Proceedings of the 10^{th} Brazilian Congress of Thermal Sciences and Engineering – ENCIT 2004
Federal University of Rio de Janeiro, Brazil, Nov. 29 – Dec. 03, 2004
Invited Lecture

Paper CIT04-IL16

ITERATIVE METHODS AS DYNAMICAL SYSTEMS WITH FEEDBACK CONTROL

Amit Bhaya
Dept. of Electrical Engineering (PEE) and Center for High Performance Computing (NACAD), COPPE, Federal University of Rio de Janeiro, P.O. Box 68504, RJ 21945-970, BRAZIL.
amit@nacad.ufrj.br

Eugenius Kaszkurewicz
Dept. of Electrical Engineering (PEE) and Center for High Performance Computing (NACAD), COPPE, Federal University of Rio de Janeiro, P.O. Box 68504, RJ 21945-970, BRAZIL.
eugenius@nacad.ufrj.br

Abstract. *It is shown how standard iterative methods for solving linear and nonlinear equations can be approached from the point of view of control. Appropriate choices of control Liapunov functions lead to both continuous and discrete-time versions of the well known Newton-Raphson and conjugate gradient algorithms as well as their common variants. Insights into these algorithms that result from the control approach are discussed.*

keywords: *Control Liapunov function, iterative methods, convergence, stability, Newton's method, conjugate gradient algorithm.*

1. Introduction

At the risk of oversimplification, it can be said that the design of a successful numerical algorithm usually involves the choice of some parameters in such a way that a suitable measure of some residue or error decreases to a reasonably small value as fast as possible. Although this is the case with most numerical algorithms, they are usually analyzed on a case by case basis: there is no general framework to guide the beginner, or even the expert, in the choice of these parameters. At a more fundamental level, one can even say that the very choice of strategy that results in the introduction of the parameters to be chosen is not usually discussed.

Control theory, once again oversimplifying considerably, is concerned with the problem of *regulation*. In the six decades or so of development of mathematical control theory, several approaches have been developed to the systematic introduction and choice of feedback control parameters in the regulation problem. The object of this paper is to show that one of these approaches—the *control Liapunov function* approach—can be used in a simple and systematic manner to motivate and derive several iterative methods by viewing them as dynamical systems with feedback control. In the context of iterative methods, truncation, roundoff and approximations play the role of disturbances, the effects of which should be minimized by good numerical algorithms.

Related approaches have been put forward in the literature. Continuous algorithms have been investigated in the Russian literature (Gavurin, 1958; Alber, 1971; Tsypkin, 1971) as well as the Western literature (Boggs and Dennis, Jr., 1976; Smale, 1976; Hirsch and Smale, 1979) and the references therein. More recently, Chu, 1988 has developed a systematic approach to the continuous realization of several iterative processes in numerical linear algebra. A control approach to iterative methods is mentioned in Krasnosel'skii et al., 1989, but not developed as in this paper. As far as using Liapunov methods in the analysis of iterative methods is concerned, contributions have been made both in the Russian literature (Evtushenko and Zhadan, 1975; Venets and Rybashov, 1977) as well as the Western literature (Hurt, 1967; Ortega, 1973). Thus the novelty of this paper is that both control and Liapunov approaches are combined to give a unified treatment of some basic linear and nonlinear iterative methods in numerical analysis.

2. Preliminaries

This section gives a whirlwind introduction to some basic concepts and terminology in stability and control theory in order to make this paper self-contained.

Stability of dynamical systems

Consider the system of autonomous or time-invariant differential equations

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

where the dot represents differentiation with respect to the variable t, thought of as time. The initial condition is, without loss of generality, specified at $t = 0$ as $\mathbf{x}(0) = \mathbf{x}_0$. The system (1) is called a *continuous dynamical system* and its solution referred to as a *trajectory*. Trajectories display different qualitative behavior: *transients* decay to zero as t tends to infinity and *steady states* remain after transients have decayed. Constant solutions, i.e., those for which, for all $t > 0$, $\mathbf{x}(t) = \mathbf{x}^*$, satisfy $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}^*) = \mathbf{0}$ and are distinguished from other types of steady states by the name *equilibrium*. The other important types of steady states, periodic and chaotic, will not concern us here.

The equilibrium $\mathbf{x}^*$ is called *stable* if for all $\varepsilon > 0$, there exists $\delta > 0$ such that $\|\mathbf{x}-\mathbf{x}_0\| < \delta$ implies that $\|\mathbf{x}(t)-\mathbf{x}^*\| < \varepsilon$ for all $t > 0$. It is called *locally asymptotically stable* if it is stable and if there exists $\delta > 0$ such that $\|\mathbf{x} - \mathbf{x}_0\| < \delta$ implies that $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*$. If δ can be chosen arbitrarily large, the stability is said to be *global*. In fact, all these stability notions should, more accurately, be qualified with the word *Liapunov*, but, for brevity, we will drop this qualifier whenever possible.

Assuming that $\mathbf{f}$ is differentiable in a neighborhood of $\mathbf{x}^*$ and denoting its derivative with respect to $\mathbf{x}$ as $\mathbf{D_f}(\mathbf{x})$, we can write

$$\mathbf{f}(\mathbf{x}(t)) = \mathbf{f}(\mathbf{x}^* + \mathbf{x}(t) - \mathbf{x}^*) = \mathbf{f}(\mathbf{x}^*) + \mathbf{D_f}(\mathbf{x}^*)(\mathbf{x}(t) - \mathbf{x}^*) + \mathrm{o}(\mathbf{x}(\mathrm{t}) - \mathbf{x}^*), \tag{2}$$

where $\mathrm{o}(\mathbf{x}(\mathrm{t}) - \mathbf{x}^*))$ denotes the fact that $\lim_{\|x\|\to 0}[(\mathbf{x}(t) - \mathbf{x}^*)/\|\mathbf{x}\|] = \mathbf{0}$. Defining $\mathbf{e}(t) := \mathbf{x}(t) - \mathbf{x}^*$, i.e., changing coordinates so that the equilibrium $\mathbf{x}^*$ occurs at the origin:

$$\dot{\mathbf{e}}(t) = \mathbf{D_f}(\mathbf{x}^*)\mathbf{e}(t) + \mathrm{o}(\mathbf{e}(\mathrm{t})). \tag{3}$$

If all the eigenvalues of the Jacobian matrix $\mathbf{D_f}(\mathbf{x}^*)$ have strictly negative real parts, then the equilibrium point ($\mathbf{0}$ for (3) and $\mathbf{x}^*$ for (1)) is locally asymptotically stable, so that if $\|\mathbf{x}_0 - \mathbf{x}^*\| < \delta$, then $\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}^*$. The system

$$\dot{\mathbf{e}}(t) = \mathbf{D_f}(\mathbf{x}^*)\mathbf{e}(t) \tag{4}$$

is referred to as the *linearization* of (1) about the equilibrium point $\mathbf{x}^*$.

Definitions that are analogous can be made for discrete dynamical systems and we will not repeat them here, referring the reader to Hurt, 1967; Ortega, 1973; Boggs, 1976 for details. Instead, we will state the main Liapunov stability theorem that we will need for the discrete-time case. A *nonautonomous* or *time-varying discrete dynamical system* is defined by the recurrence

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k), \tag{5}$$

where, for each k in $\mathbb{N}$, $\mathbf{f}_k : \mathcal{D} \to \mathbb{R}^n$ is continuously differentiable in a neighborhood of the origin, and $\mathcal{D} \subset \mathbb{R}^n$.

Liapunov stability theorem for nonautonomous discrete-time system

Theorem 2.1 *Let $\mathbf{x}^* = \mathbf{0}$ be an equilibrium of (5). Then the* zero solution *of (5), $\mathbf{x}(k) \equiv \mathbf{x}^* = \mathbf{0}, \forall k$, is* globally asymptotically stable *if there exists a scalar-valued function, $V(\mathbf{x}_k, k)$, called a* Liapunov function, *defined on an open set $\mathcal{D}$ and continuous in $\mathbf{x}_k$, such that*

1. *$V(\mathbf{0}, k) = \mathbf{0}$, for all $k \geq k_0$;*
2. *$V(\mathbf{x}_k, k) > 0$ for all $\mathbf{x}_k \neq \mathbf{0}$ in $\mathcal{D}$ and for all $k \geq k_0$.*
3. *$\Delta V(\mathbf{x}_k, k) := V(\mathbf{x}_{k+1}, k+1) - V(\mathbf{x}_k, k) < 0$, $\forall \mathbf{x}_k \in \mathcal{D}\backslash\mathbf{0}$, $\forall k \geq k_0$;*
4. *$0 < W(\|\mathbf{x}_k\|) < V(\mathbf{x}_k)$ for all $k \geq k_0$, where $W(\tau)$ is a positive continuous function defined on $\mathbb{R}$, satisfying $W(0) = 0$ and as $\tau \to \infty$, $W(\tau) \to \infty$ monotonically.*

Remark: The choice of the equilibrium at $\mathbf{0}$ is merely a matter of convenience and the same definitions could be made for an equilibrium at $\mathbf{x} = \mathbf{x}^*$. In this case, we will say, for brevity, that V is a *Liapunov function at* $\mathbf{x}^*$.

Elements of feedback control terminology

The bare minimum of control terminology is introduced below in order to make this paper self-contained. One branch of control theory, called *state space control*, is concerned with the study of dynamical systems of the form (see Figure 1)

$$\begin{aligned} \mathbf{x}^+ &= \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} \\ \mathbf{y} &= \mathbf{H}\mathbf{x} + \mathbf{J}\mathbf{u} \end{aligned} \tag{6}$$

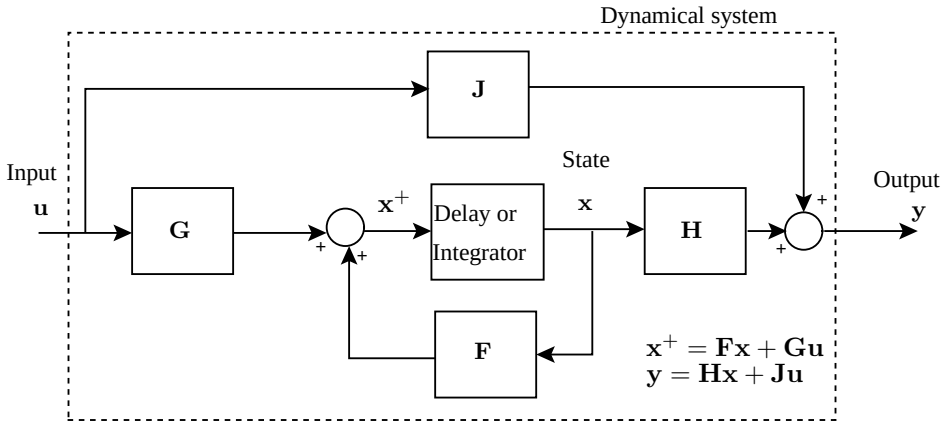

Figure 1: State space representation of a dynamical system, thought of as a transformation between the input $\mathbf{u}$ and the output $\mathbf{y}$. The vector $\mathbf{x}$ is called the state. The quadruple $\{\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{J}\}$ will denote this dynamical system and serves as the building block for the standard feedback control system depicted in Figure 2.

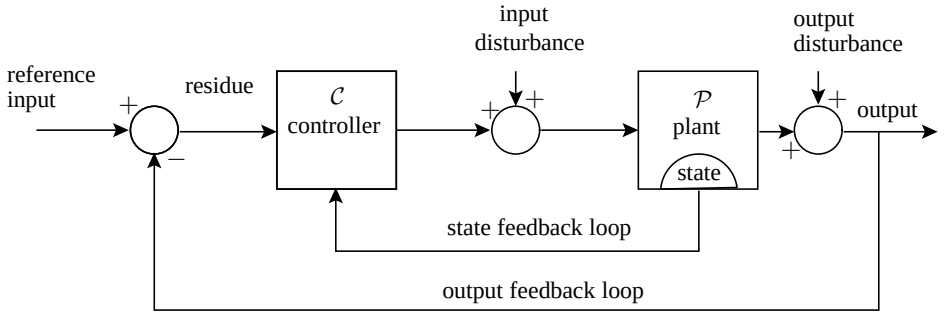

Figure 2: A standard feedback control system, denoted as $S(\mathcal{P}, \mathcal{C})$. The plant, object of the control, will represent the problem to be solved, while the controller is a representation of the algorithm designed to solve it. Note that the plant and controller will, in general, be dynamical systems of the type *(6)*, represented in Figure *1*, i.e., $\mathcal{P} = \{\mathbf{F}_p, \mathbf{G}_p, \mathbf{H}_p, \mathbf{J}_p\}$, $\mathcal{C} = \{\mathbf{F}_c, \mathbf{G}_c, \mathbf{H}_c, \mathbf{J}_c\}$. The semicircle labelled state in the plant box indicates that the plant state vector is available for feedback.

and their feedback interconnections in order to achieve some basic properties, such as stability of the interconnected system, often referred to as a *closed loop system*, specially when in the form of Figure 2. Note that $\mathbf{x}^+$ can represent either $\mathrm{d}\mathbf{x}/\mathrm{d}t$ or $\mathbf{x}(k+1)$. In the former case, the variable t is thought of as time, (6) is said to be a *continuous time system* and the central block in Figure 1 represents an integrator; in the latter case, k is a discrete time variable, the system is said to be a *discrete time system* and the central block represents a delay of one unit.

If the dynamical system is *linear*, the transformations $\{\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{J}\}$ are all linear and representable by matrices: $\mathbf{F}$ is called the *system matrix*, $\mathbf{G}$ the *input matrix*, $\mathbf{H}$ the *output matrix* and $\mathbf{J}$ the *feedforward matrix*. If one or more of $\mathbf{F}$, $\mathbf{G}$, $\mathbf{H}$, $\mathbf{J}$ is nonlinear, the dynamical system is nonlinear, and the nonlinear transformations will be denoted by the corresponding lower case boldface letters. If the transformations vary as a function of time, this is denoted by the appropriate subscript t or k.

If the matrices $\mathbf{F}, \mathbf{G}, \mathbf{H}$ and $\mathbf{J}$ are constant, the system is called *time invariant* (or *autonomous* or *stationary*); otherwise the system is called *time varying* (or *nonautonomous* or *nonstationary*). In the latter case, the matrices are subscripted: with k in the discrete time case, and with t in the continuous time case. Finally, if the matrices $\mathbf{F}, \mathbf{G}$, and $\mathbf{H}$ are zero, the system is said to be *static* or *memoryless*; otherwise, it is called *dynamic*. Thus the quadruple $\{\mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{J}\}$ is used as a convenient shorthand for (6). Finally, the word *decoupled* is used to indicate that a certain matrix is diagonal. For instance, a controller described by the quadruple $\{\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{I}\}$ would be called static and decoupled. The term *multivariable* is sometimes used to denote the fact that some matrix is not diagonal.

A standard feedback control system consists of the interconnection of two systems of the type (6) in the configuration shown in Figure 2. One or both of the feedback loops may be present.

In closing, we mention, extremely briefly, some of the main problems that control theory deals with in the context of the system $S(\mathcal{P}, \mathcal{C})$ of Fig. 2. The problem of *regulation* is that of designing a controller $\mathcal{C}$ such that the output of the system always returns to the value of the reference input, usually considered constant, in the face of nonzero initial conditions, or given some classes of input and output disturbances. The closely related problem of *asymptotic tracking* is that of choosing a controller that makes the output follow or track a class of time-varying inputs asymptotically. The problem of *stabilization* is that of choosing a controller so that a possibly unstable plant (i.e., one for which, in the absence

of any control (= *in open loop*), a bounded input can lead to an unbounded output) leads to a stable configuration $S(\mathcal{P}, \mathcal{C})$. Another type of stabilization problem has to do with the notion of Liapunov stability discussed above. Here one basic problem is to choose the plant input as a linear function of the plant state so that the resulting system with this *state feedback* is stable. Succinctly, given $\mathbf{x}^+ = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$, we set $\mathbf{u} = \mathbf{K}\mathbf{x}$, so that the resulting system is $\mathbf{x}^+ = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{x}$, and the question then is, can $\mathbf{K}$ be chosen so that the eigenvalues of $\mathbf{A} + \mathbf{B}\mathbf{K}$ can be 'placed' within stability regions in the complex plane. The answer is yes, provided that the matrices $\mathbf{A}, \mathbf{B}$ satisfy a certain rank condition, which, surprisingly, is equivalent to the property of being able to choose a control that takes the system (6) from an arbitrary initial state to an arbitrary final state—the latter property is called *controllability*. Some additional details on the concepts mentioned above are given in the sections where they are used below, but the interested reader without a control background is referred to Sontag, 1998 for a mathematically sophisticated introduction or to Kailath, 1980; Delchamps, 1988; Callier and Desoer, 1991; Terrell, 1999 for more elementary approaches.

3. Iterative methods as dynamical systems with feedback control – general nonlinear case

Standard iterative methods for solving nonlinear equations can be approached from the point of view of control. In order to motivate the study of general iterative methods as discrete dynamical systems with control, we start out with a discussion of how one might arrive at a continuous-time dynamical system[1] that finds the zeros of a given nonlinear vector function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$. In other words, the problem is to find a vector $\mathbf{x} \in \mathbb{R}^n$ such that

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}. \tag{7}$$

For a general nonlinear function $\mathbf{f}$, several solutions will, in general, exist. For the moment, we will content ourselves with attempting to find at least one, when it exists. Let $\mathbf{x}, \mathbf{r} \in \mathbb{R}^n$ such that

$$\mathbf{r} = -\mathbf{f}(\mathbf{x}). \tag{8}$$

The variable $\mathbf{r}$ is, in fact, the familiar *residue* of numerical analysis, since its norm can be interpreted as a measure of how far the current guess $\mathbf{x}$ is from a zero of $\mathbf{f}(\cdot)$, i.e., $\mathbf{r} := \mathbf{0} - \mathbf{f}(\mathbf{x})$. The other names that it goes by are *error* or *deviation*. Note that if $\mathbf{f} = \mathbf{A}\mathbf{x} - \mathbf{b}$, then zeroing the residue $\mathbf{r} := \mathbf{b} - \mathbf{A}\mathbf{x}$ corresponds to solving the classical linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$.

In order to introduce control concepts, the first step is to observe that, if the residue $\mathbf{r}$ is thought of as a time-dependent variable that is to be driven to zero, then the variable $\mathbf{x}$ is correspondingly driven to a solution of (7). The second step is to assume that this will be done using a suitably defined control variable $\mathbf{u}$, acting directly on the variable $\mathbf{x}$. In control terms, this is written as the following simple nonlinear dynamical system (Fig. 3).

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{u}, \quad \text{state equation} \tag{9}$$

$$\mathbf{y} = \mathbf{f}(\mathbf{x}), \quad \text{output equation.} \tag{10}$$

Furthermore, from (8) and (10) the output $\mathbf{y}$ is the negative of the residue $\mathbf{r}$:

$$\mathbf{y} = -\mathbf{r}. \tag{11}$$

The problem of finding a zero of $\mathbf{f}(\cdot)$ can now be formulated in control terms as follows. Find a control $\mathbf{u}$ that will drive the output (= −residue) to zero and, consequently, the state variable $\mathbf{x}$ to the desired solution. In other words, this is a *regulation problem*, where the output must be regulated to a reference signal, which in this case is zero: a glance at Figure 3 will make this description clear. From the point of view of control, a natural idea is to feedback the output variable $\mathbf{y}$ in order to drive it to zero. In other words, a *feedback law* of the following type is being chosen:

$$\mathbf{u} = -\mathbf{K}(\mathbf{x})\mathbf{y} = \mathbf{K}(\mathbf{x})\mathbf{r}. \tag{12}$$

Thus the *closed loop system* has the form (see Figure 3)

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{K}(\mathbf{x})\mathbf{r}. \tag{13}$$

The problem of choosing the feedback gain $\mathbf{K}(\mathbf{x})$ is solved using a control Liapunov function, and it is convenient to carry out the design in terms of the residual vector $\mathbf{r}$. In order to do this, it must be assumed that a local change of coordinates is possible from the variable $\mathbf{x}$ to the variable $\mathbf{r}$. Since $\mathbf{r} = -\mathbf{f}(\mathbf{x})$, by the inverse function theorem, if it is assumed that the Jacobian of $\mathbf{f}(\cdot)$ is invertible, then $\mathbf{f}$ itself is locally invertible, i.e., the desired change of coordinates exists. Accordingly, taking the time derivative of (8) leads to the following equation

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = -\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\mathbf{D}_{\mathbf{f}}(\mathbf{x})\dot{\mathbf{x}}, \tag{14}$$

[1] Other terms that have been, or are, in fashion, are analog circuits or analog computers, or, more recently, neural networks.

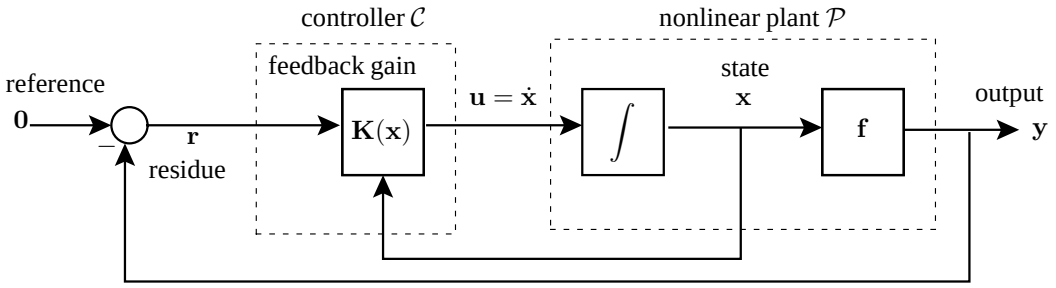

Figure 3: A continuous realization of a general iterative method represented as a feedback control system. The plant, object of the control, represents the problem to be solved, while the controller is a representation of the algorithm designed to solve it. As quadruples, $\mathcal{P} = \{\mathbf{0}, \mathbf{I}, \mathbf{f}, \mathbf{0}\}$, and $\mathcal{C} = \{\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{K}(\mathbf{x})\}$. This standard feedback control system confi guration is denoted as $S(\mathcal{P}, \mathcal{C})$ for arbitrary choices of $\mathcal{P}, \mathcal{C}$.

where $\mathbf{D_f}(\mathbf{x})$ denotes the Jacobian matrix of $\mathbf{f}$ at $\mathbf{x}$ and $\dot{\mathbf{x}}$ denotes the time derivative of $\mathbf{x}$. Substituting (13) in (14) leads to the following equation.

$$\frac{\mathrm{d}\mathbf{r}}{\mathrm{d}t} = -\mathbf{D_f}(\mathbf{x})\mathbf{K}(\mathbf{x})\mathbf{r}. \tag{15}$$

Note that the right hand side of (15) depends on both $\mathbf{x}$ and $\mathbf{r}$. Under the assumption that $\mathbf{f}$ is locally invertible in the neighborhood of the desired solution, (15) can be written in terms of the variable $\mathbf{r}$ alone. This implies that, whatever the choice of the matrix $\mathbf{K}$, convergence results based on (15) are local, unless $\mathbf{f}$ is globally invertible.

Some choices of $\mathbf{K}(\mathbf{x})$ that ensure convergence are clear by inspection; however, a specific control Liapunov function $V(\mathbf{r})$ will be used to justify these choices. Let

$$V(\mathbf{r}) := \|\mathbf{r}\|_2^2 = \mathbf{r}^T\mathbf{r}. \tag{16}$$

Then the time derivative of V along the trajectories of (15), denoted $\dot{V}$, is given by

$$\dot{V} = -\mathbf{r}^T([\mathbf{D_f}(\mathbf{x})\mathbf{K}(\mathbf{x})]^T + \mathbf{D_f}(\mathbf{x})\mathbf{K}(\mathbf{x}))\mathbf{r}. \tag{17}$$

In order for the system to be asymptotically stable, it is necessary to choose $\mathbf{K}(\mathbf{x})$ in such a way that $\dot{V}$ becomes negative definite. Since it has been assumed that the Jacobian is invertible in some neighborhood of the initial condition $\mathbf{x}_0$, two choices suggest themselves immediately. Let

$$\mathbf{K}(\mathbf{x}) = \alpha\mathbf{D}_\mathbf{f}^{-1}(\mathbf{x}), \tag{18}$$

where α is a positive scalar. Then

$$\dot{V} = -2\alpha\mathbf{r}^T\mathbf{r}, \tag{19}$$

which is clearly a negative definite function.

Another choice of $\mathbf{K}(\mathbf{x})$ is as follows. Let

$$\mathbf{K}(\mathbf{x}) = \alpha\mathbf{D}_\mathbf{f}^T(\mathbf{x}). \tag{20}$$

For this choice

$$\dot{V} = -2\alpha\mathbf{r}^T\mathbf{D_f}(\mathbf{x})\mathbf{D}_\mathbf{f}^T(\mathbf{x})\mathbf{r}, \tag{21}$$

which is negative definite by the hypothesis: $\mathbf{D_f}(\mathbf{x})$ invertible implies that the symmetric matrix $\mathbf{D_f}(\mathbf{x})\mathbf{D}_\mathbf{f}^T(\mathbf{x})$ is, in fact, positive definite.

The first choice leads to the closed-loop equation given by

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = -\alpha\mathbf{D}_\mathbf{f}^{-1}(\mathbf{x})\mathbf{f}(\mathbf{x}), \tag{22}$$

which is recognizable as a continuous realization of the familiar *discrete Newton iteration* studied further below. The second choice, (20), is a continuous realization of a version of the *Fridman algorithm* (Fridman, 1961; Maruster, 2001; Ortega and Rheinboldt, 1970; Dennis, Jr. and Schnabel, 1996). For brevity, in what follows, we will also use the term *continuous algorithm*. Note that, for the choice (18) of $\mathbf{K}$, (15) can be solved explicitly to yield $\mathbf{r}(t) = e^{-\alpha t}\mathbf{r}_0$, showing that the continuous Newton algorithm (22) has the pleasant property that the residue goes exponentially to zero, with the rate determined by α.

We may summarize the developments above as follows. The problem of finding a zero of a function is mapped into the problem of making the trajectories of an associated dynamical system converge locally to the desired zero. This dynamical system is also referred to as a continuous algorithm and, more specifically, each choice of a specific matrix, called the stabilizing feedback matrix $\mathbf{K}(\mathbf{x})$, corresponds to a different continuous algorithm. The constant solution (equal to the desired zero for all time) of the system is locally asymptotically stable, i.e., all trajectories that start from initial conditions sufficiently close to the desired zero converge to it asymptotically. In the particular case of the continuous Newton algorithm (22), the convergence is actually exponential.

In this development, the feedback gain matrix $\mathbf{K}(\cdot)$ was chosen as a function of $\mathbf{x}$ alone, but the derivation should convince the reader that it is possible to allow dependence on other variables such as time t, provided that $\dot{V}$ is guaranteed to be negative definite. Examples of this will be given below.

After this short motivation in terms of continuous algorithms, we now turn to discrete or iterative algorithms, maintaining our feedback control point of view. Starting afresh and taking equation (8) as the starting point, a Taylor expansion of $\mathbf{r}$ around $\mathbf{x}$, keeping only the first order term, can be written as follows:

$$\mathbf{r}(\mathbf{x}+\Delta\mathbf{x}) = \mathbf{r}(\mathbf{x}) - \mathbf{D}_\mathbf{f}(\mathbf{x})\Delta\mathbf{x}, \tag{23}$$

where $\mathbf{D}_\mathbf{f}$ is the Jacobian of $\mathbf{f}$ as before. Some notation is needed.

$$\begin{aligned}
\mathbf{x}_k &:= \mathbf{x} \\
\Delta\mathbf{x}_k &:= \Delta\mathbf{x} \\
\mathbf{x}_{k+1} &:= \mathbf{x}+\Delta\mathbf{x} = \mathbf{x}_k + \Delta\mathbf{x}_k =: \mathbf{x}_k + \mathbf{u}_k \\
\mathbf{u}_k &:= \Delta\mathbf{x}_k \\
\mathbf{r}_k &:= \mathbf{r}(\mathbf{x}) = \mathbf{r}(\mathbf{x}_k) = -\mathbf{f}(\mathbf{x}_k) \\
\mathbf{r}_{k+1} &:= \mathbf{r}(\mathbf{x}+\Delta\mathbf{x}) = \mathbf{r}(\mathbf{x}_{k+1}) \\
\mathbf{D}_{\mathbf{f},k} &:= \mathbf{D}_\mathbf{f}(\mathbf{x}_k) \\
\mathbf{K}_k &:= \mathbf{K}(\mathbf{x}) = \mathbf{K}(\mathbf{x}_k) \\
\mathbf{y}_k &:= \mathbf{y}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_k).
\end{aligned} \tag{24}$$

In terms of these variables, the following discrete dynamical system is obtained from (23):

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \mathbf{D}_{\mathbf{f},k}\mathbf{u}_k, \tag{25}$$

where $-\mathbf{r}_k$ is the output and $\mathbf{u}_k := \Delta\mathbf{x}_k$ is the increment that is being controlled, in analogy with the continuous-time case. The control law is a feedback law as in (12), i.e.,

$$\mathbf{u}_k = -\mathbf{K}_k\mathbf{y}_k = \mathbf{K}_k\mathbf{r}_k. \tag{26}$$

Substituting this feedback control law in (25) yields the following analog of (15), although it should be remembered that (25) is actually a first order approximation.

$$\mathbf{r}_{k+1} = (\mathbf{I} - \mathbf{D}_{\mathbf{f},k}\mathbf{K}_k)\mathbf{r}_k. \tag{27}$$

Rewriting (27) as $\mathbf{r}_{k+1} - \mathbf{r}_k = -\mathbf{D}_{\mathbf{f},k}\mathbf{K}_k\mathbf{r}_k$, i.e., $\Delta\mathbf{r}_k = -\mathbf{D}_{\mathbf{f},k}\mathbf{K}_k\mathbf{r}_k$, and observing from (23) that $\Delta\mathbf{r}_k = -\mathbf{D}_{\mathbf{f},k}\Delta\mathbf{x}_k$, it follows that $-\mathbf{D}_{\mathbf{f},k}\Delta\mathbf{x}_k = -\mathbf{D}_{\mathbf{f},k}\mathbf{K}_k\mathbf{r}_k$. Since the Jacobian $\mathbf{D}_{\mathbf{f},k}$ is assumed invertible for all k, this can be written as the discrete-time analog of (13) as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{K}_k\mathbf{r}_k, \tag{28}$$

or, yet again, from (24), as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{K}_k\mathbf{f}(\mathbf{x}_k). \tag{29}$$

Remarks: Notice that, in the discrete-time case, the basic feedback system structure of Fig. 3 has been maintained, and the standard discrete algorithm (28), corresponds to a discretization of (13), using the forward Euler method with stepsize equal to unity. Thus the discretization of (22), using the forward Euler method, results in the standard discrete Newton iterative method. This raises the question of applying different approximation methods to the left hand side of (22) in order to get corresponding discrete iterative methods that belong to the class of Newton methods, but have different convergence properties. Deeper discussion of this point will take us too far afield, so we will refer the reader to Brezinski, 2001 and earlier papers (Boggs, 1971; Boggs and Dennis, Jr., 1976; Incerti et al., 1979) for details. Essentially, Brezinski, 2001 shows that: (i) the Euler method applied to (13) is 'optimal' in the sense that explicit r-stage Runge–Kutta methods of order strictly greater than one cannot have a superlinear order of convergence; and (ii) suitable choice of a variable step

size results in most of the known and popular methods. We will follow this line of reasoning, adopting the unifying view of the step size as a control input.

The problem, as before, is to choose the feedback gain matrix $\mathbf{K}_k$ in such a way as to make (locally) all trajectories of (29) converge to the desired zero, $\mathbf{x}^*$, and, in addition, meet other convergence criteria, such as rate of convergence. Once again, a *control Liapunov function* is used to do this, based on an analysis of (27).

Thus it is now opportune to define the concept of a control Liapunov function (CLF), following Sontag, 1998, in order to formalize what has been done so far as well as to provide a framework for later developments.

Definition 3.1 *Consider the dynamical system*

$$\mathbf{x}_{k+1} = \Phi(\mathbf{x}_k, \mathbf{u}_k), \tag{30}$$

where $\mathbf{x} \in \mathbb{R}^n$, the control input $\mathbf{u}$ is a vector in $\mathbb{R}^{n_i}$ and the function $\Phi : \mathcal{M} \times \mathbb{R}^{n_i} \to \mathbb{R}^n$ is smooth in both arguments with $\mathbf{f}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$. Consider also a C^1 proper function $V : \mathcal{M} - \{\mathbf{0}\} \to \mathbb{R}^+$, with $V(\mathbf{0}) = 0$, which, for all $\mathbf{x}_k \in \mathcal{M} - \{\mathbf{0}\}$, satisfies

$$\Delta V := V(\mathbf{x}_{k+1}) - V(\mathbf{x}_k) = V(\Phi(\mathbf{x}_k, \mathbf{u}(\mathbf{x}_k))) - V(\mathbf{x}_k) < 0, \tag{31}$$

for suitable values of the control input $\mathbf{u}(\mathbf{x}_k) \in \mathbb{R}^{n_i}$. Such a function $V(\cdot)$ is called a control Liapunov function *for the system (30).*

In order to have the stabilizing control given in terms of state feedback, it is also desirable to compute, if possible, a smooth function $\mathbf{G}(\mathbf{x}) : \mathcal{M} - \{\mathbf{0}\} \to \mathbb{R}^{n_i}$ (with $\mathbf{G}(\mathbf{0}) = \mathbf{0}$) such that

$$\mathbf{u}_k = -\mathbf{G}(\mathbf{x}_k) \tag{32}$$

globally asymptotically stabilizes the zero solution of (30), with a specified rate of convergence. In other words, the control Liapunov function is used as a tool to find the appropriate stabilizing state feedback. For more on control Liapunov functions, see Sontag, 1989; Amicucci et al., 1997.

The best way to understand this definition is to see it in action. We will show that we can arrive at the Newton iteration and its variants by the analysis of the first order approximation (27), rather than the analysis of (29). Consider the system (27) and let

$$V := \mathbf{r}_k^T \mathbf{P} \mathbf{r}_k, \tag{33}$$

where $\mathbf{P}$ is a symmetric positive definite matrix, be a candidate for a control Liapunov function. From (33) and (27) it follows that

$$\Delta V := V(\mathbf{r}_{k+1}) - V(\mathbf{r}_k) \tag{34}$$

$$= -2\mathbf{r}_k^T \mathbf{P} \mathbf{D}_{\mathbf{f},k} \mathbf{K}_k \mathbf{r}_k + \mathbf{r}_k^T \mathbf{K}_k^T \mathbf{D}_{\mathbf{f},k}^T \mathbf{P} \mathbf{D}_{\mathbf{f},k} \mathbf{K}_k \mathbf{r}_k. \tag{35}$$

In order that (27) be asymptotically stable, it is necessary to choose the feedback gain matrix $\mathbf{K}_k$ such that ΔV defined in (34) becomes negative definite. Two solutions suggest themselves immediately, once again, under the assumption that the Jacobian $\mathbf{D}_{\mathbf{f},k}$ be invertible. Let

$$\mathbf{K}_k = \alpha \mathbf{D}_{\mathbf{f},k}^{-1}, \tag{36}$$

where α is a scalar to be chosen. Substituting this choice in (35) yields

$$\Delta V = -(2\alpha - \alpha^2) \mathbf{r}_k^T \mathbf{P} \mathbf{r}_k, \tag{37}$$

which shows that ΔV will be negative provided that

$$0 < \alpha < 2. \tag{38}$$

Having motivated the choice of the matrix $\mathbf{K}_k$ by analyzing the first order approximation (27), we now substitute this choice in (29), and the resulting discrete iteration for $\mathbf{x}$ turns out to be

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{D}_{\mathbf{f},k}^{-1} \mathbf{f}(\mathbf{x}_k), \tag{39}$$

which, for $\alpha = 1$, is recognized as the classic *Newton-Raphson iterative method* to find the roots of $\mathbf{f}(\mathbf{x}) = \mathbf{0}$.

Remarks: From the expression (37) for ΔV, it is clear that the choice $\alpha = 1$ is optimal, since it maximizes the decrease in the norm of $\mathbf{r}_k$. Indeed, rewriting (37) as $V(\mathbf{r}_{k+1}) - V(\mathbf{r}_k) = (\alpha^2 - 2\alpha) V(\mathbf{r}_k)$, it appears that, for $\alpha = 1$, $V(\mathbf{r}_{k+1}) = 0$, implying that $\mathbf{r}_{k+1} = \mathbf{0}$. This, of course, is only true for the residue $\mathbf{r}$ in (27), which is the first order approximation (23). The real residue, corresponding to the nonlinear Newton iteration (39) is given by

$$\mathbf{r}_{k+1} = -\mathbf{f}(\mathbf{x}_{k+1}) = -\mathbf{f}(\mathbf{x}_k - \mathbf{D}_{\mathbf{f},k}^{-1} \mathbf{f}(\mathbf{x}_k)),$$

which is zero only to first order because

$$\mathbf{f}(\mathbf{x}_k - \mathbf{D}_{\mathbf{f},k}^{-1}\mathbf{f}(\mathbf{x}_k)) = \mathbf{f}(\mathbf{x}_k) - \mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^{-1}\mathbf{f}(\mathbf{x}_k) + \text{h.o.t} = \mathbf{0} + \text{h.o.t.},$$

where h.o.t denotes higher order terms. Note also that the choice of the matrix $\mathbf{P}$ does not affect the analysis; any positive definite matrix will do and, in particular, $\mathbf{P} = \mathbf{I}$ (i.e., the 2-norm) is a convenient choice.

From the argument leading to the choice (36), it can be seen that if $\mathbf{K}_k$ is chosen as a sufficiently good approximation of the inverse of the Jacobian, i.e., such that ΔV in (35) remains negative, then this ensures local asymptotic stability of the zero solution of the linearization (27) and consequently of the desired equilibrium of (39). To be more specific, for the scalar iteration

$$x_{k+1} = x_k + u_k f(x_k) \tag{40}$$

various well known choices of u_k can be arrived at by analyzing the residual (linearized) iteration (scalar version of (27) with $\mathbf{K}_k = u_k\mathbf{I}$):

$$r_{k+1} = (1 - f'(x_k)u_k)r_k, \quad [\text{where } f' := \mathrm{d}f/\mathrm{d}x] \tag{41}$$

using the control Liapunov function $V(r_k) = r_k^2$. Rather than repeat the analysis here, we give some of the results of this analysis in Table 1. More details on the order of convergence and choices of u_k for higher-order methods that work when f' is not invertible (e.g., when f has a multiple zero), such as the Halley and Chebyshev methods, can be found in Brezinski, 2001, where u_k is regarded as a nonstationary step-size for an Euler method (and accordingly denoted as h_k).

Using the same quadratic Liapunov function (33) and now assuming that the Jacobian $\mathbf{D}_{\mathbf{f},k}$ is limited in norm, in addition to being invertible, another solution is possible, this time with α as a function of k, denoted α_k. Let

$$\mathbf{K}_k := \alpha_k \mathbf{D}_{\mathbf{f},k}^T, \tag{42}$$

Then

$$\Delta V_k := -\mathbf{r}_k^T \left(2\alpha_k \mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T - \alpha_k^2 \mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\right)\mathbf{r}_k \tag{43}$$

Calculating $\frac{\partial \Delta V_k}{\partial \alpha_k}$ and setting it to zero yields:

$$\alpha_k := \frac{\mathbf{r}_k^T \mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{r}_k}{\mathbf{r}_k^T \mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{r}_k} \tag{44}$$

as the choice of α_k which, substituted into (43) yields

$$\Delta V_k = -\frac{(\mathbf{r}_k^T \mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{r}_k)^2}{\mathbf{r}_k^T \mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{r}_k} < 0. \tag{45}$$

It is easy to check that the first three conditions of theorem 2.1 are satisfied for V chosen as in (33), the fourth condition is satisfied by taking

$$W(\|\mathbf{r}_k\|) = [\lambda_{\min}(\mathbf{P}) - \epsilon]\|\mathbf{r}_k\|^2,$$

for small $\varepsilon > 0$.

The resulting algorithm is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{D}_{\mathbf{f},k}^T \mathbf{f}(\mathbf{x}_k), \tag{46}$$

where α_k is defined as in (44) and an initial condition is specified.

We now show how to obtain another method under an additional assumption, namely that the Jacobian is bounded in norm by M, i.e.,

$$\forall k, \ \ \|\mathbf{D}_{\mathbf{f},k}\| < M. \tag{47}$$

Table 1: Showing the choices of control u_k in (40) that lead to the common variants of the Newton method for scalar iterations.

Choice of u_k	**Name of method**
$-1/f'(x_k)$	Newton
$-(x_k - x_{k-1})/[f(x_k) - f(x_{k-1})]$	Secant
$f(x_k)/[f(x_k + f(x_k)) - f(x_k)]$	Steffensen

This technical assumption guarantees that the fourth condition of theorem 2.1 is satisfied, even if a time varying Liapunov function is chosen. Specifically, since the matrix $\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T$ is being assumed invertible, it follows that $\mathbf{P}_k = (\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T)^{-1}$ is a valid choice in order to define the Liapunov function $V_k = \mathbf{r}_k^T\mathbf{P}_k\mathbf{r}_k$. If this is done, the iterative method obtained from (46) is the *Fridman method* (Fridman, 1961; Ortega and Rheinboldt, 1970; Maruster, 2001):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\mathbf{f}(\mathbf{x}_k)^T\mathbf{f}(\mathbf{x}_k)}{\mathbf{f}(\mathbf{x}_k)^T\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{f}(\mathbf{x}_k)}\mathbf{D}_{\mathbf{f},k}^T\mathbf{f}(\mathbf{x}_k). \tag{48}$$

The control formulation can also suggest new algorithms. Suppose, for instance, that the controller $\{\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{K}_k\}$ is replaced by $\{\mathbf{0},\mathbf{0},\mathbf{0},\mathbf{K}_k\mathbf{P}\}$, where $\mathbf{P}$ is the positive definite matrix that defines the Liapunov function. The reason for this choice will become clear right away. Then, another choice of α_k is possible (Silveira, 1980):

$$\alpha_k := \frac{\mathbf{r}_k^T\mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{P}\mathbf{r}_k}{1+\mathbf{r}_k^T\mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{P}\mathbf{r}_k}. \tag{49}$$

Substituting this time-varying feedback gain into (43) yields, after some algebra,

$$\Delta V_k := -\alpha_k\mathbf{r}_k^T\mathbf{P}\mathbf{D}_{\mathbf{f},k}\mathbf{D}_{\mathbf{f},k}^T\mathbf{P}\mathbf{r}_k - \alpha_k^2 < 0. \tag{50}$$

It is now clear that the introduction of $\mathbf{P}$ in the controller results in a negative definite first term in the expression for ΔV_k. Arguments similar to those just made above allow the conclusion that ΔV_k is negative definite. The resulting algorithm is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k\mathbf{D}_{\mathbf{f},k}^T\mathbf{P}\mathbf{r}_k, \tag{51}$$

where α_k is defined as in (49) and an initial condition is specified.

Connection between CLFs of a continuous algorithm and its discrete version

Consider the ODE

$$\dot{\mathbf{x}} = -\mathbf{G}(\mathbf{x}), \qquad \mathbf{x}(\mathbf{0}) = \mathbf{x}_0 \tag{52}$$

as well as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - t_k\mathbf{G}(\mathbf{x}_k), \qquad \mathbf{x}_0 \text{ given}, \tag{53}$$

where $\mathbf{G} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ is continuous and D is an open convex subset of $\mathbb{R}^n$.

Remarks: (i) Euler's method applied to (52) with a variable step size t_k yields (53); (ii) Since all iterative methods can be expressed in the form (53), (52) can be considered as the prototype *continuous analog* of (53), also referred to as a *continuous algorithm*; (iii) it is often easier to work with (52) to obtain qualitative information on its behavior and then to use this to analyze the iterative method (53). Also, as Alber, 1971 pointed out "*theorems concerning the convergence of these (continuous) methods and theorems concerning the existence of solutions of equations and of minimum points of functionals are formulated under weaker assumptions than is the case for the analogous discrete processes.*"

Boggs, 1976 observed that it is sometimes difficult to find an appropriate Liapunov function, but that it is often easier to find a Liapunov function for the continuous counterpart (52) and then use the same function for (53). His result and its simple proof are reproduced below.

Theorem 3.2 (Boggs, 1976) *Let V be a Liapunov function for (52) at $\mathbf{x}^*$. Assume that $\frac{\partial V}{\partial x}$ is Lipschitz continuous with constant K on D. Suppose that there is a constant c independent of $\mathbf{x}$ such that $\frac{\partial V}{\partial x}^T\mathbf{G}(\mathbf{x}) \geq c\|\mathbf{G}(\mathbf{x})\|^2$. Then there are constants $\underline{t}$ and $\overline{t}$ such that V is a Liapunov function for (53) at $\mathbf{x}^*$ for $t_k \in [\underline{t},\overline{t}]$. Furthermore, $\overline{t} < 2c/K$.*

Proof. It only needs to be shown that (31) is satisfied for (52). Observe that

$$\begin{aligned}\Delta V &= V(\mathbf{x}_k - t_k\mathbf{G}(\mathbf{x}_k)) - V(\mathbf{x}_k)\\ &= \{V(\mathbf{x}_k - t_k\mathbf{G}(\mathbf{x}_k)) - V(\mathbf{x}_k) + t_k\tfrac{\partial V}{\partial x}^T(\mathbf{x}_k)\mathbf{G}(\mathbf{x_k})\}\\ &\quad +[V(\mathbf{x}_k) - t_k\tfrac{\partial V}{\partial x}^T(\mathbf{x}_k)\mathbf{G}(\mathbf{x}_k)] - V(\mathbf{x}_k).\end{aligned}$$

By the Lipschitz condition and by Ortega and Rheinboldt, 1970, 15,Thm.3,2.12, the term in braces is bounded by $(1/2)Kt_k^2\|G(\mathbf{x}_k)\|^2$. Therefore,

$$\begin{aligned}\Delta V &\leq -t_k\tfrac{\partial V}{\partial x}^T\mathbf{G}(\mathbf{x}_k) + (1/2)Kt_k^2\|\mathbf{G}(\mathbf{x}_k)\|^2\\ &\leq [-t_kc + (1/2)Kt_k^2]\|\mathbf{G}(\mathbf{x}_k)\|^2,\end{aligned}$$

which is strictly less than zero if $t_kc > (1/2)Kt_k^2$. Choose $\overline{t} < 2c/K$ and $\underline{t}$ such that $0 < \underline{t} < \overline{t} < 2c/K$; and therefore, for $t \in [\underline{t},\overline{t}]$ the result follows. ■

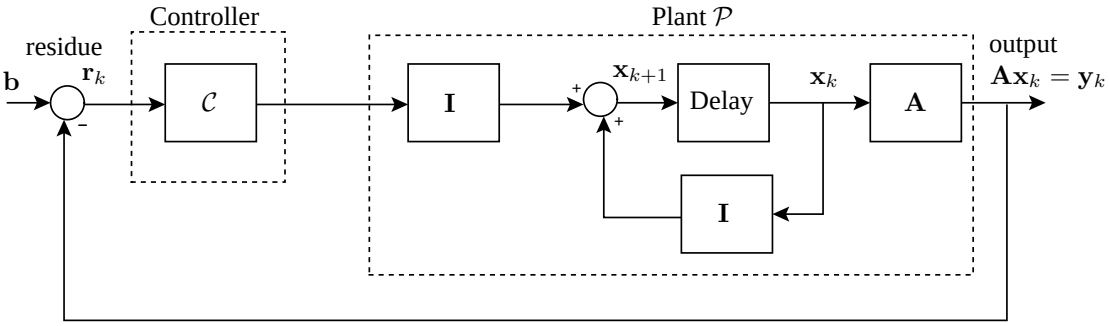

Figure 4: A general linear iterative method to solve the linear system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$ represented in standard feedback control configuration. The plant is always $\mathcal{P} = \{\mathbf{I}, \mathbf{I}, \mathbf{A}, \mathbf{0}\}$, whereas different choices of the controller $\mathcal{C}$ lead to different iterative methods.

Remarks: For the case of steepest descent, $\mathbf{G}(\mathbf{x}) = \nabla \mathbf{f}(\mathbf{x})$ and $\frac{\partial V}{\partial x}^T \mathbf{G}(\mathbf{x}) = \|\mathbf{G}(\mathbf{x})\|^2$, so that $c = 1$, and the steplengths are restricted to the interval $[\underline{t}, 2/K]$.

Clearly, the steplength can be identified with the control input and theorem 3.2 is then seen as a result giving sufficient conditions under which a CLF for the continuous time system (52) works for its discrete counterpart (53). Note that the control or stepsize (t_k) is restricted to lie in a bounded interval—a situation which is quite common in control as well. Boggs, 1976 uses theorem 3.2 to analyze the Ben-Israel iteration for nonlinear least squares problems—thus his analysis may be viewed as another application of the CLF approach.

We make brief mention of another connection between continuous algorithms, numerical methods for ODEs and fixed point iterations. A *fixed point iteration* can be regarded as the discrete dynamical system

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k), \quad k = 0, 1, \ldots, \mathbf{x}_0 \text{ given.} \tag{54}$$

Now consider the related system

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h(\mathbf{f}(\mathbf{x}_k) - \mathbf{x}_k) = (1-h)\mathbf{x}_k + h\mathbf{f}(\mathbf{x}_k), \tag{55}$$

which can be regarded as an overrelaxed version of (54) with relaxation parameter h (usually $h \in (0, 1]$). On the other hand, rewriting (55) as

$$\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h} = \mathbf{f}(\mathbf{x}_k) - \mathbf{x}_k, \tag{56}$$

it can be viewed as the Euler method applied to the ODE

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) - \mathbf{x}(t). \tag{57}$$

This ODE has been studied in Borkar and Soumyanath, 1997 where it is shown that nonexpansivity of $\mathbf{f}$ is sufficient to ensure that all trajectories of (57) converge to the set of fixed points of $\mathbf{f}$, which is assumed closed and nonempty. Other connections are discussed in Brezinski, 2001. Note that, once again, (57) provides an example of a continuous algorithm that requires less stringent conditions for convergence than its discrete counterpart (see Borkar and Soumyanath, 1997 for further discussion of this point).

Finally, it should be mentioned that the Liapunov technique is extremely powerful and can be used, among other things, to determine basins of convergence, as well as to analyze the effects of roundoff errors. This has been done mainly in Hurt, 1967 as well as in Boggs, 1976 to which we refer the reader.

4. Iterative methods for linear systems as feedback control systems

This section specializes the discussion of the previous section, focussing on iterative methods to solve linear systems of equations of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{58}$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$. First, assuming that $\mathbf{A}$ is nonsingular, equation (58) has a unique solution $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b} \in \mathbb{R}^n$, which it is desired to find, without explicitly inverting the matrix $\mathbf{A}$. In applications where the matrix $\mathbf{A}$ is large and sparse (roughly speaking, this means $n > 10^4$ with $O(n)$ nonzero entries although, of course, the notion of 'large' depends on the computer available for the solution of the system), it is well known that iterative methods based on intensive use of matrix-vector products are much more efficient than direct methods such as Gaussian elimination, specially in parallel computing environments.

Following the discussion of the previous section, a general linear iterative method to solve (58) can be described by a recurrence of the form (28), reproduced here for convenience:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{K}\mathbf{r}_k, \quad k = 0, 1, 2, \ldots \tag{59}$$

where $\mathbf{K}$ is a real $n \times n$ matrix and where the residue $\mathbf{r}_k$, in each iteration, with respect to the equation (58), is defined by:

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k. \tag{60}$$

Exactly as in the previous section, it is possible to associate a discrete-time dynamical feedback system to the iterative method (59), and in consequence equation (61) can be viewed as a closed loop dynamical system with a block diagram representation depicted in Figure 4, where $\mathcal{C} = \{\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{K}\}$. The observant reader will note that it is a discrete version of Figure 3 in which the plant is linear.

Defining $\mathbf{y}_k := \mathbf{A}\mathbf{x}_k$ as the output vector of $S(\mathcal{P}, \mathcal{C})$, consider the constant vector $\mathbf{b}$ as the constant input to this system. The vector $\mathbf{r}_k$ represents the error between the input $\mathbf{b}$ and the output $\mathbf{y}_k$ vectors. The numerical problem of solving the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ is thus equivalent to the problem known in control terminology as the *regulation* problem of forcing the output $\mathbf{y}$ to become asymptotically equal to the constant input $\mathbf{b}$, by a suitable choice of controller. When this is achieved, the state vector $\mathbf{x}$ reaches the desired solution of the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Substituting the expression for $\mathbf{r}_k$ into equation (59), the iterative equation is obtained in the so called *output feedback form*, i.e.,

$$\mathbf{x}_{k+1} = (\mathbf{I} - \mathbf{K}\mathbf{A})\mathbf{x}_k + \mathbf{K}\mathbf{b}. \tag{61}$$

Notice that this corresponds to the choice of a static controller $\mathcal{C} = \{\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{K}\}$ and the iterative method (61) corresponds to this particular choice of controller $\mathcal{C}$. We exemplify this here by the classical *Jacobi iterative method*, described by the recurrence equation:

$$\mathbf{x}_{k+1} = \mathbf{H}\mathbf{x}_k + \mathbf{D}^{-1}\mathbf{b}, \tag{62}$$

where $\mathbf{H} = \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F})$ and the matrices $\mathbf{D}$, $\mathbf{E}$, and $\mathbf{F}$ are, respectively, strictly diagonal, lower and upper triangular matrices obtained by splitting matrix $\mathbf{A}$ as $\mathbf{A} = -\mathbf{E} + \mathbf{D} - \mathbf{F}$.

Equating (61) and the classical Jacobi iterative equation (62), the relationship between the corresponding matrices is given by:

$$\mathbf{H} = (\mathbf{I} - \mathbf{K}\mathbf{A}); \quad \mathbf{K} = \mathbf{D}^{-1} \tag{63}$$

Other examples are as follows. If $\mathbf{K} = (\mathbf{D} - \mathbf{E})^{-1}$, then the recurrence (61) represents the *Gauss-Seidel* iterative method; if $\mathbf{K} = \left(\omega^{-1}\mathbf{D} - \mathbf{E}\right)^{-1}$, then it represents the *Successive Overrelaxation (SOR)* method; and finally, if $\mathbf{K} = \omega\mathbf{D}^{-1}$, then it represents the *Extrapolated Jacobi* method. This set of examples should make it clear that all these classical methods correspond to the choice of a static controller $\mathcal{C} = \{\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{K}\}$ – the particular choice of $\mathbf{K}$ distinguishes one method from another. The formulation of iterative methods for linear systems as feedback control systems presented here was initiated in Schaerer and Kaszkurewicz, 2001, where shooting methods for ODEs are also analyzed from this perspective. In order to complete the analysis, observe that, in all the cases considered above, the evolution of the residue $\mathbf{r}_k$ is given by the linear recurrence equation below, derived from (59) by multiplying both sides by $\mathbf{A}$ and subtracting each side from the vector $\mathbf{b}$.

$$\mathbf{r}_{k+1} = (\mathbf{I} - \mathbf{A}\mathbf{K})\mathbf{r}_k. \tag{64}$$

From (64) it is clear that convergence of the linear iterative method is ensured if the matrix $\mathbf{S} := (\mathbf{I} - \mathbf{A}\mathbf{K})$ has all its eigenvalues within the unit disk (i.e., is Schur stable). Observe that (64) can be viewed as the dynamical system $\{\mathbf{I}, \mathbf{A}, \mathbf{0}, \mathbf{0}\}$ subject to state feedback with gain matrix $\mathbf{K}$. Thus there exists a state feedback gain $\mathbf{K}$ that results in arbitrary placement of the eigenvalues of the closed-loop matrix $\mathbf{S} = \mathbf{I} - \mathbf{A}\mathbf{K}$ if and only if the pair $\{\mathbf{I}, \mathbf{A}\}$ of the quadruple $\{\mathbf{I}, \mathbf{A}, \mathbf{0}, \mathbf{0}\}$ is controllable (which it clearly is). Actually, it is possible to state a slightly more general form of this lemma, showing that the less stringent requirement of stabilizability also implies that the matrix $\mathbf{A}$ must be nonsingular.

Lemma 4.1 *There exists a matrix* $\mathbf{K}$ *such that* $\rho(\mathbf{S}) = \rho(\mathbf{I} - \mathbf{A}\mathbf{K}) < 1$ *if and only if the matrix* $\mathbf{A}$ *is nonsingular.*

Proof. ("if"): Choose $\mathbf{K} = \mathbf{A}^{-1}$.

("only if"): Note first that if $\mathbf{A}$ is singular, then, for all matrices $\mathbf{K}$, the product $\mathbf{A}\mathbf{K}$ is also singular, and, moreover, rank $\mathbf{A}\mathbf{K} \leq$ rank $\mathbf{A}$. Thus, it suffices to observe the following (contrapositive) statement: Given a singular matrix $\mathbf{Z} \in \mathbb{R}^{n \times n}$ with rank $\mathbf{Z} = p$, the matrix $\mathbf{I} - \mathbf{Z}$ has $n - p$ eigenvalues equal to 1, hence $\rho(\mathbf{I} - \mathbf{Z}) \geq 1$. This is clearly true because the eigenvalues of $(\mathbf{I} - \mathbf{Z})$ are those of $-\mathbf{Z}$ shifted to the right by 1. Since $\mathbf{Z}$ has $n - p$ eigenvalues equal to zero, this completes the proof. ■

Remarks. Notice that the particular choice $\mathbf{K} = \mathbf{A}^{-1}$ makes all the eigenvalues of matrix $\mathbf{S}$ equal to zero, implying that the iterative scheme (64) will converge in one iteration. This is, of course, only a theoretical remark, since if the inverse of matrix $\mathbf{A}$ were in fact available, it would be enough to compute $\mathbf{A}^{-1}\mathbf{b}$ in order to solve the linear system $\mathbf{Ax} = \mathbf{b}$ and unnecessary to resort to any iteration. In fact, the problem of solving a linear system $\mathbf{Ax} = \mathbf{b}$ without inverting $\mathbf{A}$ can be stated in control terms as that of 'emulating' $\mathbf{A}^{-1}$ without actually computing it, and this is exactly what iterative methods do. Another remark is that the convergence in one iteration, or more generally in a finite number of iterations, is just a question of making the iteration matrix in (64) *nilpotent*, with the index of nilpotency representing an upper bound on the number of iterations required to zero the residue. This is clearly the problem of *dead beat control*, with the restriction that it is not allowed to invert the matrix $\mathbf{A}$.

Lemma 4.1 says that stabilizability of the pair $\{\mathbf{I}, \mathbf{A}\}$ implies that the matrix $\mathbf{A}$ must be nonsingular. Another result of this nature is that controllability of the pair $\{\mathbf{A}, \mathbf{b}\}$ implies that the system $\mathbf{Ax} = \mathbf{b}$ possesses a unique solution (de Souza and Bhattacharyya, 1981). Actually, there are deeper connections here with Krylov subspaces which we will not dwell on here, however see Ipsen and Meyer, 1998.

The next natural question is whether it is possible to do better with other choices of controller. We first consider the case in which matrix $\mathbf{K}$ is no longer a constant and is, in fact, dependent on the state $\mathbf{x}$ or the iteration counter k. As in the previous section, it is possible to derive a multitude of iterative methods of the type

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{K}_k\mathbf{r}_k \tag{65}$$

that produce approximations of the solution $\mathbf{x}^*$, the fixed point of (65), under the assumption that $\mathbf{K}_k \neq \mathbf{0}$.

In particular, in many iterative methods, the matrix $\mathbf{K}_k$ is chosen as $\alpha_k\mathbf{I}$, leading to

$$\mathbf{x}_{k+1} = (\mathbf{I} - \alpha_k\mathbf{A})\mathbf{x}_k + \alpha_k\mathbf{b}, \tag{66}$$

where α_k is a scalar sequence and $\mathbf{I}$ is an identity matrix of appropriate dimension. One method differs from another in the way in which the scalars α_k are chosen; e.g., if the α_ks are precomputed (from arguments involving clustering of eigenvalues of the iteration matrix), we get the class of Chebyshev type 'semi-iterative' methods; if the α_k are computed in terms of the current values of $\mathbf{r}_k$, the resulting class is referred to as adaptive Richardson, etc.). This is analyzed further in the next subsection.

4.1. Control Liapunov functions and the design of minimal residual methods

Considering the matrix $\mathbf{K}_k$ in (65) given by $\mathbf{K}_k = \alpha_k\mathbf{I}$, it is convenient to rewrite (65) in terms of the residue $\mathbf{r}_k$ as follows.

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{r}_k. \tag{67}$$

In control jargon, now thinking of the parameter α_k as a control u_k, equation (67) describes a *bilinear* system. Since the system is no longer linear or time-invariant, straightforward eigenvalue analysis is no longer applicable. A control Liapunov function is used to design an asymptotically stabilizing state feedback control for (67) that drives $\mathbf{r}_k$ to the origin and thus solves the original problem (58).

Consider the control Liapunov function candidate $V(\mathbf{r}_k) := \langle \mathbf{r}_k, \mathbf{r}_k \rangle = \mathbf{r}_k^T\mathbf{r}_k$. Then, from (67),

$$\begin{aligned} \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle &= \langle \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{r}_k, \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{r}_k \rangle & (68) \\ &= \langle \mathbf{r}_k, \mathbf{r}_k \rangle - 2\alpha_k\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle + \alpha_k^2\langle \mathbf{A}\mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle, & (69) \end{aligned}$$

from which it follows that

$$\Delta V := V(\mathbf{r}_{k+1}) - V(\mathbf{r}_k) = -\alpha_k \left(2\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle - \alpha_k\langle \mathbf{A}\mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle\right). \tag{70}$$

From this expression it is clear that the choice

$$\alpha_k = \frac{\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle}{\langle \mathbf{A}\mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle} \tag{71}$$

leads to

$$\Delta V = -\frac{\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle^2}{\langle \mathbf{A}\mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle} < 0, \tag{72}$$

showing that the candidate control Liapunov function works and that (71) is the appropriate choice of feedback control. Furthermore, ΔV is strictly negative unless $\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle = 0$. One way of saying that this possibility is excluded is to say zero does not belong to the *field of values* of $\mathbf{A}$ (Greenbaum, 1997). In other words, the control Liapunov function proves that the residual vector $\mathbf{r}_k$ decreases monotonically to the zero vector.

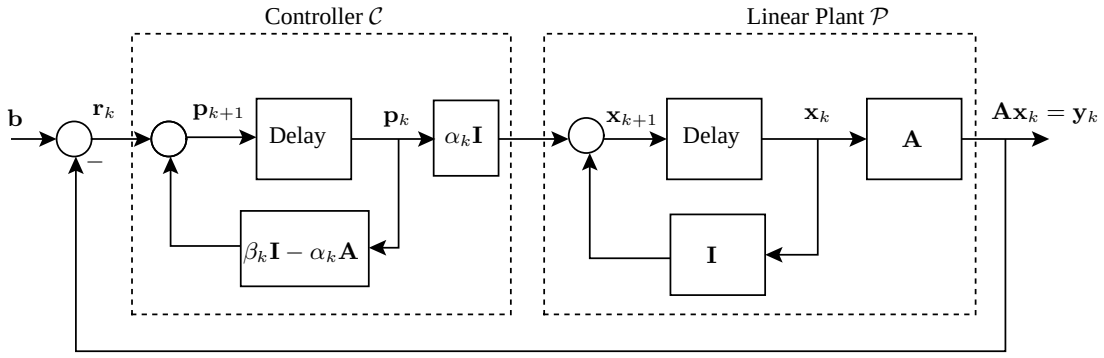

Figure 5: The conjugate gradient method represented as the standard plant $\mathcal{P} = \{\mathbf{I}, \mathbf{I}, \mathbf{A}, \mathbf{0}\}$ with dynamic nonstationary controller $\mathcal{C} = \{(\beta_k\mathbf{I} - \alpha_k\mathbf{A}), \mathbf{I}, \alpha_k\mathbf{I}, \mathbf{0}\}$ in the variables $\mathbf{p}_k$, $\mathbf{x}_k$.

Remarks: Note that the stabilizing feedback control α_k is a nonlinear function of the state, which should not be too surprising, since the system being stabilized is not linear, but bilinear. This choice is a special case of a number of methods and is called Orthomin(1) (Greenbaum, 1997).

A small change in the candidate Liapunov function, together with the assumption that the matrix $\mathbf{A}$ is positive definite, leads to another well known method. Since $\mathbf{A}$ is positive definite, $\mathbf{A}^{-1}$ exists and the following choice is legitimate

$$V(\mathbf{r}_k) := \langle \mathbf{r}_k, \mathbf{A}^{-1}\mathbf{r}_k \rangle. \tag{73}$$

Repeating the steps above, it is easy to arrive at

$$\Delta V = -\alpha_k \left(2\langle \mathbf{A}\mathbf{r}_k, \mathbf{A}^{-1}\mathbf{r}_k \rangle - \alpha_k \langle \mathbf{A}\mathbf{r}_k, \mathbf{A}^{-1}\mathbf{A}\mathbf{r}_k \rangle \right), \tag{74}$$

from which it follows, in exact analogy to the development above, that

$$\alpha_k = \frac{\langle \mathbf{A}\mathbf{r}_k, \mathbf{A}^{-1}\mathbf{r}_k \rangle}{\langle \mathbf{A}\mathbf{r}_k, \mathbf{A}^{-1}\mathbf{A}\mathbf{r}_k \rangle} = \frac{\langle \mathbf{r}_k, \mathbf{r}_k \rangle}{\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle} \tag{75}$$

is the appropriate choice of feedback control that makes $\Delta V < 0$ that, in fact, corresponds to *Richardson's iterative method* for symmetric matrices (Young, 1989; Varga, 2000; Saad and van der Vorst, 2000), sometimes also qualified with the adjectives *adaptive* and *parameter free*, since the α_ks are calculated in feedback form. Another useful way to look at this method is to observe that if the problem of solving the linear system is identified with that of minimizing the quadratic form $\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle - 2\langle \mathbf{b}, \mathbf{x} \rangle$ (which attains its minimum where $\mathbf{A}\mathbf{x} = \mathbf{b}$), then the negative gradient of this function at $\mathbf{x} = \mathbf{x}_k$ is $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$. Thus, this method is often called the *steepest descent method*.

4.2. The conjugate gradient method viewed as proportional-derivative control

In a survey of the top ten algorithms of the century, Krylov subspace methods have a prominent place (Dongarra and Sullivan, 2000; van der Vorst, 2000). This section shows that the formal conjugate gradient method, one of the best known Krylov subspace methods, is also easily arrived at from a control viewpoint. This has the merit of demystifying the conjugate gradient method in addition to providing some insights as to why it has certain desirable properties, such as speed and robustness in the face of roundoff errors.

The conjugate gradient method is conveniently viewed as an acceleration of the steepest descent method, which was presented above as an example of the standard feedback control system $S(\mathcal{P}, \mathcal{C})$ with the controller $\{\mathbf{0}, \mathbf{0}, \mathbf{0}, \alpha_k(\mathbf{r}_k)\}$, which is referred to as a *proportional controller*. The acceleration is achieved by using a discrete version of a classical control strategy for faster 'closed-loop' response (i.e., acceleration of convergence to the solution): this strategy is known as *derivative action* in the controller. The development of this approach is as follows.

Consider that a new method is to be derived from the steepest descent method by adding a new term that is proportional to a discrete derivative of the state vector $\mathbf{x}_k$. In other words, the new increment $\Delta\mathbf{x}_k := \mathbf{x}_{k+1} - \mathbf{x}_k$ is a linear combination of the steepest descent direction $\mathbf{r}_k$ and the previous increment or discrete derivative of the state $\mathbf{x}_k - \mathbf{x}_{k-1}$. Putting in scalar gains α_k and γ_k, this can be expressed mathematically as follows.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k[\mathbf{r}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1})]. \tag{76}$$

This can be rewritten as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k, \tag{77}$$

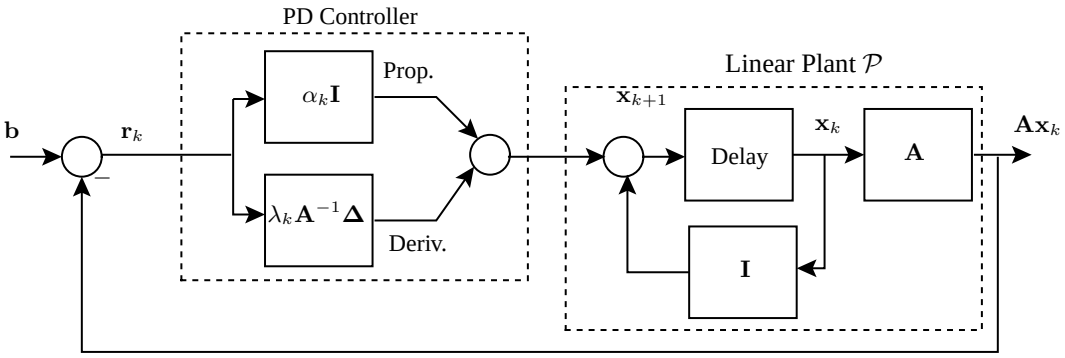

Figure 6: The conjugate gradient method represented as the standard plant $\mathcal{P}$ with a nonstationary proportional-derivative controller in the variables $\mathbf{r}_k$, $\mathbf{x}_k$, where $\lambda_k = \beta_{k-1}\alpha_k/\alpha_{k-1}$. This block diagram emphasizes the conceptual proportional-derivative structure of the controller: of course, the calculations represented by the derivative block, $\lambda_k\mathbf{A}^{-1}\mathbf{\Delta}$, are carried out using formulas (88),(93) that do not involve inversion of the matrix $\mathbf{A}$.

where

$$\begin{aligned}\mathbf{p}_k &= \mathbf{r}_k + \gamma_k(\mathbf{x}_k - \mathbf{x}_{k-1}) = \mathbf{r}_k + \gamma_k\alpha_{k-1}\mathbf{p}_{k-1} \\ &= \mathbf{r}_k + \beta_{k-1}\mathbf{p}_{k-1}\end{aligned} \tag{78}$$

Combining these formulas leads to

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k \tag{79}$$

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k \tag{80}$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k \tag{81}$$

which are the standard CG formulas analyzed below, now utilizing a control Liapunov approach.

The conjugate gradient method for the system (58) (i.e., $\mathbf{A}\mathbf{x} = \mathbf{b}$), with the additional assumption that $\mathbf{A}$ symmetric and positive definite, can be written as follows (Saad, 1996, Algo.6.17,p.179).

The Conjugate Gradient Algorithm
Compute $\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$, $\mathbf{p}_0 := \mathbf{r}_0$.
For $k = 0, 1, \ldots,$ until convergence
Do:
$\alpha_k := \langle\mathbf{r}_k, \mathbf{r}_k\rangle/\langle\mathbf{A}\mathbf{p}_k, \mathbf{p}_k\rangle$
$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k\mathbf{p}_k$
$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k$
$\beta_k := \langle\mathbf{r}_{k+1}, \mathbf{r}_{k+1}\rangle/\langle\mathbf{r}_k, \mathbf{r}_k\rangle$
$\mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k$
EndDo

From the control viewpoint taken here, one approach to understanding this algorithm is to think of the 'parameters' α_k and β_k as scalar control inputs. The motivation for doing this is the observation that the systems to be controlled then belong to the class of *bilinear* systems. More precisely, taking $\mathbf{r}_k$ and $\mathbf{p}_k$ as the state variables, the heart of the CG algorithm above is the following pair of interconnected bilinear systems.

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k \tag{82}$$

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k \tag{83}$$

The control objective is to choose the scalar controls α_k, β_k so as to drive the state vectors $\mathbf{r}_k$ and $\mathbf{p}_k$ to zero. The analysis will be carried out in terms of the variables $\mathbf{r}_k$ and $\mathbf{p}_k$. Provided that α_k is not identically zero, it is easy to see that the equilibrium solution of this system is the zero solution $\mathbf{r}_k = \mathbf{p}_k = \mathbf{0}$ for all k. Thus the objective is to show that the same control Liapunov function approach that has been successfully applied to other iterative methods above can also be used here to motivate the particular choice of α_k and β_k that result in stability of the zero solution. The analysis proceeds in two stages. In the first stage, a choice of α_k guided by a control Liapunov function is shown to result in a decrease of a suitable norm of $\mathbf{r}_k$. In the second stage, a second control Liapunov function orients the choice of β_k that results in a decrease of a suitable norm of $\mathbf{p}_k$. The conclusion is that $\mathbf{r}_k$ and $\mathbf{p}_k$ both converge to zero, as required.

Since $\mathbf{A}$ is a real positive definite matrix, so is $\mathbf{A}^{-1}$ and both matrices define weighted 2-norms. The control Liapunov method is used to choose the controls, using the $\mathbf{A}^{-1}$-norm for (82) and the $\mathbf{A}$-norm for (83). Before proceeding, it should

be pointed out that these choices are arbitrary, and that exactly the same control Liapunov argument with different choices of norms lead to different methods.

Thus the first step is to calculate the $\mathbf{A}^{-1}$-norm of both sides of (82) in order to choose a control α_k that will result in the reduction of this norm of $\mathbf{r}$ to zero.

$$\begin{aligned} \langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\mathbf{r}_{k+1}\rangle &= \langle \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k, \mathbf{A}^{-1}(\mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k)\rangle & (84)\\ &= \langle \mathbf{r}_k, \mathbf{A}^{-1}\mathbf{r}_k\rangle - 2\alpha_k\langle \mathbf{r}_k, \mathbf{p}_k\rangle + \alpha_k^2\langle \mathbf{A}\mathbf{p}_k, \mathbf{p}_k\rangle & (85)\end{aligned}$$

This can be written as

$$\Delta V_{\mathbf{r}} := \langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\mathbf{r}_{k+1}\rangle - \langle \mathbf{r}_k, \mathbf{A}^{-1}\mathbf{r}_k\rangle = -2\alpha_k\langle \mathbf{r}_k, \mathbf{p}_k\rangle + \alpha_k^2\langle \mathbf{A}\mathbf{p}_k, \mathbf{p}_k\rangle. \quad (86)$$

The (optimal) choice of α_k is found from the calculation

$$\frac{\partial \Delta V}{\partial \alpha_k} = -2\langle \mathbf{r}_k, \mathbf{p}_k\rangle + 2\alpha_k\langle \mathbf{A}\mathbf{p}_k, \mathbf{p}_k\rangle \quad (87)$$

so that $\frac{\partial \Delta V}{\partial \alpha_k} = 0$ when

$$\alpha_k = \frac{\langle \mathbf{r}_k, \mathbf{p}_k\rangle}{\langle \mathbf{A}\mathbf{p}_k, \mathbf{p}_k\rangle}. \quad (88)$$

This choice of α_k is optimal in the sense that it makes ΔV as negative as possible. In other words, it makes the reduction in the $\mathbf{A}^{-1}$-norm of $\mathbf{r}$ as large as possible

$$\Delta V = -\frac{\langle \mathbf{r}_k, \mathbf{p}_k\rangle^2}{\langle \mathbf{A}\mathbf{p}_k, \mathbf{p}_k\rangle}. \quad (89)$$

This derivation of α_k also gives a clue as to the robustness of the CG method, since the argument so far has not used any information on the properties of the vectors $\mathbf{p}_k$ (such as $\mathbf{A}$-orthogonality). This indicates that, in a finite precision implementation, even when properties such as $\mathbf{A}$-orthogonality are lost, the choice of α_k in (88) ensures that the $\mathbf{A}^{-1}$-norm of $\mathbf{r}$ will decrease.

Proceeding with the analysis, consider the "$\mathbf{p}_k$-subsystem" subject to the control β_k. The $\mathbf{A}$-norm of both sides of (83) is calculated in order to choose an appropriate control input β_k.

$$\begin{aligned} \langle \mathbf{p}_{k+1}, \mathbf{A}\mathbf{p}_{k+1}\rangle &= \langle \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k, \mathbf{A}(\mathbf{r}_{k+1} + \beta_k \mathbf{p}_k)\rangle & (90)\\ &= \langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{r}_{k+1}\rangle + 2\beta_k\langle \mathbf{p}_k, \mathbf{A}\mathbf{r}_{k+1}\rangle + \beta_k^2\langle \mathbf{p}_k, \mathbf{A}\mathbf{p}_k\rangle. & (91)\end{aligned}$$

Using the same line of argument as above, calculate

$$\frac{\partial \|\mathbf{p}_{k+1}\|_{\mathbf{A}}^2}{\partial \beta_k} = 2\langle \mathbf{p}_k, \mathbf{A}\mathbf{r}_{k+1}\rangle + 2\beta_k\langle \mathbf{p}_k, \mathbf{A}\mathbf{p}_k\rangle, \quad (92)$$

so that

$$\beta_k = -\frac{\langle \mathbf{p}_k, \mathbf{A}\mathbf{r}_{k+1}\rangle}{\langle \mathbf{p}_k, \mathbf{A}\mathbf{p}_k\rangle} \quad (93)$$

is an optimal choice of control, resulting in

$$\|\mathbf{p}_{k+1}\|_{\mathbf{A}}^2 = \|\mathbf{r}_{k+1}\|_{\mathbf{A}}^2 - \frac{\langle \mathbf{p}_k, \mathbf{A}\mathbf{r}_{k+1}\rangle^2}{\langle \mathbf{p}_k, \mathbf{A}\mathbf{p}_k\rangle}. \quad (94)$$

Since the second term is negative (except at the solution $\mathbf{p}_k = \mathbf{0}$), this results in the inequality

$$\|\mathbf{p}_{k+1}\|_{\mathbf{A}} < \|\mathbf{r}_{k+1}\|_{\mathbf{A}}. \quad (95)$$

From (89) and the equivalence of norms, it can be concluded that $\mathbf{r}_{k+1}$ decreases in any induced norm (in particular in the $\mathbf{A}$-norm). Thus (95) implies that $\mathbf{p}_{k+1}$ decreases in $\mathbf{A}$-norm, although not necessarily monotonically, concluding the proof. ■

Remarks: Equations (88) and (93) are equivalent, when the orthogonality relations are valid (Greenbaum, 1997), to the more commonly used but less obvious forms $\alpha_k = \langle \mathbf{r}_k, \mathbf{r}_k\rangle / \langle \mathbf{p}_k, \mathbf{A}\mathbf{p}_k\rangle$ and $\beta_k = \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1}\rangle / \langle \mathbf{r}_k, \mathbf{r}_k\rangle$.

The Orthomin(2) algorithm (Greenbaum, 1997) differs from the standard CG algorithm only in the choice of the controls α_k and β_k. From the viewpoint adopted here, it can be said that the difference lies in the choice of the norms

used for the control Liapunov functions for the $\mathbf{r}$ and $\mathbf{p}$ subsystems. More precisely, consider the algorithm (coupled bilinear systems) below.

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \tag{96}$$
$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} - \beta_k \mathbf{p}_k \tag{97}$$

Suppose that the 2-norm is used as the control Liapunov function for the $\mathbf{r}$ subsystem and the 2-norm of $\mathbf{A}\mathbf{p}$ (recall that for the Orthomin(2) method it is not assumed that the matrix $\mathbf{A}$ is symmetric) is the control Liapunov function for the $\mathbf{p}$ subsystem. A calculation that is strictly analogous to the one above for the CG method shows that this choice of norms results in

$$\alpha_k = \frac{\langle \mathbf{r}_k, \mathbf{A}\mathbf{p}_k \rangle}{\langle \mathbf{A}\mathbf{p}_k, \mathbf{A}\mathbf{p}_k \rangle}, \quad \beta_k = \frac{\langle \mathbf{A}\mathbf{p}_k, \mathbf{A}\mathbf{r}_{k+1} \rangle}{\langle \mathbf{A}\mathbf{p}_k, \mathbf{A}\mathbf{p}_k \rangle}, \tag{98}$$

which is exactly the Orthomin(2) choice of α_k and β_k (see Greenbaum, 1997).

The CLF proof of the CG choices of α_k, β_k allows another observation that, to the authors' knowledge, has not been made in the literature. Consider the following variant of the CG algorithm.

$$\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \tag{99}$$
$$\mathbf{p}_{k+1} = \mathbf{r}_k + \beta_k \mathbf{p}_k \tag{100}$$

In this version of CG, the second equation (in $\mathbf{p}$) has been modified and does not make use of the iterate $\mathbf{r}_{k+1}$ computed (sequentially) "before" it, but instead uses the iterate $\mathbf{r}_k$. In this sense, this version may be thought of as a Jacobi version of the standard 'Gauss-Seidel-like' CG algorithm. The analysis of the standard CG algorithm made above may be repeated almost verbatim, leading to the conclusion that the choices:

$$\alpha_k = \frac{\langle \mathbf{r}_k, \mathbf{p}_k \rangle}{\langle \mathbf{A}\mathbf{p}_k, \mathbf{p}_k \rangle}, \quad \beta_k = \frac{\langle \mathbf{p}_k, \mathbf{A}\mathbf{r}_k \rangle}{\langle \mathbf{p}_k, \mathbf{A}\mathbf{p}_k \rangle} \tag{101}$$

(the only difference is in the numerator of β_k) ensure that $\mathbf{r}_k$ is a decreasing sequence in $\mathbf{A}$-norm and, furthermore that $\|\mathbf{p}_{k+1}\|_{\mathbf{A}}^2 < \|\mathbf{r}_k\|_{\mathbf{A}}^2$, implying that $\mathbf{p}_k$ is also a decreasing sequence, although it decreases slower than it would in the standard CG method (for which the inequality $\|\mathbf{p}_{k+1}\|_{\mathbf{A}}^2 < \|\mathbf{r}_{k+1}\|_{\mathbf{A}}^2$ was obtained). This confirms the conventional wisdom that Gauss-Seidelization is conducive to faster convergence.

A block diagram representation is helpful in order to interpret what has just been done, both in terms of the taxonomy of iterative methods proposed as well as in terms of making the controller structure explicit. Comparing the block diagrams of Figures 5 and 4, it becomes clear that, although the box representing the plant (i.e., problem or equation to be solved) has remained the same, the box representing the controller (i.e., solution method) is considerably more sophisticated with respect to the simple controllers studied in section 4. It is, in fact, a dynamic time-varying or nonstationary controller. The upshot of the increased sophistication is that the method (conjugate gradient) is more efficient. In fact, it is well known that, in infinite precision, CG is actually a direct method (converges in n steps for an $n \times n$ matrix $\mathbf{A}$) (Kelley, 1995). In control terms, this last observation can be rephrased by saying that the "CG controller" achieves so called *dead beat control* in n steps.

The continuous version of the CG algorithm and its connection to the well known backpropagation with momentum method (much used in neural network training) is discussed in Bhaya and Kaszkurewicz, 2004.

Finally, the interested reader is invited to compare the control approach developed above with other didactic approaches to the conjugate gradient algorithm, such as Schönauer and Weiss, 1995; Shewchuk, 1994, or an analysis from a z-transform signal processing perspective (Chang and Willson, 2000). In our view, the control approach is natural and this is borne out by its simplicity.

5. Concluding remarks

The block diagram representation has the virtue of allowing us to make a clear separation between the problem and the algorithm, making it easy to classify as well as generalize the strategies used in the algorithm. Taking the example of linear iterative methods, we see a progression of successively more complex controllers – constant (α), nonstationary or time-varying ($\alpha_k \mathbf{I}$), multivariable ($\mathbf{K}$), multivariable time-varying ($\alpha_k \mathbf{K}_k$) and dynamic, leading to most of the standard iterative methods in a natural manner. For linear iterations, the results of this paper lead to a 'dictionary' relating controller choice to numerical algorithm that we present in Table 2, which makes reference to Figure 3, and uses the terminology of Kelley, 1995; Saad and van der Vorst, 2000.

The standardized control Liapunov function (CLF) analysis technique leads to the conventional choices of control parameters. It is worthy of note that 2-norms, possibly weighted with a diagonal or positive definite matrix, usually work as CLFs. This is in sharp contrast with the situation for an arbitrary nonlinear dynamical system, for which, as a rule, considerable ingenuity is required to find a suitable CLF. Another consequence of the relative ease in finding quadratic

Table 2: Taxonomy of linear iterative methods from a control perspective, with reference to Figure 3. Note that $\mathcal{P} = \{\mathbf{I}, \mathbf{I}, \mathbf{A}, \mathbf{0}\}$ in all cases.

Controller $\mathcal{C}$	**Controller Type**	**Class of method**	**Specific methods**
$\{\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{I}\}$	Static, stationary	Richardson	
$\{\mathbf{0}, \mathbf{0}, \mathbf{0}, \alpha_k\mathbf{I}\}$	static, nonstationary	Adaptive Richardson	Chebyshev
$\{\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{K}\}$	static, stationary	Preconditioned Richardson	Jacobi, Gauss-Seidel, SOR, extrap. Jacobi
$\{\mathbf{0}, \mathbf{0}, \mathbf{0}, \alpha_k\mathbf{K}_k\}$	static, nonstationary	Adaptive Preconditioned Richardson	
$\{\mathbf{0}, \mathbf{0}, \mathbf{0}, \alpha_k(\mathbf{r}_k)\mathbf{I}\}$	static, nonstationary	Steepest descent	Orthomin(1)
$\{\beta_k\mathbf{I} - \alpha_k\mathbf{A}, \mathbf{I}, \mathbf{I}, \mathbf{0}\}$, in variables $\mathbf{p}_k, \mathbf{x}_k$	Dynamic, nonstationary	Conjugate Gradient	CG, Orthomin(2), Orthodir
α_k, β_k, in variables $\mathbf{r}_k, \mathbf{x}_k$	Proportional-derivative, nonstationary	Conjugate Gradient	CG, Orthomin, Orthodir
$\{\alpha_k\mathbf{I}, \mathbf{I}, \beta_k\mathbf{I}, \mathbf{0}\}$	Dynamic, nonstationary	Second order Richardson	Frankel

CLFs is that each of these leads to a different algorithm, so that there is scope for devising new algorithms, showing that the CLF approach has an inherent richness. The control Liapunov approach is easily generalizable to a Hilbert space setting, following the work on iterative methods for operators by Kantorovich and Akilov, 1982, Krasnosel'skii et al., 1989 and others.

This paper has concentrated on showing that variable parameters such as step size or 'gains' (e.g., α_k, β_k in the CG method) are fruitfully interpreted as control inputs to be chosen by a CLF analysis. Of course, other control techniques for analysis and design of controllers are also available, opening up the possibility for further work on subjects not touched on in this paper, such as the second order Richardson methods (last entry in Table 2) or Chebyshev iterative methods, which use a precomputed sequence of α_ks (second entry in Table 2).

There is much scope for other applications of control theory in the design and analysis of numerical algorithms. A classic example of this is due to Gustafsson et al., 1988, where a control model leads to an improved integration routine for ODEs. A feedback control based analysis of Gauss-Newton recursive methods that is similar to the one made in this paper can be found in Rupp and Sayed, 1996. Many other applications could be cited but are omitted here for lack of space.

Some disclaimers should also be made here. Although, the control approach provides guidelines for algorithm design, it does not free the designer of the need for a careful analysis of issues such as roundoff error (robustness), computational complexity, order of convergence, etc. It should also be noted that many standard solutions of control problems are infeasible in numerical analysis because they would involve more computation for their implementation than standard numerical methods for the solution of the original problem. Here the challenge is for control theorists to develop limited complexity controllers and, to some extent, driven by technological needs such as miniaturization and low energy consumption, this is now being researched in control theory. Robust control theory has been well developed in the last few decades and a natural follow up to the ideas in this paper would be to apply this theory to the analysis of robustness of numerical algorithms to perturbations such as roundoff, truncation, etc.

In conclusion, a quote from Krasnosel'skii et al., 1989 is appropriate: *Each class of iteration procedure has its advantages, its drawbacks and its specific applications. The problem of choosing the optimal method for the approximate solution of a concrete instance of an equation of the type (58) is not only far from being solved, but is even far from being clearly posed.*

Our hope is that this paper has made a contribution in the direction suggested by Krasnosel'skii in the above quote.

6. Acknowledgements

This research was partially financed by Project Nos. 140811/2002-8, 551863/2002-1, 471262/03-0 of CNPq, and also by the agencies CAPES & FAPERJ.

The authors would like to thank Profs. Daniel B. Szyld and José Mario Martínez for useful remarks on a draft version of this paper and the latter also for bringing the work of Boggs to our attention.

7. References

Alber, Y. I., 1971, Continuous processes of the Newton type, "Differential Equations", Vol. 7, No. 11, pp. 1461–1471.

Amicucci, G. L., Monaco, S., and Normand-Cyrot, D., 1997, Control Lyapunov stabilization of affine discrete-time systems, "Proc. of the 36th IEEE Conference on Decision and Control", pp. 923–924.

Bhaya, A. and Kaszkurewicz, E., 2004, Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method, "Neural Networks", Vol. 17, No. 1, pp. 65–71.

Boggs, P. T., 1971, The solution of nonlinear systems of equations by A-stable integration techniques, "SIAM J. Numer. Anal.", Vol. 8, No. 4, pp. 767–785.

Boggs, P. T., 1976, The convergence of the Ben-Israel iteration for nonlinear least squares problems, "Mathematics of Computation", Vol. 30, No. 135, pp. 512–522.

Boggs, P. T. and Dennis, Jr., J. E., 1976, A stability analysis for perturbed nonlinear analysis methods, "Mathematics of Computation", Vol. 30, No. 134, pp. 199–215.

Borkar, V. S. and Soumyanath, K., 1997, An analog scheme for fixed point computation—Part I: Theory, "IEEE Trans. Circuits and Systems–I: Fundamental Theory and Applications", Vol. 44, No. 4, pp. 351–355.

Brezinski, C., 2001, Dynamical systems and sequence transformations, "J. Physics A: Mathematical and General", Vol. 34, No. 48, pp. 10659–10669, Special Issue dedicated to "Symmetries and Integrability of Difference Equations (SIDE IV)".

Callier, F. M. and Desoer, C. A., 1991, "Linear System Theory", Springer Verlag, New York.

Chang, P. S. and Willson, A. N., 2000, Analysis of Conjugate Gradient Algorithms for Adaptive Filtering, "IEEE Trans. Signal Processing", Vol. 48, No. 2, pp. 409.

Chu, M. T., 1988, On the continuous realization of iterative processes, "SIAM Review", Vol. 30, No. 3, pp. 375–387.

de Souza, E. and Bhattacharyya, S. P., 1981, Controllability and the linear equation $Ax = b$, "Lin. Algebra Appl.", Vol. 36, pp. 97–101.

Delchamps, D. F., 1988, "State Space and Input-Output Linear Systems", Springer-Verlag, New York.

Dennis, Jr., J. E. and Schnabel, R. B., 1996, "Numerical methods for unconstrained optimization and nonlinear equations", Vol. 16 of "Classics in Applied Mathematics", SIAM, Philadelphia, Corrected republication of work published by Prentice-Hall, 1983.

Dongarra, J. and Sullivan, F., 2000, Guest Editors' Introduction to the top 10 algorithms, "Computing in Science and Engineering", Vol. 2, No. 1, pp. 22–23.

Evtushenko, Y. G. and Zhadan, V. G., 1975, Application of the method of Lyapunov functions to the study of the convergence of numerical methods, "USSR Computational Mathematics and Mathematical Physics", Vol. 15, No. 1, pp. 96–108, Zh. Vychisl. Mat. Mat. Fiz., pp. 101-112 (Russian edition).

Fridman, V., 1961, An iteration process with minimum errors for a nonlinear operator equation, "Dokl. Akad. Nauk. SSSR", Vol. 139, pp. 1063–1066.

Gavurin, M. K., 1958, Nonlinear functional equations and continuous analogs of iterative methods, "Izv. Vyssh. Uchebn. Zaved. (Matematika)", Vol. 5, No. 6, pp. 18–31, (in Russian).

Greenbaum, A., 1997, "Iterative Methods for Solving Linear Systems", SIAM, Philadelphia.

Gustafsson, K., Lundh, M., and Söderlind, G., 1988, A PI stepsize control for the numerical solution of ordinary differential equations, "BIT", Vol. 28, pp. 270–287.

Hirsch, M. W. and Smale, S., 1979, On algorithms for solving $f(x) = 0$, "Communications on Pure and Applied Mathematics", Vol. XXXII, pp. 281–312.

Hurt, J., 1967, Some stability theorems for ordinary difference equations, "SIAM J. Numer. Anal.", Vol. 4, No. 4, pp. 582–596.

Incerti, S., Parisi, V., and Zirilli, F., 1979, A new method for solving nonlinear simultaneous equations, "SIAM J. Numer. Anal.", Vol. 16, No. 5, pp. 779–789.

Ipsen, I. C. F. and Meyer, C. D., 1998, The idea behind Krylov methods, "American Mathematical Monthly", Vol. 105, No. 10, pp. 889–899.

Kailath, T., 1980, "Linear Systems", Prentice Hall, Englewood Cliffs, N.J.

Kantorovich, L. V. and Akilov, G., 1982, "Functional Analysis", Pergamon Press, Oxford.

Kelley, C. T., 1995, "Iterative Methods for Linear and Nonlinear Equations", Vol. 16 of "Frontiers in Applied Mathematics", SIAM, Philadelphia.

Krasnosel'skii, M. A., Lifshits, J. A., and Sobolev, A. V., 1989, "Positive Linear Systems: The method of positive operators", Heldermann Verlag, Berlin.

Maruster, S., 2001, The stability of gradient-like methods, "Applied Mathematics and Computation", Vol. 117, pp. 103–115.

Ortega, J. M., 1973, Stability of difference equations and convergence of iterative processes, "SIAM J. Numer. Anal.", Vol. 10, No. 2, pp. 268–282.

Ortega, J. M. and Rheinboldt, W. C., 1970, "Iterative solutions of nonlinear equations in several variables", Academic Press, N.Y.

Rupp, M. and Sayed, A. H., 1996, Robustness of Gauss-Newton recursive methods: a deterministic feedback analysis, "Signal Processing", Vol. 50, No. 3, pp. 165–187.

Saad, Y., 1996, "Iterative methods for sparse linear systems", The PWS series in Computer Science, PWS Publishing Co., Boston.

Saad, Y. and van der Vorst, H. A., 2000, Iterative solution of linear systems in the 20th century, "J. Computational and Applied Mathematics", Vol. 123, No. 1-2, pp. 1–33.

Schaerer, C. and Kaszkurewicz, E., 2001, The shooting method for the solution of ordinary differential equations: a Control-Theoretical Perspective, "Internat. J. Systems Science", Vol. 32, No. 8, pp. 1047–1053.

Schönauer, W. and Weiss, R., 1995, An engineering approach to generalized conjugate gradient methods and beyond, "Applied Numerical Mathematics", Vol. 19, No. 3, pp. 175–206.

Shewchuk, J. R., 1994, An introduction to the conjugate gradient method without the agonizing pain, Technical Report CMU-CS-94-125, Carnegie Mellon University, Pittsburgh, PA, Available at http://www.cs.cmu.edu/˜quake/papers.html.

Silveira, H. M., 1980, Stability in the determination of algorithms to find roots of nonlinear systems of equations, "Proc. of the 3rd Brazilian Conference on Automatic Control", pp. 87–91, Rio de Janeiro. In Portuguese. Original title: *Estabilidade em determinação de algoritmos para pesquisa de raízes de sistemas de equações não-lineares*.

Smale, S., 1976, A convergent process of price adjustment and global Newton methods, "J. Math. Economics", Vol. 3, pp. 107–120.

Sontag, E. D., 1989, A universal construction of Artstein's theorem on nonlinear stabilization, "Systems and Control Letters", Vol. 13, pp. 117–123.

Sontag, E. D., 1998, "Mathematical Control Theory: Deterministic Finite Dimensional Systems", Vol. 6 of "Texts in Applied Mathematics", Springer-Verlag, New York, second edition.

Terrell, W. J., 1999, Some Fundamental Control Theory I: Controllability, Observability, and Duality, "American Mathematical Monthly", Vol. 106, No. 8, pp. 705–719.

Tsypkin, Y. Z., 1971, "Adaptation and Learning in Automatic Systems", Vol. 73 of "Mathematics in Science and Engineering", Academic Press, New York, First published in Russian under the title *Adaptatsia i obuchenie v avtomaticheskikh sistemakh*, Nauka, Moscow, 1968.

van der Vorst, H. A., 2000, Krylov subspace iteration, "Computing in Science and Engineering", Vol. 2, No. 1, pp. 32–37.

Varga, R. S., 2000, “Matrix Iterative Analysis”, Vol. 27 of “Springer Series in Computational Mathematics”, Springer, Berlin, second edition.

Venets, V. I. and Rybashov, M. V., 1977, The method of Lyapunov functions in the study of continuous algorithms of mathematical programming, “USSR Computational Mathematics and Mathematical Physics”, Vol. 17, No. 3, pp. 64–73, Zh. Vychisl. Mat. Mat. Fiz., pp. 622-633 (Russian edition).

Young, D. M., 1989, A historical overview of iterative methods, “Computer Physics Communications”, Vol. 53, No. 1-3, pp. 1–17.