Lyapunov function based step size control for numerical ODE solvers with application to optimization algorithms

Lars Grüne
University of Bayreuth
Bayreuth, Germany
lars.gruene@uni-bayreuth.de

Iasson Karafyllis
Technical University of Crete
Chania, Greece
ikarafyl@enveng.tuc.gr

Abstract. We present and analyze an abstract step size selection algorithm which ensures asymptotic stability of numerical approximations to asymptotically stable ODEs. A particular implementation of this algorithm is proposed and tested with two numerical examples. The application to ODEs solving nonlinear optimization problems on manifolds is explained and illustrated by means of the Rayleigh flow for computing eigenvalues of symmetric matrices.

1 Introduction

Step size control algorithms are nowadays standard in numerical methods for solving ordinary differential equations (ODEs). Due to the fact that the characteristics of the vector field depend on the state (and possibly also on time), adaptive step sizes should be used as the solution evolves. Using efficient implementations, the additional computational effort for the online computation of suitable step sizes is typically negligible compared to the gain in computational efficiency. Usually, the adaptive step sizes are selected on the basis of local error estimates and the corresponding algorithms are classical and can be found in any text book on numerical methods for differential equations, as, e.g., in [4, 14, 15, 25].

While error estimation based step size selection schemes achieve very good results in ensuring accurate approximations on finite time intervals, they do not necessarily guarantee that the asymptotic behavior of the numerical solution equals that of the exact solution. In this paper, we will investigate this case for ODEs exhibiting an asymptotically stable equilibrium. It is well known that any consistent and stable numerical scheme for ODEs inherits the asymptotic stability of the original equation in a practical sense, even for more general attractors than equilibria, see for instance [11, 12, 20] and [25, Chapter 7] for fixed step size and [7, 21] for schemes with variable step size. However, in general the numerical approximation need not be asymptotically stable in the usual sense. Instead, it may happen that the numerical solution does not converge to the equilibrium but only to a small neighborhood thereof and this can happen not only for fixed step sizes but also when error based step size control techniques are used, as [18, Example 2.11] shows.

A popular approach to ensure "true" asymptotic stability of the numerical scheme is the use of specialized numerical schemes, like (typically implicit) schemes having

the A-stability or B-stability property which guarantee asymptotic stability for certain classes of ODEs, cf. e.g., [4, 15, 25], or geometric integration methods which preserve structural properties of the ODE also on infinite integration intervals, cf. e.g., [13] or [10]. Here, we build upon a different approach which was recently proposed in [18]. In this reference, general consistent Runge-Kutta schemes (explicit or implicit) were represented as hybrid control systems such that the step size selection problem could be reformulated as a nonlinear feedback stabilization problem. Consequently, nonlinear feedback design techniques like the small gain methodology or Lyapunov function based design, e.g., via backstepping, could then be applied to solve the problem under suitable assumptions on the system and, in case of Lyapunov function based design, on the corresponding Lyapunov function. Although the methods proposed in [18] may not necessarily outperform specialized taylored methods for particular problem classes, they form a systematic and versatile approach which can be applied to many different problems. While the majority of the results in [18] were focused on existence issues or explicit state dependent step size formulas, it was also observed that if a Lyapunov function for the ODE is known, then an online step size control algorithm similar to classical error estimation based step size control schemes can be designed.

In this paper, we will further investigate and refine this concept. Specifically, we will present an abstract Lyapunov function based step size control algorithm and prove asymptotic stability of the generated numerical approximations under general assumptions on the functions adjusting the step sizes. We then propose an implementation of this abstract algorithm in which the adjustment of the step sizes is obtained using ideas from consistency order based step size control. In this context it is important to note that the discretization error introduced by the Runge-Kutta scheme may not necessarily destroy asymptotic stability. On the contrary, it may well happen that the numerical approximation converges to the equilibrium at a faster rate than the exact solution and this effect may be even stronger if large time steps are used. A particular feature of our algorithm — which will also be visible in our numerical examples — is that it is able to detect this situation and then allows for large step sizes. Of course, proceeding this way, we can no longer guarantee that the numerical solution faithfully reproduces the exact solution. However, the algorithm still ensures convergence to the correct equilibrium and may thus be able to reach a small neighborhood of this equilibrium with considerably less steps than an approximation which aims at a correct reproduction of the exact solution during the transient phase.

The algorithm is thus particularly suited for applications in which the numerical computation of an asymptotically stable equilbrium — but not necessarily the path along which this equilibrium is approached — is of interest. A typical class of problems in which this is the case are ODEs which are desiged for solving nonlinear optimization problems. Since such ODEs, moreover, often come with a canonical Lyapunov function (in the simplest case the function to be optimized, itself), our algorithm is readily applicable. For optimization problems appearing in mathematical systems theory, the monograph of Helmke and Moore [16] presents a variety of ODEs for optimization and we will illustrate the use of our algorithm in this area by applying it to the Rayleigh flow for computing the minimal eigenvalue of a

symmetric matrix which is a particular example from [16].

The remainder of the paper is organized as follows. After introducing the necessary notation at the end of this introduction, Section 2 defines the systems under consideration as well as Runge-Kutta approximations and their representation via hybrid systems and introduces the precise problem formulation. Moreover, preliminary Lyapunov function results from [18] are recalled for convenience of the reader. In Section 3 we first present and analyze our abstract step size control algorithm and then discuss a particular implementation of this algorithm and illustrate its performance by means of two numerical examples. Section 4 then discusses the application nonlinear optimization, gives a brief survey of approaches from the literature to which our algorithm applies and finally illustrates the performance of our algorithm for the Rayleigh flow.

1.1 Notation

By $C^0(A\,;\,\Omega)$, we denote the class of continuous functions on $A \subseteq \mathbb{R}^n$, which take values in $\Omega \subseteq \mathbb{R}^m$. By $C^k(A\,;\,\Omega)$, where $k \geq 1$ is an integer, we denote the class of differentiable functions on A with continuous derivatives up to order k, which take values in Ω.

For a vector $x \in \mathbb{R}^n$ we denote by $\|x\|$ the Euclidean norm and by x^T its transpose. By $B_\varepsilon(x)$, where $\varepsilon > 0$ and $x \in \mathbb{R}^n$, we denote the ball of radius $\varepsilon > 0$ centered at $x \in \mathbb{R}^n$, i.e., $B_\varepsilon(x) := \{\, y \in \mathbb{R}^n \,:\, |y-x| < \varepsilon \,\}$.

$\mathbb{R}^+$ denotes the set of non-negative real numbers and $\mathbb{Z}^+$ the set of non-negative integer numbers. By $\mathcal{K}_\infty$ we denote the set of all strictly increasing and continuous functions $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ with $\rho(0) = 0$ and $\lim_{s\to+\infty} \rho(s) = +\infty$.

For every scalar continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$, $\nabla V(x)$ denotes the gradient of V at $x \in \mathbb{R}^n$, i.e., $\nabla V(x) = (\frac{\partial V}{\partial x_1}(x), \ldots, \frac{\partial V}{\partial x_n}(x))$. We say that a function $V : \mathbb{R}^n \to \mathbb{R}^+$ is positive definite if $V(x) > 0$ for all $x \neq 0$ and $V(0) = 0$. We say that a continuous function $V : \mathbb{R}^n \to \mathbb{R}^+$ is radially unbounded if for every $M > 0$ the set $\{\, x \in \mathbb{R}^n \,:\, V(x) \leq M \,\}$ is compact.

2 Setting and problem formulation

We consider autonomous differential equations of the type

$$\dot{z}(t) = f(z(t))\ ,\ z(t) \in \mathbb{R}^n \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a locally Lipschitz vector field for which there exists $x^* \in \mathbb{R}^n$ with $f(x^*) = 0$. Without loss of generality we may assume $x^* = 0$. For every $z_0 \in \mathbb{R}^n$ and $t \geq 0$, the solution of (1) with initial condition $z(0) = z_0$ will be denoted by $z(t)$ or by $z(t, z_0)$ if we want to emphasize the dependence on the initial value z_0.

2.1 Runge-Kutta schemes

A standard way of obtaining a numerical approximation of this solution is via a Runge-Kutta scheme. Here we summarize the facts for these schemes we need in this paper. Proofs and more details can be found, e.g., in the monographs [4], [14, 15] or [25].

Given an approximation $\tilde{z} \approx z(t)$ for some $t \geq 0$ and a time step $h > 0$, an approximation $\Phi(\tilde{z},h) \approx z(t+h)$ via an s-stage Runge-Kutta method is given by

$$k_j = f\left(\tilde{z} + h\sum_{l=1}^{s} a_{jl}k_l\right), \quad j = 1,\ldots,s \tag{2}$$

$$\Phi(\tilde{z},h) := \tilde{z} + h\sum_{j=1}^{s} b_j k_j \tag{3}$$

Here a_{jl}, b_j, $j,l = 1,\ldots,s$, are the *coefficients* of the scheme and $k_1,\ldots,k_s$ are called the *stages* of the scheme. Some popular examples for Runge-Kutta schemes can be found in Section 3.3, below. If $a_{jl} = 0$ for all $l \geq j$ and all $j = 1,\ldots,s$, then the scheme is called *explicit* and the equations (2) can be evaluated recursively. Otherwise, the scheme is called *implicit* and the equations (2) form a system of (in general nonlinear) equations. Under the Lipschitz assumption on f one can show using Banach's fixed point theorem that there exists a continuous function $h_{max} : \mathbb{R}^n \to \mathbb{R}^+$ such that a unique solution of (2) exists for each $\tilde{z} \in \mathbb{R}^n$ and each $h \in (0, h_{max}(\tilde{z})]$, see, e.g. [18]. In a practical implementation, (2) needs to be solved by some numerical method, e.g., a fixed-point iteration or Newton's method. Even though this introduces additional numerical effort in computing $\Phi(\tilde{z},h)$, this effort may pay off when solving stiff equations.

Given the initial time $\tau_0 = 0$, an initial value $z_0 \in \mathbb{R}^n$ and a sequence of time steps $h_i > 0$ we recursively define the times $\tau_{i+1} := \tau_i + h_i$. Then, one can generate approximations $\tilde{z}_i \approx z(\tau_i, z_0)$ at the times τ_i via the iteration

$$\tilde{z}_0 := z_0, \quad \tilde{z}_{i+1} := \Phi(\tilde{z}_i, h_i).$$

In order to analyze the convergence of a Runge-Kutta scheme, one looks at the approximation error $e_i := \|\tilde{z}_i - z(\tau_i, z_0)\|$. For estimating this error, the concept of *consistency* is used.

Definition 1. A Runge-Kutta scheme is called *consistent* with *order* $p \geq 1$, if for each compact set $K \subset \mathbb{R}^n$ there exists $\bar{h} > 0$ and a constant $C > 0$ such that the inequality

$$\|\Phi(z_0,h) - z(h,z_0)\| \leq Ch^{p+1} \tag{4}$$

holds for all $z_0 \in K$ and all $h \in (0,\bar{h}]$, where $z(h,z_0)$ denotes the solution of (1) and $\bar{h} > 0$ is chosen such that this solution exists for all $z_0 \in K$ and all $h \in (0,\bar{h}]$.

The consistency and the order of consistency depends on the coefficients of the scheme. For instance, the condition $\sum_{j=1}^{s} b_j = 1$ ensures consistency with order $p = 1$ for continuously differentiable vector fields f. Additional conditions on the coefficients a_{jl} and b_j ensure higher order convergence, i.e., (5) with $p \geq 2$, provided the vector field f is sufficiently smooth. Consistency together with a Lipschitz-type stability condition (which holds for any Runge-Kutta scheme provided f in (1) is Lipschitz) then implies convergence of the scheme. More precisely, if the scheme is consistent and if the solution $z(t,z_0)$ exists for $t \in [0,T]$, then there exists a

constant $C_T > 0$, such that for any selection of time steps $h_0, \ldots, h_{N-1} > 0$ satisfying $\tau_i \in [0,T]$ for $i = 0, \ldots, N$ the inequality

$$\max_{i=0,\ldots,N} e_i \leq C_T h^p \tag{5}$$

holds for all sufficiently small $h > 0$, where $h := \max_{i=0,\ldots,N} h_i$ and p is the order of consistency of the scheme.

2.2 Runge-Kutta schemes as hybrid systems

Our goal in this paper is to analyze the dynamical behavior of the numerical approximation, more precisely its asymptotic stability at the origin. To this end, we need to interpret the values $\tilde{z}_i$ as states of a dynamical system. This is a relatively easy task if the time steps h_i are constant, i.e., $h_i \equiv h$ for all $i = 0, \ldots, N$, since in this case $\tilde{z}_{i+1} = \Phi(\tilde{z}_i, h)$ defines a standard discrete time dynamical system. However, if the h_i are time varying — which is the case we consider in this paper — the situation becomes more complicated. Varying time steps can, for instance, be handled as part of an extended state space, cf. [22], or by defining the discrete time system on the nonuniform time grid $\{\tau_0, \tau_1, \tau_2, \ldots\}$ induced by the time steps, cf. [21] or [7]. Here, we choose another option, namely to represent the numerical approximation by a hybrid dynamical system of the form

$$\begin{aligned} &\dot{x}(t) = F(h_i, x(\tau_i)) \;,\; t \in [\tau_i, \tau_{i+1}) \\ &\tau_0 = 0 \;,\; \tau_{i+1} = \tau_i + h_i \\ &h_i = \varphi(x(\tau_i)) \exp(-u(\tau_i)) \\ &x(t) \in \mathbb{R}^n \,, u(t) \in [0, +\infty) \end{aligned} \tag{6}$$

where $\varphi \in C^0(\mathbb{R}^n; (0,r])$, $r > 0$ is a constant, $F : \bigcup_{x \in \mathbb{R}^n} ([0, \varphi(x)] \times \{x\}) \to \mathbb{R}^n$ is a (not necessarily continuous) vector field with $F(h,0) = 0$ for all $h \in [0, \varphi(0)]$, $\lim_{h \to 0^+} F(h,z) = f(z)$, for all $z \in \mathbb{R}^n$. The function $u : \mathbb{R}^+ \to \mathbb{R}^+$ is a locally bounded input to the system whose meaning will be described below.

The solution $x(t)$ of the hybrid system (6) is obtained for every such u by setting $\tau_0 = 0$, $x(0) := x_0$ and then proceeding iteratively for $i = 0, 1, \ldots$ as follows (cf. [17]):

1. Given τ_i and $x(\tau_i)$, calculate τ_{i+1} according to $\tau_{i+1} = \tau_i + \varphi(x(\tau_i)) \exp(-u(\tau_i))$
2. Compute the state trajectory $x(t)$, $t \in (\tau_i, \tau_{i+1}]$ as the solution of the differential equation $\dot{x}(t) = F(h_i, x(\tau_i))$, i.e., $x(t) = x(\tau_i) + (t - \tau_i) F(h_i, x(\tau_i))$ for $t \in (\tau_i, \tau_{i+1}]$.

We denote the resulting trajectory by $x(t, x_0, u)$ or briefly $x(t)$ when x_0 snd u are clear from the context.

Any Runge-Kutta scheme can be represented by a hybrid system (6) by defining

$$F(h,x) := h^{-1}(\Phi(x,h) - x) = \sum_{j=1}^{s} b_j k_j \tag{7}$$

Indeed, from the explicit solution formula in Step 2, above, it immediately follows that the solutions of the hybrid system using this F and $x_0 = z_0$ satisfy

$$x(\tau_i, x_0, u) = \tilde{z}_i.$$

The corresponding time steps $h_i = \varphi(x(\tau_i))\exp(-u(\tau_i))$ are determined via the state dependent function $\varphi(x(\tau_i))$ and the time dependent factor $\exp(-u(\tau_i)) \in (0,1]$. Hence, $\varphi(x(\tau_i))$ can be interpreted as the maximal allowable step size for the state $x(\tau_i)$ (with global upper bound $r > 0$) and $u(\tau_i)$ can be used to arbitrarily reduce this value. Note that for implicit Runge-Kutta schemes typically an upper bound on the step size is needed in order to ensure solvability of the system of equations (2) and the function φ can be used for this purpose, for details we refer to [18].

We will further assume that there exists a continuous, non-decreasing function $M : \mathbb{R}^+ \to \mathbb{R}^+$ such that

$$\|F(h,x)\| \le \|x\| M(\|x\|) \text{ for all } x \in \mathbb{R}^n \text{ and } h \in [0, \varphi(x)] \tag{8}$$

This condition implies that (6) has the "Boundedness-Implies-Continuation" property and thus for each locally bounded input $u : \mathbb{R}^+ \to \mathbb{R}^+$ and $x_0 \in \mathbb{R}^n$ there exists a unique absolutely continuous solution function $[0, +\infty) \ni t \to x(t) \in \mathbb{R}^n$ with $x(0) = x_0$, see [17]. Appropriate step size restriction can always guarantee that (8) holds for F from (7), cf. [18].

Modeling numerical schemes (and particularly Runge-Kutta schemes) as hybrid systems is nonstandard but has certain advantages compared to the alternative discrete time formulations approaches from [7, 21, 22]. For instance, here we aim at stability statements for all step size sequences $(h_i)_{i \in \mathbb{N}_0}$ with $h_i > 0$ and $h_i \le \varphi(x(\tau_i))$, cf. the discussion after Definition 3, below. Once φ is fixed, for the hybrid system (6) this is equivalent to ensuring the desired stability property for all locally bounded functions $u : \mathbb{R}^+ \to \mathbb{R}^+$. Hence, our hybrid approach leads to an explicit condition ("for all u") while the discrete time approach leads to a more technical implicit condition ("for all h_i satisfying $h_i \le \varphi(x(\tau_i))$"). Moreover, the formulation via hybrid models enables us to use readily available stability results from the hybrid control systems literature, while for other formulations we would have to rely on ad hoc arguments.

2.3 Problem formulation

Our general aim is to ensure asymptotic stability of the origin for (6), (7) for suitable choices of φ and all locally bounded inputs u, provided the origin is asymptotically stable for (1). To this end, we first precisely define these stability properties.

For the differential equation (1) we use the following condition, cf. [23] (see also [17, 19]).

Definition 2. We say that the origin $0 \in \mathbb{R}^n$ is *uniformly globally asymptotically stable* (UGAS) for (1) if it is

(i) *Lyapunov stable*, i.e., for each $\varepsilon > 0$ there exists $\delta > 0$ such that $\|z(t, z_0)\| \le \varepsilon$ for all $t \ge 0$ and all $z_0 \in \mathbb{R}^n$ with $\|z_0\| \le \delta$ and

(ii) *uniformly attractive*, i.e., for each $R > 0$ and $\varepsilon > 0$ there exists $T > 0$ such that $\|z(t, z_0)\| \le \varepsilon$ for all $t \ge T$ and all $z_0 \in \mathbb{R}^n$ with $\|z_0\| \le R$.

The next definition formalizes the condition that (6) is asymptotically stable for all locally bounded inputs u, cf. [17].

Definition 3. We say that the origin $0 \in \mathbb{R}^n$ is *uniformly robustly globally asymptotically stable* (URGAS) for (6) if it is

(i) *robustly Lagrange stable*, i.e., for each $R > 0$ it holds that $\sup\{\|x(t,x_0,u)\| \,|\, t \geq 0, \|x_0\| \leq R, u : \mathbb{R}^+ \to \mathbb{R}^+ \text{ locally bounded}\} < \infty$.

(ii) *robustly Lyapunov stable*, i.e., for each $\varepsilon > 0$ there exists $\delta > 0$ such that $\|x(t,x_0,u)\| \leq \varepsilon$ for all $t \geq 0$, all $x_0 \in \mathbb{R}^n$ with $\|x_0\| \leq \delta$ and all locally bounded $u : \mathbb{R}^+ \to \mathbb{R}^+$ and

(iii) *robustly uniformly attractive*, i.e., for each $R > 0$ and $\varepsilon > 0$ there exists $T > 0$ such that $\|x(t,x_0,u)\| \leq \varepsilon$ for all $t \geq T$, all $x_0 \in \mathbb{R}^n$ with $\|x_0\| \leq R$ and all locally bounded $u : \mathbb{R}^+ \to \mathbb{R}^+$.

Contrary to the ordinary differential equation (1), for the hybrid system (6) Lyapunov stability and attraction do not necessarily imply Lagrange stability. This is why — in contrast to Definition 2 — we explicitly included this property in Definition 3.

Note that our choice $\varphi \in C^0(\mathbb{R}^n;(0,r])$ implies $\inf_{x\in\mathcal{N}} \varphi(x) > 0$ for any bounded neighborhood $\mathcal{N}$ of the origin. This implies that the asymptotic stability property can be achieved for a sequence of step sizes h_i which is bounded from below by a positive value. This avoids the undesirable property that the discretization step sizes tend to 0 as $i \to +\infty$. However, as we will see, it will also be possible to make rigorous statements in situations where such a φ does not exist, cf. Theorem 8 and the discussion after its proof.

The stability property in Definition 3 is called *robust* because it requires the respective stability properties uniformly for all locally bounded inputs u and thus for all (positive) step sizes $h_i \leq \varphi(x(\tau_i))$. This is an important feature because it allows us to couple our method with other step size selection schemes. For instance, we could use the step size $\min\{\varphi(x(\tau_i)), \tilde{h}_i\}$ where $\tilde{h}_i$ is chosen such that a local error bound is guaranteed. Such methods are classical, cf. [14] or any other textbook on numerical methods for ODEs. Proceeding this way results in a numerical solution which is asymptotically stable and at the same time maintains a pre-defined accuracy. Note that our approach will not incorporate error bounds, hence the approximation may deviate from the true solution, at least in the transient phase, i.e., away from 0. On the other hand, as [18, Example 2.1] shows, local error based step size control does in general not guarantee asymptotic stability of the numerical approximation. Thus, a coupling of both approaches may be needed in order to ensure both accuracy and asymptotic stability.

Assuming that Definition 2 is satisfied, Definition 3 now gives rise to several problems. The first of these is the following existence problem.

(P1) Existence *Is there a continuous function $\varphi : \mathbb{R}^n \to (0,r]$, such that $0 \in \mathbb{R}^n$ is URGAS for system (6), (7)?*

Provided the answer to **(P1)** is positive, one may then look at the following design problems.

(P2) Global Design *Construct a continuous function* $\varphi : \mathbb{R}^n \to (0,r]$*, such that* $0 \in \mathbb{R}^n$ *is URGAS for system* (6)*,* (7)*.*

(P3) Trajectory based Design *For a given initial value* x_0*, construct a sequence of step sizes* $h_i > 0$ *satisfying* $h_i \leq \varphi(x(\tau_i))$ *for all* $i \in \mathbb{N}$ *and the function* φ *from* **(P1)**.

A variety of results related to Problems **(P1)** and **(P2)** can be found in [18]. In this context we note that any consistent and stable numerical scheme for ODEs inherits the asymptotic stability of the original equation in a practical sense, even for more general attractors than equilibria, see for instance [11, 12] or [25, Chapter 7]. Practical asymptotic stability means that the system exhibits an asymptotically stable set close to the original attractor, i.e., in our case a small neighborhood around the equilibrium point, which shrinks down to the attractor as the time step h tends to 0. In contrast to this, the property defined in Definition 3 is "true" asymptotic stability, a stronger property which cannot in general be deduced from practical stability. In [25, Chapter 5], several results for our problem for specific classes of ODEs are derived using classical numerical stability concepts like A-stability, B-stability and the like. In [18], it was observed that Problems **(P1)** and **(P2)** can be interpreted as feedback stabilization problems for the system (6), (7) in which φ plays the role of the stabilizing feedback law. Consequently, various methods from nonlinear control theory, like small-gain and Lyapunov function techniques, have been applied to these problems in [18] generalizing the results from [25, Chapter 5] to more general classes of systems and to systems with different structural properties. In contrast to [18], in this paper our focus lies on Problem **(P3)** and applications thereof.

2.4 Lyapunov functions

Lyapunov functions are the main technical tool we are going to use in this paper. In this section we collect and extend some results from [18] which form the basis for our algorithm and analysis. The first lemma gives a sufficient Lyapunov condition for the URGAS property for hybrid systems of the form (6). For its proof we refer to [18, Lemma 4.1].

Lemma 4. *Consider system* (6) *and suppose that there exist a continuous, positive definite and radially unbounded function* $V : \mathbb{R}^n \to \mathbb{R}^+$ *and a continuous, positive definite function* $W : \mathbb{R}^n \to \mathbb{R}^+$ *such that for every* $x \in \mathbb{R}^n$ *the following inequality holds for all* $h \in [0, \varphi(x)]$*.*

$$V(x + hF(h,x)) \leq V(x) - hW(x) \tag{9}$$

Then the origin $0 \in \mathbb{R}^n$ *is URGAS for system* (6)*.*

In the following section, we will use inequality (9) in order to construct adaptive step sizes h_i online while computing the numerical solution. To this end, we need to know a Lyapunov function V. Similar to [18], we will use a Lyapunov function for the continuous-time system (1) for this purpose. Such a Lyapunov is defined as follows.

Definition 5. A positive definite, radially unbounded function $V \in C^1(\mathbb{R}^n;\mathbb{R}^+)$ is called a *Lyapunov function* for system (1) if the inequality

$$\nabla V(x) f(x) < 0 \tag{10}$$

holds for all $x \in \mathbb{R}^n \setminus \{0\}$.

As we will see in the proof of Theorem 8, below, such a Lyapunov function for (1) can indeed be used in order to establish (9) for F from (7). As a prerequisite for this proof, in the remainder of this section we establish bounds on the decay of V along the solutions of (1). To this end, observe that the equation

$$\lim_{\substack{h \to 0 \\ h>0}} \frac{V(z(h,x)) - V(x)}{h} = \nabla V(x) f(x)$$

(which follows by the chain rule) together with $\nabla V(x) f(x) < 0$ for $x \neq 0$ implies that for each $\lambda \in (0,1)$, each $x \in \mathbb{R}^n$ and each sufficiently small $h > 0$ the inequality

$$V(z(h,x)) - V(x) \leq h\lambda \nabla V(x) f(x) \tag{11}$$

holds. The following lemma makes the statement "sufficiently small $h > 0$" in this observation more precise.

Lemma 6. *Let $V \in C^1(\mathbb{R}^n;\mathbb{R}^+)$ be a Lyapunov function for system (1) and $\lambda \in (0,1)$. Then the following statements hold.*

(i) For each two constants $R > \varepsilon > 0$ there exists $h_{\varepsilon,R} > 0$ such that (11) holds for all $x \in \mathbb{R}^n$ with $R \geq \|x\| \geq \varepsilon$ and all $h \in (0, h_{\varepsilon,R}]$.

(ii) Assume that $W(x) := -\nabla V(x) f(x)$ is locally Lipschitz and that there exist constants $b > 1$, $\varepsilon, c > 0$ and a continuous positive definite function $\ell : \mathbb{R}^n \to \mathbb{R}^+$ satisfying

$$\ell(x) \geq \sup \left\{ \frac{|W(y) - W(z)|}{\|y - z\|} : y, z \in \mathbb{R}^n, y \neq z, \max\{\|y\|, \|z\|\} \leq b\|x\| \right\}$$

for all $x \in \mathbb{R}^n \setminus \{0\}$ and

$$\|x\| \, \ell(x) \leq c W(x) \tag{12}$$

for all $x \in B_\varepsilon(0)$. Then there exists a continuous function $\varphi \in C^0(\mathbb{R}^n;(0,r])$ for some $r > 0$ such that (11) holds for all $x \in \mathbb{R}^n$ and all $h \in (0, \varphi(x))$.

Proof. (i) Fix an arbitrary $\eta \in (0,\varepsilon)$ and consider the compact sets $K := \{x \in \mathbb{R}^n \,|\, R \geq \|x\| \geq \varepsilon\}$ and $K_\eta := \{x \in \mathbb{R}^n \,|\, R + \eta \geq \|x\| \geq \varepsilon - \eta\}$ and the map $x \mapsto \nabla V(x) f(x)$. This map is continuous and attains negative values on K, hence the value $\alpha := \max_{x \in K} \nabla V(x) f(x)$ exists and satisfies $\alpha < 0$. Moreover, since any continuous map is uniformly continuous on each compact set, the map is uniformly continuous on K_η, i.e., for given $\varepsilon' > 0$ there exists $\delta > 0$ such that

$$|\nabla V(x) f(x) - \nabla V(y) f(y)| \leq \varepsilon'$$

holds for all $x, y \in K_\eta$ with $\|x - y\| \leq \delta$.

Since the vector field f is also continuous, its norm $\|f(x)\|$ is bounded on K_η by some $M > 0$ and thus for all $t \in [0, \eta/M]$ we obtain that $z(t,x) \in K_\eta$ and $\|z(t,x) - x\| \leq tM$ for all $x \in K$.

Now we set $\varepsilon' = (\lambda - 1)\alpha$, pick $\delta > 0$ from the uniform continuity property (without loss of generality we may choose $\delta \leq \eta$) and set $h_{\varepsilon,R} := \delta/M$. Then for all $x \in K$ and all $t \in (0, h_{\varepsilon,R}]$ we obtain $\|z(t,x) - x\| \leq \delta$ and thus for all $h \in (0, h_{\varepsilon,R}]$ we can estimate

$$\begin{aligned} V(z(h,x)) - V(x) &= \int_0^h \nabla(z(t,x)) f(z(t,x)) dt \\ &\leq \int_0^h \nabla(x) f(x) + \varepsilon' dt \; = h\,\nabla(x) f(x) + h(\lambda - 1)\alpha \\ &\leq h\nabla(x) f(x) + h(\lambda - 1)\nabla(x) f(x) \; = \; h(1-\lambda)\nabla(x) f(x) \end{aligned}$$

which shows the claim.

(ii) Follows from [18, Lemma 4.3]. □

3 Lyapunov function based step size control

In this section we present our Lyapunov function based step size control algorithm. We start by stating and analyzing an "abstract" version of this algorithm and then describe the details of our implementation and illustrate it by means of two numerical examples.

3.1 An abstract step size control algorithm

The following algorithm provides an abstract step size selection method based on a Runge-Kutta scheme for (1), expressed via (7) as a hybrid system (6), a Lyapunov function V for (1) according to Definition 5 and its derivative ∇V. Moreover, we assume that we have defined two functions

$$h_{reduce} : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^+ \quad \text{and} \quad h_{new} : \mathbb{R}^+ \times \mathbb{R}^n \to \mathbb{R}^+,$$

which for a given $h > 0$ and $x \in \mathbb{R}^n$ produce a reduced step size $h_{reduce}(h,x) < h$ and a new step size $h_{new}(h,x) > 0$. In order to simplify the presentation, the algorithm uses a maximal step size $h_{max} > 0$ which does not depend on the state x, but the inclusion of a state dependent upper bound is straightforward.

Algorithm 7. **(Lyapunov function based step size control algorithm)**

Input: Initial value $x_0 \in \mathbb{R}^n$, initial step size $h_0 > 0$, maximal step size $h_{max} > 0$, parameter $\lambda \in (0,1)$

(0) set $x(0) := x_0$, $\tau_0 := 0$, $i := 0$
(1) set $h_i := \min\{h_i, h_{max}\}$ and compute $x(\tau_i + h_i) := \Phi(x(\tau_i), h_i)$
(2) while $V(x(\tau_i + h_i)) - V(x(\tau_i)) > \lambda h \nabla V(x(\tau_i)) f(x(\tau_i))$
(3) set $h_i := h_{reduce}(h_i, x(\tau_i))$ and recompute $x(\tau_i + h_i) := \Phi(x(\tau_i), h_i)$
(4) end while
(5) set $h_{i+1} := h_{new}(h_i, x(\tau_i))$, $\tau_{i+1} := \tau_i + h_i$, $i := i+1$ and go to (1)

Note that this algorithm does not have a termination criterion and hence — in principle — produces infinitely many values $x(\tau_i)$. Of course, if only a solution on some interval $[0,T]$ is desired, the algorithm could be modified accordingly.

Since the above algorithm only produces one single sequence of step sizes h_i for each initial value x_0, it does not make sense to talk about robust stability concepts anymore. Moreover, since the Lyapunov function condition in step (2) is only ensured at the discrete times τ_i, we cannot in general expect stability properties for all $t \geq 0$ (although under suitable conditions they could be recovered by continuity arguments, see, e.g., [24]). However, under appropriate assumptions on h_{reduce} and h_{new} we can still recover the required properties from Definition 3 at the discrete time instants τ_i, as the following theorem states.

Theorem 8. *Consider Algorithm 7 in which Φ is a consistent Runge-Kutta scheme with order $p \geq 1$. Let $V \in C^1(\mathbb{R}^n;\mathbb{R}^+)$ be a Lyapunov function for system (1) and $\lambda \in (0,1)$. Assume that $h_{\max} > 0$ is chosen such that $\Phi(x,h)$ is defined for all $x \in \mathbb{R}^n$ and $h \in (0,h_{max}]$ and that the functions h_{reduce} and h_{new} satisfy*

- *there exist real values $0 < \rho_1 \leq \rho_2 < 1$ such that $h_{reduce}(x,h) \in [\rho_1 h, \rho_2 h]$ holds for all $x \in \mathbb{R}^n$ and all $h > 0$ for which the condition in Step (2) of Algorithm 7 is satisfied*
- *$h_{new}(x,h) \geq h$ holds for all $x \in \mathbb{R}^n$ and all $h > 0$ for which the condition in Step (2) of Algorithm 7 is not satisfied.*

Then, for each initial value Algorithm 7 generates an infinite sequence of time steps h_i (i.e., the while loop in steps (2)–(4) always terminates) and at the times τ_i the resulting trajectories are

(i) Lagrange stable*, i.e., for each $R > 0$ it holds that* $\sup\{\|x(\tau_i,x_0)\| \,|\, i \in \mathbb{N}, \|x_0\| \leq R\} < \infty$.

(ii) Lyapunov stable*, i.e., for each $\varepsilon > 0$ there exists $\delta > 0$ such that $\|x(\tau_i,x_0)\| \leq \varepsilon$ for all $i \in \mathbb{N}$ and all $x_0 \in \mathbb{R}^n$ with $\|x_0\| \leq \delta$*

(iii) uniformly attractive*, i.e., for each $R > 0$ and $\varepsilon > 0$ there exists $T > 0$ such that $\|x(\tau_i,x_0)\| \leq \varepsilon$ for all $\tau_i \geq T$ and all $x_0 \in \mathbb{R}^n$ with $\|x_0\| \leq R$.*

If, in addition, there exists a continuous function $\varphi \in C^0(\mathbb{R}^n;(0,r])$ for some $r > 0$ such that

$$V(\Phi(x,h)) - V(x) \leq h\lambda \nabla V(x) f(x) \tag{13}$$

holds for all $x \in \mathbb{R}^n$ and all $h \in (0,\varphi(x))$, then for each initial value $x_0 \in \mathbb{R}^n$ there exists $h_{min} > 0$ such that $h_i \geq h_{min}$ holds for all $i \in \mathbb{N}$. In particular, in this case the sequence of times τ_i generated by Algorithm 7 tends to ∞ as $i \to \infty$.

Proof. Let $(a_k)_{k\in\mathbb{N}}$ be an arbitrary sequence with $0 < a_{k+1} < a_k$, $\lim_{k\to\infty} a_k \to 0$ and pick $\tilde{\lambda} \in (\lambda,1)$ arbitrarily. Define the sets $M := \{x \in \mathbb{R}^n \,|\, V(x) \leq a_1\}$, $M_k := \{x \in \mathbb{R}^n \,|\, V(x) \in [a_{k+1},a_k]\}$ and let $k \in \mathbb{N}$ be arbitrary. Since V is continuous, positive definite and radially unbounded, it follows from Lemma 3.5 in [19] that there exist functions $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ with

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|) \quad \text{for all } x \in \mathbb{R}^n. \tag{14}$$

This implies that the sets M and M_k are compact and there exists $R_k > \varepsilon_k > 0$ such that $R_k \geq \|x\| \geq \varepsilon_k$ holds for all $x \in M_k$. Thus we can apply Lemma 6(i) which

implies that there exists $h_{\varepsilon_k,R_k} \in (0,h_{max}]$ such that the condition (11) holds for $\tilde{\lambda}$ in place of λ for all $x \in M_k$ and all $h \in (0,h_{\varepsilon_k,R_k}]$.
Since the scheme is consistent and V is Lipschitz, this implies

$$V(\Phi(h,x)) - V(x) \le V(z(h,x)) - V(x) + LCh^{p+1} \le h\tilde{\lambda}\nabla V(x)f(x) + LCh^{p+1}$$

for all $x \in M_k$ and all $h \in (0,\min\{\bar{h},h_{\varepsilon_k,R_k}\}]$, where $\bar{h}$ is the step size from the consistency condition for the compact set M and L is the Lipschitz constant of V on the set $K = \{\Phi(x,h)\,|\,x \in M, h \in [0,\bar{h}]\} \cup \{z(h,x)\,|\,x \in M, h \in [0,\bar{h}]\}$, which is compact since $M \times [0,h]$ is compact and both $z(\cdot,\cdot)$ and $\Phi(\cdot,\cdot)$ are continuous. Setting

$$\gamma_k := \max_{x \in M_k} \nabla V(x)f(x) < 0 \quad \text{and} \quad h_k' := \left(\frac{(\lambda - \tilde{\lambda})\gamma}{LC}\right)^{\frac{1}{p}} > 0,$$

for $\bar{h}_k := \min\{\bar{h},h_{\varepsilon,R},h'\}$ we thus obtain

$$V(\Phi(x,h)) - V(x) \le h\lambda\nabla V(x)f(x) \le h\lambda\gamma_k \tag{15}$$

for all $x \in M_k$ and all $h \in (0,\bar{h}_k]$.
Now consider the while loop in the steps (2)–(4) of Algorithm 7 for some $x(\tau_i) \in M_k$ and denote by $h_{i,old}$ the time step for which step (1) is executed before entering this loop. Inequality (15) and the inequalities for h_{reduce} then implies that for any $x(\tau_i) \in M_k$ the loop terminates with a time step $h_i \ge \min\{h_{i,old},\rho_1\bar{h}_k\}$ for which $V(\Phi(x(\tau_i),h_i)) \le V(x(\tau_i)) \le h_i\lambda\gamma_k$ holds. Moreover, since $h_{new}(h_i,x(\tau_i)) \ge h_i$, the subsequent time steps h_j, $j \ge i+1$, will satisfy the same lower bound as h_i as long as $x_j \in M_k$ holds. Hence, as long as $x_j \in M_k$ holds, $V(x(\tau_i))$ decreases monotonically with a fixed amount of decay and a uniform positive lower bound on the time step. Thus, by definition of the sets M_k, for each $k \in \mathbb{N}$ there exists a time $t_k > 0$ such that whenever $x(\tau_i) \in M_k$ there exists a $\tau_j \le \tau_i + t_k$ such that either $x(\tau_j) = 0$ (and thus $x(\tau_m) = 0$ for all $\tau_m \ge \tau_j$) or $x(\tau_j) \in M_l$ holds for some $l \ge k+1$. By induction, for each $k \in \mathbb{N}$ one thus obtains a time $T_k > 0$ such that for each $x_0 \in M$ there exists some $i_k \in \mathbb{N}$ such that $\tau_{i_k} \le T_k$ and either $x(\tau_i) \in M_k$ or $x(\tau_i) = 0$ holds for all $i \ge i_k$.
The three stability properties (i)–(iii) are now readily concluded from this property:
(i) Given $R > 0$ we choose $a_1 = \alpha_2(R)$ which implies that each $x_0 \in \mathbb{R}^n$ with $\|x_0\| \le R$ lies in M. Then the whole solution $x(\tau_i)$ lies in M which implies $\|x(\tau_i)\| \le \alpha_1^{-1}(\alpha_2(R))$ which implies Lagrange stability.
(ii) Given $\varepsilon > 0$ we choose $\delta = \alpha_2^{-1}(\alpha_1(\varepsilon))$. Then the inequality needed for Lyapunov stability follows as in (i) with δ in place of R.
(iii) Given R and ε, choose a_1 as in (i) and k so large that $a_k \le \alpha_1(\varepsilon)$ holds. Then, for $T = T_k$ and all $\tau_i \ge T$, the solution $x(\tau_i)$ is either equal to 0 or lies in M_l for some $l \ge k$. This implies $\|x(\tau_i)\| \le \alpha_1^{-1}(V(x(\tau_i))) \le \alpha_1^{-1}(a_k) = \varepsilon$ for all $\tau_i \ge T$.
We finish the proof by proving the additional property of the h_i in case that $\varphi \in C^0(\mathbb{R}^n;(0,r])$ exists such that (13) holds. To this end, pick an arbitrary $x_0 \in \mathbb{R}^n$ and choose a_1 so large that $x_0 \in M$ holds. Then, (13) implies that the values $\bar{h}_k$ defined in the proof, above, can be bounded from below by $h^* := \min_{x \in M} \varphi(x) > 0$. By induction, the inequality $h_i \ge \min\{h_{i,old},\rho_1\bar{h}_k\}$ then implies $h_i \ge \min\{h_0,\rho_1 h^*\} > 0$ for all $i \in \mathbb{N}$ which yields the desired positive lower bound on the step sizes h_i. □

We note that various sufficient conditions ensuring the existence of $\varphi \in C^0(\mathbb{R}^n;(0,r])$ with (13) can be found in [18, Section 4]. We emphasize, however, that even without this condition the stability properties defined in Theorem 8 and in particular the convergence $x(\tau_i) \to 0$ is ensured. In particular, the numerical solution will converge to the origin even if Problem **(P1)** does not have a solution.

3.2 Implementation of the algorithm

There are various ways of defining h_{reduce} and h_{new} such that the conditions of Theorem 8 are satisfied. A simple way, proposed in [18] is to define

$$h_{reduce}(h,x) := h/2 \quad \text{and} \quad h_{new}(h,x) := 2h.$$

This choice produces reasonable results (cf. [18]) but due to the "try and error" nature of the resulting algorithm it has the disadvantage that typically the while loop is executed at least once for each i, implying that Φ is usually evaluated at least twice for each i.

A more efficient way of defining h_{reduce} and h_{new} is obtained by using ideas from the classical error estimation based step size control, cf. e.g. [14, Section II.4]. To this end, define the Lyapunov differences

$$\Delta V(x,h) := V(z(h,x)) - V(x) \quad \text{and} \quad \widetilde{\Delta V}(x,h) := V(\Phi(x,h)) - V(x)$$

for the exact solution and the numerical approximation. If V and f are sufficiently smooth, then for a p-th order scheme there exists a constant $c \in \mathbb{R}$ such that the approximate equality

$$\widetilde{\Delta V}(x,h) \approx \Delta V(x,h) + ch^{p+1}$$

holds for all sufficiently small $h > 0$. Hence, we can approximately compute c as

$$c \approx \frac{\widetilde{\Delta V}(x,h) - \Delta V(x,h)}{h^{p+1}}.$$

We now intend to find a time step $\tilde{h} > 0$ such that

$$\widetilde{\Delta V}(x,\tilde{h}) \le \lambda \tilde{h} \nabla V(x) f(x)$$

holds, which will be approximately satisfied if the inequality

$$\Delta V(x,\tilde{h}) + c\tilde{h}^{p+1} \le \lambda \tilde{h} \nabla V(x) f(x)$$

holds, i.e, if

$$\tilde{h} \le \left(\frac{\lambda \nabla V(x) f(x) - \Delta V(x,\tilde{h})/\tilde{h}}{c} \right)^{\frac{1}{p}}$$

holds. Inserting the approximate value for c, we obtain the condition

$$\tilde{h} \le h \left(\frac{\lambda \nabla V(x) f(x) - \Delta V(x,\tilde{h})/\tilde{h}}{\widetilde{\Delta V}(x,h)/h - \Delta V(x,h)/h} \right)^{\frac{1}{p}}.$$

This suggests to use the expression on the right hand side of this inequality as a candidate for a new step size in our algorithm for $h = h_i$. However, this expression is implicit (as it contains the unknown $\tilde{h}$) and contains the values $\Delta V(x,h)$ which are not available in practice as they depend on the exact solution.

Both problems vanish if we replace the term $\Delta V(x,h)$ by its first order approximation $h\nabla V(x)f(x)$ (both for h and $\tilde{h}$) which leads to the expression

$$h\left(\frac{(\lambda-1)\nabla V(x)f(x)}{\widetilde{\Delta V}(x,h)/h-\nabla V(x)f(x)}\right)^{\frac{1}{p}}.$$

Although the first order approximation $\Delta V(x,h) \approx h\nabla V(x)f(x)$ introduces an error of higher order than the error of the scheme, the resulting step size control mechanism shows very good results (cf. the discussion at the end of Example 9), probably due to the fact that the choice of $\lambda < 1$ introduces some tolerance against additional approximation errors.

For the practical implementation, we moreover need to take into account that the denominator $\widetilde{\Delta V}(x,h)/h - \nabla V(x)f(x)$ may become negative or 0 — this happens if the discretization error speeds up the convergence to the origin instead of slowing down the convergence. To this end, we replace the denominator by $\max\{\widetilde{\Delta V}(x,h)/h - \nabla V(x)f(x), eps(\lambda-1)\nabla V(x)f(x)\}$, where $eps > 0$ is a small positive constant. Finally, in order to compensate for the various approximations during the derivation, we multiply our formula with a security factor $\rho \in (0,1)$. All in all, we end up with

$$h_{reduce}(h,x) := \rho h\left(\frac{(\lambda-1)\nabla V(x)f(x)}{\max\{\widetilde{\Delta V}(x,h)/h-\nabla V(x)f(x), eps(\lambda-1)\nabla V(x)f(x)\}}\right)^{\frac{1}{p}}. \tag{16}$$

For h_{new} we may use the same formula, i.e.,

$$h_{new}(h,x) := h_{reduce}(h,x), \tag{17}$$

although this formula does not rigorously ensure the condition $h_{new}(x,h) \geq h$ imposed in Theorem 8 (it would satisfy this condition if all approximate equations in the derivation were exact). As an alternative, one might use the definition $h_{new}(h,x) := \max\{h, h_{reduce}(h,x)\}$, however, the difference between these two choices turned out to be marginal in our numerical simulations and since (17) yields a slightly lower number of evaluations of Φ we have used this variant in our simulations.

3.3 Examples

In our simulations we run Algorithm 7 with (16) for the Euler, the Heun and the classical Runge-Kutta scheme. All these schemes are explicit and thus satisfy $a_{jl} = 0$ for all $j,l = 1,\ldots,s$ with $l \geq j$. The Euler scheme is a scheme of order $p = 1$ with $s = 1$ stages and coefficient $b_1 = 1$, the Heun scheme is an $s = 2$ stage scheme of order $p = 2$ with

$$s = 2,\ a_{21} = 1,\ b_1 = b_2 = 1/2$$

and the classical Runge-Kutta scheme (henceforth abbreviated as RK4) is of order $p = 4$ with $s = 4$ stages and

$$a_{21} = a_{32} = 1/2, a_{43} = 1, a_{41} = a_{42} = a_{31} = 0, b_1 = b_4 = 1/6, b_2 = b_3 = 1/3.$$

The standard parameters in all examples were $h_0 = 0.1$, $h_{max} = 1$, and $\rho = 0.9$ and $eps = 0.01$ in Formula 16.

Example 9. The first example we investigate is the 2d differential equation

$$\dot{z}_1 = -z_1 + z_2^2, \quad \dot{z}_2 = -z_2 - z_1 z_2$$

with Lyapunov function $V(x) = \|x\|^2$, cf. [18, Example 4.15]. The Figures 1–3 show simulation results (phase portrait, Lyapunov function $V(x(\tau_i))$ over time and time steps) on the time interval $[0,20]$ with $\lambda = 0.5$ and $x_0 = (5,5)^T$.

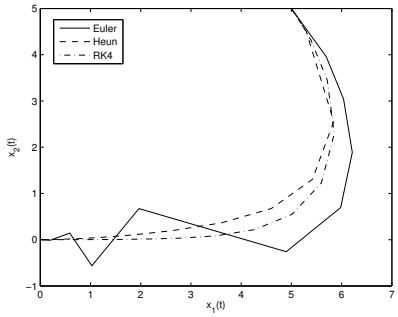

Figure 1: Phase portrait for Example 9 with $\lambda = 0.5$ and $x_0 = (5,5)^T$

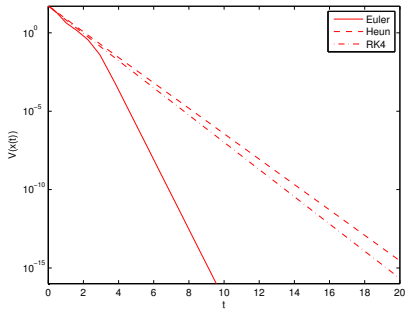

Figure 2: Lyapunov function (logarithmic scale) for Ex. 9, $\lambda = 0.5$, $x_0 = (5,5)^T$

All solutions approach the equilibrium $x = 0$ very quickly. The total number of steps for the three schemes on the time interval $[0,20]$ were 28 for the Euler scheme, 42 for the Heun scheme and 52 for the RK4 scheme. Here, two facts are worth noting.

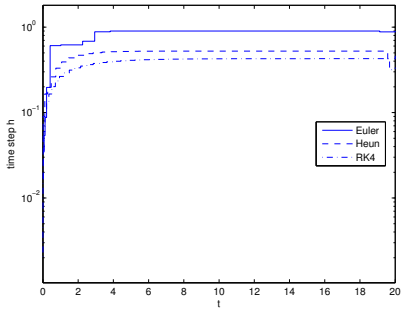

Figure 3: Time steps (logarithmic scale) for Example 9 with $\lambda = 0.5$ and $x_0 = (5,5)^T$

First, although the Euler scheme is the scheme with the lowest order, it allows for the largest steps and needs the fewest total number of steps. This is due to the fact that for asymptotic stability not only the size of the error matters but also the direction in which the error distorts the solution. Here, the error in the Euler scheme speeds up the convergence towards 0 and hence there is no reason for the scheme to reduce the time step. In contrast to this, while the errors for the Heun and the RK4 scheme are smaller, they have a tendency to slow down the convergence to 0 and hence the time steps have to be chosen more cautiously.

Second, it is clearly visible that our step size control Algorithm 7 does by no means ensure that the numerical solution is close to the exact solution during the transient phase, i.e., until a small neighborhood of the equilibrium is reached. In fact, the three numerical solutions differ considerably and the Euler solution actually looks quite irregular. This is not a drawback of our algorithm but actually intended, since all the algorithm cares about is the convergence to $x = 0$ which is perfectly achieved for all schemes. If, in addition, a faithful reproduction of the exact solution during the transient phase is desired, our step size control algorithm could be coupled with traditional error estimation based techniques.

In order to illustrate the effects of different choices of $\lambda \in (0,1)$, Figure 4 shows the time steps for the RK4 scheme for $\lambda = 0.1, 0.5, 0.9$.

As expected, the time steps become the smaller the closer the value λ is to 1, i.e., the more decay of the Lyapunov function is supposed to be guaranteed. The total number of steps for the simulations on the time interval $[0,20]$ was 28 for $\lambda = 0.1$, 52 for $\lambda = 0.5$ and 290 for $\lambda = 0.9$.

In order to investigate the efficiency of Formula (16), we have changed the definition of h_{new} in (17) by using Formula (16) with $\rho = 1.1$ instead of 0.9 (the ρ in the formula for h_{reduce} remains unchanged). With this enlarged ρ, it turns out that the condition in step (2) of Algorithm 7 is violated in more than 90% of the iterations (similar results have been obtained for Example 10, below). In contrast to that, with the choice of $\rho = 0.9$ this typically happens in less than 5% of the iterations, which shows that Formula (16) with $\rho = 0.9$ very precisely predicts a good (i.e., small enough but not too small) time step h_{i+1} for the next time step.

Example 10. Our second example is [18, Example 4.11(ii)], given by the 2d differ-

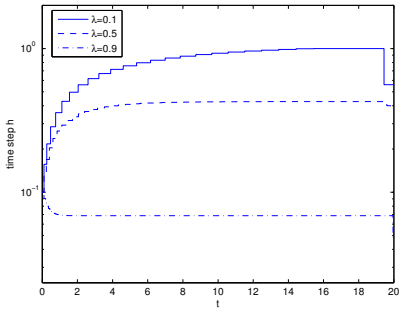

Figure 4: Time steps from Algorithm 7 (logarithmic scale) using the RK4 scheme applied to Example 9 with $\lambda = 0.1, 0.5, 0.9$ and $x_0 = (5,5)^T$

ential equation

$$\dot{z}_1 = -\|z\|^2 z_1 + z_2, \quad \dot{z}_2 = -z_1 - \|z\|^2 z_2$$

with Lyapunov function $V(z) = \|z\|^2$. Since the vector field for this example satisfies $\|f(z)\| = O(\|z\|)$ in a neighborhood of $z = 0$, one can show that in a neighborhood of $z = 0$ the consistency error of a p-th order Runge-Kutta scheme satisfies

$$\|\Phi(z_0,h) - z(h,z_0)\| = O(h^{p+1}\|z_0\|)$$

which, since V is quadratic, implies

$$\begin{aligned} \left|\widetilde{\Delta V}(z_0,h) - \Delta V(z_0,h)\right| &= \left|\Big(V(\Phi(z_0,h)) - V(z_0)\Big) - \Big(V(z(h,z_0)) - V(z_0)\Big)\right| \\ &= O(h^{p+1}\|z_0\|^2). \end{aligned}$$

On the other hand, writing the system in polar coordinates one verifies that

$$\Delta V(z_0,h) = O(h\|z_0\|^4),$$

again in a neighborhood of 0. Hence, for each fixed $h > 0$ and all z_0 sufficiently close to 0 the inequality

$$\widetilde{\Delta V}(z_0,h) < 0 \tag{18}$$

can *not* be guaranteed from these estimates, since the Lyapunov difference consistency error $|\widetilde{\Delta V}(z_0,h) - \Delta V(z_0,h)|$ is not guaranteed to be smaller than the exact decay $\Delta V(z_0,h)$. Since the analysis in [18] uses similar estimates, this explains why none of the sufficient conditions in [18] guaranteeing asymptotic stability for $h \not\to 0$ is satisfied for this example.

However, the fact that (18) is not guaranteed by this analysis does, of course, not imply that this inequality does not hold. Indeed, the fact that the difference $\|\Phi(z_0,h) - z(h,z_0)\|$ is large does not necessarily imply that the difference $|\widetilde{\Delta V}(z_0,h) - \Delta V(z_0,h)|$ is large: it may well happen that the error included in $\Phi(z_0,h)$ is large compared to $\Delta V(z_0,h)$ but nevertheless does not act destabilizing, because it changes the exact solution $z(h,z_0)$ into a direction in which V does not grow — or does even further

decrease. In fact, we already observed this behavior in Example 9 for the Euler scheme and will also observe it for this example, but now for the RK4 scheme.

The Figures 5–7 show the simulation results (phase portrait, Lyapunov function $V(x(\tau_i))$ over time and the time steps) on the time interval $[0,200]$ with $\lambda = 0.5$ and $x_0 = (5,5)^T$.

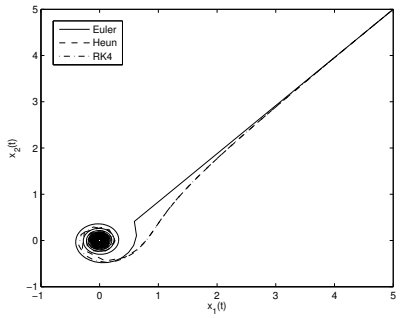

Figure 5: Phase portrait for Example 10 with $\lambda = 0.5$ and $x_0 = (5,5)^T$

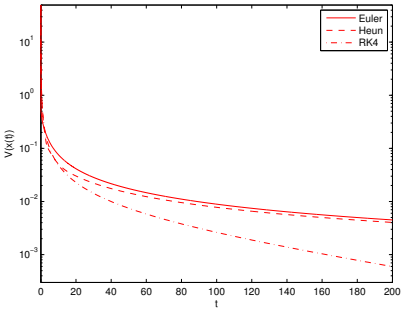

Figure 6: Lyapunov function (logarithmic scale) for Ex. 10, $\lambda = 0.5$, $x_0 = (5,5)^T$

The total number of steps is 24925 for the Euler scheme, 621 for the Heun scheme and 240 for the RK2 scheme. Hence, in this example the benefit of using higher order schemes is clearly visible.

However, the advantage of the RK4 is not only due to the higher order. Looking at the step sizes one sees that for the Euler and the Heun scheme the step size is strictly decreasing after the first couple of time steps. Longer simulations indicate that the sequences indeed converge to 0 which is in accordance with the discussion above, i.e., that decay of the Lyapunov function can only be guaranteed for vanishing step size h if the discretization error acts destabilizing, which appears to be the case for these two schemes. In contrast to this, the error in the RK4 scheme has a stabilizing effect, because we observe a much faster decay of the Lyapunov function V than in

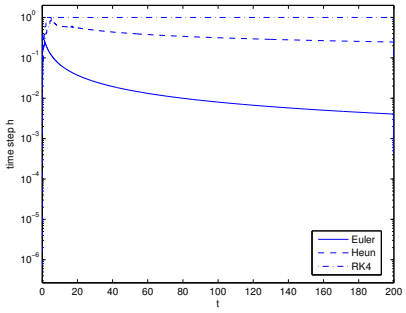

Figure 7: Time steps (logarithmic scale) for Example 10, $\lambda = 0.5$, $x_0 = (5,5)^T$

the other examples (even faster than for the exact solutions), while the step sizes are constantly equal to the maximal allowed step size $h_{max} = 1$.

Summarizing, our examples show that the step size control scheme manages to obtain asymptotically stable solutions for different numerical schemes. A particular feature of the scheme is that step sizes are only reduced if a large error has destabilizing effect, while the scheme allows for large step sizes (and errors) as long as they do not affect the decay of the Lyapunov function.

4 Application to optimization

An obvious limitation of Algorithm 7 is that a Lyapunov function for (1) needs to be known. There are, however, several settings in which a Lyapunov function is known and yet finding a solution of (1) which converges to an equilibrium x^* of f (which in this section is typically $\neq 0$) is a meaningful problem. Examples can be found in [18, Section 6]. Here we focus on the solution of a nonlinear optimization problem (also called a nonlinear program), which are defined as follows.

$$\begin{aligned} &\min_{x\in\mathbb{R}^m} F(x) \\ &\text{subject to} \\ &h_i(x) = 0, \quad i = 1,\dots,m \\ &g_j(x) \le 0, \quad j = 1,\dots,k. \end{aligned} \tag{19}$$

Here $F : \mathbb{R}^m \to \mathbb{R}$, $h_i, g_j : \mathbb{R}^m \to \mathbb{R}$ for $i = 1,\dots,p$ and $j = 1,\dots,q$ are twice continuously differentiable functions.

The Problem (19) is well posed, e.g., if its feasible set

$$\Omega := \{x \in \mathbb{R}^m \,|\, h_i(x) = 0, i = 1,\dots,p,\, g_j(x) \le 0, j = 1,\dots,q\}$$

is nonempty and F is radially unbounded, or if Ω is nonempty and compact.

4.1 Differential equation approach to nonlinear optimization

The idea to solve (19) which fits our setting is now as follows: Design a differential equation

$$\dot{z} = f(z) \tag{20}$$

with state $z = (x, \bar{x}) \in \mathbb{R}^{m+k}$, which exhibits an asymptotically stable equilibrium $z^* = (x^*, \bar{x}^*)$ such that x^* is a minimizer of (19).

In order to explain how this can be accomplished, let us first look at an unconstrained problem, i.e., a problem (19) with $p = q = 0$. Then, a candidate for (20) (with $z = x \in \mathbb{R}^m$) is the (negative) gradient flow

$$f(x) := -\nabla F(x).$$

Using $V(x) := f(x) - f(x^*)$, we obtain

$$\nabla V(x) f(x) = -(\nabla F(x))^2$$

and if f is radially unbounded and $\nabla F(x) \neq 0$ for all $x \neq x^*$ (which means that x^* is the only critical point of F), then V is a Lyapunov function for f and global asymptotic stability of x^* follows from standard Lyapunov function arguments. Moreover, even though x^* and $f(x^*)$ are unknown, the Lyapunov function $V(x) := f(x) - f(x^*)$ can be used in Algorithm 7 since the term $f(x^*)$ vanishes in the derivative $\nabla V(x)$ and cancels out in the Lyapunov difference $V(\Phi(h,x)) - V(x)$. For further information on this approach for unconstrained problems we refer to [9].

For constrained problems, there are different ways to incorporate h_i and g_j into the definition of f in (20). Typically, these approaches include the constraints via suitable penalization terms in (20). In order to illustrate this concept in a simple setting, let us consider the case where no inequality constraints are present (i.e., $q = 0$) and the equality constraints are linear, i.e., of the form $Ax = b$ for a matrix A and a vector b of suitable dimension. For this setting, it was shown in [18, Section 6.1] that — under appropriate conditions on F and A — the system

$$\dot{z} = \begin{pmatrix} -(\nabla^2 F(x)(\nabla F(x) + \bar{x}^T A)^T + A^T(Ax - b)) \\ -A(\nabla F(x) + \bar{x}^T A)^T \end{pmatrix} =: f(z)$$

has a unique asymptotically stable equilibrium $z^* = (x^*, \bar{x}^*)$ where x^* minimizes F. The corresponding Lyapunov function $V(z) = \|\nabla F(x) - \bar{x}^T A\|^2 + \|Ax - b\|^2$ does not require the knowledge of x^* and is thus implementable in Algorithm 7. Similar constructions can be made for more general constraints, see, e.g., [1, 2, 3, 26, 27], however, not all of these approaches provide a Lyapunov function implementable in Algorithm 7 and sometimes the dynamics are only locally defined. Of course, suitable conditions on the data of (19) are needed in order to ensure that the (extended) minimizer is indeed an asymptotically stable equilibrium of (20). For this purpose, linear independence conditions on the derivatives of the constraint functions and sufficient conditions like KKT or Fritz John conditions can be employed, which we will not discuss in detail here (the interested reader is referred, e.g., to [5, 6, 8]). However, the interplay between these conditions for the approaches just cited and Algorithm 7 is still largely unexplored and will be addressed in future research.

Finally, we remark that — unless certain convexity assumptions are satisfied — is in general overly optimistic to assume that the global optimum x^* is the only equilibrium of (20). However, as our example in the next section shows, one may still be able to solve (20) using Algorithm 7 if the initial value lies in the domain of attraction. Again, the precise behavior of Algorithm 7 in this situation is subject to future research.

4.2 Optimization on manifolds: the example of the Rayleigh flow

Optimization problems of the type (19) can be designed on order to solve various problems in systems theory. A comprehensive account of such techniques can be found in Helmke and Moore [16]. For many of the problems discussed in this monograph gradient flows are presented and analyzed. Typically, the optimization problems presented in [16] are posed on suitable manifolds $M \subset \mathbb{R}^n$ and the constraints in (19) then represent the condition $x \in M$.

As an example, let us look at the problem of computing the smallest eigenvalue λ_{min} of a symmetric real matrix $A \in \mathbb{R}^{n\times n}$. The minimal eigenvalue can then be computed as the minimum of the Rayleigh quotient

$$r_A(x) = \frac{x^T Ax}{\|x\|^2}$$

over all $x \in M = \mathbb{S}^{n-1} := \{x \in \mathbb{R}^n \,|\, \|x\| = 1\}$ and the minimizer $x^* \in \mathbb{S}^{n-1}$ is an associated eigenvector. Hence, λ_{min} and an associated eigenvector can be computed by solving (19) with $F(x) = x^T Ax$ and $h_1(x) = \|x\|^2 - 1$.

The gradient flow associated to this minimization problem is the Rayleigh flow

$$\dot{x} = -(A - r_A(x)I_n)x, \tag{21}$$

where I_n is the $n \times n$ identity matrix and the derivative of r_A at a point $x \in \mathbb{S}^{n-1}$ applied to ξ from the tangent space $T_x\mathbb{S}^{n-1}$ is given by

$$\nabla r_A(x)\xi = 2x^T A\xi$$

(details on the derivation of these formulas can be found in [16, Section 1.3]).

Similar gradient flows are provided and analyzed in [16] for various other problems on manifolds M. All these flows have in common that the solution of the gradient flow stays in M, i.e., that the vector field in (20) satisfies $f(x) \in T_xM$ for all $x \in M$. Hence, for the exact solution to (20) the constraints are automatically satisfied.

However, when applying a Runge-Kutta scheme to (20), due to the discretization error $x \in M$ does in general not imply $\Phi(h,x) \in M$. One remedy for this problem is to incorporate the constraints which keep the system on M into the definition of f in (20) and to consider Φ as an "exterior" approximation to the gradient flow on M in the ambient $\mathbb{R}^n$. However, our attempt to do this for the Rayleigh flow so far did not yield satisfactory results, since the solution deteriorates due to additional equilibria appearing outside $M = \mathbb{S}^{n-1}$.

Hence, as an alternative approach we use an "interior" approximation, in which we modify Φ in order to keep the numerical solution on M (for more information on this approach see [13, Chapter IV] and for its relation to differential algebraic equations see [15, Chapter VII]). This approach is possible if we can define (and implement) a projection operator $\mathbb{P}$ which maps each point in a neighborhood $\mathcal{N}$ of M to M. For $M = \mathbb{S}^{n-1}$ such a projection is simply given by $\mathbb{P}x = x/\|x\|$ for all $x \in \mathcal{N} = \mathbb{R}^n \setminus \{0\}$. Then, we may replace $\Phi(h,x)$ by $\mathbb{P}\Phi(h,x)$ and if $\mathbb{P}$ satisfies the inequality $\|\mathbb{P}x - x\| \le C\|y - x\|$ for all $x \in \mathcal{N}$, all $y \in M$ and some constant $C > 0$,

then one easily verifies that for sufficiently small $h > 0$ the projected approximation $\mathbb{P}\Phi(x,h)$ is well defined and consistent with the same order of consistency as $\Phi(x,h)$. Proceding this way leads to very good results for the Rayleigh flow, as the following example shows.

Example 11. We applied Algorithm 7 with $\mathbb{P}\Phi = \Phi/\|\Phi\|$ and Φ obtained by applying the Euler, Heun and RK4 scheme introduced on Section 3.3 to the Rayleigh flow (21). As Lyapunov function we use $V(x) = r_A(x) - \lambda_{min}$, which, as explained in Section 4.1, can be implemented in Algorithm 7 without the knowlegde of λ_{min}. Here, we use the (randomly chosen) symmetric 3×3 matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 4 \\ 3 & 4 & 11 \end{pmatrix}$$

with $\lambda_{min} \approx -0.046732641945883$ and associated eigenvector

$$x^* \approx \begin{pmatrix} 0.954876958271786 \\ -0.242466419355919 \\ -0.171522680851079 \end{pmatrix} \in \mathbb{S}^{n-1}.$$

Since the Rayleigh flow has several equilbria (in fact, each eigenvalue of A is a critical value of r_A), the system is not UGAS. However, it is still UGAS on each compact subset of the domain of attraction of either x^* or $-x^*$ and if we start in such a set then the guaranteed decay of V ensures that we stay in this set. Moreover, since the set of exceptional points (i.e., the set of initial values for which the solution does not converge to either x^* or $-x^*$) is a set of lower dimension, picking a "random" initial value (in our simulation $x_0 = (1,0,0)^T$) the probability of starting in a compact subset of the domain of attraction is very high. Due to this fact, we did not observe any problems in our numerical simulations (which showed comparable results for several other matrices we have tested).

The Figures 8–10 show the phase portrait (projected into the (x_1,x_2)-plane), the values $r_A(x(t)) - \lambda_{min}$ and the corresponding time steps. For each scheme, the simulation was stopped if the condition $|r_A(x(\tau_{i+1})) - r_A(x(\tau_i))| < 10^{-10}$ was satisfied.

The total number of steps in the computation was 13 for the Euler scheme, 32 for the Heun scheme and 26 for RK4. The value $\lambda = 0.4$ in the simulations was chosen because it turned out to yield termination in a smaller number of steps than larger or smaller choices of λ.

It is interesting to note that — as in Example 9 — the Euler scheme turns out to yield the best results since it delivers the approximation of the minimal eigenvalue λ_{min} up to the desired accuracy of 10^{-10} in the smallest number of steps, even though the solution (as clearly visible in Figure 8) is obviously not a good approximation during the transient phase. Hence, the example shows that if the emphasis lies on a numerically cheap computation of the minimum, i.e., the equilibrium, then high order schemes may not necessarily be advantageous.

For the Euler scheme (and to a lesser extent also for the Heun scheme), Figure 10 shows that the step size constantly changes between larger and smaller values.

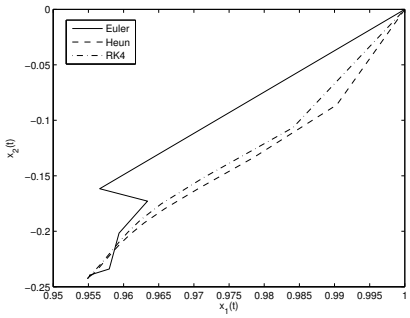

Figure 8: Phase portrait (in (x_1, x_2)-plane) for Ex. 11 with $\lambda = 0.4$ and $x_0 = (1,0,0)^T$

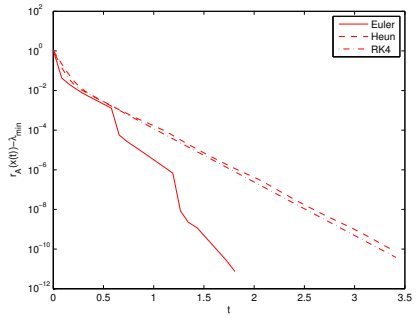

Figure 9: $r_A(x(t)) - \lambda_{min}$ (logarithmic scale) for Ex. 11, $\lambda = 0.4$, $x_0 = (1,0,0)^T$

This behavior is typical for the application of Lyapunov based step size control with explicit schemes to stiff equations (cf. e.g., [18, Example 6.1]) which may lead to the conjecture that the Rayleigh flow for the particular matrix A we have chosen is a moderately stiff system (even though we did not check this rigorously).

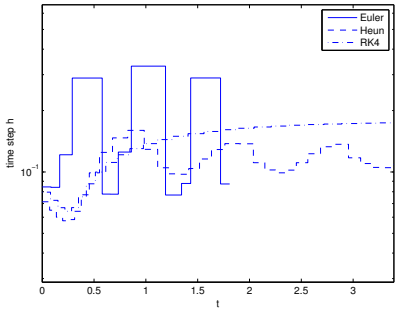

Figure 10: Time steps (logarithmic scale) for Example 11, $\lambda = 0.4$, $x_0 = (1,0,0)^T$

Since the Rayleigh flow is a well studied systems and there are many known techniques for its discretization (again, we refer to [16]), we do not expect Algorithm 7 to outperform more sophisticated methods particularly taylored for the Rayleigh flow. Still, our methods produces very reasonable results and moreover provides valuable insights into the performance of different discretization methods.

References

[1] A. S. Antipin. Minimization of convex functions on convex sets by means of differential equations. *Differ. Equ.*, 30:1365–1375, 1995.

[2] A. A. Brown and M. C. Bartholomew-Biggs. ODE versus SQP methods for constrained optimization. *J. Optim. Theory Appl.*, 62(3):371–386, 1989.

[3] A. Cabot. The steepest descent dynamical system with control. Applications to constrained minimization. *ESAIM Control Optim. Calc. Var.*, 10(2):243–258 (electronic), 2004.

[4] P. Deuflhard and F. Bornemann. *Scientific computing with ordinary differential equations*, volume 42 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2002. Translated from the 1994 German original by Werner C. Rheinboldt.

[5] A. V. Fiacco and G. P. McCormick. *Nonlinear programming*, volume 4 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 1990.

[6] R. Fletcher. *Practical methods of optimization*. Wiley-Interscience, New York, second edition, 2001.

[7] B. M. Garay and K. Lee. Attractors under discretization with variable stepsize. *Discrete Contin. Dyn. Syst.*, 13(3):827–841, 2005.

[8] P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization*. Academic Press Inc., London, 1981.

[9] B. S. Goh. Algorithms for unconstrained optimization problems via control theory. *J. Optim. Theory Appl.*, 92(3):581–604, 1997.

[10] V. Grimm and G. R. W. Quispel. Geometric integration methods that preserve Lyapunov functions. *BIT*, 45(4):709–723, 2005.

[11] L. Grüne. *Asymptotic behavior of dynamical and control systems under perturbation and discretization*, volume 1783 of *Lecture Notes in Mathematics*. Springer, Berlin, 2002.

[12] L. Grüne. Attraction rates, robustness, and discretization of attractors. *SIAM J. Numer. Anal.*, 41(6):2096–2113, 2003.

[13] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations*. Springer, Berlin, second edition, 2006.

[14] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving ordinary differential equations. I Nonstiff Problems*. Springer, Berlin, second edition, 1993.

[15] E. Hairer and G. Wanner. *Solving ordinary differential equations. II Stiff and differential-algebraic problems*. Springer, Berlin, second edition, 1996.

[16] U. Helmke and J. B. Moore. *Optimization and dynamical systems*. Communications and Control Engineering Series. Springer-Verlag London Ltd., London, 1994. With a foreword by R. Brockett.

[17] I. Karafyllis. A system-theoretic framework for a wide class of systems. I. Applications to numerical analysis. *J. Math. Anal. Appl.*, 328(2):876–899, 2007.

[18] I. Karafyllis and L. Grüne. Feedback stabilization methods for the numerical solution of ordinary differential equations. *Discrete Contin. Dyn. Syst. Ser. B*, 16(1):283–317, 2011.

[19] H. K. Khalil. *Nonlinear systems*. Prentice Hall, Upper Saddle River, third edition, 2002.

[20] P. E. Kloeden and J. Lorenz. Stable attracting sets in dynamical systems and in their one-step discretizations. *SIAM J. Numer. Anal.*, 23(5):986–995, 1986.

[21] P. E. Kloeden and B. Schmalfuss. Lyapunov functions and attractors under variable time-step discretization. *Discrete Contin. Dynam. Systems*, 2(2):163–172, 1996.

[22] H. Lamba. Dynamical systems and adaptive timestepping in ODE solvers. *BIT*, 40(2):314–335, 2000.

[23] Y. Lin, E. D. Sontag, and Y. Wang. A smooth converse Lyapunov theorem for robust stability. *SIAM J. Control Optim.*, 34(1):124–160, 1996.

[24] D. Nešić, A. R. Teel, and E. D. Sontag. Formulas relating $\mathscr{KL}$ stability estimates of discrete-time and sampled-data nonlinear systems. *Syst. Control Lett.*, 38(1):49–60, 1999.

[25] A. M. Stuart and A. R. Humphries. *Dynamical systems and numerical analysis*. Cambridge University Press, Cambridge, 1996.

[26] H. Yamashita. A differential equation approach to nonlinear programming. *Math. Programming*, 18(2):155–168, 1980.

[27] L. Zhou, Y. Wu, L. Zhang, and G. Zhang. Convergence analysis of a differential equation approach for solving nonlinear programming problems. *Appl. Math. Comput.*, 184(2):789–797, 2007.