Control-theoretic forward error analysis of iterative numerical algorithms

Ammar Hasan, Eric C. Kerrigan, and George A. Constantinides

***Abstract*—It has been known for at least five decades that control theory can be used to study iterative algorithms. However, little work can be found in the control systems literature on numerical algorithms, especially on the study of finite precision effects. In this paper, we consider numerical iterative algorithms in finite precision as dynamical systems and study the effects of finite precision using control theory. By using the control tools of input-to-state stability and results from the study of quantization in control systems, we present new systematic ways to find bounds on the forward error for iterative algorithms. The advantages of the proposed schemes are shown by applying them to find bounds for the classical iterative methods for solving a system of linear equations.**

***Index Terms*—Numerical algorithms as dynamical systems, finite precision, forward error analysis, input-to-state stability, quantization.**

I. Introduction

MANY iterative numerical algorithms can be considered as dynamical systems, and therefore can be studied using control systems theory. Although having known this fact for many years [1], [2] and the possible potential of this approach to provide us with new tools for analysis of algorithms, this idea has not been very popular in the control systems literature.

In recent years, interest has been increasing in this application of control theory [3]–[10]. Most of this work is focused on design and analysis of iterative algorithms in exact precision. In this paper we will use control-theoretic ideas to study the effects of *finite* precision in algorithms.

Algorithms are implemented on digital hardware, which have a finite precision (inability to store all real numbers). Due to finite precision, computations in algorithms can have errors and the behavior of an algorithm can be quite different from that in exact precision. Therefore, the study of finite precision effects in an algorithm is vital. We consider algorithms as dynamical systems and consider the errors due to finite precision as disturbances in the dynamical system. Since control theory provides many tools to study the effects of disturbances, we use control theoretic ideas to study the effects of finite precision in numerical algorithms.

The solution obtained by an algorithm in finite precision can be different from the exact solution of the problem. The error in the solution obtained in finite precision is called the forward error. The process of finding a bound on the forward error is called forward error analysis. Forward error analysis is one of the methods to establish the obtainable accuracy of solutions for an algorithm in finite precision. In this paper we present two schemes for forward error analysis based on control-theoretic ideas. The first scheme is based on the control concept of input-to-state stability (ISS). It is shown that if the algorithms can be represented by a dynamical system that is input-to-state stable, then we can obtain a bound on the forward error. The second scheme is based on results from the study of quantization effects in control systems. We apply the proposed schemes to the the classical iterative methods for solving a system of linear equations. The obtained bounds are compared with bounds given in the numerical analysis literature and are numerically shown to be tighter.

In [5], [7] authors have also studied the effects of finite precision errors using control theory. In [5] it has been shown that if the finite precision errors in iterations of Newton's method are bounded, then the forward error will also be bounded. However, they have not given a bound for either the finite precision errors or the forward error. In [7] the authors have looked at the robustness of algorithms in finite precision. They have proposed the use of control Lyapunov functions to find a bound on the maximum finite precision error that would guarantee the algorithm remains stable. In contrast, in this paper we are concerned with bounds on the difference between the exact solution of a problem and the solution obtained by an algorithm in finite precision.

The use of ISS in the study of numerical methods has also been explored before, but in a different context. In [11] the author has discussed how ISS can be used to show that a numerical approximation of continuous-time dynamical system has the same qualitative behavior as the original system. However, we are concerned with the forward error bound of numerical algorithms that can be represented by discrete-time dynamical systems.

A practical application of forward error analysis is in the design of custom hardware. One of the important research areas in numerical analysis and computing is to look for algorithms and computational hardware that give low computational time for solving numerical problems. The need for fast algorithms is driven by many applications especially ones with real-time constraints. In the past few years the

A. Hasan is with the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, U.K. and also with the National University of Sciences and Technology, Islamabad, Pakistan. ammar.hasan07@imperial.ac.uk

E. Kerrigan is with the Department of Aeronautics and the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, U.K. e.kerrigan@imperial.ac.uk

G. Constantinides is with the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, U.K. g.constantinides@imperial.ac.uk

This work was funded by the EPSRC under grant numbers EP/G031576/1, EP/I020357/1, EP/I010236/1 and the EC FP7 grant EMBOCON.

trend in computer hardware design to speedup algorithms is to develop computer architectures that are customized for a specific application [12]. Some examples include graphics processing units (GPUs), application-specific integrated circuit (ASIC), field-programmable gate arrays (FPGAs), etc. It has been shown that the above listed and other application specific architectures can give significant speedup over general purpose processors for several applications [12]–[14]. One of the parameters in hardware design is the number representation. In general purpose processors this is usually fixed to IEEE double precision, which is a 64 bit floating point number representation. However, in a custom design we can choose the number representation arbitrarily. Floating point number representation consist of three parts: sign bit, exponent and mantissa. The hardware resources required to implement a circuit grow asymptotically quadratically with the number of bits used to represent the mantissa or the mantissa width [13]. A circuit for lower mantissa width utilizes less hardware resources. The saved hardware resources can be used to implement parallel computational blocks thus increasing the speed of overall computations. On the other hand, the unit roundoff is inversely related to the mantissa width and decreasing the mantissa width increases the roundoff errors in arithmetic computations. This suggests that there is a trade-off in the decision of number representation. A discussion on this trade-off can be found in [15]. The best choice for the mantissa width would be the lowest one to guarantee that the error in the solution of numerical problems of interest will be no greater than some desired value. The forward error bound, if not too conservative, would be appropriate for this task since it provides a relation between the forward error and the unit roundoff. We believe that our effort to explore new ways of forward error analysis, which may result in a tighter bound, is a step towards the application of forward error bounds in custom hardware design. Another point to note is that in this application we decide the number representation during chip design, i.e. before actually solving the actual numerical problems. Therefore, for this application the computational complexity of forward error analysis is not as important as the tightness of the bound.

Notation: We denote matrices with capital letters; vectors, scalars and functions with small letters and sequences with small and bold letters. We denote the vector Euclidean norm and the induced matrix 2-norm with $\|\cdot\|_2$. We denote the vector infinity norm, sequence l_∞-norm and the induced matrix infinity norm with $\|\cdot\|_\infty$. We use $|\cdot|$ to denote the component-wise absolute value of a matrix or a vector.

II. Algorithms as dynamical systems

In this section we discuss how iterative algorithms can be represented as dynamical systems.

A. State space representation for exact arithmetic

Consider an iterative algorithm and a dynamical system with state vector x_k and dynamics $x_{k+1} = f(x_k)$. If, for some initial state $x_0 = \xi$, the state x_k of the dynamical system is equal to the k^{th} iterate of the algorithm, for all k, then we will say that the dynamical system represents the algorithm in state space form when the initial state is $x_0 = \xi$.

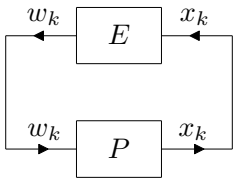


Fig. 1. Algorithm in finite precision as a dynamical system using I/O maps

B. State space representation for finite precision

To represent an algorithm in finite precision we also have to incorporate finite precision errors in the state space model. This could be done by considering the finite precision errors as disturbance inputs w_k in the dynamical model. A dynamical system $x_{k+1} = f(x_k, w_k)$ can be used to represent an algorithm in finite precision if for each iteration the iterates of the algorithm and the finite precision errors are equal to the state vector x_k and disturbance inputs w_k of the dynamical system, respectively.

C. Input-output map representation

An algorithm in finite precision can be represented by the system in Figure 1, which is a feedback interconnection of two input-output maps. In the figure, system P represents the input-output map of dynamics $x_{k+1} = f(x_k, w_k)$ and system E is the input-output map between the iterates of the algorithm and the finite precision errors w_k.

III. Forward error analysis based on input-to-state stability

In this section we present the first control-theoretic scheme of finding a forward error bound. Let us define the solution error as $e_k := x_k - x^*$, where x_k is the approximate solution at iteration k of an algorithm and x^* is the exact solution of a numerical problem. Given a dynamical system representation of an algorithm in state space form, $x_{k+1} := f(x_k, w_k)$, we can always find the dynamics for the solution error as $e_{k+1} = f_e(e_k, w_k) := f(e_k + x^*, w_k) + x^*$.

Proposition 1: Assume that the algorithm in finite precision can be considered as a dynamical system in state space form with dynamics $x_{k+1} := f(x_k, w_k)$. Also assume that x^* is an equilibrium point of the dynamical system when there is no disturbance, i.e. $w_k = 0$, and x^* is equal to the solution of the numerical problem.

If the dynamical system for the solution error $e_{k+1} = f_e(e_k, w_k)$, which has an equilibrium at the origin for $w_k = 0$, is input-to-state stable (ISS) [16], then the forward error, which is the norm of the solution error, is bounded by an expression of the following type:

$$\|e_k\| \leq \zeta_1(\|e_0\|, k) + \zeta_2(\|\mathbf{w}\|_\infty)$$

for each k, where $\mathbf{w} := (w_0, w_1, w_2, \ldots)$ is a sequence of input disturbances, $\|\mathbf{w}\|_\infty := \sup_k \|w_k\|$, ζ_1 is a $\mathcal{KL}$-function and ζ_2 is a $\mathcal{K}$-function.

Proof: The proof follows by the ISS [16] property. ■

Corollary 1: We have the following bound:

$$\lim_{k\to\infty} \|e_k\| \leq \zeta_2(\|\mathbf{w}\|_\infty).$$

Since the norm of e_k is the forward error, the above-mentioned proposition provides a systematic way for forward error analysis of iterative numerical algorithms.

In the above proposition, to find a bound on the forward error we need a bound on $\|\mathbf{w}\|_\infty$ that is independent of x_k or e_k. Since the finite precision errors w_k are a result of the computations involved in a single iteration of an algorithm they usually depend on the iterate x_k. We now give a result that could help us find a bound on $\|\mathbf{w}\|_\infty$ that is independent of the state vector.

Proposition 2: Assume that the algorithm in finite precision can be considered as a dynamical system with input-output maps as in Figure 1. If the following hold:

1) System P is finite gain l_∞-stable [17, Sec. 6.7.1], i.e.
$$\|\mathbf{x}\|_\infty \leq \gamma_1 \|\mathbf{w}\|_\infty + \beta_1$$
for all input sequences $\mathbf{w}$, where $\mathbf{x} := (x_0, x_1, x_2, \ldots)$, $\mathbf{w} := (w_0, w_1, w_2, \ldots)$ and γ_1 and β_1 are positive scalars;
2) System E is finite gain l_∞-stable, i.e.
$$\|\mathbf{w}\|_\infty \leq \gamma_2 \|\mathbf{x}\|_\infty + \beta_2$$
for all sequences $\mathbf{x}$, where γ_2 and β_2 are positive scalars;
3) The small gain condition is satisfied, i.e. $\gamma_1\gamma_2 < 1$,

then the output of system E, which is equal to the finite precision errors, is bounded and we have

$$\|\mathbf{w}\|_\infty \leq \frac{1}{(1-\gamma_1\gamma_2)}(\gamma_2\beta_1 + \beta_2).$$

Proof: The relationship follows by applying the small gain theorem [17, Sec. 6.7.3]. ■

IV. Forward error analysis based on the study of quantization effects

In this section we propose a forward error analysis scheme for iterative numerical algorithms that is based on results from the study of quantization effects in control systems. Quantization refers to the restriction of a variable to a discrete set of values and can arise in control systems due to digital implementation, analog-to-digital converters, digital-to-analog converters, etc. Due to finite precision, digital hardware can only represent a finite set of real numbers. In this sense finite precision and quantization are similar.

There is a lot of literature on the study of effects of quantization in control systems. A more detailed discussion on the sources and effects of quantization in control systems can be found in [18] and references therein. Among the literature, we have found that the work presented in [19] and [20] can be used for forward error analysis. In this paper we will present a new scheme based on the results given by Miller et al. [19]. For details on how the results in [20] can be used for forward error analysis, see [21].

A. Miller et al.'s work

Miller et al. [19] have used a Lyapunov-based approach to find ultimate bounds on solutions of perturbed linear control systems, where the perturbation arises due to quantization. They have considered a discrete-time linear system

$$z_{k+1} = Gz_k \tag{1}$$

and a nonlinear and possibly discontinuous perturbation of (1) given by

$$z_{k+1} = Gz_k + p(z_k) + \alpha_k, \tag{2}$$

where $z_k \in \mathbb{R}^n$, $G \in \mathbb{R}^{n\times n}$, $p : \mathbb{R}^n \to \mathbb{R}^n$ and $\alpha_k \in \mathbb{R}^n$ for $k = 1, 2, \cdots$. By comparing (1) and (2) we note that the perturbation is assumed to be additive and is represented by $(p(z_k) + \alpha_k)$. The term α_k represents the part of the perturbation that approaches zero as the time index k approaches infinity and the term $p(z_k)$ represents any perturbation that depends on the state and is possibly non-zero at the origin.

The following theorem, which is given in [19], can be used to find error bounds for iterative algorithms:

Theorem 1: Suppose that the magnitude of all the eigenvalues of matrix G of system (1) are less than one, $\alpha_k \to 0$ as $k \to \infty$ and p satisfies

$$\|p(z_k)\|_2 \leq \epsilon_1 \|z_k\|_2 + \epsilon_2 \tag{3}$$

for some small positive constants ϵ_1 and ϵ_2. If ϵ_1 is sufficiently small, then there exists a ψ (depending only on G, ϵ_1 and ϵ_2) such that system (2) is uniformly ultimately bounded with bound $\psi\epsilon_2$.

The definition of a uniformly ultimately bounded system given in [22, Sec. 6.1] is

Definition 1: A nonlinear system $z_{k+1} = f(z_k, k)$ is said to be uniformly ultimately bounded with bound β if for any $a > 0$ and for every $K \in \mathbb{Z}_{\geq 0}$, there exists an $l(a) > 0$, independent of K, such that $\|z_{k+K}\| \leq \beta$ for all $\|z_K\| \leq a$ and $k \geq l(a)$, where $\|\cdot\|$ denotes any norm on $\mathbb{R}^n$.

By an analysis similar to the one given in [19], it can be shown that a possible value for ψ is

$$\psi = \frac{\rho_1\rho_2}{1-\nu_1}, \tag{4}$$

where $\nu_1 := (\nu + \rho_1\rho_2\epsilon_1)$, ν is a constant such that $0 < \nu < 1$ and such that for a Lyapunov function $V(\cdot)$ of system (1) it holds that $V(z_{k+1}) - V(z_k) \leq (\nu - 1)V(z_k)$ for all z_k, ρ_1 is a constant such that $V(z_k) \leq \rho_1 \|z_k\|_2$ for all z_k, ρ_2 is a constant such that $\|z_k\|_2 \leq \rho_2 V(z_k)$ for all z_k, and ϵ_1 is sufficiently small such that $\nu_1 < 1$. Therefore, by Theorem 1 we have the following bound:

$$\lim_{k\to\infty} \|z_k\|_2 \leq \frac{\rho_1\rho_2}{1-\nu_1}\epsilon_2. \tag{5}$$

B. A forward error analysis scheme for iterative algorithms

We have defined the solution error e_k as the difference between the k^{th} iterate x_k of an algorithm and the solution x^* of a numerical problem, i.e. $e_k := x_k - x^*$. If we can represent an iterative algorithm in exact precision as a dynamical system

$$x_{k+1} := f(x_k), \tag{6}$$

TABLE I
CLASSICAL ITERATIVE METHODS

Matrix M	**Method**	**Convergence conditions**
D	Jacobi	A is SDD
$(D+L)$	Gauss-Seidel	A is SDD or SPD
$(\omega^{-1}D+L)$	Successive over-relaxation	A is SPD and $0<\omega<2$

SDD = strictly diagonally dominant; SPD = symmetric positive definite

or in finite precision as

$$x_{k+1} := f(x_k, w_k), \tag{7}$$

where w_k represents the finite precision error, then by subtracting x^* from both sides of (6) or (7) we can also find the dynamics of the solution error e_k

$$e_{k+1} := f_e(e_k) := f(e_k + x^*) - x^*, \tag{8}$$

or

$$e_{k+1} := f_e(e_k, w_k) := f(e_k + x^*, w_k) - x^*. \tag{9}$$

If systems (8)–(9) have the same structure as systems (1)–(2), then we can apply Theorem 1 and use (5) to find a bound on the norm of e_k. Therefore, we can use the results given in [19] to have a systematic way of forward error analysis of algorithms.

V. BOUNDS FOR THE CLASSICAL ITERATIVE METHODS

In this section we will apply the proposed forward error analysis schemes on the classical iterative methods [23, Sec. 10.1] for solving a system of linear equations $Ax^* = b$, where $A \in \mathbb{R}^{n\times n}$, $x^* \in \mathbb{R}^n$ and $b \in \mathbb{R}^n$. The iterations of the classical iterative methods are as follows:

$$x_0 = 0, \quad x_{k+1} = M^{-1}(Nx_k + b), \tag{10}$$

where $M \in \mathbb{R}^{n\times n}$ and $N \in \mathbb{R}^{n\times n}$ are matrices such that $A = M - N$. Table I lists some of the classical iterative methods [23, Sec. 10.1]. In the table, D is the diagonal of the matrix A, L is the strictly lower triangle part of A and ω is a scalar. If the convergence conditions listed in the table are satisfied, then matrices A and M are invertible, the magnitudes of all the eigenvalues of the matrix $M^{-1}N$ are less than one and the algorithm converges to the solution $x^* = A^{-1}b$. In the sequel we will assume that the convergence conditions are satisfied.

A. Finite precision error analysis for a single iteration

If the algorithm is implemented in floating point, then there are finite precision errors in the calculations in each iteration of the algorithm. These finite precision errors can be represented as an additive error [24, Sec 17.2] and we have the expression

$$x_{k+1} = \mathrm{fl}(M^{-1}(Nx_k + b)) = M^{-1}(Nx_k + b) + w_k, \tag{11}$$

where operator $\mathrm{fl}(\cdot)$ represents calculation in floating point and w_k represents the finite precision errors in the whole calculation.

Using a finite precision error analysis similar to the one in [24, Sec. 17.2] we can obtain the following bounds on w_k:

$$\begin{aligned} \|w_k\|_\infty &\leq c_1(1+\gamma_x)\left\|M^{-1}\right\|_\infty \\ &\quad \times(\|N\|_\infty + \|M\|_\infty)\|x^*\|_\infty, \end{aligned} \tag{12}$$

$$\begin{aligned} \|w_k\|_2 &\leq n^{1/2}c_1(1+\gamma_x)\left\|M^{-1}\right\|_\infty \\ &\quad \times(\|N\|_\infty + \|M\|_\infty)\|x^*\|_\infty, \end{aligned} \tag{13}$$

where $c_1 := 1.01h\mu^2 + 1.01h\mu + \mu$, h is maximum number of nonzero elements in any row of A, and

$$\gamma_x := \sup_k \left(\frac{\|x_k\|_\infty}{\|x^*\|_\infty}\right).$$

The details of finding the above bounds can be found in [25].

B. Forward error analysis using input-to-state stability

The classical iterative methods in finite precision can be represented as the dynamical system

$$x_{k+1} = f(x_k, w_k) = M^{-1}(Nx_k + b) + w_k. \tag{14}$$

To find the dynamics of the solution error $e_k := x_k - x^*$, we subtract x^* from both sides of the above equation to obtain

$$e_{k+1} = M^{-1}Ne_k + w_k. \tag{15}$$

Straightforward analysis leads to

$$\|e_k\|_2 \leq \left\|\left(M^{-1}N\right)^k\right\|_2 \|e_0\|_2 + \sum_{j=0}^{\infty}\left\|\left(M^{-1}N\right)^j\right\|_2 \|\mathbf{w}\|_\infty.$$

Since all the eigenvalues of $M^{-1}N$ have magnitude less than one, $\left\|\left(M^{-1}N\right)^k\right\|_2 \to 0$ as $k \to \infty$. Moreover, the series in the equation converges [24, Prob. 17.1], therefore (15) is input-to-state stable.

A bound on the finite precision error is given by (12). This bound is independent of the state, therefore we can also find a bound on the maximum finite precision error over all iterations that is independent of the state as

$$\begin{aligned} \|\mathbf{w}\|_\infty &\leq c_1(1+\gamma_x)\left\|M^{-1}\right\|_\infty \\ &\quad \times(\|N\|_\infty + \|M\|_\infty)\|x^*\|_\infty. \end{aligned} \tag{16}$$

If the bound on $\|w_k\|_\infty$ was dependent on the state, we could have used Proposition 2 to find a bound on $\|\mathbf{w}\|_\infty$.

Since we have a bound on $\|\mathbf{w}\|_\infty$ that is independent of the state and (15) is ISS, we can use Proposition 1 to obtain the following forward error bound for the classical iterative methods

$$\|e_k\|_\infty \leq \left\|(M^{-1}N)^k\right\|_\infty \|e_0\|_\infty + \frac{\|\mathbf{w}\|_\infty}{1 - \|M^{-1}N\|_\infty}, \tag{17}$$

where the bound on $\|\mathbf{w}\|_\infty$ is given by (16). For $k \to \infty$ the forward error bound becomes

$$\lim_{k\to\infty}\|e_k\|_\infty \leq \frac{1}{1 - \|M^{-1}N\|_\infty}\|\mathbf{w}\|_\infty. \tag{18}$$

Using (16) we get

$$\begin{aligned} \lim_{k\to\infty}\|e_k\|_\infty &\leq c_1(1+\gamma_x)(\|N\|_\infty + \|M\|_\infty) \\ &\quad \times\frac{\left\|M^{-1}\right\|_\infty}{1 - \|M^{-1}N\|_\infty}\|x^*\|_\infty. \end{aligned} \tag{19}$$

C. Forward error analysis using Miller et al.'s technique

Following the discussion in Section IV-B we can write the dynamics of the solution error e_k for classical iterative methods in finite precision as (15) and exact arithmetic as

$$e_{k+1} := M^{-1}Ne_k. \tag{20}$$

Comparing (20) and (15) with (1)–(2), we note that matrix G in (1) is $M^{-1}N$ and the additive perturbation is w_k. The perturbation w_k does not have a part that approaches zero as k approaches infinity, therefore we take $\alpha_k \equiv 0$ and $p(z_k) \equiv w_k$. Using the finite precision error bound (13) we get

$$\|w_k\|_2 \quad \leq \quad \epsilon_1 \|e_k\|_2 + \epsilon_2,$$

where

$$\epsilon_1 := 0, \tag{21}$$

$$\begin{aligned}\epsilon_2 \quad := \quad & n^{1/2}c_1(1+\gamma_x)\left\|M^{-1}\right\|_\infty \\ & \times(\|N\|_\infty + \|M\|_\infty)\|x^*\|_\infty. \end{aligned} \tag{22}$$

If all the eigenvalues of $M^{-1}N$ have magnitude less than one, which is the case here, then from standard Lyapunov theory we know that a possible Lyapunov function for system (20) is

$$V(e_k) = e_k^T P e_k,$$

where T in superscript denotes transposition and $P \in \mathbb{R}^{n\times n}$ is the solution of the discrete-time Lyapunov equation

$$(M^{-1}N)^T P(M^{-1}N) - P = -Q,$$

for some positive definite $Q \in \mathbb{R}^{n\times n}$ [17, Sec. 5.9]. For this choice of Lyapunov function we have

$$\rho_1 \leq \lambda_{\max}(P), \quad \rho_2 \leq \frac{1}{\lambda_{\min}(P)}, \quad \nu \leq 1 - \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)},$$

where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the largest and the smallest eigenvalue of a symmetric matrix, respectively.

If

$$\nu_1 := (\nu + \rho_1\rho_2\epsilon_1) = 1 - \frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} + \frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}\epsilon_1 < 1,$$

then using Theorem 1 we have

$$\lim_{k\to\infty}\|e_k\|_2 \leq \frac{\frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}}{\frac{\lambda_{\min}(Q)}{\lambda_{\max}(P)} - \frac{\lambda_{\max}(P)}{\lambda_{\min}(P)}\epsilon_1}\epsilon_2. \tag{23}$$

Using (21) and (22) we get

$$\begin{aligned}\lim_{k\to\infty}\|e_k\|_2 \quad \leq \quad & \frac{\lambda_{\max}(P)^2}{\lambda_{\min}(Q)\lambda_{\min}(P)}n^{1/2}c_1(1+\gamma_x)\left\|M^{-1}\right\|_\infty \\ & \times(\|N\|_\infty + \|M\|_\infty)\|x^*\|_\infty. \end{aligned} \tag{24}$$

D. Forward error analysis of classical iterative methods in the numerical analysis literature

A forward error bound for classical iterative methods is given in Section 17.2 of [24]. The bound (12) is used for finite precision error in a single iteration. The obtained forward error bound is

$$\begin{aligned}\lim_{k\to\infty}\|e_k\|_\infty \quad \leq \quad & c_1(1+\gamma_x)(\|N\|_\infty + \|M\|_\infty) \\ & \times\frac{\left\|M^{-1}\right\|_\infty}{1-\|M^{-1}N\|_\infty}\|x^*\|_\infty, \end{aligned} \tag{25}$$

with the condition that $\left\|M^{-1}N\right\|_\infty < 1$.

E. Comparison

In this section we compare the obtained forward error bounds. The expression of the bound obtained using input-to-state stability (19) is interestingly the same as the expression of the bound given by Higham (25). To compare the bound based on the work by Miller et al. we use numerical results.

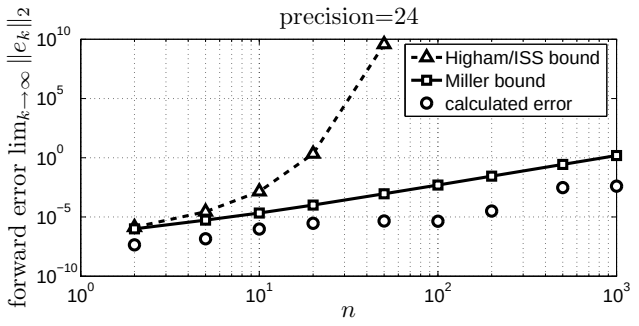
As an example, we consider systems of linear equations with tridiagonal $n \times n$ matrices with 2 as the diagonal entries and -1 as the off-diagonal entries. These matrices have certain properties that make them very interesting from a numerical analysis perspective. First of all, these matrices arise in the discretization of partial differential equations [26, Ch. 1], i.e. they can arise in problems involving real physical systems. Moreover, these matrices are invertible, positive definite, symmetric, sparse (for $n > 5$) and diagonally dominant. We also consider systems of linear equations with tridiagonal matrices with 4 as the diagonal entries and -1 as the off-diagonal entries. Besides all the other properties mentioned earlier, these matrices are also strictly diagonally dominant. For the right hand side vector in the system of linear equations we consider a unit vector with the n^{th} element equal to 1.

We will compare the algorithms for the Guass-Seidel method. Since our example systems of linear equations have symmetric and positive definite matrices, convergence is guaranteed for the Gauss-Seidel method.

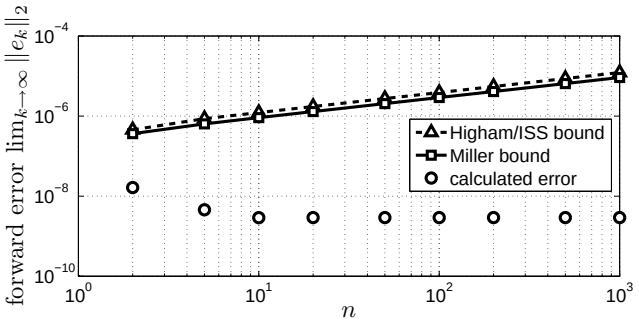
To calculate the bound obtained using Miller's technique (24) we need a matrix Q. Although any positive definite matrix can be chosen as Q, we would like to select a value that gives a lower bound. To calculate the bound we select Q by random search over one hundred random values. The random values of Q are generated by setting $Q = R^T R$, where R is a non-singular matrix with each element as an independent random number with a standard Gaussian distribution. We also take $Q := I - (M^{-1}N)(M^{-1}N)^T$ as one of our test cases in the random search. This choice of Q results in $P = I$, which has a unity condition number and could result in a tighter bound.

The calculated error bounds for a 24 bit precision are shown in Figure 2. In the figure we have also shown the errors in the calculated solution to give some idea of the conservativeness of these bounds. These errors were obtained by simulating the Guass-Seidel algorithm at a precision of 24 bits using the Multiple Precision Toolbox [27]. In the figure, we observe that the Higham/ISS bound is not reasonable for larger values of n for the case of 2 as diagonal entries. This is due to the fact that for the considered example the values of $1-\left\|M^{-1}N\right\|_\infty$, which is in the denominator of the expression of the bound obtained using ISS and the bound give by Higham, becomes close to zero. From Figure 2 we also observe that the bound based on Miller et al.'s approach (24) is the least conservative.

Although Higham's bound (or the bound obtained using ISS) is not the tightest bound, we note that the simplicity of the expression (25) allows us to easily comment on the dependence of the bound on problem size n and machine precision μ.



(a) matrix with 2 as diagonal entries



(b) matrix with 4 as diagonal entries

Fig. 2. Comparison of forward error bounds for the Gauss-Seidel method

VI. Conclusions and future work

In this paper we have considered numerical iterative algorithms in finite precision as dynamical systems with disturbances. We have shown that some results from control theory can be used to study algorithms in finite precision. We have focused on the forward error analysis of iterative algorithms and presented two control-theoretic schemes for forward error analysis. The first scheme is based on the control concept of input-to-state stability, while the second scheme is based on results from the study of quantization effects in control systems. The proposed schemes are applied to the classical iterative methods. The obtained forward error bounds are numerically shown to be tighter than the bounds in the numerical analysis literature.

The proposed forward error analysis schemes have been applied on an algorithm that can be represented by a linear dynamical system. Since we have used control tools that are also applicable to nonlinear systems, the proposed schemes may also be applied to algorithms that can be represented by a nonlinear dynamical system. However, we will have to show that the dynamical system used to represent the algorithm is Lyapunov stable or ISS depending on the forward error analysis scheme that is used.

We have only focused on the forward error analysis of algorithms. Backward error analysis of an algorithm is also important since it guarantees numerical stability of the algorithm [24, Sec. 1.5]. It would be interesting to find a control-theoretic method for backward error analysis of algorithms.

References

[1] James Hurt. Some stability theorems for ordinary difference equations. *SIAM Journal on Numerical Analysis*, 4(4):582–596, 1967.

[2] James M. Ortega. Stability of difference equations and convergence of iterative processes. *SIAM Journal on Numerical Analysis*, 10(2):268–282, 1973.

[3] A. Bhaya and E. Kaszkurewicz. A control-theoretic approach to the design of zero finding numerical methods. *IEEE Transactions on Automatic Control*, 52(6):1014–1026, 2007.

[4] Amit Bhaya and Eugenius Kaszkurewicz. *Control Perspectives on Numerical Algorithms and Matrix Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[5] K. Kashima and Y. Yamamoto. System theory for numerical analysis. *Automatica*, 43(7):1156–1164, 2007.

[6] R. Brockett. Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. *Linear Algebra and its Applications*, 146(1):79–91, 1991.

[7] Oumar Diene and Amit Bhaya. A study of the robustness of iterative methods for linear systems. In *8th International Conference on Numerical Analysis and Applied Mathematics*, Rethymno, Greece, 2009.

[8] J. P. Chehab and Jacques Laminie. Differential equations and solution of linear systems. *Numerical Algorithms*, 40(2):103–124, 2005.

[9] M. T. Chu. Linear algebra algorithms as dynamical systems. *Acta Numerica*, 17:1–87, 2008.

[10] Uwe Helmke, Jens Jordan, and Alexander Lanzon. A control theory approach to linear equation solvers. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, 2006.

[11] Lars Grüne. *Asymptotic behavior of dynamical and control systems under perturbation and discretization*. Springer, 2002.

[12] S. Che, J. Li, J.W. Sheaffer, K. Skadron, and J. Lach. Accelerating compute-intensive applications with GPUs and FPGAs. In *Symposium on Application Specific Processors*, pages 101 –107, Anaheim, California, USA, 2008.

[13] George A. Constantinides. Tutorial paper: Parallel architectures for model predictive control. In *Proceedings of the European Control Conference*, pages 138–143, Budapest, Hungary, 2009.

[14] R. Weber, A. Gothandaraman, R.J. Hinde, and G.D. Peterson. Comparing hardware accelerators in scientific applications: A case study. *IEEE Transactions on Parallel and Distributed Systems*, 22(1):58–68, 2011.

[15] A. Roldao Lopes, A. Shahzad, George A. Constantinides, and Eric C. Kerrigan. More flops or more precision? Accuracy parameterizable linear equations solvers for model-predictive control. In *Proceedings of the IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA, USA, 2009.

[16] Z. P. Jiang and Yuan Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37(6):857–869, 2001.

[17] M. Vidyasagar. *Non-linear Systems Analysis*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 2002.

[18] Hernan Haimovich. *Quantisation Issues in Feedback Control*. PhD thesis, The University of Newcastle, 2006.

[19] R. K. Miller, M. S. Mousa, and A. N. Michel. Quantization and overflow effects in digital implementations of linear dynamic controllers. *IEEE Transactions on Automatic Control*, 33(7):698–704, 1988.

[20] E. Kofman, H. Haimovich, and M. M. Seron. A systematic method to obtain ultimate bounds for perturbed systems. *International Journal of Control*, 80(2):167–178, 2007.

[21] Ammar Hasan, Eric C. Kerrigan, and George A. Constantinides. Quantization in control systems and forward error analysis of iterative numerical algorithms. In *UKACC International Conference on Control 2010*, Coventry, UK, 2010.

[22] Anthony N. Michel, Ling Hou, and Derong Liu. *Stability of dynamical systems: continuous, discontinuous, and discrete systems*. Birkhäuser, 2008.

[23] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, USA, 1996.

[24] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 2002.

[25] Ammar Hasan. *Control Theoretic Analysis and Design of Numerical Algorithms*. PhD thesis, Imperial College London, 2012.

[26] Gilbert Strang. *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007.

[27] B. Barrowes. Multiple Precision Toolbox. `http://www.sondette.com/math/mp_toolbox.html`, February 2010.