveness with existing SVD software packages. We briefly outline the ideas below, which also provides another example of Figure 1.1 on how a differential system might be carefully discretized and implemented to become an effective algorithm.

It will be most convenient if we present the interrelationships in matrix form, even though the actual computation should involve only a few scalars. For each ℓ, define two sequences of scalars,

$$q_i^{[\ell]} := \frac{1}{\delta^{[\ell]}}\big(1+\delta^{[\ell]}u_{2i-2}^{[\ell]}\big)\big(1+\delta^{[\ell]}u_{2i-1}^{[\ell]}\big), \quad i=1,\ldots,n, \tag{6.21}$$

$$e_j^{[\ell]} := \delta^{[\ell]}u_{2j-1}^{[\ell]}u_{2j}^{[\ell]}, \quad j=1,\ldots n-1, \tag{6.22}$$

and assemble them into two $n \times n$ bidiagonal matrices,

$$L^{[\ell]} := \begin{bmatrix} q_1^{[\ell]} & 0 & & & 0 \\ 1 & q_2^{[\ell]} & & & \\ & & \ddots & & \\ & & \ddots & & \\ & & & 1 & q_n^{[\ell]} \end{bmatrix}, \tag{6.23}$$

$$R^{[\ell]} := \begin{bmatrix} 1 & e_1^{[\ell]} & & & \\ 0 & 1 & & & \\ & & \ddots & \ddots & \\ & & & & e_{n-1}^{[\ell]} \\ & & & & 1 \end{bmatrix}. \tag{6.24}$$

From the relationship (6.20), it is readily verifiable that the matrix equation

$$L^{[\ell+1]}R^{[\ell+1]} = R^{[\ell]}L^{[\ell]} - \left(\frac{1}{\delta^{[\ell]}} - \frac{1}{\delta^{[\ell+1]}}\right)I_n \tag{6.25}$$

holds for all ℓ. It should not be a surprise to discover that the above formulation corresponds to the so-called *progressive qd algorithm* already described by Rutishauser (1954, 1960).

As a matter of fact, equation (6.25) is even more closely related to the so-called *differential quotient-difference algorithm with shift* (*dqds*) proposed by Fernando and Parlett (1994) and implemented in Parlett and Marques (2000). More specifically, if we abbreviate the left-hand side of the *vdLV* scheme in (6.20) as

$$w_k^{[\ell]} := u_k^{[\ell]}\bigl(1 + \delta^{[\ell]} u_{k-1}^{[\ell]}\bigr), \tag{6.26}$$

and introduce the tridiagonal matrix $Y^{[\ell]}$ defined by

$$Y^{[\ell]} := L^{[\ell]} R^{[\ell]} - \frac{1}{\delta^{[\ell]}} I_n, \tag{6.27}$$

then we find from (6.20) that $Y^{[\ell]}$ can be expressed in the form

$$Y^{[\ell]} = \begin{bmatrix} w_1^{[\ell]} & w_1^{[\ell]} w_2^{[\ell]} & 0 & & 0 \\ 1 & w_2^{[\ell]} + w_3^{[\ell]} & w_3^{[\ell]} w_4^{[\ell]} & & \\ & & \ddots & & \\ & & \ddots & & \\ & & \ddots & & \\ 0 & & & & w_{2n-3}^{[\ell]} w_{2n-2}^{[\ell]} \\ 0 & & & 1 & w_{2n-2}^{[\ell]} + w_{2n-1}^{[\ell]} \end{bmatrix}, \tag{6.28}$$

and that the relationship

$$Y^{[\ell+1]} = R^{[\ell]} Y^{[\ell]} {R^{[\ell]}}^{-1} \tag{6.29}$$

holds for all ℓ. Clearly, all matrices in the sequence $\{Y^{[\ell]}\}$ are isospectral. To connect back to our original goal of computing the singular values, observe that $w_k^{[\ell]} > 0$ as long as $u_k^{[0]} > 0$ and $\delta^{[\ell]} > 0$, which can easily be achieved. We thus can symmetrize the tridiagonal matrix $Y^{[\ell]}$ by a diagonal similarity transformation,

$$Y_S^{[\ell]} := {D^{[\ell]}}^{-1} Y^{[\ell]} D^{[\ell]}, \tag{6.30}$$

with

$$D^{[\ell]} := \operatorname{diag}\left\{ \prod_{i=1}^{n-1} \sqrt{w_{2i-1}^{[\ell]} w_{2i}^{[\ell]}},\ \prod_{i=2}^{n-1} \sqrt{w_{2i-1}^{[\ell]} w_{2i}^{[\ell]}}, \ldots, \sqrt{w_{2n-3}^{[\ell]} w_{2n-2}^{[\ell]}}, 1 \right\}.$$

Again, it is easy to check that the positivity of $w_k^{[\ell]}$ guarantees that $Y_S^{[\ell]}$ enjoys a Cholesky decomposition

$$Y_S^{[\ell]} = {B^{[\ell]}}^{\top} B^{[\ell]}, \tag{6.31}$$

with

$$B^{[\ell]} := \begin{bmatrix} \sqrt{w_1^{[\ell]}} & \sqrt{w_2^{[\ell]}} & & & & \\ 0 & \sqrt{w_3^{[\ell]}} & \sqrt{w_4^{[\ell]}} & & & \\ & & & \ddots & & \\ & & & \ddots & & \\ & & & & \sqrt{w_{2n-3}^{[\ell]}} & \sqrt{w_{2n-2}^{[\ell]}} \\ & & & & & \sqrt{w_{2n-1}^{[\ell]}} \end{bmatrix}. \tag{6.32}$$

The above recurrence relationships, all derived from an integrable discretization (6.20) of the Lotka–Volterra equation (6.3), have useful application to the SVD computation. We summarize the discussion thus far in the following theorem.

Theorem 6.1. Given the boundary conditions $u_0^{[\ell]} \equiv 0$ and $u_{2n}^{[\ell]} \equiv 0$, let the sequence $\{u_k^{[\ell]}\}$ be generated by the scheme (6.20). Then the singular values of the bidiagonal matrices $\{B^{[\ell]}\}$ which is defined in (6.32) with its entries $\{w_k^{[\ell]}\}$ given by (6.26) are invariant in ℓ.

For our application, we are interested in computing the singular values of a given matrix B_0. Thus, we need to make sure that the initial values for the iterative scheme (6.20) should be

$$u_k^{[0]} := \frac{b_k(0)^2}{1 + \delta^{[0]} u_{k-1}^{[0]}}, \quad k = 1, 2, \ldots, 2n-1. \tag{6.33}$$

The calculation of $u_k^{[\ell+1]}$ proceeds in the fashion depicted in Figure 6.1, where the quantity

$$v_k^{[\ell]} := u_k^{[\ell]} \bigl(1 + \delta^{[\ell]} u_{k+1}^{[\ell]}\bigr) \tag{6.34}$$

is an intermediate value listed for convenience, but is also used later. The bold-faced arrows point to the input and output in one step of the calculation. The shaded region indicates the array of initial values and progresses downward as ℓ increases. In the meantime, it is important to note that the boundary conditions from the two vertical boxes in Figure 6.1 help to make the computation explicit in ℓ.

Convergence theory and stability analysis of the *vdLV* scheme are well established in the series of papers referred to earlier and, in particular, the book by Nakamura (2006). It has been proved, for example, that with initial values (6.33) and any step sizes $\delta^{[\ell]} > 0$, the sequence $\{u_1^{[\ell]}, u_3^{[\ell]}, \ldots, u_{2n-1}^{[\ell]}\}$ converges to the squares of singular values of B_0 in descending order while

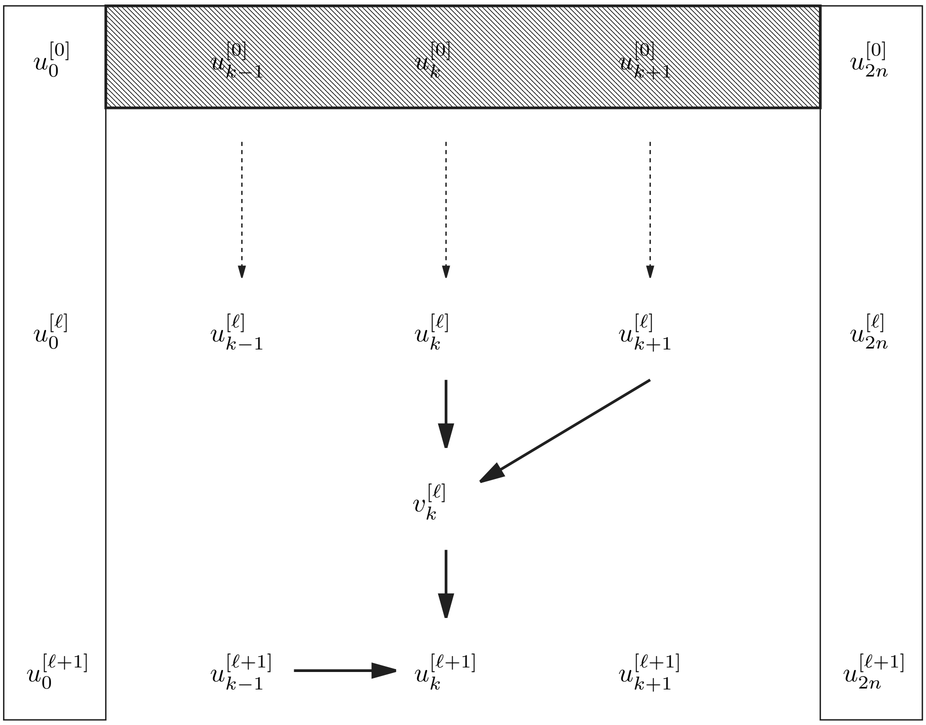

Figure 6.1. Computing $u_k^{[\ell+1]}$ via *vdLV*.

$u_{2k}^{[\ell]}$ converges to 0 for all k as ℓ goes to infinity. The *vdLV* scheme (6.20) enjoys additional nice features: no subtraction is involved and all quantities are bounded by $\|B_0\|$, implying its numerical stability.

What we have shown thus far is that the Lotka–Volterra equation gives rise to, on one hand, the iterates of the standard SVD algorithm when its solution is sampled at integer times and, on the other hand, an entirely different iterative scheme when the differential system is discretized under some proper conditions. The relationship (6.25) indicates that the *vdLU* scheme is algebraically equivalent to the *dqds* with the shift

$$s := \frac{1}{\delta^{[\ell]}} - \frac{1}{\delta^{[\ell+1]}}. \tag{6.35}$$

However, up to this point, we have not given any clear strategy on how the step size $\delta^{[\ell]}$ should be selected in the *vdLV* scheme. In the case of constant step size $\delta^{[\ell]} \equiv \delta$, Iwasaki and Nakamura (2002) have shown that the convergence is linear, with asymptotic convergence factor given by

$$\alpha = \max_{k=1,\dots,n-1} \frac{\sigma_{k+1} + \frac{1}{\delta}}{\sigma_k + \frac{1}{\delta}},$$

where $\sigma_1 > \sigma_2 > \cdots > \sigma_n$ are the singular values of B_0. It implies that larger step sizes might reduce the value of α to a certain extent. Linear convergence with the built-in shift (6.35) certainly cannot make the *vdLV* algorithm efficient enough.

Strictly speaking, the shift (6.35) has never entered into the matrix $B^{[\ell]}$ effectually. In the case of constant step size, $s = 0$. In the case of variable step size, the effect of s is diminished as $\delta^{\ell]}$ is increased. The true shift that is really needed should be of the form (Iwasaki and Nakamura 2006)

$$\overline{B}^{[\ell]^\top}\overline{B}^{[\ell]} = B^{[\ell]^\top}B^{[\ell]} - \theta^{[\ell]^2}, \tag{6.36}$$

while we keep the bidiagonal form

$$\overline{B}^{[\ell]} := \begin{bmatrix} \sqrt{\overline{w}_1^{[\ell]}} & \sqrt{\overline{w}_2^{[\ell]}} & & & & \\ 0 & \sqrt{\overline{w}_3^{[\ell]}} & \sqrt{\overline{w}_4^{[\ell]}} & & & \\ & & & \ddots & & \\ & & & \ddots & & \\ & & & & \sqrt{\overline{w}_{2n-3}^{[\ell]}} & \sqrt{\overline{w}_{2n-2}^{[\ell]}} \\ & & & & & \sqrt{\overline{w}_{2n-1}^{[\ell]}} \end{bmatrix}. \tag{6.37}$$

Upon comparing the entries, we find the nonlinear relationship that

$$\overline{w}_{2k}^{[\ell]} + \overline{w}_{2k+1}^{[\ell]} = w_{2k}^{[\ell]} + w_{2k+1}^{[\ell]} - \theta^{[\ell]^2}, \tag{6.38}$$

$$\overline{w}_{2k-1}^{[\ell]}\overline{w}_{2k}^{[\ell]} = w_{2k-1}^{[\ell]}w_{2k}^{[\ell]}, \quad k = 0, \ldots, n-1, \tag{6.39}$$

with $\overline{w}_0^{[\ell]} = w_0^{[\ell])} \equiv 0$. Though nonlinear, this relationship is a bijection correspondence between $\big(w_1^{[\ell]}, \ldots, w_{2n-1}^{[\ell]}\big)$ and $\big(\overline{w}_1^{[\ell]}, \ldots, \overline{w}_{2n-1}^{[\ell]}\big)$. The nonlinear map in (6.38) and (6.39) can easily be carried out by recurrence for computation, starting at the vertical box on the left with zero boundary conditions and progressing to the right, as indicated in Figure 6.2.

Recall that

$$v_k^{[\ell]} = w_k^{[\ell+1]},$$

by definitions in (6.26) and (6.34), and that

$$u_k^{[\ell+1]} = \frac{w_k^{[\ell+1]}}{1 + \delta^{[\ell+1]}u_{k-1}^{[\ell+1]}},$$

$$\begin{tikzcd}
w_0^{[\ell]} & w_{2k-1}^{[\ell]} \arrow[dr] & w_{2k}^{[\ell]} \arrow[d] \arrow[dr, dashed] & w_{2k+1}^{[\ell]} \arrow[d, dashed] \\
\overline{w}_0^{[\ell]} \arrow[r, dotted] & \overline{w}_{2k-1}^{[\ell]} \arrow[r] & \overline{w}_{2k}^{[\ell]} \arrow[r, dashed] & \overline{w}_{2k+1}^{[\ell]}
\end{tikzcd}$$

Figure 6.2. Updating $w_k^{[\ell]}$ to $\overline{w}_k^{[\ell]}$ with shift (dashed line (6.38); solid line (6.39)).

by the *vdLV* scheme (6.20). The modified scheme with shift becomes

$$u_k^{[\ell+1]} = \frac{\overline{w}_k^{[\ell+1]}}{1 + \delta^{[\ell+1]} u_{k-1}^{[\ell+1]}}. \tag{6.40}$$

This variant, called the *mdLVs*, has been studied thoroughly in Iwasaki and Nakamura (2006). The diagram in Figure 6.1 is therefore modified to become Figure 6.3. Be aware of the possible 'psychological illusion' perceived in Figure 6.3. It does appear that the emphasis is on the computation of $u_k^{[\ell+1]}$. However, the diagram can also be interpreted as a path to advance $w_k^{[\ell]}$ and $\overline{w}_k^{[\ell]}$ to $w_k^{[\ell+1]}$ and $\overline{w}_k^{[\ell+1]}$, respectively, whereas $u_k^{[\ell]}$ should be regarded as an intermediate value for convenience.

Many more research results and interesting properties could have been described. Singular vector computation by taking advantage of the *mdLVs* scheme, for example, is another important topic. However, to stay within the theme of this article, we shall stop short of giving more detailed shift strategies and convergence analysis which are available in the literature (Nakamura 2006). Suffice it to say that numerical experiments reported in Takata, Iwasaki, Kimura and Nakamura (2005, 2006) seem to suggest strongly that the resulting algorithm is competitive in both speed and accuracy with existing SVD packages.

The discourse presented in Sections 5 and 6 appears verbose. However, these two sections manifest a successful story about viewing numerical linear algebra algorithms as dynamical systems. We hope to have accomplished two goals through this important deliberation.

First, powerful discrete dynamical systems such as the *QR* algorithm and the SVD algorithm do have their continuous counterparts, namely, the Toda lattice and the Lotka–Volterra equation, which often arise from seemingly rather distinct fields of disciplines. We think it is truly remarkable that

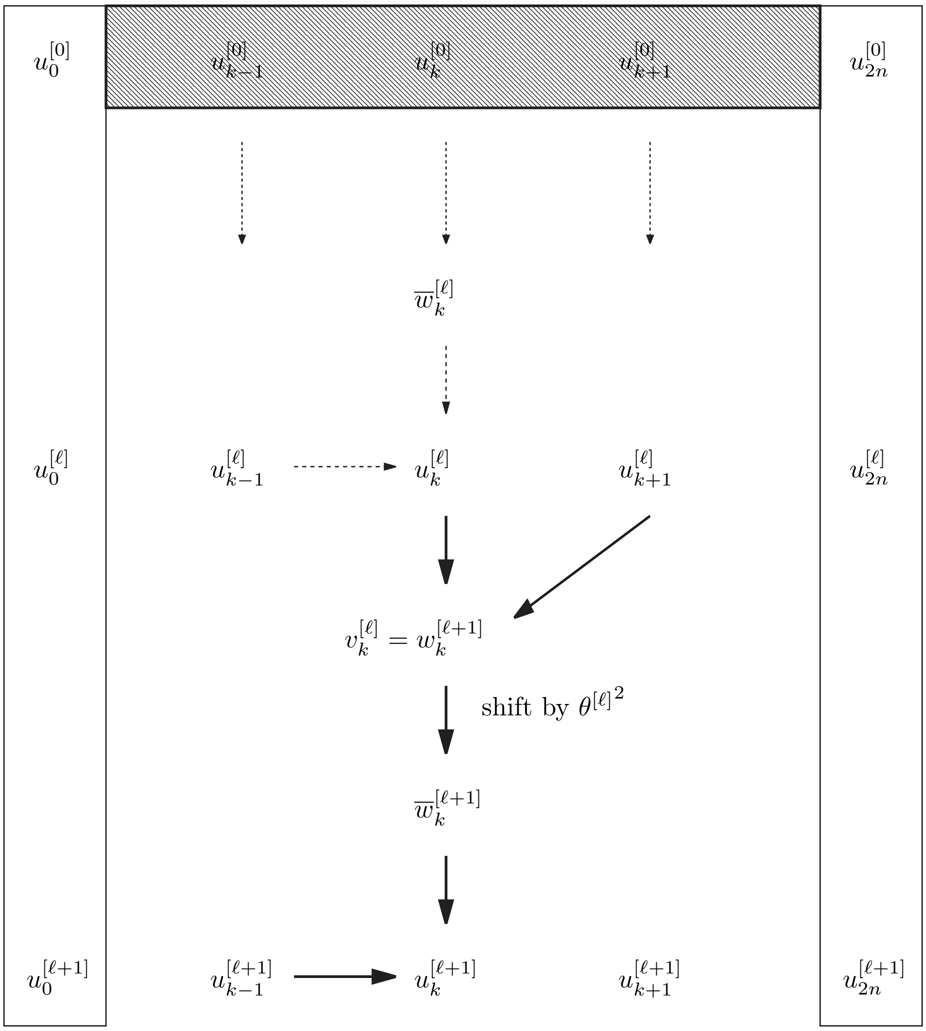

Figure 6.3. Computing $u_k^{[\ell+1]}$ via *mdLVs*.

diverse topics, such as soliton theory, integrable systems, continuous fractions, τ functions, orthogonal polynomials, the Sylvester identity, moments, and Hankel determinants, can all play together, intertwine, and eventually lead to the fact abstractly, but literally, that the eigenvalues and the singular values of a given matrix can be expressed as the limit of some closed-form formulas! We hope to have offered complete particulars on the determinantal solutions to the Toda lattice and the Lotka–Volterra equation, which we know are tied to the eigenvalues and singular values of the underlying matrix.

Secondly, via a rather thorough description of the *vdLV* scheme, we hope to have demonstrated our point in Figure 1.1 that a careful discretization of a continuous dynamical system may indeed lead to an effective numerical algorithm. By a 'careful discretization', it is important to note that the discrete scheme (6.20) maintains its complete integrability, which in the limit is the same integrability as that of the original Lotka–Volterra equation. Integrability-preserving discretization seems to be the key to success here, though a great many details such as shift strategies and implementation tactics also demand considerable attention. It is interesting to note the route we have taken, from the classical Golub–Kahan algorithm to the Lotka–Volterra equation, to the *dLV* scheme, to the *dqds* algorithm, and then to a brand new method, *mdLVs*, for computing the singular value decomposition.

7. Dynamical systems as group actions

Linear transformation is one of the simplest, yet most profound, ways to describe the relationship between two vector spaces. Over linear subspaces with a countable basis, linear transformations can be conveniently represented by matrices. It is often desirable to represent a linear transformation in some characteristic way, leading to the notion of identifying a matrix by its *canonical form.* The canonical form, most frequently expressed in terms of matrix decomposition, can facilitate discussions that, otherwise, would be complicated and involved. For years researchers have taken great steps to describe, analyse, and modify algorithms to reduce a matrix to its canonical form.

Many types of canonical forms exist in the literature. Those feasible for numerical computation include the spectral decomposition for symmetric matrices, the singular value decomposition for rectangular matrices, and the Schur decomposition for general square matrices (Golub and Van Loan 1996, Horn and Johnson 1990). The Jordan canonical form, on one hand, is perhaps the most fundamental and classical in matrix theory. On the other hand, the Jordan decomposition is generally considered a 'taboo' for numerical computation because it is hard to distinguish between eigenvalues that are repeated exactly and eigenvalues that are clustered closely together (Beelen and Van Dooren 1990, Golub and Wilkinson 1976). Nonetheless, it might be worthwhile to mention the notion of *pejorative manifold* proposed in an unpublished paper by Kahan (1972), who argues that multiple roots are well behaved under perturbation when the multiplicity structure is preserved. Loosely speaking, it is suggested that problems that are sensitive to arbitrary perturbations might be less sensitive to structured perturbations. Exploiting this idea, Zeng and Li (2007) have recently proposed an interesting approach to tackle the Jordan decomposition.

Thus far, most matrix decompositions are processed through iterative procedures whose success is made evident by the many available discrete methods. Our goal in this section is to recast some of those iterative schemes as dynamical systems via group actions.

We need to adjust our mind-set before continuing: the meaning of a canonical form should be understood with a much broader field of view than just matrix factorizations. Arnold (1988) asks a question in a similar spirit:

> 'What is *the simplest form* to which a family of matrices depending smoothly on the parameters can be reduced by *a change of coordinates* depending smoothly on the parameters?'

Obviously, an essential component to any answer is a qualification of the simplest form to which, and a mechanism by which, the coordinates are continuously changed. Before being specific about the qualification and the mechanism, we may categorically characterize the proposed procedure, whether discrete or continuous in nature, as a realization process. The canonical form, or the simplest form, that the process intends to realize ultimately should be interpreted broadly as any 'mode' from which we gain the agility to think and draw conclusions. Some useful modes, as well as the mechanisms to realize these modes, will be exemplified in the subsequent discussion.

The precise meaning of our points above will become clear later, but for now we hastily point out that, as a whole, the procedure to find the simplest form in most applications appears to follow the orbit of a certain matrix group action on the underlying matrix. This connection should not come as a surprise because the representation of a group by its homomorphisms into bijective linear maps over a certain vector space is well known (Curtis 1984, Shaw 1982, Smirnov 1970). For groups whose elements depend on continuously varying parameters, so do the corresponding matrix representations. The obvious advantage of this tie is that we have the group structure on one side and the matrix structure on the other side. A *matrix group*, that is, a subset of non-singular matrices which are closed under matrix multiplication and inversion, does form a Lie group (Howe 1983). The well-developed Lie theory therefore lends us greater advantages over iterations lacking this structure.

The question then becomes: What canonical form can a matrix, or a family of matrices, be linked to by the orbit of a group action? The choice of the group, the definition of the action, and the intended targets will constrain the various paths of transitions and, thus, the algorithms. Earlier work along these lines can be found in Della-Dora (1975). We will try to expound the various aspects of the recent development and applications in this direction. Some newly developed dynamical systems seem able to offer promising channels to tackle some linear algebra problems that, otherwise, are difficult to solve by iterative means.

7.1. Group actions and canonical forms

In a dynamical system, the state variable gets evolved in accordance with a certain rule. How the rule of transition is defined determines the dynamical behaviour. The emphasis of this section is on a specific rule characterized by group actions.

Given a group G and a set $\mathbb{V}$, a *group action* of G on $\mathbb{V}$ is a map $\mu : G \times \mathbb{V} \longrightarrow \mathbb{V}$ satisfying the associative law

$$\mu(gh, \mathbf{x}) = \mu(g, \mu(h, \mathbf{x})), \quad g, h \in G, \tag{7.1}$$

Table 7.1. Examples of classical matrix groups over $\mathbb{R}$.

Group	Subgroup	Notation	Characteristics
general linear		$\mathcal{G}l(n)$	$\{A \in \mathbb{R}^{n\times n} \mid \det(A) \neq 0\}$
	special linear	$\mathcal{S}l(n)$	$\{A \in \mathcal{G}l(n) \mid \det(A) = 1\}$
upper triangular		$\mathcal{U}(n)$	$\{A \in \mathcal{G}l(n) \mid A$ is upper triangular$\}$
	unipotent	$\mathcal{U}nip(n)$	$\{A \in \mathcal{U}(n) \mid a_{ii} = 1$ for all $i\}$
orthogonal		$\mathcal{O}(n)$	$\{Q \in \mathcal{G}l(n) \mid Q^\top Q = I\}$
generalized orthogonal		$\mathcal{O}_S(n)$	$\{Q \in \mathcal{G}l(n) \mid Q^\top SQ = S\}$, S is a fixed symmetric matrix
	symplectic	$\mathcal{S}p(2n)$	$\mathcal{O}_J(2n), \quad J := \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$
	Lorentz	$\mathcal{L}or(n,k)$	$\mathcal{O}_L(n+k)$, $L := \operatorname{diag}\{\underbrace{1,\ldots,1}_{n}, \underbrace{-1,\ldots-1}_{k}\}$
affine		$\mathcal{A}ff(n)$	$\left\{\begin{bmatrix} A & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mid A \in \mathcal{G}l(n), \mathbf{t} \in \mathbb{R}^n\right\}$
	translation	$\mathcal{T}rans(n)$	$\left\{\begin{bmatrix} I & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mid \mathbf{t} \in \mathbb{R}^n\right\}$
	isometry	$\mathcal{I}som(n)$	$\left\{\begin{bmatrix} Q & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \mid Q \in \mathcal{O}(n), \mathbf{t} \in \mathbb{R}^n\right\}$
product of G_1 and G_2		$G_1 \times G_2$	$\{(g_1, g_2) \mid g_1 \in G_1, g_2 \in G_2\}$, $(g_1, g_2) * (h_1, h_2) := (g_1h_1, g_2h_2)$, G_1 and G_2 are given groups
automorphism		$\mathbb{G}_M$	$\{A \in \mathcal{G}l(n) \mid \langle A\mathbf{x}, A\mathbf{y}\rangle_M = \langle \mathbf{x}, \mathbf{y}\rangle_M\}$, $\langle \mathbf{x}, \mathbf{y}\rangle_M = \mathbf{x}^\top M\mathbf{y}$ M is a given matrix

and the identity property

$$\mu(e, \mathbf{x}) = \mathbf{x}, \tag{7.2}$$

where e is the identity element in G, for all $\mathbf{x} \in \mathbb{V}$. Given a fixed $\mathbf{x} \in \mathbb{V}$, the *orbit of* $\mathbf{x}$ associated to an action μ of G is defined to be the set

$$\mathrm{Orb}_G(\mathbf{x}) := \{\mu(g, \mathbf{x}) | g \in G\}. \tag{7.3}$$

For our applications, we are interested in using matrix groups and various actions to help transform a given matrix into an appropriate canonical form. The transformation is to take place along the associated orbit of the given matrix. To get this idea going, we need four components working together: a group that characterizes the coordinates to be used, an action that constrains the transformations to be allowed, a canonical form that sets the goal to be reached, and a rule that delineates the path to be followed. Each of these four components affects the final result.

For demonstration, Table 7.1 is a short list of matrix groups compiled from the books of Baker (2002), Chu and Golub (2005) and Curtis (1984). We remark that the automorphism group $\mathbb{G}_M$ associated with a non-degenerate bilinear form $\langle \mathbf{x}, \mathbf{y} \rangle_M = \mathbf{x}^\top M \mathbf{y}$ contains as special cases the orthogonal group and the symplectic group (Mackey, Mackey and Tisseur 2003).

Table 7.2 typifies some group actions that have been commonly used in numerical linear algorithm algorithms. Traditionally, numerical analysts prefer to use the orthogonal group for actions because of its cost efficiency and numerical stability. Such a restriction, however, could have limited the canonical forms that we otherwise would be able to reach by different groups.

Table 7.2. Examples of group actions and their applications.

Set $\mathbb{V}$	Group G	Action $\mu(g, A)$	Application
$\mathbb{R}^{n \times n}$	any subgroup	$g^{-1}Ag$	conjugation
$\mathbb{R}^{n \times n}$	$\mathcal{O}(n)$	$g^\top Ag$	orthogonal similarity
$\underbrace{\mathbb{R}^{n \times n} \times \ldots \times \mathbb{R}^{n \times n}}_{k}$	any subgroup	$(g^{-1}A_1 g, \ldots, g^{-1}A_k g)$	simultaneous reduction
$\mathbb{S}(n) \times \mathbb{S}_{PD}(n)$	any subgroup	$(g^\top Ag, g^\top Bg)$	symm. positive definite pencil reduction
$\mathbb{R}^{n \times n} \times \mathbb{R}^{n \times n}$	$\mathcal{O}(n) \times \mathcal{O}(n)$	$(g_1^\top Ag_2, g_1^\top Bg_2)$	QZ decomposition
$\mathbb{R}^{m \times n}$	$\mathcal{O}(m) \times \mathcal{O}(n)$	$g_1^\top Ag_2$	singular value decomp.
$\mathbb{R}^{m \times n} \times \mathbb{R}^{p \times n}$	$\mathcal{O}(m) \times \mathcal{O}(p) \times \mathcal{G}l(n)$	$(g_1^\top Ag_3, g_2^\top Bg_3)$	generalized singular value decomp.

Table 7.3 makes evident the wide scope of canonical forms that group actions can (or be desired to) accomplish, ranging from a typical structure with a specified pattern of zeros, such as a diagonal, tridiagonal, or triangular matrix, to a matrix with a specified construct, such as Toeplitz, Hamiltonian, stochastic, or other linear varieties, to a matrix with a specified algebraic constraint, such as low rank or non-negativity.

With the group, action and orbit in place, we finally need a properly defined dynamical system, either continuous or discrete, so that its integral curves or iterates stay on the specified orbit and connect one state to the next state. The Toda lattice and the Lotka–Volterra equation discussed earlier serve as typical examples in this regard, although in both cases the group actions are built into the dynamical systems and are not exploited explicitly. We shall develop a general framework of the projected gradient

Table 7.3. Examples of canonical forms used in practice.

Canonical form	Also known as	Action
bidiagonal J	quasi-Jordan decomp., $A \in \mathbb{R}^{n\times n}$	$P^{-1}AP = J$, $P \in \mathcal{G}l(n)$
diagonal Σ	sing. value decomp., $A \in \mathbb{R}^{m\times n}$	$U^\top AV = \Sigma$, $(U,V) \in \mathcal{O}(m) \times \mathcal{O}(n)$
diagonal pair (Σ_1, Σ_2)	gen. sing. value decomp., $(A,B) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{p\times n}$	$(U^\top AX, V^\top BX) = (\Sigma_1, \Sigma_2)$, $(U,V,X) \in \mathcal{O}(m) \times \mathcal{O}(p) \times \mathcal{G}l(n)$
upper quasi-triangular H	real Schur decomp., $A \in \mathbb{R}^{n\times n}$	$Q^\top AQ = H$, $Q \in \mathcal{O}(n)$
upper quasi-triangular H upper triangular U	gen. real Schur decomp., $A, B \in \mathbb{R}^{n\times n}$	$(Q^\top AZ, Q^\top BZ) = (H, U)$, $Q, Z \in \mathcal{O}(n)$
symmetric Toeplitz T	Toeplitz inv. eigenv. prob., $\{\lambda_1, \ldots, \lambda_n\} \subset \mathbb{R}$ is given	$Q^\top \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\}Q = T$, $Q \in \mathcal{O}(n)$
non-negative $N \geq 0$	non-neg. inv. eigenv. prob., $\{\lambda_1, \ldots, \lambda_n\} \subset \mathbb{C}$ is given	$P^{-1} \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\}P = N$, $P \in \mathcal{G}l(n)$
linear variety X with fixed entries at fixed locations	matrix completion prob., $\{\lambda_1, \ldots, \lambda_n\} \subset \mathbb{C}$ is given $X_{i_\nu, j_\nu} = a_\nu, \nu = 1, \ldots, \ell$	$P^{-1}\{\lambda_1, \ldots, \lambda_n\}P = X$, $P \in \mathcal{G}l(n)$
nonlinear variety with fixed singular values and eigenvalues	test matrix construction, $\Lambda = \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\}$ and $\Sigma = \operatorname{diag}\{\sigma_1, \ldots \sigma_n\}$ are given	$P^{-1}\Lambda P = U^\top \Sigma V$ $P \in \mathcal{G}l(n), \quad U, V \in \mathcal{O}(n)$
maximal fidelity	structured low-rank approx. $A \in \mathbb{R}^{m\times n}$	$\big(\operatorname{diag}(USS^\top U^\top)\big)^{-1/2}USV^\top$, $(U,S,V) \in \mathcal{O}(m) \times \mathbb{R}^k_\times \times \mathcal{O}(n)$

approach in the next section to help to construct other useful dynamical systems. The projected gradient flows from continuous group actions are often easy to formulate and analyse, and are sometimes able to tackle problems that are seemingly impossible to resolve by conventional discrete methods.

An area that has been active for research, and remains open for further work, is to develop numerical algorithms that can effectively trace dynamical systems arising from various group actions. We note that there are many new techniques developed recently for dynamical systems on Lie groups, including the RK-MK methods (Engø 2003, Munthe-Kaas 1998), Magnus and Fer expansions (Blanes, Casas, Oteo and Ros 1998, Zhang and Deng 2005), and so on. A good collection of Lie structure-preserving algorithms and pertaining references can be found in the seminal review paper by Iserles, Munthe-Kaas, Nørsett and Zanna (2000) and the book by Hairer, Lubich and Wanner (2006). These new geometric integration techniques certainly can benefit the computations needed for the projected gradient flow, but still we are seeking a method that also takes into account the descent property of a gradient flow. For a gradient flow where finding its stable equilibrium point is the ultimate goal of computation, recall that the pseudo-transient continuation described in Section 2.2 has been suggested as a possible numerical method.

7.2. Projected gradient flows

The idea of projected gradient flows stems from the constrained least-squares approximation to a desirable canonical form. From a given matrix A in a subset $\mathbb{V}$ of matrices of fixed sizes, the constraint on the variable is that the transformation of A must be limited to the orbit $\operatorname{Orb}_G(A)$ determined by a prescribed continuous matrix group G and a group action $\mu : G \times \mathbb{V} \longrightarrow \mathbb{V}$. The objective function itself is built with two additional limitations. One is a differentiable map $f : \mathbb{V} \longrightarrow \mathbb{V}$ designed to regulate certain 'inherent' properties such as symmetry, diagonal, isospectrality, low rank, or other algebraic conditions. The other is a projection map $P : \mathbb{V} \longrightarrow \mathbb{P}$, where $\mathbb{P}$ denotes the subset of matrices in $\mathbb{V}$ carrying a certain desirable structure, that is, the canonical form. The set $\mathbb{P}$ could be a singleton, an affine subspace, or a cone, or other geometric entities. Consider the functional $F : G \longrightarrow \mathbb{R}$ where

$$F(Q) := \frac{1}{2}\|f(\mu(Q,A)) - P(\mu(Q,A))\|_F^2. \tag{7.4}$$

The goal is to minimize F over the group G. The meaning of this constrained minimization is that, while staying in the orbit of A under the action of μ and maintaining the inherent property guaranteed by the function f, we look for the element $Q \in G$ so that the matrix $f(\mu(Q,A))$ best realizes the desired canonical structure in the sense of least squares.

In principle, the functional (7.4) can be minimized by conventional optimization techniques which mostly are iterative in nature. However, we find that the projected gradient flow approach can conveniently be formulated as a dynamical system,

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = -\mathrm{Proj}_{\mathcal{T}_Q G} \nabla F(Q), \tag{7.5}$$

where $\mathcal{T}_Q G$ and $\nabla F(Q)$ stand for the tangent space of the group G and the gradient of the objective functional F at Q, respectively.

One advantage of working with a matrix group is that its tangent spaces at every element g have the same structure as the tangent space $\mathfrak{g} = \mathcal{T}_e G$ at the identity element e of G. More specifically, the tangent space at any element Q in G is a translation of $\mathfrak{g}$ via the relationship

$$\mathcal{T}_Q G = Q\mathfrak{g}. \tag{7.6}$$

Thus the projection in (7.5) is fairly easy to do once the tangent space $\mathfrak{g}$ is identified.

It might be instructive to illustrate the idea of projection by the following calculation (Chu and Driessel 1990). By (7.6), the tangent space of $\mathcal{O}(n)$ at any orthogonal matrix Q is

$$\mathcal{T}_Q \mathcal{O}(n) = Qo(n),$$

where $o(n)$ denotes the subspace of all skew-symmetric matrices in $\mathbb{R}^{n\times n}$. It can easily be argued that the normal space of $\mathcal{O}(n)$ at any orthogonal matrix Q is

$$\mathcal{N}_Q \mathcal{O}(n) = Qo(n)^{\perp},$$

where the orthogonal complement $o(n)^{\perp}$ is precisely the subspace of all symmetric matrices. The space $\mathbb{R}^{n\times n}$ can be split as the direct sum of

$$\mathbb{R}^{n\times n} = Qo(n) \oplus Qo(n)^{\perp}.$$

Any $X \in \mathbb{R}^{n\times n}$ therefore has the unique orthogonal splitting

$$X = Q(Q^{\top}X) = Q\left\{\frac{1}{2}(Q^{\top}X - X^{\top}Q)\right\} + Q\left\{\frac{1}{2}(Q^{\top}X + X^{\top}Q)\right\}.$$

The projection of X onto the tangent space $\mathcal{T}_Q \mathcal{O}(n)$ is therefore given by the formula

$$\mathrm{Proj}_{\mathcal{T}_Q \mathcal{O}(n)} X = Q\left\{\frac{1}{2}(Q^{\top}X - X^{\top}Q)\right\}. \tag{7.7}$$

For other groups, the projection can be done in a similar way.

We briefly touch upon the realm of differential geometry with two remarks. First, the notion of ‘projected’ gradient described above is indeed the ‘ordinary’ gradient with respect to the Killing form or the normal metric on the tangent space $\mathfrak{g}$ (Edelman, Arias and Smith 1999, Tam 2004).

Secondly, the set $\mathfrak{g}$ is a *Lie subalgebra*, that is, its elements are closed under the Lie bracket operation. The Lie subalgebra $\mathfrak{g}$ can be characterized as the logarithm of G in the sense that

$$\mathfrak{g} = \{M \in \mathbb{R}^{n \times n} \mid \exp(tM) \in G, \text{ for all } t \in \mathbb{R}\}. \tag{7.8}$$

The exponential map $\exp : \mathfrak{g} \to G$, as we have seen in Theorem 5.1, is a central step from a Lie algebra $\mathfrak{g}$ to the associated Lie group G (Celledoni and Iserles 2000, Howe 1983). Since exp is a local homeomorphism which maps a neighbourhood of the zero O in the algebra $\mathfrak{g}$ onto a neighbourhood of the identity e in the group G, any dynamical system in G, in the neighbourhood of e, would therefore have a corresponding dynamical system in $\mathfrak{g}$, in the neighbourhood of O. Because of this, the decomposition we have observed in Section 5.1 can be interpreted as follows. It is known that the Lie group $\mathcal{G}l(n)$ can be decomposed as the product of two Lie subgroups in the neighbourhood of the identity matrix I if and only if the corresponding tangent space $gl(n)$ of real-valued $n \times n$ matrices can be decomposed into the sum of two Lie subalgebras. By the decomposition property and the reversal property in Theorem 5.1, the Lie structure is apparently not needed for isospectral flows. A subspace decomposition of $gl(n)$ as is indicated in (5.8) suffices to guarantee a factorization of a *one-parameter semigroup* in the neighbourhood of I as the product of two non-singular matrices, that is, the decomposition indicated in (5.10).

Before we talk about specific applications, a misconception about the gradient flow (2.10) in general and the projected gradient flow (7.5) in particular must be clarified. It is true that the objective value $F(\mathbf{x}(t))$ is non-increasing in t if $\mathbf{x}(t)$ follows the gradient flow (2.10). If F is further known to be bounded below, the $F(\mathbf{x}(t))$ converges to a limit value. However, the flow $\mathbf{x}(t)$ itself might not converge at all. Examples can be constructed to show the case that a local minimum of an infinitely differentiable objective function F may not be an equilibrium point of the differential system (2.10). Likewise, a stable equilibrium point of (2.10) may not be a local minimum of F at all (Absil and Kurdyka 2006). A cone-shaped minaret with outside spiral ramp, or a helicoid, can be modified to serve as examples where a gradient flow converges to a limit cycle. The important message we want to convey is that infinite smoothness of the gradient vector field is not sufficient to guarantee the convergence of a gradient trajectory. A sufficient condition that happens to fit our applications is the analyticity of the objective function. More specifically, the Łojasiewicz–Simon theorem asserts that if the objective function F is real analytic, then the trajectory of a gradient flow cannot have more than one limit point (Chill 2003, Łojasiewicz 1963, Simon 1983). Furthermore, under the analyticity assumption, a stable equilibrium point of the differential system (2.10) is a local minimum of F, and *vice versa* (Absil, Mahony and

Andrews 2005, Absil and Kurdyka 2006). In our applications, group actions, linear projections and squares of the Frobenius norm are naturally analytic. Our gradient flows are defined by an analytic vector field, so convergence is ensured.

7.3. Applications

From the framework outlined above, projected gradient dynamical systems can be tailored to meet the need arising from various circumstances. We shall demonstrate four interesting designs in this section. Many additional applications and the associated dynamical systems can be found in the literature. See, for instance, the problems discussed in the paper by Brockett (1993) and the book by Helmke and Moore (1994). Our intention in this section is to demonstrate the versatility of projected gradient flows. Some applications can be solved more efficiently by other means, but there are problems where the continuous dynamical systems approach is particularly easy to formulate and compute.

Example 7.1. Given a symmetric matrix Λ and a desirable structure $\mathbb{P}$, suppose we want to find a symmetric matrix that is closest to $\mathbb{P}$ and has the same spectrum as Λ (Chu and Driessel 1990). By defining the isospectral matrix $X := Q^\top \Lambda Q$ with $Q \in \mathcal{O}(n)$, the objective functional $F : \mathcal{O}(n) \to \mathbb{R}$ is taken to be

$$F(Q) := \frac{1}{2}\|Q^\top \Lambda Q - P(Q^\top \Lambda Q)\|_F^2, \tag{7.9}$$

where the Frobenius norm of a real matrix M is, as usual, defined by

$$\|M\|_F = \sqrt{\operatorname{trace}(MM^\top)}.$$

It can be verified that the projected gradient flow (7.5) on the group $\mathcal{O}(n)$ is equivalent to the isospectral flow,

$$\frac{\mathrm{d}X}{\mathrm{d}t} = [X, [X, P(X)]], \tag{7.10}$$

on the orbit $\mathrm{Orb}_{\mathcal{O}(n)}(\Lambda)$.

With different choices of Λ and $\mathbb{P}$, the dynamical system (7.10) enjoys different interpretation of applications. For example, if $P(X) = \operatorname{diag}(X)$, then $X(t)$ stands for a continuous Jacobi-type flow that gradually reduces the off-diagonal elements of X while maintaining isospectrality. As another example, by specifying the structure retained in $\mathbb{P}$, the flow (7.10) offers an avenue to tackle various kinds of very difficult structured inverse eigenvalue problems (Chu and Golub 2002).

The so-called double bracket flow by Brockett (1991) corresponds to the special case where $\mathbb{P} = \{N\}$ contains a single constant symmetric matrix N

and hence $P(Q^\top \Lambda Q) \equiv N$. The resulting qualitative behaviour is relatively easier to analyse, but this seemingly ingenuous nearest matrix approximation to a fixed matrix has the following sorting property, which appears universal in a wide spectrum of applications, including the interior-point algorithm (Faybusovich 1991), the *QR* algorithm (Deift *et al.* 1983), moment maps (Bloch, Brockett and Ratiu 1992) and many others (Helmke and Moore 1994).

Theorem 7.2. Suppose that both $\Lambda = \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\}$ and the spectrum of N have distinct elements. Then $X = Q^\top \Lambda Q$ is the unique nearest matrix to N on the isospectral orbit of Λ if and only if the columns of $Q^\top$ are the orthonormal eigenvectors of N, corresponding to eigenvalues arranged in the same order as $\{\lambda_1, \ldots, \lambda_n\}$.

We have to mention one remarkable connection. If Λ is a tridiagonal matrix to begin with and if $N = \operatorname{diag}\{n, n-1, \ldots, 2, 1\}$, then the double bracket flow becomes exactly the Toda lattice that has been discussed in great length in Section 5.1. The sorting property asserted in Theorem 7.2 therefore explains the sorting property of the *QR* algorithm. It is interesting that 'the same set of equations is thus Hamiltonian and a gradient flow on the isospectral set' (Bloch *et al.* 1992).

Given the wide range of applications, an effective way of integrating either the isospectral dynamical system (7.10) for $X(t)$ over the orbit or the associated parameter dynamical system for $Q(t)$ over the group therefore would be extremely useful and desirable. We think that an efficient discretization would probably not come from the traditional numerical ODE approaches, but rather could be more in line with the *vdLV* approach, where a certain structure is preserved. On the other hand, it is worth noting that the versatile double bracket flow $\frac{\mathrm{d}X}{\mathrm{d}t} = [X, [X, N]]$ might be handled differently. By representing the isospectral solution in the form $X(t) = \mathrm{e}^{\Omega(t)} X_0 E^{-\Omega(t)}$, Iserles (2002) has developed an interesting approach to the discretization of $X(t)$. Specifically, each term in the Taylor series expansion of $\Omega(t)$ can be constructed explicitly and recursively by means of rooted trees with bicolour leaves.

Example 7.3. In analogy to Example 7.1, we could also consider the nearest approximation by iso-singular-value matrices. Given a rectangular matrix Σ of size $m \times n$ and a desirable structure $\mathbb{P}$ over $\mathbb{R}^{m\times n}$, all matrices on the orbit $\operatorname{Orb}_{\mathcal{O}(m)\times\mathcal{O}(n)}(\Lambda) := \{X = U^\top \Sigma V | U \in \mathcal{O}(m), V \in \mathcal{O}(n)\}$ have the same singular values as Σ. The objective functional $F : \mathcal{O}(m) \times \mathcal{O}(n) \to \mathbb{R}$ defined by

$$F(U, V) := \|U^\top \Sigma V - P(U^\top \Sigma V)\|_F^2 \tag{7.11}$$

is meant to best approach the structure $\mathbb{P}$ while maintaining the singular values. A continuous transformation $X := U^\top \Sigma V$ is governed by the dynamical system

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \{X(X^\top P(X) - P(X)^\top X) - (XP(X)^\top - P(X)X^\top)X\}, \tag{7.12}$$

which, at first glance, is not exactly in the double bracket form. However, upon recasting the original action of equivalence $U^\top \Sigma V$ by the product group $\mathcal{O}(m) \times \mathcal{O}(n)$ as a new action of conjugation,

$$\begin{bmatrix} U^\top & 0 \\ 0 & V^\top \end{bmatrix} \begin{bmatrix} 0 & \Sigma \\ \Sigma^\top & 0 \end{bmatrix} \begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix},$$

by a subgroup of $\mathcal{O}(m+n)$, Tam (2004) has observed that (7.12) can indeed be written in a double bracket form,

$$\frac{\mathrm{d}\mathfrak{X}}{\mathrm{d}t} = [\mathfrak{X}, [\mathfrak{X}, \mathfrak{P}(\mathfrak{X})]], \tag{7.13}$$

with the definition

$$\mathfrak{X} := \begin{bmatrix} 0 & X \\ X^\top & 0 \end{bmatrix},$$

$$\mathfrak{P}(\mathfrak{X}) := \begin{bmatrix} 0 & P(X) \\ P(X)^\top & 0 \end{bmatrix}.$$

Some applications of the gradient flow (7.12) include a sorting property similar to Theorem 7.2 if $\mathbb{P}$ consists of a single constant matrix (Chu and Driessel 1990, Smith 1991), structured inverse singular value problems, and a Jacobi-type algorithm if $P(X) = \operatorname{diag}(X)$. In the last case, the corresponding dynamical system is

$$\frac{\mathrm{d}X}{\mathrm{d}t} = \{X(X^\top \operatorname{diag}(X) - \operatorname{diag}(X)^\top X) - (X \operatorname{diag}(X)^\top - \operatorname{diag}(X)X^\top)X\}.$$

It is interesting to note that by merely a change of sign in the above equation, we obtain the system

$$\begin{aligned} \frac{\mathrm{d}X}{\mathrm{d}t} &= \{X(X^\top \operatorname{diag}(X) - \operatorname{diag}(X)^\top X) + (X \operatorname{diag}(X)^\top - \operatorname{diag}(X)X^\top)X\} \\ &= XX^\top \operatorname{diag}(X) - \operatorname{diag}(X)X^\top X, \end{aligned} \tag{7.14}$$

which is precisely the SVD flow (6.2). Recall that the SVD flow was originally formulated with the intention to preserve the bidiagonal structure if Σ is bidiagonal to begin with. The fact that the SVD flow can be expressed differently as in (7.14) is interesting. At present, whether (7.14) is just an algebraic coincidence or is a result of a deeper theory is not clear to us.

Example 7.4. Consider the classical matrix nearness problem of finding the closest normal matrix to a given matrix $A \in \mathbb{C}^{n\times n}$ (Higham 1989, Ruhe

1987). This problem is equivalent to minimizing the functional

$$F(U) = \frac{1}{2}\|U^*AU - \operatorname{diag}(U^*AU)\|_F^2, \tag{7.15}$$

subject to the constraint that $U \in \mathbb{C}^{n\times n}$ is unitary. Once the minimizer $\tilde{U}$ of (7.15) is found, the nearest normal matrix to A is given by $\tilde{U}\operatorname{diag}(\tilde{U}^*A\tilde{U})\tilde{U}^*$.

The objective function (7.15) is similar to (7.9) except that we are dealing with complex-valued matrices. A projected gradient flow,

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = \left[Z, \frac{[Z, \operatorname{diag}(Z^*)] - [Z, \operatorname{diag}(Z^*)]^*}{2}\right], \tag{7.16}$$

for the complex matrix $Z = U^*AU$ can be derived as the action of the unitary group over $\mathbb{C}^{n\times n}$. One advantage of this differential equation approach is that many theoretical results concerning the nearest normal matrix approximation which have been challenging to matrix theorists can be obtained naturally from analysing the equilibrium point of the dynamical system (Chu 1991, Ruhe 1987).

Example 7.5. We now illustrate how the 'regulator' of f in (7.4) comes into play in some applications. Given two vectors $\mathbf{a}, \boldsymbol{\lambda} \in \mathbb{R}^n$, the Schur–Horn theorem states that there exists a Hermitian matrix H with eigenvalues $\boldsymbol{\lambda}$ and diagonal entries $\mathbf{a}$ if and only if $\boldsymbol{\lambda}$ is majorized by $\mathbf{a}$ (Horn and Johnson 1990). The harder part of this classical result is the inverse problem of construct a symmetric matrix with prescribed diagonal entries $\mathbf{a}$ and spectrum $\{\lambda_1, \ldots, \lambda_n\}$. We recast the inverse problem as the problem of minimizing the functional

$$F(Q) := \frac{1}{2}\|\operatorname{diag}(Q^\top \Lambda Q) - \operatorname{diag}(\mathbf{a})\|_F^2, \tag{7.17}$$

subject to $Q \in \mathcal{O}(n)$. Note that we have taken $f(X) = \operatorname{diag}(X)$ for the isospectral matrices $X := Q^\top \Lambda Q$. It can be shown that the projected gradient flow becomes a double bracket equation (Chu 1995):

$$\frac{\mathrm{d}X}{\mathrm{d}t} = [X, [X, \operatorname{diag}(\mathbf{a}) - \operatorname{diag}(X)]]. \tag{7.18}$$

Stability analysis at the equilibrium yields an easy existence proof of the Schur–Horn theorem.

We should re-emphasize that, unless special care is given to the discretization and implementation, the differential equation approach generally is not necessarily the most effective numerical means for solving problems. For the Schur–Horn problem, a finite-step recursive algorithm is computationally more efficient (Zha and Zhang 1995).

7.4. Generalization beyond group actions

The primary purpose of employing group actions in linear transformations is to keep eigenvalues or singular values invariant under the change of coordinates. It sometimes becomes desirable to keep other properties invariant. In many cases, the notion of gradient flows can be generalized to other geometric entities that do not hold any group structure. Examples of applications include the Stiefel manifold for the orthonormal Procrustes problem, or the more general Penrose regression problem (Chu and Trendafilov 2001), the convex set of positive definite real symmetric matrices for the balanced realization (Helmke, Moore and Perkins 1994), the Grassmann manifold for the geometric optimization methods (Edelman *et al.* 1999), the manifold of oblique matrices for the multi-dimensional scaling (Cox and Cox 1994, Del Buono and Lopez 2002) or the data fitting on the unit sphere (Chu, Del Buono, Lopez and Politi 2005), the cone of non-negative matrices for inverse eigenvalue problem (Chu and Guo 1998, Orsi 2006), and so on.

We wrap up this section by demonstrating one of these generalizations. At first glance, no group structure is involved in the formulation of the dynamical system. We then modify the coordinate systems to bring in group actions.

Example 7.6. The non-negative inverse eigenvalue problem concerns the construction of a entry-wise non-negative matrix $A \in \mathbb{R}^{n\times n}$ with a prescribed set $\{\lambda_1, \ldots, \lambda_n\} \subset \mathbb{C}$, closed under conjugation, as its spectrum. This has been a classical but hard problem, long investigated by many matrix theorists. The inadequacy of the current development is evidenced by the fact that the necessary condition for solvability is usually too general, while the sufficient condition is too specific (Chu and Golub 2005).

Recently it has been proved that, given an arbitrary $(n-1)$-tuple

$$\Omega = (\lambda_2, \ldots, \lambda_n) \in \mathbb{C}^{n-1},$$

whose components are closed under complex conjugation, there exists a unique positive real number $\mathcal{R}(\Omega)$, called the *minimal realizable spectral radius of* Ω, such that the set $\{\lambda_1, \ldots, \lambda_n\}$ is precisely the spectrum of a certain $n \times n$ non-negative matrix with λ_1 as its spectral radius if and only if $\lambda_1 \geq \mathcal{R}(\Omega)$. Employing any existing necessary conditions as a mode of checking criteria, Chu and Xu (2005) have proposed a simple bisection procedure to approximate the location of $\mathcal{R}(\Omega)$. As an immediate application, it offers a quick numerical way to check whether a given n-tuple could be the spectrum of a certain non-negative matrix. However, even after a potential spectrum is identified as feasible, very few general numerical procedures are available for the actual construction of non-negative matrices. Generalizing the above ideas and taking the advantage of its easy formulation, a gradient flow can come to serve this purpose (Chu and Guo 1998).

Since the spectrum is closed under complex conjugation, we may assume a real-valued matrix J to carry the prescribed spectrum. We cast the inverse problem as a constrained minimization problem by working with two matrix parameters (g, R),

$$\begin{aligned} &\text{minimize} && F(g,R) := \frac{1}{2}\|gJg^{-1} - R \circ R\|_F^2, \\ &\text{subject to} && g \in \mathcal{G}l(n),\ R \in gl(n), \end{aligned}$$

where $\circ$ denotes the component-to-component Hadamard product. The idea behind $F(g, R)$ is similar to that in (7.4), except that this time we want to minimize the distance between the orbit $\mathrm{Orb}_{\mathcal{G}l(n)}(J)$ and the cone of non-negative matrices. The constraints literally do not exist because both $\mathcal{G}l(n)$ and $gl(n)$ are open sets. No projection onto the constraints is needed. The steepest descent flow for $F(g, R)$ is given by straightforward calculation,

$$\frac{\mathrm{d}g}{\mathrm{d}t} = \big[(gJg^{-1})^\top, \alpha(g,R)\big] g^{-\top}, \tag{7.19}$$

$$\frac{\mathrm{d}R}{\mathrm{d}t} = 2\alpha(g,R) \circ R, \tag{7.20}$$

with $\alpha(g,R) := gJg^{-1} - R \circ R$.

The requirement of computing g^{-1} in the gradient flow is worrisome. We can diminish concern at the cost of re-parametrizing g by its analytic singular value decomposition (Bunse-Gerstner, Byers, Mehrmann and Nichols 1991, Wright 1992). Suppose $g(t) = X(t)S(t)Y(t)^\top$ is the singular value decomposition of $g(t)$, where $S(t)$ is a diagonal matrix with elements from the multiplicative group $\mathbb{R}_\times$ of non-zero real numbers and $X(t)$ and $Y(t)$ are elements from the orthogonal group $\mathcal{O}(n)$. From the relationship of derivatives,

$$X^\top \frac{\mathrm{d}g}{\mathrm{d}t} Y = \underbrace{X^\top \frac{\mathrm{d}X}{\mathrm{d}t}}_{Z} S + \frac{\mathrm{d}S}{\mathrm{d}t} + S \underbrace{\frac{\mathrm{d}Y^\top}{\mathrm{d}t} Y}_{W}, \tag{7.21}$$

we can specify the dynamics of evolution for the parameters (X, S, Y). In particular, let $\Upsilon := X^\top \frac{\mathrm{d}g}{\mathrm{d}t} Y$, where $\frac{\mathrm{d}g}{\mathrm{d}t}$ is given by (7.20). Given initial values $(X(0), S(0), Y(0))$, we see that the equation for $S(t)$ is readily available,

$$\frac{\mathrm{d}S}{\mathrm{d}t} = \mathrm{diag}(\Upsilon), \tag{7.22}$$

whereas the two equations

$$\frac{\mathrm{d}X}{\mathrm{d}t} = XZ, \tag{7.23}$$

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = YW \tag{7.24}$$

can also be defined, since the skew-symmetric matrices Z and W can be retrieved from off-diagonal elements of Υ and S. In total, we have constructed a gradient flow for the objective function F in terms of the four matrix parameters (X, S, Y, R) that evolve on the manifold $\mathcal{O}(n) \times \mathbb{R}^n_{\times} \times \mathcal{O}(n) \times gl(n)$.

8. Structure-preserving dynamical systems

The notion of structure preservation has been put into practice in numerical linear algebra since its very early stage of development. The upper Hessenberg form has been used in the QR algorithm, the upper Hessenberg/triangular form in the QZ algorithm, and the bidiagonal form in the SVD algorithm (Golub and Van Loan 1996), to mention a few. These structures are not only preserved throughout the iterative processes, but also play a fundamental role in making the algorithms effective for computation.

Each of the three above-mentioned iterative schemes has a corresponding continuous analogue. It is well known that the generalized Toda flow preserves the tridiagonal form for symmetric matrices and the upper Hessenberg form for general matrices (Chu 1988, Watkins and Elsner 1988). The QZ flow and the SVD flow, on the other hand, were designed specifically to preserve the upper Hessenberg/triangular and the bidiagonal structures, respectively. Recall that the Lotka–Volterra equation discussed extensively in Section 6 is precisely the SVD flow applied to bidiagonal matrices.

As before, the meaning of structure should be interpreted broadly to include any invariant properties under the flow. The Toda flow, therefore, preserves at least two structures: the spectrum and the upper Hessenberg form. Likewise, the SVD flow preserves the singular values and the bidiagonal form. It then becomes interesting to ask whether there are other structures invariant under these flows. To distinguish these special matrix forms from other invariant properties to be discussed later, we shall use the term *zero structure* to refer collectively to any specific zero pattern of a matrix. The flip side of the question is equally interesting and perhaps more important: Given a set of structures related to a fixed matrix, can a dynamical system, continuous or discrete, be designed to preserve the specified structures?

The importance of structure preservation goes far beyond the realm of linear algebra alone. There are properties other than zero structures that we want to maintain. Stability and passivity preservation, for example, are highly desirable in model reduction (Antoulas 2005). Standard simplicity preservation allows a doubling algorithm to effectively separate stable and unstable eigenvalues when solving the discrete algebraic Riccati equation (Lin and Xu 2006). See also an interesting discussion by Mackey *et al.* (2003) for structured matrices arising in the context of a bilinear or sesquilinear form. A quick search for the key phrase 'structure preserving' over the internet brings up a wide range of applications across multiple disciplines. We will not and are unable to review the various situations in the literature where structure preservation is essential. However, it might be safe to state that structure preservation is essential in applications because it often makes possible more efficient computation, improves physical feasibility or interpretability, and is more robust.

In this section, we shall explore dynamical systems that preserve some interesting structures arising from linear algebra. We intend to disclose some of the structures that are elusive from consideration of the dynamical systems. Be warned that we have to pose several observations as conjectures because no mathematical proofs are available at present. Even so, numerical experiments strongly suggest that these conjectures should be true.

8.1. Staircase structure

The upper Hessenberg form is actually a special case of the more general form known as the staircase structure. Given a matrix $A = [a_{ij}] \in \mathbb{R}^{m\times n}$, define the step index for each column by

$$t_k(A) := \max\Big\{k, \max_{k<i\leq m}\big\{i|a_{ik} \neq 0\big\}\Big\}, \quad k = 1,\dots,n. \tag{8.1}$$

We say that A is in *staircase form* if and only if

$$t_k(A) \leq t_{k+1}(A), \quad k = 1,\dots,n-1. \tag{8.2}$$

Both of the following matrices, for example,

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & 0 & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}, \quad \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times \end{bmatrix}}_{\text{full staircase}}$$

are staircase matrices with step indices $\{1, 3, 4, 4, 5\}$. When there are no zero elements above the stairs, we say that the matrix is of *full staircase.*

Recall that the QR algorithm is the most efficient method for eigenvalue computation due to its stability and isospectrality. The following result by Arbenz and Golub (1995) identifies the zero structure that is preserved under the QR algorithm when applied to symmetric matrices.

Theorem 8.1. Assume that A_0 is symmetric. Let $\{A_k\}$ be the iterates generated by the QR algorithm (5.2). Then the following are true.

(1) If A_0 is reducible by some permutation matrix P, that is,

$$PA_0P^\top = \begin{bmatrix} A_{01} & A_{02} \\ 0 & A_{03} \end{bmatrix},$$

then each A_k is also reducible by means of the same permutation P.

(2) If A_0 is irreducible, then the zero pattern of A_0 is preserved throughout $\{A_k\}$ if and only if A_0 is a full staircase matrix.

Consider the zero structure of the following two 7×7 symmetric matrices,

$$\begin{bmatrix} \times & 0 & \times & 0 & \times & 0 & \times \\ 0 & \times & 0 & \times & 0 & \times & 0 \\ \times & 0 & \times & 0 & \times & 0 & \times \\ 0 & \times & 0 & \times & 0 & \times & 0 \\ \times & 0 & \times & 0 & \times & 0 & \times \\ 0 & \times & 0 & \times & 0 & \times & 0 \\ \times & 0 & \times & 0 & \times & 0 & \times \end{bmatrix}, \quad \begin{bmatrix} \times & 0 & \times & 0 & \times & \times & \times \\ 0 & \times & 0 & \times & 0 & \times & 0 \\ \times & 0 & \times & 0 & \times & 0 & \times \\ 0 & \times & 0 & \times & 0 & \times & 0 \\ \times & 0 & \times & 0 & \times & 0 & \times \\ \times & \times & 0 & \times & 0 & \times & 0 \\ \times & 0 & \times & 0 & \times & 0 & \times \end{bmatrix}, \tag{8.3}$$

which differ only at the $(1,6)$ and $(6,1)$ positions. The QR algorithm using these two matrices as the initial values produces very different behaviour. Theorem 8.1 asserts that the zero pattern for the left matrix is preserved because it is reducible, but the zero pattern for the right matrix is totally destroyed even after one iteration.

For non-symmetric matrices, the reducibility is not guaranteed to be preserved. However, the staircase form remains a sufficient, but not necessary, condition for shape preservation under the QR algorithm. Given the close relationship between the QR algorithm and the Toda flow, it should not be surprising that if X_0 is a staircase matrix, then so is $X(t)$ under the dynamical system (5.14) (Ashlock, Driessel and Hentzel 1997, Chu and Norris 1988).

For the generalized eigenvalue problem,

$$A_0\mathbf{x} = \lambda B_0\mathbf{x}, \tag{8.4}$$

a typical iterative scheme is the QZ algorithm. For practical purposes, the matrix A_0 is usually first reduced to an upper Hessenberg form and B_0 to an upper triangular form by orthogonal equivalence transformations. The basic idea behind the QZ algorithm is to simulate the effect of the QR algorithm on the matrix $B_0^{-1}A_0$ (assuming B_0 is invertible) without explicitly forming the inverse or the product. Throughout the QZ iteration, a critical component in the algorithm is that the upper Hessenberg/triangular structure is preserved.

Suppose now that a smooth orthogonal equivalence transformation has been applied to the pencil $B_0\lambda - A_0$,

$$\mathscr{L}(t) = Q(t)(B_0\lambda - A_0)Z(t), \quad Q(t), Z(t) \in \mathcal{O}(n). \tag{8.5}$$

Upon differentiation, the isospectral flow $\mathscr{L}(t)$ is necessarily governed by a differential system of the form

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}t} = \mathscr{L}R - L\mathscr{L}, \quad \mathscr{L}(0) = B_0\lambda - A_0, \tag{8.6}$$

where the coordinate transformation must satisfy the system

$$\begin{aligned}\frac{\mathrm{d}Q}{\mathrm{d}t} &= -LQ,\\ \frac{\mathrm{d}Z}{\mathrm{d}t} &= ZR,\end{aligned}$$

with some $L, R \in o(n)$. The choice of skew-symmetric matrix parameters $L(t)$ and $R(t)$ determines the dynamics. Write

$$\begin{aligned}X(t) &= Q(t)A_0Z(t),\\ Y(t) &= Q(t)B_0Z(t).\end{aligned}$$

To mimic the QZ algorithm, we prefer to choose $L(t)$ and $R(t)$ so that the resulting vector fields $\frac{\mathrm{d}X}{\mathrm{d}t}$ and $\frac{\mathrm{d}Y}{\mathrm{d}t}$ remain upper Hessenberg/triangular whenever $X(t)$ and $Y(t)$ are, respectively. Among many possibilities, one selection out of naïveté but with proper symmetry is the choice

$$L = \Pi_0(XY^{-1}), \tag{8.7}$$

$$R = \Pi_0(Y^{-1}X), \tag{8.8}$$

where the operator Π_0 is given in (5.13). Define the QZ flow accordingly by

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}t} = \mathscr{L}\Pi_0(Y^{-1}X) - \Pi_0(XY^{-1})\mathscr{L}, \quad \mathscr{L}(0) = B_0\lambda - A_0. \tag{8.9}$$

Note that if $X(t)$ and $Y(t)$ are upper Hessenberg/triangular, then both $L(t)$ and $R(t)$ are tridiagonal. Note also that if we define

$$E(t) := X(t)Y^{-1}(t), \tag{8.10}$$

$$F(t) := Y^{-1}(t)X(t), \tag{8.11}$$

then it can readily be proved that

$$\frac{\mathrm{d}E}{\mathrm{d}t} = [E, \Pi_0(E)], \tag{8.12}$$

$$\frac{\mathrm{d}F}{\mathrm{d}t} = [F, \Pi_0(F)]. \tag{8.13}$$

In other words, the QZ flow (8.9) is related to the QZ algorithm in the same way as the Toda flow is related to the QR algorithm. The convergence of the QZ flow therefore follows naturally from the dynamics of the Toda flow (Chu 1986*a*).

Thus far, the peculiar right-hand sides of (8.9) are designed solely for the purpose of maintaining the upper Hessenberg/triangular form. However, one interesting phenomenon as a by-product is worth mentioning. It has been observed that the QZ flow and, consequently, the corresponding QZ algorithm preserve the staircase structure. A more precise description of our empirical observation is given in the following conjecture, of which a rigorous proof has not been established at present.

Conjecture 8.2. If both A_0 and B_0 are staircase matrices, not necessarily of the same pattern, then the structures of A_0 and B_0 are preserved by $X(t)$ and $Y(t)$ under the QZ flow defined by (8.9), respectively.

We elaborate on the implication of Conjecture 8.2 a little bit more. The distinct zero patterns of the two matrices,

$$A_0 = \begin{bmatrix} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & 0 & \times \end{bmatrix}, \quad B_0 = \begin{bmatrix} \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & 0 & \times & \times \end{bmatrix},$$

for example, are preserved, respectively, in the QZ flow. It is not obvious why the separate stair structures are kept without interference. It is amazing that the procedure of 'mixing' Y^{-1}, which is usually full and dense with the structured X followed by the operations in the way specified in (8.9), will eventually separate and give back the original staircase structures of X and Y, respectively. Direct manipulation is hard to come by, because algebraic expression would be considerably complicated. Perhaps it is for this reason that the staircase structure has been reticent thus far. Although not necessarily of practical value, such a structure-preserving property of the QZ flow (and of the QZ algorithm) is mathematically intriguing.

An equally interesting structure-preserving property is also found in the SVD flow (6.2). Our original idea in deriving this particular matrix form of dynamical system was simply to maintain the bidiagonal structure (Chu 1986*b*). Because of this property, the SVD flow is reduced to the Lotka–Volterra equation (6.3) when B_0 is bidiagonal to begin with. Surprisingly, if we continue to use the SVD flow in its matrix form (6.2), then we have empirical evidence to support the following conjecture.

Conjecture 8.3. Suppose B_0 is a staircase matrix. Then the SVD flow $B(t)$ defined by (6.2) and the corresponding SVD algorithm maintains the same staircase structure.

For small size matrices, the validity of Conjecture 8.3 can be proved by an *ad hoc* calculation. We are curious whether there is a more elegant way to validate this conjecture in general.

Finally, we remark that the staircase form is only a sufficient condition for shape preservation under the SVD flow. There are other structures invariant under the dynamical system (6.2). The chessboard structure of the left matrix in (8.3), for example, is preserved under the SVD flow, but unlike the symmetric QR flow, the SVD flow does not preserve the reducibility.

8.2. Lancaster structure

The *Lancaster structure* of three given matrices M_0, C_0 and K_0 in $\mathbb{R}^{n\times n}$ refers to a linear pencil of the form (Gohberg, Lancaster and Rodman 1982)

$$\mathfrak{L}(\lambda) := \mathfrak{L}(\lambda; M_0, C_0, K_0) = \begin{bmatrix} C_0 & M_0 \\ M_0 & 0 \end{bmatrix} \lambda - \begin{bmatrix} -K_0 & 0 \\ 0 & M_0 \end{bmatrix}. \tag{8.14}$$

The matrices need not have any additional properties such as symmetry or positive definiteness. The Lancaster structure consists of more than just zero patterns. It also requires the matrix M_0 to appear at three specified locations. It is easy to see that the linear pencil (8.14) is equivalent to the quadratic pencil,

$$\mathfrak{Q}(\lambda) := \mathfrak{Q}(\lambda; M_0, C_0, K_0) = \lambda^2 M_0 + \lambda C_0 + K_0, \tag{8.15}$$

in the sense that

$$\left(\begin{bmatrix} C_0 & M_0 \\ M_0 & 0 \end{bmatrix} \lambda - \begin{bmatrix} -K_0 & 0 \\ 0 & M_0 \end{bmatrix} \right) \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = 0 \tag{8.16}$$

if and only if

$$\begin{cases} (\lambda C_0 + K_0)\mathbf{u} + \lambda M_0 \mathbf{v} = 0, \\ \qquad \lambda M_0 \mathbf{u} - M_0 \mathbf{v} \qquad = 0. \end{cases} \tag{8.17}$$

Indeed, if M_0 is non-singular, then we know further that $\mathbf{v} = \lambda \mathbf{u}$. Obviously, the Lancaster structure implies that if $\mathfrak{Q}(\lambda)$ is self-adjoint, then so is $\mathfrak{L}(\lambda)$. The eigen-information $(\lambda, \mathbf{u}) \in \mathbb{C}\times\mathbb{C}^n$ of the quadratic pencil $\mathfrak{Q}(\lambda)$ is critical to the understanding of the dynamical system

$$M_0\ddot{\mathbf{x}} + C_0\dot{\mathbf{x}} + K_0\mathbf{x} = f(t), \tag{8.18}$$

which arises frequently in many important applications, including applied mechanics, electrical oscillations, vibro-acoustics, fluid mechanics, and signal processing (Tisseur and Meerbergen 2001).

We are interested in the Lancaster structure because, in contrast to the common knowledge that generally no three matrices can be diagonalized simultaneously by equivalence transformations, it has been shown that for almost all quadratic pencils there exist real-valued $2n \times 2n$ real matrices Π_ℓ and Π_r such that

$$\Pi_\ell^\top \mathfrak{L}(\lambda)\Pi_r = \mathfrak{L}(\lambda; M_D, C_D, K_D), \tag{8.19}$$

where M_D, C_D, K_D are all real-valued $n \times n$ diagonal matrices. In other words, almost all n-degree-of-freedom second-order systems can be reduced to n totally independent single-degree-of-freedom second-order subsystems by real-valued isospectral transformations (Chu and Del Buono 2008*a*, Garvey, Friswell and Prells 2002*a*, 2002*b*). Such an isospectral transformation is significant in that it links the dynamical behaviour of a multiple-degree-of-freedom system directly to that of a system consisting of n independent single-degree-of-freedom subsystems. It breaks down the interlocking connectivity in the original system into totally disconnected subsystems while preserving the entire spectral properties. Thus it will be of great value in practice if the transformations Π_ℓ and Π_r can be found from any given pencil. We may consider (8.19) as a special kind of canonical form for the linear pencil (8.14).

The current theory of existence expresses Π_ℓ and Π_r in terms of the complete spectral information of $\mathfrak{L}(\lambda)$. The need for spectral information in the construction of Π_ℓ and Π_r is certainly not practical. Employing the notion of structure-preserving isospectral flows, it is possible to construct Π_ℓ and Π_r numerically without knowing the spectral information.

We first explore the 'orbit' of $\mathfrak{L}(\Lambda)$ under (Lancaster) structure-preserving equivalence transformations. Denote

$$\Pi_\ell = \begin{bmatrix} \ell_{11} & \ell_{12} \\ \ell_{21} & \ell_{22} \end{bmatrix}, \quad \Pi_r = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}, \tag{8.20}$$

where each ℓ_{ij} or r_{ij} is an $n \times n$ matrix. In order to maintain the Lancaster structure in the transformation $\Pi_\ell^\top \mathfrak{L}(\Lambda)\Pi_r$, it is necessary that the following five equations hold:

$$\begin{aligned} -\ell_{11}^\top K_0 r_{12} + \ell_{21}^\top M_0 r_{22} &= 0, \\ -\ell_{12}^\top K_0 r_{11} + \ell_{22}^\top M_0 r_{21} &= 0, \\ \ell_{12}^\top C_0 r_{12} + \ell_{22}^\top M_0 r_{12} + \ell_{12}^\top M_0 r_{22} &= 0, \\ \ell_{11}^\top C_0 r_{12} + \ell_{21}^\top M_0 r_{12} + \ell_{11}^\top M_0 r_{22} &= \ell_{12}^\top C_0 r_{11} + \ell_{22}^\top M_0 r_{11} + \ell_{12}^\top M_0 r_{21} \\ &= -\ell_{12}^\top K_0 r_{12} + \ell_{22}^\top M_0 r_{22}. \end{aligned} \tag{8.21}$$

Ultimately, in order to produce the canonical form, the matrices Π_ℓ and Π_r

must be such that the left-hand sides of the following three expressions,

$$\begin{aligned} -\ell_{12}^\top K_0 r_{12} + \ell_{22}^\top M_0 r_{22} &= M_D, \\ \ell_{11}^\top C_0 r_{11} + \ell_{21}^\top M_0 r_{11} + \ell_{11}^\top M_0 r_{21} &= C_D, \\ \ell_{11}^\top K_0 r_{11} - \ell_{21}^\top M_0 r_{21} &= K_D, \end{aligned} \tag{8.22}$$

are diagonal matrices. The conditions (8.21) and (8.22) together constitute a homogeneous second-degree polynomial system of $8n^2 - 3n$ equations in $8n^2$ unknowns. It is not obvious how the nonlinear algebraic system could be solved analytically, but the underdetermined system does imply that there is plenty of room to choose the transformation matrices Π_ℓ and Π_r. In particular, a smooth path connecting (M_0, C_0, K_0) to (M_D, C_D, K_D) can be defined.

To characterize the path, denote the Lancaster pair in (8.14) by (A_0, B_0), where

$$A_0 = \begin{bmatrix} -K_0 & 0 \\ 0 & M_0 \end{bmatrix}, \quad B_0 = \begin{bmatrix} C_0 & M_0 \\ M_0 & 0 \end{bmatrix}. \tag{8.23}$$

We now develop two one-parameter families $T_\ell(t)$ and $T_r(t)$ in $\mathbb{R}^{2n\times 2n}$ of structure-preserving transformations starting with $T_\ell(0) = T_r(0) = I_{2n}$. Assume that these families of transformations act on (A_0, B_0) via the form

$$\begin{aligned} A(t) &= T_\ell^\top(t) A_0 T_r(t), \\ B(t) &= T_\ell^\top(t) B_0 T_r(t), \end{aligned}$$

respectively. Clearly, regardless of how $T_\ell(t)$ and $T_R(t)$ are defined, the transformed pencil $(A(t), B(t))$ is isospectral to (A_0, B_0) for any t. For simplicity, we limit ourselves to a special class of transformations where matrices $T_\ell(t)$ and $T_r(t)$ are governed by the dynamical systems

$$\frac{\mathrm{d}T_\ell(t)}{\mathrm{d}t} = T_\ell(t)\mathcal{L}(t) = T_\ell(t) \begin{bmatrix} L_{11}(t) & L_{12}(t) \\ L_{21}(t) & L_{22}(t) \end{bmatrix}, \tag{8.24}$$

$$\frac{\mathrm{d}T_r(t)}{\mathrm{d}t} = T_r(t)\mathcal{R}(t) = T_r(t) \begin{bmatrix} R_{11}(t) & R_{12}(t) \\ R_{21}(t) & R_{22}(t) \end{bmatrix}, \tag{8.25}$$

respectively, where each $L_{ij}(t)$ or $R_{ij}(t)$, $i, j = 1, 2$, is a $n \times n$ real one-parameter matrix yet to be defined. Upon substitution, we observe that the pencil

$$\mathscr{L}(t) = B(t)\lambda - A(t)$$

must satisfy the equation

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}t} = \mathcal{L}^\top \mathscr{L} + \mathscr{L}\mathcal{R}, \quad \mathscr{L}(0) = \mathfrak{L}(\lambda).$$

It is interesting to note that these differential equations are similar to those discussed by Bloch and Iserles (2006), which led to a Lie–Poisson system. By insisting that $(A(t), B(t))$ maintains the Lancaster structure throughout the transformation, that is,

$$A(t) = \begin{bmatrix} K(t) & 0 \\ 0 & -M(t) \end{bmatrix}, \quad B(t) = \begin{bmatrix} C(t) & M(t) \\ M(t) & 0 \end{bmatrix}, \tag{8.26}$$

we see that the entries of $\mathcal{L}(t)$ and $\mathcal{R}(t)$ should satisfy

$$\begin{align} R_{12} &= -DM, \tag{8.27}\\ R_{21} &= DK, \tag{8.28}\\ L_{12} &= D^\top M^\top, \tag{8.29}\\ L_{21} &= -D^\top K^\top, \tag{8.30}\\ L_{11} - L_{22} &= D^\top C^\top, \tag{8.31}\\ R_{11} - R_{22} &= -DC, \tag{8.32} \end{align}$$

where $D \in \mathbb{R}^{n\times n}$ is an arbitrary matrix parameter. Note that hidden in (8.31) and (8.32) are two other free matrix parameters denoted by N_L and N_R, respectively.

There are several possible ways to choose the parameters and to arrange the diagonal blocks of $\mathcal{L}(t)$ and $\mathcal{R}(t)$. For instance, corresponding to the choice

$$\mathcal{L} = \begin{bmatrix} D^\top & 0 \\ 0 & D^\top \end{bmatrix} \begin{bmatrix} \frac{C^\top}{2} & M^\top \\ -K^\top & -\frac{C^\top}{2} \end{bmatrix} + \begin{bmatrix} N_L^\top & 0 \\ 0 & N_L^\top \end{bmatrix}, \tag{8.33}$$

$$\mathcal{R} = \begin{bmatrix} D & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} -\frac{C}{2} & -M \\ K & \frac{C}{2} \end{bmatrix} + \begin{bmatrix} N_R & 0 \\ 0 & N_R \end{bmatrix}, \tag{8.34}$$

an isospectral flow of the triplet $(M(t), C(t), K(t))$ can be defined by the autonomous system

$$\begin{aligned} \frac{\mathrm{d}K}{\mathrm{d}t} &= \frac{1}{2}(CDK - KDC) + N_L^\top K + KN_R, \\ \frac{\mathrm{d}C}{\mathrm{d}t} &= (MDK - KDM) + N_L^\top C + CN_R, \\ \frac{\mathrm{d}M}{\mathrm{d}t} &= \frac{1}{2}(MDC - CDM) + N_L^\top M + MN_R. \end{aligned} \tag{8.35}$$

Furthermore, by assuming $N_R(t) = N_L(t)$, the symmetry retained in the matrix parameter D has the effect of preserving the symmetry for the flow $(M(t), K(t), C(t))$ defined by the dynamical system (8.35). The various symmetry-preserving properties are summarized in Table 8.1.

The remaining task is to ‘control’ the free matrix parameters in such a way that the structure-preserving isospectral flow $(A(t), B(t))$ converges to the

Table 8.1. Preserving symmetry of $(M(t), C(t), K(t))$ by $D(t)$, if $N_R(t) = N_L(t)$.

$D(t)$	$M(t)$	$C(t)$	$K(t)$
skew-symmetric	symmetric	symmetric	symmetric
symmetric	symmetric	skew-symmetric	symmetric
symmetric	skew-symmetric	skew-symmetric	skew-symmetric
skew-symmetric	skew-symmetric	symmetric	skew-symmetric

canonical form (8.19). Consider the idea of minimizing a given sufficiently smooth objection function $f : \mathbb{R}^n \to \mathbb{R}$ whose state variable $\mathbf{x} \in \mathbb{R}^n$ is constrained to the integral curve of

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = g(\mathbf{x})\mathbf{u}, \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{8.36}$$

where $\mathbf{g} : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ is a fixed function and $\mathbf{u}(t) \in \mathbb{R}^m$ is the control. For minimization, one way to choose the control $\mathbf{u}$ is to make the vector $\dot{\mathbf{x}}$ as close to $-\nabla f(\mathbf{x})$ as possible. This amounts to the selection of the least squares solution $\mathbf{u}$ defined by

$$\mathbf{u}(t) = -g(\mathbf{x}(t))^\dagger \nabla f(\mathbf{x}(t)), \tag{8.37}$$

where $g(\mathbf{x})^\dagger$ stands for the Moore–Penrose generalized inverse of $g(\mathbf{x})$. It follows that the closed-loop[1] dynamical system,

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -g(\mathbf{x})g(\mathbf{x})^\dagger \nabla f(\mathbf{x}), \tag{8.38}$$

defines a descent flow $\mathbf{x}(t)$ for the objective function $f(\mathbf{x})$.

For our application, we wish the structure-preserving isospectral flow $(M(t), C(t), K(t))$ to be driven to diagonal matrices. However, unlike the isospectral flow by orthogonal transformations, our flow $(M(t), C(t), K(t))$ preserves only the Lancaster structure but not the norm. Thus, we seek matrix parameters N_R, N_L and D to minimize the function

$$\begin{aligned} f(K, C, M) := \frac{1}{2}\big\{\|\mathrm{offdiag}(M)\|_F^2 + \|\mathrm{offdiag}(C)\|_F^2 + \|\mathrm{offdiag}(K)\|_F^2\big\} \\ + \delta h(\mathrm{diag}(M), \mathrm{diag}(C), \mathrm{diag}(K)), \end{aligned} \tag{8.39}$$

subject to the condition that (M, C, K) is governed by the differential system (8.35). The crux of choosing this particular objective function is to minimize

[1] The system (8.36) is 'closed-loop' in the sense that it is now self-contained: the reference to $\mathbf{u}$ is no longer needed directly.

the off-diagonal entries of (M, C, K) while using the function h to regulate the behaviour of the diagonal entries by a factor of δ. Note that we may rewrite the dynamical system (8.35) in the same control scheme,

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{vec}(M)\\ \mathbf{vec}(C)\\ \mathbf{vec}(K)\end{bmatrix} = \begin{bmatrix}\frac{1}{2}(K\otimes C - C\otimes K) & K\otimes I & I\otimes K\\ K\otimes M - M\otimes K & C\otimes I & I\otimes C\\ \frac{1}{2}(C\otimes M - M\otimes C) & M\otimes I & I\otimes M\end{bmatrix}\begin{bmatrix}\mathbf{vec}(D)\\ \mathbf{vec}(N_L^\top)\\ \mathbf{vec}(N_R)\end{bmatrix},$$

as that of (8.36). The above-mentioned control strategy fits perfectly. In this way, we have developed a 'controlled' gradient flow which not only preserves both the Lancaster structure and the isospectrality, but also moves in the direction of total decoupling of a quadratic pencil. More detailed discussion can be found in Chu and Del Buono (2008*b*).

8.3. Hamiltonian structure

A matrix $\mathcal{H} \in \mathbb{R}^{2n\times 2n}$ is said to be *Hamiltonian* if it satisfies the relationship $(\mathcal{H}J)^\top = \mathcal{H}J$, where

$$J := \begin{bmatrix}0 & I_n\\ -I_n & 0\end{bmatrix}.$$

It is easy to see that a Hamiltonian matrix must have the structure

$$\mathcal{H} = \begin{bmatrix}M & P\\ Q & -M^\top\end{bmatrix}, \quad P \text{ and } Q \text{ are symmetric.} \tag{8.40}$$

Likewise, a *skew-Hamiltonian* matrix $\mathcal{W}$ satisfies $(\mathcal{W}J)^\top = -\mathcal{W}J$, and has the structure

$$\mathcal{W} = \begin{bmatrix}M & F\\ G & M^\top\end{bmatrix}, \quad F \text{ and } G \text{ are skew-symmetric.} \tag{8.41}$$

Without causing ambiguity, we shall refer to a form of either (8.40) or (8.41) collectively as a *Hamiltonian structure*. We shall call up the more specific reference to a Hamiltonian matrix or a skew-Hamiltonian matrix only when a clear distinction is necessary. The notation $\mathcal{H}$ and $\mathcal{W}$, specifically reserved for the Hamiltonian matrix and the skew-Hamiltonian matrix, respectively, should offer a clue as to which structure we are referring to in the context.

Matrices with Hamiltonian structure arise from a variety of applications, including systems and controls, algebraic Riccati equations, and quadratic eigenvalue problems (Benner, Kressner and Mehrmann 2005). Inherent in the Hamiltonian structure are many interesting properties. For example, the eigenvalues of $\mathcal{H}$ are symmetric with respect to the imaginary axis, and the eigenvalues of $\mathcal{W}$ have even algebraic and geometric multiplicities. These properties are often tied to the physical settings that lead to the underlying structure. For feasibility and interpretability, therefore, any transformation of $\mathcal{H}$ or $\mathcal{W}$ should respect the original Hamiltonian structure. Because conventional algorithms usually fail to meet this requirement, there has been considerable research effort to derive special methods for matrices with Hamiltonian structure. Some principal references will be given in the course of our presentation. Needless to say, special methods mean more delicate manipulations. The description of these methods are usually quite involved.

In this section, we are mainly interested in deriving continuous dynamical systems that mimic existing iterative schemes. In contrast to the iterative methods, most of our Hamiltonian structure-preserving dynamical systems can be characterized as a single line equation. Nonetheless, despite the fact that our extensive numerical experiments have given convincing evidence for the resulting dynamical behaviour, a major drawback in our current work is the lack of a complete asymptotic analysis of these differential systems. We have to leave these gaps as conjectures in this presentation.

To maintain the Hamiltonian structure, it is typical in practice that a similarity transformation of $\mathcal{H}$ or $\mathcal{W}$ should involve only symplectic matrices $S \in \mathbb{R}^{2n\times 2n}$. A symplectic matrix S must satisfy the condition

$$S^\top J S = J, \tag{8.42}$$

which naturally implies the symmetry $SJS^\top = J$ as well. Recall that we mentioned earlier in Table 7.1 that symplectic matrices form a group $\mathcal{S}p(2n)$. For numerical stability, it is often further required that the transformation matrix S be orthogonal symplectic.

The following three facts, leading to the particular structure called the *real Schur–Hamiltonian form* in the first two cases and the *URV form* in the third case, play fundamental roles in the computation of eigenvalues for matrices with Hamiltonian structure.

Theorem 8.4. Given $\mathcal{H}, \mathcal{W} \in \mathbb{R}^{2n\times 2n}$ which are Hamiltonian and skew-Hamiltonian matrices, respectively, then we have the following.

(1) (Paige and Van Loan 1981) If $\mathcal{H}$ has no purely imaginary eigenvalues, then there exists an orthogonal symplectic matrix $U \in \mathbb{R}^{2n\times 2n}$ such that $\widetilde{\mathcal{H}} = U^\top \mathcal{H} U$ is Hamiltonian and is of the form

$$\widetilde{\mathcal{H}} = \begin{bmatrix} R & P \\ 0 & -R^\top \end{bmatrix}, \tag{8.43}$$

where P is symmetric and R is upper quasi-triangular.

(2) (Van Loan 1984) There exists an orthogonal symplectic matrix $U \in \mathbb{R}^{2n\times 2n}$ such that $\widetilde{\mathcal{W}} = U^\top \mathcal{W} U$ is skew-Hamiltonian, and is of the form

$$\widetilde{\mathcal{W}} = \begin{bmatrix} R & F \\ 0 & R^\top \end{bmatrix}, \tag{8.44}$$

where F is skew-symmetric and R is upper quasi-triangular.

(3) (Benner *et al.* 2005) There exist orthogonal symplectic matrices $U, V \in \mathbb{R}^{2n\times 2n}$ such that $\widehat{\mathcal{H}} = U^\top \mathcal{H} V$ is of the form

$$\widehat{\mathcal{H}} = \begin{bmatrix} T & N \\ 0 & R^\top \end{bmatrix}, \tag{8.45}$$

where N has no particular structure, T is upper triangular and R is upper quasi-triangular.

Evidently, being able to reduce a matrix of Hamiltonian structure to its Schur–Hamiltonian form is sufficient for retrieving eigenvalue information. Most existing numerical methods for eigenvalue problems with Hamiltonian structure consist of two steps: first, endeavour to obtain the reduced form and, secondly, employ some classical iterative schemes to solve the reduced eigenproblem.

Currently, stable procedures for computing eigenvalues of skew-Hamiltonian matrices are well developed (Benner *et al.* 2005, Van Loan 1984). For Hamiltonian matrices, the task is much harder. The prevailing idea is to square a Hamiltonian $\mathcal{H}$ due to the fact that $\mathcal{H}^2$ is skew-Hamiltonian. Indeed, by (8.45), we see that $\mathcal{H}^2$ can be factorized as

$$U^\top \mathcal{H}^2 U = \begin{bmatrix} -TR & TN^\top - NT^\top \\ 0 & -R^\top T^\top \end{bmatrix}, \tag{8.46}$$

showing that the eigenvalues of $\mathcal{H}$ are the square roots of the eigenvalues from the matrix $-TR$. The $2n \times 2n$ eigenvalue problem is therefore effectively halved. A QZ-type algorithm can be applied to find the eigenvalues of the product TR without explicitly forming the product. Implementation details can be found in the paper by Benner and Kressner (2006). A similar idea but with improved invariant subspace computation is explored in Chu, Liu and Mehrmann (2007). We shall present in the following an interesting contrast that a continuous approach is easier to formulate for the Hamiltonian eigenproblem than for the skew-Hamiltonian eigenproblem.

In a spirit similar to that of the QR, the QZ or the SVD algorithms, we are interested in deriving dynamical systems that can realize the Schur–Hamiltonian form or its like. Towards that end, we need to understand how a smooth curve $S(t)$ moves on the manifold of symplectic group $\mathcal{S}p(2n)$. It suffices to know that the tangent space $\mathfrak{g} = \mathcal{T}_{I_{2n}}\mathcal{S}p(2n)$ for $\mathcal{S}p(2n)$ at the identity is simply the collection of Hamiltonian matrices. The tangent vectors of $S(t)$ must be given by

$$\frac{\mathrm{d}S}{\mathrm{d}t} = S\mathfrak{K}, \quad (\text{or } \mathfrak{K}S), \tag{8.47}$$

where $\mathfrak{K}$ is Hamiltonian. If the symplectic $S(t)$ is also orthogonal, then the

Hamiltonian matrix $\mathfrak{K}$ must be of the special form

$$\mathfrak{K} = \begin{bmatrix} M & -Q \\ Q & M \end{bmatrix}, \tag{8.48}$$

where M is skew-symmetric and Q is symmetric.

We demonstrate a simple application of (8.48) to the Hamiltonian eigenproblem. Given a matrix $\mathcal{H}_0 \in \mathbb{R}^{2n\times 2n}$, consider a special kind of Lax dynamical system described in (5.4),

$$\frac{\mathrm{d}X}{\mathrm{d}t} = [X, \mathcal{P}_0(X)], \quad X(0) = \mathcal{H}_0, \tag{8.49}$$

where the operator $\mathcal{P}_0$ acting on X is defined to be the skew-symmetric matrix,

$$\mathcal{P}_0(X) := \begin{bmatrix} 0 & -X_{21}^\top \\ X_{21} & 0 \end{bmatrix}, \tag{8.50}$$

if X is partitioned into four blocks of size $n \times n$,

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}.$$

Following (5.6), define the parameter dynamical system

$$\frac{\mathrm{d}g}{\mathrm{d}t} = g\mathcal{P}_0(X), \quad g(0) = I_{2n}. \tag{8.51}$$

Note that if $\mathcal{P}_0(X)$ is Hamiltonian, then $g(t)$ is automatically orthogonal symplectic. In particular, if $\mathcal{H}_0$ is Hamiltonian to begin with, then we know by Theorem 5.1 that $X(t) = g^\top(t)\mathcal{H}_0 g(t)$ remains Hamiltonian for all t. Under some mild conditions, it can be proved that $X(t)$ converges to an upper block triangular form, that is, $X_{21}(t) \longrightarrow 0$ as $t \longrightarrow \infty$ (Chu and Norris 1988). Though the limit point of the isospectral flow (8.49) is not exactly of the Schur–Hamiltonian form, it suffices to halve the Hamiltonian eigenproblem. The flow approach is remarkably simple, given that in the literature the Hamiltonian eigenproblem is known to be notoriously hard to solve by iterative methods.

Unfortunately, the corresponding $\mathcal{P}_0(X)$ is not Hamiltonian if X is skew-Hamiltonian. The simple dynamical system (8.49) therefore cannot preserve the skew-Hamiltonian structure. Since the skew-Hamiltonian eigenproblem is supposed to be relatively easier to handle than the Hamiltonian eigenproblem by iterative methods, it becomes interesting to ask whether the Schur–Hamiltonian form of a skew-Hamiltonian matrix $\mathcal{W}_0$ can ever be realized continuously. We offer a partial answer that looks pleasingly neat in theory, but is probably of little use in practice.

It is known that every real skew-Hamiltonian matrix has a real Hamiltonian square root (Faßbender, Mackey, Mackey and Xu 1999). Thus, given a skew-Hamiltonian matrix $\mathcal{W}_0$, if we define $\mathcal{H}_0$ to be its real Hamiltonian square root and define $X(t)$ according to (8.49), then the corresponding $\mathcal{W}(t) = X^2(t)$ is skew-Hamiltonian and will converge to an upper block triangular form. In particular, the very same parameter $g(t)$ defined in (8.51) (in terms of the Hamiltonian square root $X(t)$) serves as the continuous coordinate transformation for $\mathcal{W}(t) = g^\top(t)\mathcal{W}_0 g(t)$ and leads to convergence. It is not difficult to verify that symbolically we can write the motion of $\mathcal{W}(t)$ via the dynamical system

$$\frac{\mathrm{d}\mathcal{W}}{\mathrm{d}t} = [\mathcal{W}, \mathcal{P}_0(\mathcal{W}^{1/2})], \quad \mathcal{W}(0) = \mathcal{W}_0, \tag{8.52}$$

where $\mathcal{W}^{1/2}$ represents the real Hamiltonian square root of $\mathcal{W}$. We hasten to point out that caution must be taken in the above expression because a skew-Hamiltonian matrix $\mathcal{W}$ has infinitely many Hamiltonian square roots (Faßbender *et al.* 1999).

In the Lax dynamical system (5.14), the operation $\Pi_0(X)$ provides the magic of convergence to the real Schur form for a general square matrix X_0. We seek a similar dynamical system that converges to the real Schur–Hamiltonian for a Hamiltonian matrix $\mathcal{H}_0$. The operator $\mathcal{P}_1$ applied to a Hamiltonian matrix X via the definition

$$\mathcal{P}_1(X) := \begin{bmatrix} \Pi_0(X_{11}) & -X_{21} \\ X_{21} & \Pi_0(X_{11}) \end{bmatrix} \tag{8.53}$$

appears to be a compromise between the overall $\Pi_0(X)$ required by the *QR* flow for reaching sensible convergence and the form (8.48) required by the orthogonal symplecticity for keeping the Hamiltonian structure. The two operators Π_0 and $\mathcal{P}_1$ for a Hamiltonian matrix X differ only in the $(2,2)$-block. We propose the dynamical system

$$\frac{\mathrm{d}\mathcal{H}}{\mathrm{d}t} = [\mathcal{H}, \mathcal{P}_1(\mathcal{H})], \quad \mathcal{H}(0) = \mathcal{H}_0, \tag{8.54}$$

for finding the real Schur–Hamiltonian form of a Hamiltonian matrix $\mathcal{H}_0$. The following conjecture characterizes the convergence behaviour we have observed numerically, but we cannot offer a theoretical proof for the present.

Conjecture 8.5. Suppose $\mathcal{H}_0$ is Hamiltonian with no purely imaginary eigenvalues. Then the solution flow $\mathcal{H}(t)$ of (8.54) remains Hamiltonian and converges to the real Schur–Hamiltonian form as is specified in (8.43).

If the square root is interpreted in the same way as in (8.52), then a similar conjecture can be made for the system

$$\frac{\mathrm{d}\mathcal{W}}{\mathrm{d}t} = [\mathcal{W}, \mathcal{P}_1(\mathcal{W}^{1/2})], \quad \mathcal{W}(0) = \mathcal{W}_0. \tag{8.55}$$

The solution flow $\mathcal{W}(t)$ preserves the skew-Hamiltonian structure of an initial matrix $\mathcal{W}_0$ and converges to the real Schur skew-Hamiltonian form as is characterized in (8.44).

Regarding the *URV* decomposition, it is necessary that a flow $X(t) = U^\top(t)X_0V(t)$ satisfies a differential equation of the form

$$\frac{\mathrm{d}X}{\mathrm{d}t} = XR - LX, \quad X(0) = X_0, \tag{8.56}$$

where the coordinate transformations are governed by

$$\frac{\mathrm{d}U}{\mathrm{d}t} = -UL^\top, \tag{8.57}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} = VR, \tag{8.58}$$

with L and R to be determined. The setting thus far is very similar to that of the SVD flow. Let the operator $\mathcal{P}_3$ denote a generalization of $\mathcal{P}_0$ in that the partition of X is not necessarily at the midpoint of its diagonal. In particular, the off-diagonal block X_{21} can be of size $(2n-k)\times k$ with $k \leq n$. Consider the dynamical system

$$\frac{\mathrm{d}X}{\mathrm{d}t} = X\mathcal{P}_3(X^\top X) - \mathcal{P}_3(XX^\top)X, \quad X(0) = X_0, \tag{8.59}$$

for a general $2n \times 2n$ matrix X_0, Note that (8.59) is analogous to the SVD flow (6.2) except that P_3 is used in the place of Π_0. Clearly, $X(t)$ maintains the same singular values as X_0. Numerical experiments support the following conjecture, which seems new and interesting.

Conjecture 8.6. Given a general $2n\times 2n$ matrix X_0 with distinct singular values and an integer $k \leq n$, the solution flow $X(t)$ of (8.59) converges to a block diagonal matrix $\operatorname{diag}\{\widehat{X}_{11}, \widehat{X}_{22}\}$ of size $k \times k$ and $(2n-k)\times(2n-k)$, respectively. Furthermore, the singular values of $\widehat{X}_{11}$ are the first k largest singular values of X_0.

The coordinate transformations involved in Conjecture 8.6 are orthogonal similarity at most. To really achieve the *URV* decomposition specified in Theorem 8.4 part (3) for a Hamiltonian matrix $\mathcal{H}_0$, we have to employ orthogonal symplectic transformations. The clue comes at recognizing from (8.46) that the U transformation that does the *URV* decomposition for $\mathcal{H}_0$ should be the same U transformation that does the real Schur–Hamiltonian form for $\mathcal{H}_0$. That is, by Conjecture 8.5, $L = \mathcal{P}_1(U^\top\mathcal{H}_0U)$. Similarly, the V matrix in the *URV* decomposition should be the same V matrix that transforms $\mathcal{H}_0^\top$ to *lower* quasi-triangular Schur–Hamiltonian form. That is, by defining the operator

$$\mathcal{P}_2(X) := \begin{bmatrix} -\Pi_0(X_{11}^\top) & X_{12} \\ -X_{12} & -\Pi_0(X_{11}^\top) \end{bmatrix} \tag{8.60}$$

for a given Hamiltonian matrix X, we take $R = \mathcal{P}_2(V^\top \mathcal{H}_0^\top V)$. We are interested in a *URV* flow $X(t) = U^\top(t)\mathcal{H}_0 V(t)$. From the relations

$$\begin{aligned} U^\top \mathcal{H}_0^2 U &= XJX^\top J, \\ V^\top \mathcal{H}_0^2 V &= X^\top JXJ, \end{aligned}$$

we can express the *URV* flow symbolically through the autonomous dynamical system

$$\frac{\mathrm{d}X}{\mathrm{d}t} = X\mathcal{P}_2((X^\top JXJ)^{1/2}) - \mathcal{P}_1((XJX^\top J)^{1/2})X, \quad X(0) = \mathcal{H}_0, \tag{8.61}$$

where again $\mathcal{W}^{1/2}$ represents a proper real Hamiltonian square root of the skew-Hamiltonian matrix $\mathcal{W}$.

Hamiltonian structure-preserving differential systems like (8.49), (8.54), or even (8.61) might not be practically useful right away, but they neatly represent complicated dynamics that otherwise will be quite tedious, if not formidable, to describe by iterative procedures. Maybe, and only maybe, these flows could be suitably discretized and lead to effective numerical algorithms. One precedent is the realization of the *vdLV* algorithm for the Lotka–Volterra equation which, when first proposed two decades ago, was regarded as 'impractical' as well. These flows might be worth further investigation.

8.4. Hamiltonian pencils

We have already seen linear pencils with the Lancaster structure resulting from a special linearization of a quadratic pencil. There are also linear pencils with the Hamiltonian structure. To start off, two different definitions in the literature must be carefully differentiated from each other. First, a linear pencil $B\lambda - A$ is said to be *Hamiltonian* if and only if

$$BJA^\top = -AJB^\top. \tag{8.62}$$

This definition is equivalent to saying that the product $B^{-1}A$ is Hamiltonian, provided B^{-1} exists (Lin, Mehrmann and Xu 1999). If λ is an eigenvalue of a Hamiltonian pencil, then so are $-\lambda, \overline{\lambda}, -\overline{\lambda}$. Secondly, a linear pencil $B\lambda - A$ is said to be *skew-Hamiltonian/Hamiltonian* (sHH) if and only if B is skew-Hamiltonian and A is Hamiltonian (Mehl 1999). Pencils with the sHH structure appear in gyroscopic systems, structural mechanics, linear response theory, quadratic optimal control problems and many other applications (Benner, Byers, Mehrmann and Xu 2002, Mehrmann and Watkins 2000). Although it is a natural generalization in mathematics, we have rarely seen Hamiltonian/Hamiltonian (HH) pencils in applications. One indicator that an HH pencil is probably too general to deserve any special attention is the fact that the HH structure does not generally carry any additional symmetric properties in its spectrum. We note, for example, that any self-adjoint quadratic pencil (8.15) can be linearized as the pencil

$$\begin{bmatrix} M_0 & 0 \\ -C_0 & -M_0 \end{bmatrix} \lambda - \begin{bmatrix} 0 & M_0 \\ K_0 & 0 \end{bmatrix}, \tag{8.63}$$

which is equivalent to the Lancaster pair (8.14), is of the HH structure, and can literally have arbitrary eigenvalues.

In analogy to (8.5), the one-parameter isospectral flow

$$\mathscr{L}(t) = Q(t)\big(B_0\lambda - A_0\big)Z(t)$$

should satisfy a differential equation of the form

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}t} = \mathscr{L}R - L\mathscr{L}, \quad \mathscr{L}(0) = B_0\lambda - A_0, \tag{8.64}$$

where the coordinate transformations are governed by

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = -LQ, \tag{8.65}$$

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = ZR. \tag{8.66}$$

with L and R to be determined. So far, this setting is similar to the QZ flow except that the definition of the two matrices L and R needs to be further specified. The conventional condition that L and R be skew-symmetric so that $Q(t)$ and $Z(t)$ are orthogonal is certainly assumed in all cases, but we are more interested in specifying conditions on L and R so as to maintain the Hamiltonian structure. Besides, we are further interested in using L and R to establish limiting behaviour of $\mathscr{L}(t)$ that might be of some practical usages. We outline some general ideas below.

We first consider the sHH pencils. Suppose that $\mathscr{L}(0) = \mathcal{W}_0\lambda - \mathcal{H}_0$ is of the sHH structure to begin with. Write

$$\mathscr{L}(t) = \mathcal{W}(t)\lambda - \mathcal{H}(t).$$

In order that $\mathscr{L}(t)$ maintains the sHH structure for all t, it is necessary that $\mathcal{W}R - L\mathcal{W}$ and $\mathcal{H}R - L\mathcal{H}$ remain skew-Hamiltonian and Hamiltonian, respectively. A straightforward algebraic manipulation shows that a sufficient condition for this to happen is that

$$L = JR^\top J. \tag{8.67}$$

Consequently, $Q(t)$ and $Z(t)$ can be interchanged via the relationship

$$Z(t) = JQ^\top(t)J, \tag{8.68}$$

$$Q(t) = JZ^\top(t)J. \tag{8.69}$$

Only one coordinate transformation of either (8.65) or (8.66) is needed for the isospectral flow of an sHH pencil.

For any given $2n \times 2n$ matrix X, define a new operator $\mathcal{P}_4$ by

$$\mathcal{P}_4(X) := \begin{bmatrix} \Pi_0(X_{11}) & -X_{21}^\top \\ X_{21} & -\Pi_0(X_{22}^\top) \end{bmatrix}. \tag{8.70}$$

Observe that $\mathcal{P}_4(X)$ is almost identical to $\Pi_0(X)$ except for a 'twist' at the $(2,2)$ block. Take the definitions

$$R := \mathcal{P}_4(\mathcal{W}^{-1}\mathcal{H}), \tag{8.71}$$

$$L := \mathcal{P}_4(\mathcal{H}\mathcal{W}^{-1}). \tag{8.72}$$

It is easy to see that the relationship

$$\mathcal{H}\mathcal{W}^{-1} = J(\mathcal{W}^{-1}\mathcal{H})^\top J \tag{8.73}$$

holds for every sHH pencil. A direct substitution then shows that the sufficient condition (8.67) is satisfied. In this way, we find that the dynamical system

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}t} = \mathscr{L}\mathcal{P}_4(\mathcal{W}^{-1}\mathcal{H}) - \mathcal{P}_4(\mathcal{H}\mathcal{W}^{-1})\mathscr{L}, \quad \mathscr{L}(0) = B_0\lambda - A_0, \tag{8.74}$$

defines an sHH flow which can be expressed as

$$\mathscr{L}(t) = JZ^\top(t)J(\mathcal{W}_0\lambda - \mathcal{H}_0)Z(t). \tag{8.75}$$

Had $\mathcal{P}_4$ been taken as Π_0, we would have precisely the standard QZ flow described earlier and the convergence behaviour of the QZ flow is well understood. With the little flip at the $(2,2)$ block in $\mathcal{P}_4$, we maintain the sHH structure and we can almost expect that a similar convergence behaviour will occur. We conceive the following conjecture from our numerical observation. Its assertion is in agreement with the sHH Schur form characterized in Benner *et al.* (2002). If the convergence can be proved, then we have a very simple way to realize the canonical form.

Conjecture 8.7. Suppose $\mathscr{L}(0)$ is an sHH pencil to begin with. Then the flow (8.75) with R defined by (8.71) maintains the sHH structure and converges to the canonical form

$$\widetilde{\mathscr{L}} = \begin{bmatrix} \widetilde{\mathcal{W}}_{11} & \widetilde{W}_{12} \\ 0 & \widetilde{W}_{11}^\top \end{bmatrix}\lambda - \begin{bmatrix} \widetilde{\mathcal{H}}_{11} & \widetilde{\mathcal{H}}_{12} \\ 0 & -\widetilde{\mathcal{H}}_{11}^\top \end{bmatrix},$$

where $\widetilde{W}_{11}$ and $\widetilde{H}_{11}$ are upper quasi-triangular, $\widetilde{W}_{12}$ is skew-symmetric, and $\widetilde{H}_{12}$ is symmetric, respectively.

We next consider the Hamiltonian pencils. It is easy to verify that $B\lambda - A$ is Hamiltonian if and only if $Q(B\lambda - A)Z$ is Hamiltonian for arbitrary nonsingular Q and symplectic Z. In order to maintain the Hamiltonian pencil, the R matrix in (8.66) must be Hamiltonian, but there is no restriction on L in (8.65). The only concern is somehow to ensure nice convergence.

For Hamiltonian pencils, both $B^{-1}A$ and $A^{-1}B$ are Hamiltonian matrices, but AB^{-1} and BA^{-1} are not. Based on our past experience, we thus propose to take $R = \mathcal{P}_1(B^{-1}A)$, which is a compromise of $\Pi_0(B^{-1}A)$ with the restriction (8.48) and makes Z orthogonal symplectic. There are no restrictions on L, so we use the QZ flow as a guide. In all, we propose the differential equation

$$\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}t} = \mathscr{L}\mathcal{P}_1(B^{-1}A) - \Pi_0(AB^{-1})\mathscr{L}, \tag{8.76}$$

which differs from the QZ flow defined in (8.9) at the $\mathcal{P}_1$ operator but keeps the pencil flow $\mathscr{L}(t)$ Hamiltonian for all t.

The limiting behaviour of (8.76) is somewhat more complicated to describe. For convenience, let Ξ denote the unit perdiagonal matrix whose entries are all zero but 1's along the north-east to south-west diagonal. We introduce the notion that a matrix X is *upper-left quasi-triangular* if the product $X\Xi$ is upper(-right) quasi-triangular in the usual sense. Again, the following conjecture is observed in our numerical experiments, but we have no proof for the moment.

Conjecture 8.8. Suppose the pencil $B_0\lambda - A_0$ is Hamiltonian. Then the flow defined by (8.76) remains a Hamiltonian pencil. Furthermore, we have the following.

(1) Suppose that $B_0\lambda - A_0$ has no purely imaginary eigenvalues. Then $\mathscr{L}(t)$ converges to the canonical form

$$\widehat{\mathscr{L}} = \begin{bmatrix} \widehat{B}_{11} & \widehat{B}_{12} \\ 0 & \widehat{B}_{22} \end{bmatrix} \lambda - \begin{bmatrix} \widehat{A}_{11} & \widehat{A}_{12} \\ 0 & \widehat{A}_{22} \end{bmatrix},$$

where $\widehat{A}_{11}$ and $\widehat{B}_{11}$ are upper quasi-triangular matrices with 1×1 or 2×2 blocks at the same corresponding locations, and $\widehat{A}_{22}$ and $\widehat{B}_{22}$ are upper-left quasi-triangular matrices with 1×1 or 2×2 blocks at the same corresponding locations.

(2) If $B_0\lambda - A_0$ has one pair of purely imaginary eigenvalues. Then $\mathscr{L}(t)$ converges to the same canonical form as above, with the exception of a non-zero entry at the $(n+1, n)$ position which is periodic in t.

Finally, we mention the following theorem concerning a general $2n \times 2n$ pencil (Benner, Mehrmann and Xu 1998).

Theorem 8.9. Given an arbitrary real $2n \times 2n$ pencil $B_0\lambda - A_0$, there exist an orthogonal matrix Q_3 and orthogonal symplectic matrices Q_1 and Q_2 such that

$$Q_3^\top B_0 Q_1 = \begin{bmatrix} \widetilde{B}_{11} & \widetilde{B}_{12} \\ 0 & \widetilde{B}_{22}^\top \end{bmatrix}, \quad Q_3^\top A_0 Q_2 = \begin{bmatrix} \widetilde{A}_{11} & \widetilde{A}_{12} \\ 0 & \widetilde{A}_{22}^\top \end{bmatrix}, \tag{8.77}$$

where $\widetilde{B}_{ij}$, $\widetilde{A}_{ij} \in \mathbb{R}^{n\times n}$,, $\widetilde{B}_{11}$, $\widetilde{A}_{11}$, $\widetilde{B}_{22}$ are upper triangular and A_{22} is upper quasi-triangular.

Note that what is involved in Theorem 8.9 is a non-equivalence transformation, so generally it is not useful for eigenvalue preservation. However, in the case when $B_0\lambda - A_0$ is Hamiltonian, then $Q_1^\top(B_0^{-1}A_0)Q_2$ is precisely the *URV* form for the Hamiltonian matrix $B_0^{-1}A_0$. The reference to Q_3 is completely annihilated. The above result therefore has been exploited as an effective way of eigenvalue computation for Hamiltonian pencils (Benner *et al.* 1998).

We are curious as to whether the canonical form described in (8.77) can be realized continuously. Defining $\mathcal{H}(t) = B^{-1}(t)A(t)$, we have already learned that the *URV* flow $\mathcal{H}(t)$ is governed by (8.61). In particular, we know that $Q_1(t)$ and $Q_2(t)$ should be governed by

$$\frac{\mathrm{d}Q_1}{\mathrm{d}t} = Q_1\mathcal{P}_1((\mathcal{H}J\mathcal{H}^\top J)^{1/2}), \tag{8.78}$$

$$\frac{\mathrm{d}Q_2}{\mathrm{d}t} = Q_2\mathcal{P}_2((\mathcal{H}^\top J\mathcal{H}J)^{1/2}), \tag{8.79}$$

respectively. It is not immediately clear how the dynamics for $Q_3(t)$ should be defined.

Consider the product

$$Z(t) := A(t)B^{-1}(t) = Q_3^\top \underbrace{A_0Q_2Q_1^\top B_0^{-1}}_{\mathscr{Z}} Q_3 = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}.$$

Note that $\mathscr{Z}(t)$ is not necessarily isospectral in t. However, the canonical form (8.77) motivates us to hope that, as t tends to infinity, the matrix $\mathscr{Z}(t)$ would ultimately exhibit the property that $Z_{11} = \widetilde{A}_{11}\widetilde{B}_{11}^{-1}$ is upper triangular, $Z_{21} = 0$ and $Z_{22} = \widetilde{A}_{22}^\top\widetilde{B}_{22}^{-\top}$ should be lower quasi-triangular. We suspect, therefore, that $Q_3(t)$ should be governed by the dynamical system

$$\frac{\mathrm{d}Q_3}{\mathrm{d}t} = Q_3\mathcal{P}_4(Z), \tag{8.80}$$

where the operator $\mathcal{P}_4$ was defined earlier in (8.70). Assembling all together, we conjecture that the canonical form (8.77) can be realized via the dynamical system

$$\frac{\mathrm{d}A}{\mathrm{d}t} = A\mathcal{P}_2((A^\top B^{-\top}JB^{-1}AJ)^{1/2}) - \mathcal{P}_4(AB^{-1})A,\ A(0) = A_0, \tag{8.81}$$

$$\frac{\mathrm{d}B}{\mathrm{d}t} = B\mathcal{P}_1((B^{-1}AJA^\top B^{-\top}J)^{1/2}) - \mathcal{P}_4(AB^{-1})A,\ B(0) = B_0. \tag{8.82}$$

If this conjecture is true, it would nicely express the complicated iterative algorithm described in Benner *et al.* (1998) in a concise form.

Table 8.2. Hierarchy of structure-preserving dynamical systems.

Initial structure	Dynamical system	Limiting behaviour	Operator
X_0 = staircase	$\dot{X} = [X, \Pi_0(X)]$	Ashlock *et al.* (1997)	$\Pi_0(X) = X^- - (X^-)^\top$
$B_0\lambda - A_0$ = staircase	$\dot{\mathscr{L}} = \mathscr{L}\Pi_0(Y^{-1}X) - \Pi_0(XY^{-1})\mathscr{L}$	Conjecture 8.2	
B_0 = staircase	$\dot{B} = B\Pi_0(B^\top B) - \Pi_0(BB^\top)B$	Conjecture 8.3	
$B_0\lambda - A_0$ = Lancaster	$\dot{K} = \frac{1}{2}(CDK - KDC) + N_L^\top K + KN_R$ $\dot{C} = (MDK - KDM) + N_L^\top C + CN_R$ $\dot{M} = \frac{1}{2}(MDC - CDM) + N_L^\top M + MN_R$		D, N_R, N_L = controls
$\mathcal{H}_0$ = Hamiltonian	$\dot{\mathcal{H}} = [\mathcal{H}, \mathcal{P}_0(\mathcal{H})]$	Chu and Norris (1988)	$\mathcal{P}_0(X) = \begin{bmatrix} 0 & -X_{21}^\top \\ X_{21} & 0 \end{bmatrix}$
$\mathcal{W}_0$ = skew-Hamiltonian	$\dot{\mathcal{W}} = [\mathcal{W}, \mathcal{P}_0(\mathcal{W}^{1/2})]$		
$\mathcal{H}_0$ = Hamiltonian	$\dot{\mathcal{H}} = [\mathcal{H}, \mathcal{P}_1(\mathcal{H})]$	Conjecture 8.5	$\mathcal{P}_1(X) = \begin{bmatrix} \Pi_0(X_{11}) & -X_{21} \\ X_{21} & \Pi_0(X_{11}) \end{bmatrix}$
$\mathcal{W}_0$ = skew-Hamiltonian	$\dot{\mathcal{W}} = [\mathcal{W}, \mathcal{P}_1(\mathcal{W}^{1/2})]$		
X_0 = general	$\dot{X} = X\mathcal{P}_3(X^\top X) - \mathcal{P}_3(XX^\top)X$	Conjecture 8.6	$\mathcal{P}_3$ = generalized $\mathcal{P}_0$
$\mathcal{H}_0$ = Hamiltonian	$\dot{X} = X\mathcal{P}_2((X^\top JXJ)^{1/2}) - \mathcal{P}_1((XJX^\top J)^{1/2})X$	*URV* flow	$\mathcal{P}_2(X) := \begin{bmatrix} -\Pi_0(X_{11}^\top) & X_{12} \\ -X_{12} & -\Pi_0(X_{11}^\top) \end{bmatrix}$
$\mathcal{W}_0\lambda - \mathcal{H}_0$ = sHH	$\dot{\mathscr{L}} = \mathscr{L}\mathcal{P}_4(\mathcal{W}^{-1}\mathcal{H}) - \mathcal{P}_4(\mathcal{H}\mathcal{W}^{-1})\mathscr{L}$	Conjecture 8.6	$\mathcal{P}_4(X) := \begin{bmatrix} \Pi_0(X_{11}) & -X_{21}^\top \\ X_{21} & -\Pi_0(X_{22}^\top) \end{bmatrix}$
$B_0\lambda - A_0$ = Hamiltonian	$\dot{\mathscr{L}} = \mathscr{L}\mathcal{P}_1(B^{-1}A) - \Pi_0(AB^{-1})\mathscr{L}$	Conjecture 8.8	
$B_0\lambda - A_0$ = general	$\dot{A} = A\mathcal{P}_2((A^\top B^{-\top}JB^{-1}AJ)^{1/2}) - \mathcal{P}_4(AB^{-1})A$ $\dot{B} = B\mathcal{P}_1((B^{-1}AJA^\top B^{-\top}J)^{1/2}) - \mathcal{P}_4(AB^{-1})A$	not tested	

It might be helpful to summarize the different dynamical systems discussed thus far in Table 8.2. Recall that the principal consideration in formulating these flows is to preserve the structure of the initial data. The special operators in the right-hand column of the table are designed for that purpose, all of which are variations of the operator Π_0. Only a few of these systems have their asymptotic behaviour understood in the literature. Those identified by a conjecture in the table have been extensively tested by numerical integrators, but no theory of asymptotic analysis is available for the present. If any of the conjectures is true, then the corresponding dynamical system often encapsulates a fairly complicated iterative process into a nice and simple mathematical expression. Be aware that we are not implying that the flows sampled at integer times will produce the same iterates as those generated by existing discrete methods; this coincidence might be too difficult to achieve for matrices with Hamiltonian structure. The only cases we know for sure about this coincidence are the QR, QZ and SVD flows. Nor are we inferring that these structure-preserving dynamical systems can easily be discretized with the resulting iterative schemes still preserving the original structure. We must stress that the link diagram in Figure 1.1 that we frequently refer to in this paper now has an added dimension of constraint, namely, structure preservation. Thus there is much room left for further investigation of these relationships.

8.5. Group structure

Needless to say, there are far too many other applications where it is desirable that a specific structure is maintained throughout a specified dynamical system. Like the canonical forms, the notion of 'structure' should be interpreted quite liberally. We have discussed only a few cases involving the spectrum, the singular values, the staircase, or the Hamiltonian structure from the linear algebra perspective. Obviously, it is never an overstatement that preserving volume, momentum, energy, symplecticity, or other kinds of physical quantities, is an extremely important task with significant consequences. The subject is simply so wide in scope that the author must humbly admit it is beyond his comprehension. We conclude this chapter by pointing out one more structure that has recently attracted tremendous interest.

The once-abstract notion of Lie theory is now a ubiquitous framework in many disciplines of sciences and engineering applications. In Section 7 we have also demonstrated how group actions often serve as the fundamental coordinate transformations leading to canonical forms. It should not come as a surprise, but rather a necessity, that many of the dynamical systems and numerical algorithms originally developed over Euclidean space need to be redeveloped over manifolds. By studying the underlying geometry, for example, critical algorithms such as the Newton and the conjugate gradient methods can be generalized to the Grassmann and the Stiefel manifolds in a natural way (Edelman *et al.* 1999).

We illustrate in this section how the Newton dynamics can take place on a Lie group (Owren and Welfert 2000). This notion typifies what we mean by a dynamical system that respects the group structure.

Let G be a Lie group and $\mathfrak{g}$ its corresponding Lie algebra. Keep in mind that elements in G can be abstract functionals or operators. Suppose we want to find 'zero(s)' of a given map,

$$f : G \to \mathfrak{g},$$

where the iterates are to stay on the manifold G. Given a current iterate $y_n \in G$, the Newton scheme can interpreted as solving the equation

$$\mathrm{d}f_{y_n}(u_n) + f(y_n) = 0, \tag{8.83}$$

for a tangent vector $u_n \in \mathfrak{g}$ and then updating to the next iterate via the exponential map

$$y_{n+1} = y_n \exp(u_n). \tag{8.84}$$

In the above, the differential

$$\mathrm{d}f_y : \mathcal{T}_y G \to \mathfrak{g}$$

can be interpreted as

$$\mathrm{d}f_y(u) = (\mathrm{d}/\mathrm{d}t)_{t=0} f(y \exp(tu)). \tag{8.85}$$

Alternatively, since all local charts of a Lie group can be obtained by translation, we can restrict ourselves to the local charts. In particular, it suffices to consider the 'local' representation of f at y_n,

$$\tilde{f} := f \circ L_{y_n} \circ \exp, \tag{8.86}$$

where $L_z(y) = zy$ with a fixed $z \in G$. This becomes a classical algebraic equation in Euclidean space. The Newton iteration involves the steps of solving the equation

$$\mathrm{d}\tilde{f}_{v_n}(u_n) + \tilde{f}(v_n) = 0, \tag{8.87}$$

for u_n, where v_n is the local parametrization, *i.e.*, the logarithm of y_n, updating in the linear space by $v_{n+1} = v_n + u_n$, and finally advancing to the new iterate on the manifold G by defining

$$y_{n+1} = y_n \exp(v_{n+1}). \tag{8.88}$$

Note that both formulations reduce to the standard method in the Euclidean case. It can be shown that under classical assumptions the proposed methods converge quadratically (Owren and Welfert 2000).

We think this framework can be repeatedly applied to generalize other types of algorithms originally designed for Euclidean space to Lie groups. How far this generalization should go, and how practical such extensions might be, are yet to be seen.

Acknowledgement

The author is deeply indebted to Professor Yoshimasa Nakamura and his research team at the Kyoto University for enthusiastically and gracefully relaying their knowledge and experiences on the Lotka–Volterra equation to him. The amazing connection among orthogonal polynomials, moment representation, τ functions, Hankel determinantal solution, the Toda lattice and the effective *vdLV* algorithm would not have been understood by the author without their kind help.

REFERENCES

P.-A. Absil and K. Kurdyka (2006), 'On the stable equilibrium points of gradient systems', *Systems Control Lett.* **55**, 573–577.

P.-A. Absil, R. Mahony and B. Andrews (2005), 'Convergence of the iterates of descent methods for analytic cost functions', *SIAM J. Optim.* **16**, 531–547 (electronic).

N. I. Akhiezer (1965), *The Classical Moment Problem and Some Related Questions in Analysis*, Hafner, New York. Translated by N. Kemmer.

E. Allgower and K. Georg (1980), 'Simplicial and continuation methods for approximating fixed points and solutions to systems of equations', *SIAM Rev.* **22**, 28–85.

E. L. Allgower and K. Georg (2003), *Introduction to Numerical Continuation Methods*, Vol. 45 of *Classics in Applied Mathematics*, SIAM, Philadelphia, PA.

A. C. Antoulas (2005), *Approximation of Large-Scale Dynamical Systems*, Vol. 6 of *Advances in Design and Control*, SIAM, Philadelphia, PA.

A. I. Aptekarev, A. Branquinho and F. Marcellán (1997), 'Toda-type differential equations for the recurrence coefficients of orthogonal polynomials and Freud transformation', *J. Comput. Appl. Math.* **78**, 139–160.

P. Arbenz and G. H. Golub (1995), 'Matrix shapes invariant under the symmetric *QR* algorithm', *Numer. Linear Algebra Appl.* **2**, 87–93.

V. I. Arnold (1988), *Geometrical Methods in the Theory of Ordinary Differential Equations*, Vol. 250 of *Grundlehren der Mathematischen Wissenschaften* (*Fundamental Principles of Mathematical Sciences*), second edn, Springer, New York. Translated from the Russian by Joseph Szücs (József M. Szűcs).

D. A. Ashlock, K. R. Driessel and I. R. Hentzel (1997), On matrix structures invariant under Toda-like isospectral flows, in *Proc. Fifth Conference of the International Linear Algebra Society, Atlanta 1995*, Vol. 254, pp. 29–48.

A. Baker (2002), *Matrix Groups: An Introduction to Lie Group Theory*, Springer Undergraduate Mathematics Series, Springer, London.

R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. Van der Vorst (1994), *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, PA.

T. Beelen and P. Van Dooren (1990), Computational aspects of the Jordan canonical form, in *Reliable Numerical Computation*, Oxford University Press, New York, pp. 57–72.

P. Benner and D. Kressner (2006), 'Algorithm 854: Fortran 77 subroutines for computing the eigenvalues of Hamiltonian matrices II', *ACM Trans. Math. Software* **32**, 352–373.

P. Benner, R. Byers, V. Mehrmann and H. Xu (2002), 'Numerical computation of deflating subspaces of skew-Hamiltonian/Hamiltonian pencils', *SIAM J. Matrix Anal. Appl.* **24**, 165–190 (electronic).

P. Benner, D. Kressner and V. Mehrmann (2005), Skew-Hamiltonian and Hamiltonian eigenvalue problems: theory, algorithms and applications, in *Proc. Conference on Applied Mathematics and Scientific Computing*, Springer, Dordrecht, pp. 3–39.

P. Benner, V. Mehrmann and H. Xu (1998), 'A numerically stable, structure preserving method for computing the eigenvalues of real Hamiltonian or symplectic pencils', *Numer. Math.* **78**, 329–358.

A. Bhaya and E. Kaszkurewicz (2006), *Control Perspectives on Numerical Algorithms and Matrix Problems*, Vol. 10 of *Advances in Design and Control*, SIAM, Philadelphia, PA.

S. Blanes, F. Casas, J. A. Oteo and J. Ros (1998), 'Magnus and Fer expansions for matrix differential equations: The convergence problem', *J. Phys. A* **31**, 259–268.

A. M. Bloch and A. Iserles (2006), 'On an isospectral Lie-Poisson system and its Lie algebra', *Found. Comput. Math.* **6**, 121–144.

A. M. Bloch, R. W. Brockett and T. S. Ratiu (1992), 'Completely integrable gradient flows', *Comm. Math. Phys.* **147**, 57–74.

J. H. Bramble (1993), *Multigrid Methods*, Vol. 294 of *Pitman Research Notes in Mathematics Series*, Longman, Harlow.

W. L. Briggs (1987), *A Multigrid Tutorial*, SIAM, Philadelphia, PA.

R. W. Brockett (1991), 'Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems', *Linear Algebra Appl.* **146**, 79–91.

R. W. Brockett (1993), Differential geometry and the design of gradient algorithms, in *Differential Geometry: Partial Differential Equations on Manifolds, Los Angeles 1990*, Vol. 54 of *Proc. Sympos. Pure Math.*, AMS, Providence, RI, pp. 69–92.

A. Bunse-Gerstner, R. Byers, V. Mehrmann and N. K. Nichols (1991), 'Numerical computation of an analytic singular value decomposition of a matrix valued function', *Numer. Math.* **60**, 1–39.

M. P. Calvo, A. Iserles and A. Zanna (1997), 'Numerical solution of isospectral flows', *Math. Comp.* **66**, 1461–1486.

E. Celledoni and A. Iserles (2000), 'Approximating the exponential from a Lie algebra to a Lie group', *Math. Comp.* **69**, 1457–1480.

R. Chill (2003), 'On the Łojasiewicz–Simon gradient inequality', *J. Funct. Anal.* **201**, 572–601.

D. Chu, X. Liu and V. Mehrmann (2007), 'A numerical method for computing the Hamiltonian Schur form', *Numer. Math.* **105**, 375–412.

M. Chu (1986*a*), 'A continuous approximation to the generalized Schur decomposition', *Linear Algebra Appl.* **78**, 119–132.

M. T. Chu (1986*b*), 'A differential equation approach to the singular value decomposition of bidiagonal matrices', *Linear Algebra Appl.* **80**, 71–79.

M. T. Chu (1988), 'On the continuous realization of iterative processes', *SIAM Rev.* **30**, 375–387.

M. T. Chu (1991), 'A continuous Jacobi-like approach to the simultaneous reduction of real matrices', *Linear Algebra Appl.* **147**, 75–96.

M. T. Chu (1995), 'Constructing a Hermitian matrix from its diagonal entries and eigenvalues', *SIAM J. Matrix Anal. Appl.* **16**, 207–217.

M. T. Chu and N. Del Buono (2008*a*), 'Total decoupling of general quadratic pencils I: Theory', *J. Sound Vibration* **309**, 96–111.

M. T. Chu and N. Del Buono (2008*b*), 'Total decoupling of general quadratic pencils II: Structure preserving isospectral flows', *J. Sound Vibration* **309**, 112–128.

M. T. Chu and K. R. Driessel (1990), 'The projected gradient method for least squares matrix approximations with spectral constraints', *SIAM J. Numer. Anal.* **27**, 1050–1060.

M. T. Chu and R. E. Funderlic (2002), 'The centroid decomposition: Relationships between discrete variational decompositions and SVDs', *SIAM J. Matrix Anal. Appl.* **23**, 1025–1044 (electronic).

M. T. Chu and G. H. Golub (2002), Structured inverse eigenvalue problems, in *Acta Numerica*, Vol. 11, Cambridge University Press, pp. 1–71.

M. T. Chu and G. H. Golub (2005), *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*, Numerical Mathematics and Scientific Computation, Oxford University Press, New York.

M. T. Chu and Q. Guo (1998), 'A numerical method for the inverse stochastic spectrum problem', *SIAM J. Matrix Anal. Appl.* **19**, 1027–1039 (electronic).

M. T. Chu and L. K. Norris (1988), 'Isospectral flows and abstract matrix factorizations', *SIAM J. Numer. Anal.* **25**, 1383–1391.

M. T. Chu and N. T. Trendafilov (2001), 'The orthogonally constrained regression revisited', *J. Comput. Graph. Statist.* **10**, 746–771.

M. T. Chu and S.-F. Xu (2005), 'On computing minimal realizable spectral radii of non-negative matrices', *Numer. Linear Algebra Appl.* **12**, 77–86.

M. Chu, N. Del Buono, L. Lopez and T. Politi (2005), 'On the low-rank approximation of data on the unit sphere', *SIAM J. Matrix Anal. Appl.* **27**, 46–60 (electronic).

T. F. Cox and M. A. A. Cox (1994), *Multidimensional Scaling*, Vol. 59 of *Monographs on Statistics and Applied Probability*, Chapman & Hall, London.

M. L. Curtis (1984), *Matrix Groups*, second edn, Universitext, Springer, New York.

J. W. Daniel (1967), 'The conjugate gradient method for linear and nonlinear operator equations', *SIAM J. Numer. Anal.* **4**, 10–26.

E. Date, M. Kashiwara, M. Jimbo and T. Miwa (1983), Transformation groups for soliton equations, in *Nonlinear Integrable Systems: Classical Theory and Quantum Theory, Kyoto 1981*, World Scientific, Singapore, pp. 39–119.

P. Deift, J. Demmel, L. C. Li and C. Tomei (1991), 'The bidiagonal singular value decomposition and Hamiltonian mechanics', *SIAM J. Numer. Anal.* **28**, 1463–1516.

P. Deift, T. Nanda and C. Tomei (1983), 'Ordinary differential equations and the symmetric eigenvalue problem', *SIAM J. Numer. Anal.* **20**, 1–22.

N. Del Buono and L. Lopez (2002), 'Geometric integration on manifold of square oblique rotation matrices', *SIAM J. Matrix Anal. Appl.* **23**, 974–989 (electronic).

J. Della-Dora (1975), 'Numerical linear algorithms and group theory', *Linear Algebra Appl.* **10**, 267–283.

J. Demmel and W. Kahan (1990), 'Accurate singular values of bidiagonal matrices', *SIAM J. Sci. Statist. Comput.* **11**, 873–912.

J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić and Z. Drmač (1999), 'Computing the singular value decomposition with high relative accuracy', *Linear Algebra Appl.* **299**, 21–80.

R. L. Devaney (1992), *A First Course in Chaotic Dynamical Systems: Theory and Experiment*, Addison-Wesley Studies in Nonlinearity, Addison-Wesley, Reading, MA.

L. Dieci, R. D. Russell and E. S. Van Vleck (1994), 'Unitary integrators and applications to continuous orthonormalization techniques', *SIAM J. Numer. Anal.* **31**, 261–281.

A. Edelman, T. A. Arias and S. T. Smith (1999), 'The geometry of algorithms with orthogonality constraints', *SIAM J. Matrix Anal. Appl.* **20**, 303–353 (electronic).

S. Elaydi (2005), *An Introduction to Difference Equations*, Undergraduate Texts in Mathematics, third edn, Springer, New York.

K. Engø (2003), 'Partitioned Runge–Kutta methods in Lie-group setting', *BIT* **43**, 21–39.

H. Faßbender, D. S. Mackey, N. Mackey and H. Xu (1999), 'Hamiltonian square roots of skew-Hamiltonian matrices', *Linear Algebra Appl.* **287**, 125–159.

L. Faybusovich (1991), 'Hamiltonian structure of dynamical systems which solve linear programming problems', *Phys. D* **53**, 217–232.

K. V. Fernando and B. N. Parlett (1994), 'Accurate singular values and differential qd algorithms', *Numer. Math.* **67**, 191–229.

J. G. F. Francis (1961/1962), 'The *QR* transformation: A unitary analogue to the *LR* transformation I', *Comput. J.* **4**, 265–271.

R. W. Freund and N. M. Nachtigal (1991), 'QMR: A quasi-minimal residual method for non-Hermitian linear systems', *Numer. Math.* **60**, 315–339.

O. Galor (2005), 'Discrete dynamical systems', *GE, Growth, Math Methods, EconWPA*. Available at http://ideas.repec.org/p/wpa/wuwpge/0504001.html.

C. B. García and F. J. Gould (1980), 'Relations between several path following algorithms and local and global Newton methods', *SIAM Rev.* **22**, 263–274.

S. D. Garvey, M. I. Friswell and U. Prells (2002*a*), 'Co-ordinate transformations for second order systems I: General transformations', *J. Sound Vibration* **258**, 885–909.

S. D. Garvey, M. I. Friswell and U. Prells (2002*b*), 'Co-ordinate transformations for second order systems II: Elementary structure-preserving transformations', *J. Sound Vibration* **258**, 911–930.

C. W. Gear (1981), 'Numerical solution of ordinary differential equations: Is there anything left to do?', *SIAM Rev.* **23**, 10–24.

I. Gohberg, P. Lancaster and L. Rodman (1982), *Matrix Polynomials*, Academic Press, New York.

D. Goldberg (1991), 'What every computer scientist should know about floating-point arithmetic', *ACM Computing Surveys* **23**, 5–48.

G. Golub and W. Kahan (1965), 'Calculating the singular values and pseudo-inverse of a matrix', *J. Soc. Indust. Appl. Math. Ser. B, Numer. Anal.* **2**, 205–224.

G. H. Golub and C. F. Van Loan (1996), *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences, third edn, Johns Hopkins University Press, Baltimore, MD.

G. H. Golub and J. H. Wilkinson (1976), 'Ill-conditioned eigensystems and the computation of the Jordan canonical form', *SIAM Rev.* **18**, 578–619.

A. Greenbaum (1997), *Iterative Methods for Solving Linear Systems*, Vol. 17 of *Frontiers in Applied Mathematics*, SIAM, Philadelphia, PA.

J. Guckenheimer (2002), Numerical analysis of dynamical systems, in *Handbook of Dynamical Systems*, Vol. 2, North-Holland, Amsterdam, pp. 345–390.

L. A. Hageman and D. M. Young (1981), *Applied Iterative Methods*, Academic Press, New York. Also, unabridged republication of the 1981 original: Dover, Mineola, NY (2004).

E. Hairer and G. Wanner (1996), *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Vol. 14 of *Springer Series in Computational Mathematics*, second edn, Springer, Berlin.

E. Hairer, C. Lubich and G. Wanner (2006), *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, Vol. 31 of *Springer Series in Computational Mathematics*, second edn, Springer, Berlin.

E. Hairer, S. P. Nørsett and G. Wanner (1993), *Solving ordinary differential equations I: Nonstiff Problems*, Vol. 8 of *Springer Series in Computational Mathematics*, second edn, Springer, Berlin.

R. Hauser and J. Nedić (2007), 'On the relationship between the convergence rates of iterative and continuous processes', *SIAM J. Optim.* **18**, 52–64 (electronic).

U. Helmke and J. B. Moore (1994), *Optimization and Dynamical Systems*, Communications and Control Engineering Series, Springer, London.

U. Helmke and F. Wirth (2000), Controllability of the shifted inverse power iteration: the case of real shifts, in *International Conference on Differential Equations, Berlin 1999*, Vols 1, 2, World Scientific, River Edge, NJ, pp. 859–864.

U. Helmke and F. Wirth (2001), 'On controllability of the real shifted inverse power iteration', *Systems Control Lett.* **43**, 9–23.

U. Helmke, J. B. Moore and J. E. Perkins (1994), 'Dynamical systems that compute balanced realizations and the singular value decomposition', *SIAM J. Matrix Anal. Appl.* **15**, 733–754.

M. R. Hestenes and E. Stiefel (1952), 'Methods of conjugate gradients for solving linear systems', *J. Research Nat. Bur. Standards* **49**, 409–436.

N. J. Higham (1989), Matrix nearness problems and applications, in *Applications of Matrix Theory, Bradford 1988*, Vol. 22 of *Inst. Math. Appl. Conf. Ser., New Ser.*, Oxford University Press, New York, pp. 1–27.

R. Hirota, S. Tsujimoto and T. Imai (1993), Difference scheme of soliton equations, in *Future Directions of Nonlinear Dynamics in Physical and Biological Systems, Lyngby 1992*, Vol. 312 of *NATO Adv. Sci. Inst. Ser. B, Phys.*, Plenum, New York, pp. 7–15.

M. W. Hirsch and S. Smale (1979), 'On algorithms for solving $f(x) = 0$', *Comm. Pure Appl. Math.* **32**, 281–313.

R. A. Horn and C. R. Johnson (1990), *Matrix Analysis*, Cambridge University Press, Cambridge.

P. Horst (1965), *Factor Analysis of Data Matrices*, Holt, Rinehart and Winston, New York.

R. Howe (1983), 'Very basic Lie theory', *Amer. Math. Monthly* **90**, 600–623. Correction to 'Very basic Lie theory', **91** (1984), 247.

A. Iserles (2002), 'On the discretization of double-bracket flows', *Found. Comput. Math.* **2**, 305–329.

A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett and A. Zanna (2000), Lie-group methods, in *Acta Numerica*, Vol. 9, Cambridge University Press, pp. 215–365.

M. Iwasaki and Y. Nakamura (2002), 'On the convergence of a solution of the discrete Lotka–Volterra system', *Inverse Problems* **18**, 1569–1578.

M. Iwasaki and Y. Nakamura (2004), 'An application of the discrete Lotka–Volterra system with variable step-size to singular value computation', *Inverse Problems* **20**, 553–563.

M. Iwasaki and Y. Nakamura (2006), 'Accurate computation of singular values in terms of shifted integrable schemes', *Japan J. Indust. Appl. Math.* **23**, 239–259.

W. Kahan (1972), Conserving confluence curbs ill-condition. Technical report 6, Computer Science Department, University of California, Berkeley.

N. Karmarkar (1984), 'A new polynomial-time algorithm for linear programming', *Combinatorica* **4**, 373–395.

C. T. Kelley and D. E. Keyes (1998), 'Convergence analysis of pseudo-transient continuation', *SIAM J. Numer. Anal.* **35**, 508–523 (electronic).

C. T. Kelley, L.-Z. Liao, L. Qi, M. T. Chu, J. P. Reese and C. Winton (2007), Projected pseudo-transient continuation. Preprint, North Carolina State University.

M. R. S. Kulenović and O. Merino (2002), *Discrete Dynamical Systems and Difference Equations with Mathematica*, Chapman & Hall/CRC, Boca Raton, FL.

P. D. Lax (1968), 'Integrals of nonlinear equations of evolution and solitary waves', *Comm. Pure Appl. Math.* **21**, 467–490.

W.-W. Lin and S.-F. Xu (2006), 'Convergence analysis of structure-preserving doubling algorithms for Riccati-type matrix equations', *SIAM J. Matrix Anal. Appl.* **28**, 26–39 (electronic).

W.-W. Lin, V. Mehrmann and H. Xu (1999), 'Canonical forms for Hamiltonian and symplectic matrices and pencils', *Linear Algebra Appl.* **302/303**, 469–533.

S. Łojasiewicz (1963), Une propriété topologique des sous-ensembles analytiques réels, in *Les Equations aux Dérivées Partielles, Paris 1962*, Éditions du Centre National de la Recherche Scientifique, Paris, pp. 87–89.

D. S. Mackey, N. Mackey and F. Tisseur (2003), 'Structured tools for structured matrices', *Electron. J. Linear Algebra* **10**, 106–145 (electronic).

C. Mehl (1999), 'Condensed forms for skew-Hamiltonian/Hamiltonian pencils', *SIAM J. Matrix Anal. Appl.* **21**, 454–476 (electronic).

V. Mehrmann and D. Watkins (2000), 'Structure-preserving methods for computing eigenpairs of large sparse skew-Hamiltonian/Hamiltonian pencils', *SIAM J. Sci. Comput.* **22**, 1905–1925 (electronic).

G. Meurant (2006), *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*, Vol. 19 of *Software, Environments, and Tools*, SIAM, Philadelphia, PA.

A. Morgan (1987), *Solving Polynomial Systems Using Continuation for Engineering and Scientific Problems*, Prentice-Hall, Englewood Cliffs, NJ.

J. Moser (1975), 'Three integrable Hamiltonian systems connected with isospectral deformations', *Adv. Math.* **16**, 197–220.

W. A. Mulder and B. van Leer (1985), 'Experiments with implicit upwind methods for the Euler equations', *J. Comput. Phys.* **59**, 232–246.

H. Munthe-Kaas (1998), 'Runge–Kutta methods on Lie groups', *BIT* **38**, 92–111.

Y. Nakamura (2004), A new approach to numerical algorithms in terms of integrable systems, in *Proc. International Conference on Informatics Research for Development of Knowledge Society Infrastructure: ICKS 2004* (T. Ibaraki, T. Inui and K. K. Tanaka, eds), IEEE Computer Society Press, pp. 194–205.

Y. Nakamura (2006), *Functionality of Integrable Systems*, Kyoritsu Shuppan Co., Tokyo, Japan. (In Japanese.)

R. Orsi (2006), 'Numerical methods for solving inverse eigenvalue problems for nonnegative matrices', *SIAM J. Matrix Anal. Appl.* **28**, 190–212 (electronic).

J. M. Ortega and W. C. Rheinboldt (2000), *Iterative Solution of Nonlinear Equations in Several Variables*, Vol. 30 of *Classics in Applied Mathematics*, SIAM, Philadelphia, PA.

B. Owren and B. Welfert (2000), 'The Newton iteration on Lie groups', *BIT* **40**, 121–145.

C. Paige and C. Van Loan (1981), 'A Schur decomposition for Hamiltonian matrices', *Linear Algebra Appl.* **41**, 11–32.

B. N. Parlett (1974), 'The Rayleigh quotient iteration and some generalizations for nonnormal matrices', *Math. Comp.* **28**, 679–693.

B. N. Parlett and O. A. Marques (2000), An implementation of the dqds algorithm (positive case), in *Proc. International Workshop on Accurate Solution of Eigenvalue Problems, University Park 1998*, Vol. 309, pp. 217–259.

C. Pöppe (1989), 'General determinants and the τ function for the Kadomtsev–Petviashvili hierarchy', *Inverse Problems* **5**, 613–630. See also corrigenda: **5**, 1173.

F. A. Potra and S. J. Wright (2000), 'Interior-point methods', *J. Comput. Appl. Math.* **124**, 281–302.

A. Ruhe (1987), 'Closest normal matrix finally found!', *BIT* **27**, 585–598.

H. Rutishauser (1954), 'Der Quotienten-Differenzen-Algorithmus', *Z. Angew. Math. Physik* **5**, 233–251.

H. Rutishauser (1960), 'Über eine kubisch konvergente Variante der *LR*-Transformation', *Z. Angew. Math. Mech.* **40**, 49–54.

Y. Saad and M. H. Schultz (1986), 'GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems', *SIAM J. Sci. Statist. Comput.* **7**, 856–869.

G. V. Savinov (1983), 'The conjugate gradient method for systems of nonlinear equations', *J. Math. Sci.* **23**, 2012–2017. Translated from *Zap. Naučn. Sem. Leningrad. Otdel. Mat. Inst. Steklov.* (LOMI), **70** (1977), 178–183.

H. Sedaghat (2003), *Nonlinear Difference Equations: Theory with Applications to Social Science Models*, Vol. 15 of *Mathematical Modelling: Theory and Applications*, Kluwer, Dordrecht.

R. Shaw (1982), *Linear Algebra and Group Representations I: Linear Algebra and Introduction to Group Representations*, Academic Press, London.

L. Simon (1983), 'Asymptotics for a class of nonlinear evolution equations, with applications to geometric problems', *Ann. of Math.* (2) **118**, 525–571.

S. Smale (1977), Convergent process of price adjustment and global Newton methods, in *Frontiers of Quantitative Economics: Invited papers, Econometric Soc., Third World Congress, Toronto 1975*, Vol. IIIA, North-Holland, Amsterdam, pp. 191–205.

V. I. Smirnov (1970), *Linear Algebra and Group Theory*, Dover, New York.

S. T. Smith (1991), 'Dynamical systems that perform the singular value decomposition', *Systems Control Lett.* **16**, 319–327.

G. W. Stewart (1993), 'On the early history of the singular value decomposition', *SIAM Rev.* **35**, 551–566.

A. M. Stuart and A. R. Humphries (1996), *Dynamical Systems and Numerical Analysis*, Vol. 2 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press, Cambridge.

W. W. Symes (1981/82), 'The *QR* algorithm and scattering for the finite nonperiodic Toda lattice', *Phys. D* **4**, 275–280.

G. Szegő (1975), *Orthogonal Polynomials*, fourth edn. Vol. XXIII of *AMS Colloquium Series*, AMS, Providence, RI.

M. Takata, M. Iwasaki, K. Kimura and Y. Nakamura (2005), An evaluation of singular value computation by the discrete Lotka–Volterra system, in *Proc. 2005 International Conference on Parallel and Distributed Processing Techniques and Applications: PDPTA2005*, Vol. II, pp. 410–416.

M. Takata, M. Iwasaki, K. Kimura and Y. Nakamura (2006), Performance of a new scheme for bidiagonal singular value decomposition of large scale, in *Proc. IASTED International Conference on Parallel and Distributed Computing and Networks: PDCN2006*, pp. 304–309.

T.-Y. Tam (2004), 'Gradient flows and double bracket equations', *Differential Geom. Appl.* **20**, 209–224.

F. Tisseur and K. Meerbergen (2001), 'The quadratic eigenvalue problem', *SIAM Rev.* **43**, 235–286 (electronic).

A. Toselli and O. Widlund (2005), *Domain Decomposition Methods: Algorithms and Theory*, Vol. 34 of *Springer Series in Computational Mathematics*, Springer, Berlin.

S. Tsujimoto (1995), Molecule solution of hungry Volterra equations, in *Soliton Theory and Its Applications* (J. Satsuma, ed.), University of Tokyo, Japan, pp. 53–56. (In Japanese.)

S. Tsujimoto, Y. Nakamura and M. Iwasaki (2001), 'The discrete Lotka–Volterra system computes singular values', *Inverse Problems* **17**, 53–58.

Y. Z. Tsypkin (1971), *Adaptation and Learning in Automatic Systems*, Vol. 73 of *Mathematics in Science and Engineering*, Academic Press, New York. Translated from the Russian by Z. J. Nikolic.

Y. Z. Tsypkin (1973), *Foundations of the Theory of Learning Systems*, Vol. 101 of *Mathematics in Science and Engineering*, Academic Press, New York/ London. Translated from the Russian by Z. J. Nikolic.

H. A. van der Vorst (2003), *Iterative Krylov Methods for Large Linear Systems*, Vol. 13 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press, Cambridge.

C. F. Van Loan (1984), 'A symplectic method for approximating all the eigenvalues of a Hamiltonian matrix', *Linear Alg. Appl.* **61**, 233–253.

R. S. Varga (2000), *Matrix Iterative Analysis*, Vol. 27 of *Springer Series in Computational Mathematics*, expanded edn, Springer, Berlin.

D. S. Watkins (1982), 'Understanding the QR algorithm', *SIAM Rev.* **24**, 427–440.

D. S. Watkins and L. Elsner (1988), 'Self-similar flows', *Linear Algebra Appl.* **110**, 213–242.

J. Wimp (1984), *Computation with Recurrence Relations*, Applicable Mathematics Series, Pitman, Boston, MA.

K. Wright (1992), 'Differential equations for the analytic singular value decomposition of a matrix', *Numer. Math.* **63**, 283–295.

M. H. Wright (2005), 'The interior-point revolution in optimization: history, recent developments, and lasting consequences', *Bull. Amer. Math. Soc. (N.S.)* **42**, 39–56 (electronic).

S. J. Wright (1997), *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, PA.

H. Yabe and M. Takano (2004), 'Global convergence properties of nonlinear conjugate gradient methods with modified secant condition', *Comput. Optim. Appl.* **28**, 203–225.

Z. Zeng and T. Y. Li (2007), A numerical method for computing the Jordan canonical form. Preprint, Northeastern Illinois University.

H. Zha and Z. Zhang (1995), 'A note on constructing a symmetric matrix with specified diagonal entries and eigenvalues', *BIT* **35**, 448–451.

S. Zhang and Z. Deng (2005), 'Geometric integration methods for general nonlinear dynamic equation based on Magnus and Fer expansions', *Progr. Natur. Sci.* (*English edn*) **15**, 304–314.