A control theory approach to linear equation solvers

Uwe Helmke Jens Jordan Alexander Lanzon

Abstract— **We present a new iterative approach for solving linear systems of equations. Our method is inspired by feedback stabilization schemes from robust control and yields a control system, whose parameters can be tuned to achieve prescribed convergence properties. In contrast to well–known iterative solution methods for linear equations from linear algebra, such as GMRES(m) or Arnoldi's method, the proposed dynamical systems solution algorithms have the advantage of being globally convergent or having tunable convergence properties.**

Keywords— **linear equations, iterative methods, feedback control, stabilization.**

I. INTRODUCTION

Solving linear systems of equations $Ax = b$ or computing the inverse A^{-1} of a matrix are core problems of numerical linear algebra, for which powerful solution methods and software packages exist. For the inversion of moderate size matrices, standard LU or QR factorization methods work very well, but these methods are no longer applicable for large scale matrices. Krylov subspace methods such as conjugate gradient, GMRES(m) or Lanczos and Arnoldi work well for even very large systems of equations defined by positive definite symmetric or normal matrices A. However, the situation becomes more complicated (and interesting) in other cases. In fact, the dynamics of an iterative method such as e.g. GMRES(m) can be quite complex and is far from being fully understood. For indefinite symmetric matrices, GMRES(m) exhibit continua of non-trivial equilibrium points that may prevent the algorithm to converge to the desired solution. The situation becomes even worse for matrices far from normality, forcing the algorithm to loose fast local convergence or create even regimes of chaotic behavior in the phase space; see [1].

Of course, there is no reason, why there might not exist reliable linear equation solvers that would work also in those cases where the presently known methods fail. The very structure of present iterative solution methods even points to a resolution of this issue. In fact, such iterative schemes define feedback control systems and therefore can be analyzed using tools from control theory. It is not obvious whether or not the feedback control schemes implemented by current numerical linear algebra algorithms are optimal in view of control systems performance. Indeed, it is quite plausible that closed loop performance can be improved using alternative feedback control strategies; see [2], [3] for first results in this direction.

To achieve a better understanding of how to design iterative algorithms for linear equations, we propose a control theoretic approach that enables us to address the issues of stabilization of dynamics in a rather straightforward way. It also provides us with a large class of algorithms that are potentially useful to solve linear equations. We formalize their desired properties in the following definition.

Definition 1: Let m be an integer with $m < n$. An m-**control scheme** for solving $Ax = b$ is a control system

$$x_{t+1} = f(x_t, u_t) \quad (1)$$
$$y_t = Ax_t - b, \quad (2)$$

with the following properties:

- **low complexity implementation**: $f(x, u)$ should be easily computable from the variables x, u.
- **output stabilizability**: There exists an easily computable feedback control law
$$u_t = K(x_t, A^t b, \cdots, A^{t-m} b)$$
such that the output $y_t, t > m,$ of any trajectory x_t of the closed loop system converges to zero.

We do not attempt here to formalize the meaning of "easily computable". Heuristically speaking, what we want is that the computations performed by the algorithm are easier to carry out than a direct solution approach to solve $Ax = b$, or to invert A, would be. In particular, inverses of A should not appear in $f(x, u)$.

Of course, the main problem is to show that such algorithms do exist. As one can see from the definition, this is really a problem of nonlinear feedback stabilization with complexity constraints and as such, to the best of our knowledge, has not been addressed before. The earlier numerical linear algebra on the problem can be re-interpreted as saying that such algorithms do in fact exist for restricted classes of matrices (such as positive definite Hermitian matrices). Here we try to extend the scope of such investigations by introducing a larger class of algorithms and investigate their stabilization properties. These algorithms are described in the next sections.

We first review a generalization of GMRES(m) using d-stage Runge–Kutta methods. Solving an associated rational

Manuscript draft date is May 29, 2006. Corresponding author is U. Helmke. Tel: +49 931 888 5003, Fax: +49 931 888 4611, Email: helmke@mathematik.uni-wuerzburg.de.

Uwe Helmke, Jens Jordan are at Mathematisches Institut, Universität Würzburg, Am Hubland, 97074 Würzburg, Germany.

Alexander Lanzon is at the Research School of Information Sciences and Engineering, The Australian National University, Canberra ACT 0200, Australia and National ICT Australia Ltd.

approximation problem yields to algorithms with optimal local convergence rates. Proceeding in a different direction, new globally convergent schemes are proposed based on linear optimal control and matrix Riccati equations. Our final scheme contains all these methods as special cases. Thus we introduce algorithms for solving linear equations, by using a feedback interconnection scheme with a delay one element. This enables us to apply standard techniques from robust and optimal control to stabilize the dynamics. At present, algorithms that combine global stability with low computational complexity are available only for restricted classes of matrices. This is true for the known as well as for the newly proposed schemes. Nevertheless, our results suggest that such "systems theory approaches" to numerical linear algebra can yield promising solution methods in cases where existing algorithms fail.

II. RUNGE-KUTTA METHOD FOR $Ax = b$

We apply the well-known Runge-Kutta method for integrating ordinary differential equations to obtain a natural generalization of the well-known GMRES(m) method [4]. Moreover, we consider optimization tasks for the coefficients in the associated Butcher tableau to propose new iterative solution methods for linear equations $Ax = b$ with optimal performance. The d-stage Runge-Kutta method for integrating

$$\dot{x} = Ax - b \tag{3}$$

with step-size $h > 0$ is equivalent to the discrete-time system

$$\begin{aligned} x_{t+1} &= x_t + hK\beta & (4)\\ K &= (Ax_t - b)e^\top + hAKF^\top. & (5) \end{aligned}$$

Here $e^\top := (1, \cdots, 1)$, $\beta := (\beta_1, \cdots, \beta_d)^\top$, F is an arbitrary $d \times d$-matrix, and the triple (β, F, e) defines the so-called Butcher-array of $\dot{x} = Ax - b$. Here, we assumed that $e^\top \beta = 1$. Moreover, the $n \times d$-matrix K is formally defined as

$$K = \sum_{i=0}^{\infty} (hA)^i (Ax_t - b)e^\top (F^\top)^i.$$

Here we circumvent convergence issues for this series by rewriting the closed loop system in explicit form as follows. Given realizations F, β, e, consider rational transfer function

$$r(z) := \beta^\top (I - zF)^{-1} e.$$

Then $K\beta = -r(hA)(Ax_t - b)$ and therefore the d-th stage Runge-Kutta discretization scheme for (3) is given as

$$x_{t+1} = (I - hAr(hA))x_t - hr(hA)b. \tag{6}$$

Thus the autonomous part of the dynamics of (6) is given by the rational matrix valued function $R(hA)$, where $R(z) := 1 - zr(z), R(0) = 1$ is rational function of degree $\leq d$. Note, that if F is nilpotent, then r is a polynomial of degree $< d$ with $r(0) = 1$. Runge-Kutta methods with nilpotent F are called explicit Runge-Kutta methods. Morover, any such polynomial $\pi(z)$ of degree $< d$ has a representation as $\pi(z) = \beta^\top (I - zF)^{-1} e$ with a nilpotent matrix F. Thus the class of explicit d-stage Runge Kutta schemes for (3) coincides exactly with the control systems defined by the GMRES(d) scheme

$$x_{t+1} = (I - hA\pi(hA))x_t - h\pi(hA)b. \tag{7}$$

In the following we assume that A is a complex invertible matrix whose eigenvalues are contained in a compact subset $S \subset \{z \in \mathbb{C} \mid z \neq 0\}$. This situation arises often in practise, when the eigenvalues of A are not known exactly, but where bounds on the sizes of the eigenvalues are available. Let $Rat(d)$ denote the set of rational functions $R = p/q$, where p, q are real polynomials of degree $\leq d$, $q(0) = 1$ and $p(0) = 1$. Using Kalman's realization theorem from linear systems theory we see that every rational function $R \in Rat(d)$ has a realization as $R(z) := \beta^\top (I - zF)^{-1} e$ for a suitable Butcher array (β, F, e). Let $\|R\|_S := max\{|R(z)| \mid z \in S\}$ denote the H_∞-norm of R on S. We consider the task of solving the rational H_∞–approximation problem

$$\delta_d(S) := inf\{\|R\|_S \mid R \in Rat(d)\}. \tag{8}$$

Since the constant function $1 \in Rat(d)$, we have $0 \leq \delta_d(S) \leq 1$. Thus for any rational function $R \in Rat(d)$ we have

$$\|R\|_S \geq \delta_d(S).$$

Any rational function $R \in Rat(d)$ with $\|R\|_S = \delta_d(S)$ is called an *optimal Butcher scheme for* S.

Theorem 1: Let $R_h \in Rat(d)$ be an optimal Butcher scheme for hS and $r_h(z) := (R_h(z) - 1)/z$. Then the algorithm

$$x_{t+1} = (I + hAr_h(hA))x_t - hr_h(hA)b. \tag{9}$$

converges linearly to $A^{-1}b$, provided $\delta_d(hS) < 1$. Moreover, let $h_* > 0$ be chosen such that

$$h_* = \arg\min\{\delta_d(hS) \quad | \quad h > 0\}.$$

Then the Runge-Kutta scheme

$$x_{t+1} = (I + h_* Ar_*(h_* A))x_t - h_* r_*(h_* A)b \tag{10}$$

with $r_*(z) := (R_*(z) - 1)/z$ and $R_* := R_{h_*}$ converges linearly to $A^{-1}b$, provided there exists a step-size $h > 0$ with $\delta_d(hS) < 1$.

The Runge-Kutta schemes proposed in this theorem have the interpretation of being those with the best, i.e. fastest convergence rate. It is therefore called the *optimal Runge-Kutta scheme* for solving $Ax = b$. Using results from complex approximation theory one can give concrete examples of compact sets S, together for which the algorithm in the theorem is provably convergent. As mentioned before, Runge -Kutta methods contain as a special case the GMRES(m) method and in this case optimal polynomial Runge-Kutta have been already studied in the literature; see e.g. [5]. The rational approximation task is however much more complicated, but has the benefit of yielding improved convergence rates.

One may wonder how to implement such schemes, as they seem to require the inversion of polynomial matrix functions. But this can often be done in a rather straightforward way. Suppose that $R_*(z) = p(z)/q(z)$ is a coprime factorization and that ψ denotes the characteristic polynomial of F. Assume, that ψ, q are coprime. Then, solving the Bezout equation $aq + b\psi = 1$, yields a polynomial $a(z)$ of degree $< d$ with $a(hA) = q(hA)^{-1}$. Therefore the rational Runge-Kutta scheme is equivalent to a polynomial one, i.e. to GMRES(m) with $m \leq 2d$.

III. SPLITTING METHODS

We begin with a brief survey on splitting methods for solving linear equations; see [4], [6].

A *splitting method* is a discrete dynamical system of the form $x_{t+1} = (I - NA)x_t + Nb$ where N is an arbitrary nonsingular matrix. Such a method converges from any initial condition to the unique equilibrium $x_* = A^{-1}b$ if and only if the spectral radius of $(I - NA)$ satisfies $\rho(I - NA) < 1$.

Examples of common splitting methods are:

a) **Trivial method**: $N = I$ defines a splitting method which converges if and only if $\rho(I - A) < 1$.
b) **Jacobi method**: Let D be the diagonal part of A, which is assumed to be nonsingular here. Then, $N = D^{-1}$ defines a splitting method which converges if and only if $\rho(I - D^{-1}A) < 1$.
c) **Gauß-Seidel method**: Let R be the strict upper triangular part of A and assume $L = A - R$ is nonsingular here. Then, $N = L^{-1}$ defines a splitting method which converges if and only if $\rho(L^{-1}R) < 1$.

In each of the above examples, the necessary and sufficient condition for convergence has to be satisfied by the choice of N if the method is to be used. Otherwise, one has to find an appropriate N that ensures convergence of the algorithm (i.e. $\rho(I - NA) < 1$). We now show how to design such a convergent splitting method using well-known ideas from linear optimal control.

For any input matrix $B \in \mathbb{R}^{n \times m}$ consider the problem of minimizing the quadratic cost functional

$$J(u) = \frac{1}{2}\sum_{t=0}^{\infty}(\|y_t\|^2 + \|u_t\|^2) \tag{11}$$

for the linear discrete-time system

$$\begin{aligned} x_{t+1} &= (I - A)x_t + Bu_t + b && (12) \\ y_t &= Ax_t - b. && (13) \end{aligned}$$

Standard linear quadratic controller design (see [7], [8]) then leads to an explicit solution, provided the pair $(I - A, AB)$ is discrete-time stabilizable. The minimal value of the cost functional is

$$J(u^*) = \frac{1}{2}(Ax_0 - b)^T P(Ax_0 - b), \tag{14}$$

where P is the unique positive semidefinite solution of the discrete-time Riccati equation

$$\begin{aligned} P &= I_n + (I - A)^T P(I - A) + (B^T A^T P(I - A))^T \\ &\quad (I_m + B^T A^T PAB)^{-1} B^T A^T P(I - A). && (15) \end{aligned}$$

Moreover, one can explicitly calculate the optimal control sequence u^* as $u_t = -K(Ax_t - b)$ where

$$K = (I_m + B^T A^T PAB)^{-1} B^T A^T P(I - A). \tag{16}$$

This leads us to the following LQRES algorithm

(i) Choose B such that $(I - A, AB)$ is stabilizable
(ii) Calculate the unique positive definite solution of the Riccati Equation (15)
(iii) Calculate K as in (16)
(iv) Iterate the closed loop system $x_{t+1} = (I - (I - BK)A)x_t + (I - BK)b$.

This defines a globally convergent splitting method by setting $N = I - BK$, with K as above. Note that for generic choices of A step (i) is always solvable. Moreover, the freedom in choosing B can be potentially exploited to improve convergence speed. If the eigenvalues λ of A satisfy $|1 - \lambda| < 1$, then one can choose $B = 0$. Then LQRES coincides with the stable fixed point iteration $x_{t+1} = (I - A)x_t + b$, defined by the trivial method. Provided m is relatively small, step (iii) does not cause major numerical problems. Of course, the bottleneck of this method lies in solving the algebraic Riccati equation, i.e. in step (ii). Here iterative methods are useful, by solving the dynamic Riccati equation. The obvious advantage of LQRES to other linear systems solvers such as GMRES(m) is its global stability. In fact, it is a standard result from linear systems theory, that the Riccati–based optimal control strategy leads to an asymptotically stable closed loop system $x_{t+1} = (I - (I - BK)A)x_t + (I - BK)b$, see [8].

Theorem 2: Let A be invertible. The LQRES iteration

$$x_{t+1} = (I - (I - BK)A)x_t + (I - BK)b \tag{17}$$

is global assymptotically stable. In particular, it converges for any initial vector x_0 to the unique solution of equation $Ax = b$.

The disadvantage of LQRES is the expensive preconditioning process by solving an algebraic Riccati equation. For matrices of size larger then $n = 1000$ this is hardly possible using common numerical tools. Computational more attractive algorithms are based on suboptimal techniques.

It is easy to construct examples where the LQRES method is convergent but none of the other three methods ("Trivial", "Jacobi", and "Gauß-Seidel") is. Moreover, we can improve performance of the LQRES algorithm by choosing an appropriate matrix B. Simulations supporting this statement are given below.

Experiment 1: Choose

$$A = \begin{pmatrix} 1 & 2 & -2 \\ 0 & 2 & 4 \\ 0 & 0 & 3 \end{pmatrix};\ b = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix};\ B1 = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix};\ B2 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix}$$

as in [1] (where GMRES(1) and GMRES(2) do not converge).

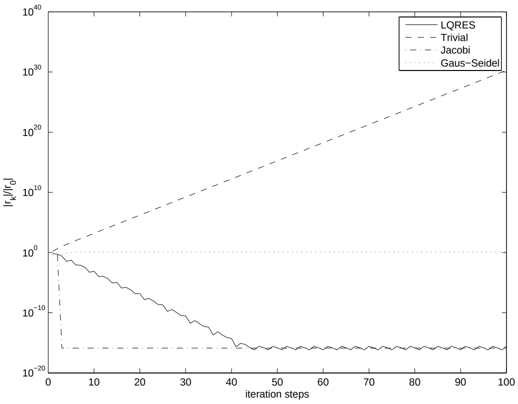

Fig. 1. Experiment 1 with $B = B1$.

Experiment 2: This example shows that appropriate choices of B can improve the performance behavior.

$$A = \begin{pmatrix} 1 & 0.2 & 0.4 \\ 0 & 1.1 & 0.3 \\ 0.1 & 0.1 & 0.9 \end{pmatrix} ; \; b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

$$B1 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} ; \; B2 = \begin{pmatrix} 1 & -1 \\ 1 & 0 \\ 1 & -10 \end{pmatrix}$$

IV. OUTPUT FEEDBACK CONTROL SCHEME

In this section we propose our main new method that contains the previous methods as special cases. It is based on general matrix shift strategies. Consider a discrete-time dynamical system P described by the state-space difference equations:

$$x_{t+1} = x_t + \Phi u_t \tag{18}$$

$$y_t = A x_t + \Xi u_t, \tag{19}$$

where $t \in \mathbb{Z}$ is the discrete-time index, $x_t \in \mathbb{C}^n$ is the state vector, $u_t \in \mathbb{C}^n$ is the vector of input signals, $y_t \in \mathbb{C}^n$ is the vector of measured outputs and matrices $A, \Phi, \Xi \in \mathbb{C}^{n \times n}$, where A, Φ is nonsingular. Consider the output feedback law:

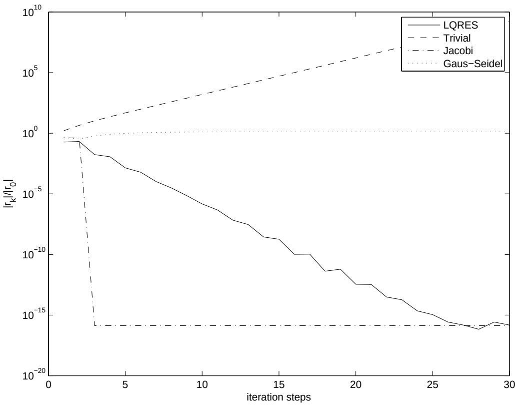

Fig. 2. Experiment 1 with $B = B2$. (Note that Jacobi converges quicker then LQRES).

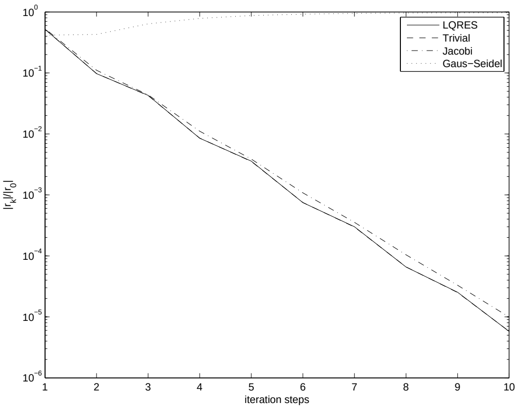

Fig. 3. Experiment 2 with $B = B1$

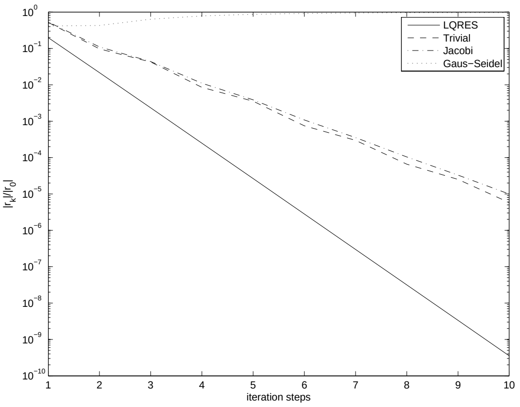

Fig. 4. Experiment 2 with $B = B2$.

$$u_{t+1} = b - y_t, \tag{20}$$

where $b \in \mathbb{C}^n$. Then equations (18) – (20) can be presented as in Figure 5. Note that the open–loop transfer function contains an integrator element with a repeated pole at 1. In fact, any stable recursive scheme for solving a linear system of equations that converges to the correct steady–state value without any steady–state error must contain an integrator in the feedback loop. The above equations (18) – (20) can be rewritten in a compact matrix form, leading to the definition of our basic control system for solving $Ax = b$.

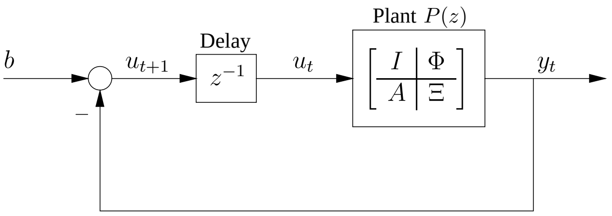
Plant $P(z)$
b
u_{t+1}
Delay
z^{-1}
u_t
$\left[\begin{array}{c|c} I & \Phi \\ \hline A & \Xi \end{array}\right]$
y_t
$-$

Fig. 5. Feedback interconnection describing the algorithm

Definition 2: Let $\Phi, \Xi \in \mathbb{C}^{n\times n}$ denote arbitrary matrices with Φ invertible. Then the associated feedback control scheme for solving $Ax = b$ is the controlled dynamical system

$$\begin{bmatrix} x_{t+1} \\ u_{t+1} \end{bmatrix} = \begin{bmatrix} I & \Phi \\ -A & -\Xi \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} b, \tag{21}$$

with output function given by the residue

$$z_t = Ax_t - b.$$

Here Φ, Ξ are a design choice.

Note, for any solution trajectory (x_t, u_t), u_t is a solution of the second order system

$$u_{t+2} + (\Xi - I)u_{t+1} + (A\Phi - \Xi)u_t = 0.$$

The main purpose of course is to tune the parameters Φ, Ξ in such a way that the output z_t of (21) goes to zero as t tends to infinity. Thus the task of solving linear equations can be reformulated as an output stabilization problem for system (21). Purely for the sake of analyzing the stability properties of (21) we consider the transformed system in the new variables $z_t = Ax_t - b$ and u_t, given as:

$$\begin{bmatrix} z_{t+1} \\ u_{t+1} \end{bmatrix} = \begin{bmatrix} I & A\Phi \\ -I & -\Xi \end{bmatrix} \begin{bmatrix} z_t \\ u_t \end{bmatrix}. \tag{22}$$

Our first observation is that Φ, Ξ act as admissible choices for (21) that drive the output z_t to zero if and only if they stabilize (22). Thus all trajectories (z_t, u_t) (22) converge to zero if and only if (x_t, u_t) converges to $(A^{-1}b, 0)$. Consequently, the control issue for (21) reduces to an internal stabilizability task for (22).

Now, let us explain the relation to previously mentioned algorithms, such as GMRES(m). Using the change of variables $(z_t, u_t) \mapsto (z_t, z_t + u_t)$ one checks that (22) becomes equivalent to

$$\begin{bmatrix} z_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} I - A\Phi & A\Phi \\ 0 & 0 \end{bmatrix} \begin{bmatrix} z_t \\ v_t \end{bmatrix}, \tag{23}$$

provided Ξ is chosen as

$$\Xi := A\Phi.$$

Moreover, by chosing $v_0 := 0$ the system is equivalent to

$$x_{t+1} = (I - \Phi A)x_t + \Phi b. \tag{24}$$

Thus, if $\Phi = \pi(A)$ is chosen as a polynomial in A, then (IV) is equivalent to GMRES(m). In contrast, the special choice $\Phi := I - BK$ yields the LQRES algorithm (17).

If the feedback interconnection of Figure 5 is designed to be internally stable through appropriate selection of a nonsingular matrix $\Phi \in \mathbb{C}^{n\times n}$ and a matrix $\Xi \in \mathbb{C}^{n\times n}$, then the final state vector $x_\infty := \lim_{t\to\infty} x_t$ must solve

$$Ax_\infty = b, \tag{25}$$

as from equations (18) – (20) it easily follows (as $t \to \infty$) that:

$$\begin{aligned} x_{t+1} - x_t \to 0 &\Leftrightarrow u_t \to 0 \\ &\Rightarrow Ax_t \to b. \end{aligned}$$

By inspection, the goal of internally stabilizing the above feedback interconnection is equivalent to chosing the matrices $\Phi \in \mathbb{C}^{n\times n}$ and $\Xi \in \mathbb{C}^{n\times n}$ such that the spectral radius function ρ satisfies

$$\begin{aligned} &\rho \begin{pmatrix} I & \Phi \\ -A & -\Xi \end{pmatrix} < 1 \\ \Leftrightarrow \quad &\det\left[Iz^2 + (\Xi - I)z + (A\Phi - \Xi)\right] \neq 0 \quad \forall |z| \geq 1. \end{aligned} \tag{26}$$

Thus we have shown the following theorem.

Theorem 3: Let $\Xi \in \mathbb{C}^{n\times n}$ and $\Phi \in \mathbb{C}^{n\times n}$ invertible such that

$$\det\left[Iz^2 + (\Xi - I)z + (A\Phi - \Xi)\right] \neq 0 \quad \forall |z| \geq 1.$$

Then any solution (x_t, u_t) of (21) satisfies

$$x_\infty := \lim_{t\to\infty} x_t = A^{-1}b, \quad \lim_{t\to\infty} u_t = 0.$$

There are several ways how a good choice of stabilizing matrices can be made. First of all, note that

$$\begin{bmatrix} I & \Phi \\ -A & -\Xi \end{bmatrix} = \begin{bmatrix} I & 0 \\ -A & 0 \end{bmatrix} + \begin{bmatrix} \Phi \\ -\Xi \end{bmatrix} \begin{bmatrix} 0 & I \end{bmatrix} \tag{27}$$

is in output injection form $\mathcal{A} + \mathcal{J}\mathcal{C}$ for the observable system

$$\mathcal{A} := \begin{bmatrix} I & 0 \\ -A & 0 \end{bmatrix}, \quad \mathcal{C} := \begin{bmatrix} 0 & I \end{bmatrix}$$

and output injection matrix

$$\mathcal{J} := \begin{bmatrix} \Phi \\ -\Xi \end{bmatrix}.$$

Thus $\Xi \in \mathbb{C}^{n\times n}, \Phi \in \mathbb{C}^{n\times n}$ are internally stabilizing if and only if they stabilize $(\mathcal{C}, \mathcal{A})$ by output injection. Therefore any stabilizing output injection gain $\mathcal{J}$ with Φ invertible will lead to a convergent algorithm for computing $A^{-1}b$.

Of course there are many possible choices for selecting of a stabilizing output injection gain $\mathcal{J}$. One way is e.g. by characterizing $\mathcal{J}$ by a linear matrix inequality and then use standard software for solving LMIs. Thus, given $\gamma < 1$ and $(\mathcal{C}, \mathcal{A})$ observable, we want to find $\mathcal{J}$ such that $\rho(\mathcal{A} + \mathcal{J}\mathcal{C}) \leq \gamma$, where $\rho(.)$ is the spectral radius. The smaller γ, the faster then the resulting convergence time constant (though still achieving linear rate of convergence). Equivalently, this amounts to solving the non-convex optimization task of finding – for given $\gamma < 1$ and $(\mathcal{C}, \mathcal{A})$ observable – output feedback injection transformations $\mathcal{J}$ and T nonsingular such that $\overline{\sigma}(T(\mathcal{A} + \mathcal{J}\mathcal{C})T^{-1}) \leq \gamma$, where $\overline{\sigma}(.)$ denotes the largest singular value.

Another possibility is to restrict the structure of admissible output injection matrices and reformulate the problem as a lower complexity robust polynomial interpolation problem. We briefly sketch how this could be done.

First note, that the spectral radius function

$$r(\Phi, \Xi) := \rho \begin{pmatrix} I & \Phi \\ -A & -\Xi \end{pmatrix}$$

is a natural measure of performance for the feedback control scheme (21). The idea therefore is to choose Φ, Ξ such that $r(\Phi, \Xi)$ becomes minimized. Clearly the choice $\Phi := A^{-1}, \Xi := I$ would be optimal as it renders $r(\Phi, \Xi) = 0$. However, as mentioned above, we cannot implement A^{-1} directly into the feedback control scheme as it is not available to us. Therefore, computationally less expensive control strategies are sought for. Given any nonsingular matrix Φ, choose

$$\Xi = p(A\Phi) \tag{28}$$

where p denotes any degree $m < n$ real polynomial. Then Φ, Ξ are internally stabilizing if and only if for any eigenvalue λ of $A\Phi$

$$z^2 + (p(\lambda) - 1)z + (\lambda - p(\lambda))$$

is a Schur polynomial. Recall, [9], that a complex quadratic polynomial

$$z^2 + a_1 z + a_0$$

is Schur stable if and only if

$$|a_0|^2 + |a_1 - \bar{a_1} a_0| < 1.$$

Theorem 4: Let Φ be any nonsingular matrix and π denote a degree m polynomial with $\Xi := A\Phi - \pi(A\Phi)$. Then Φ and Ξ are internally stabilizing if and only if for any eigenvalue λ of $A\Phi$ the following inequality in $w := \pi(\lambda)$ holds

$$|\lambda - \bar{\lambda} w - (1 - |w|^2)| < 1 - |w|^2. \ (*)$$

Moreover, if $\Phi, \Xi = A\Phi - \pi(A\Phi)$ are internally stabilizing, then any eigenvalue λ of $A\Phi$ satisfies $0 < |\lambda| < 2(1 + |\pi(\lambda)|) \leq 4$.

A special and well-known case is obtained by choosing

$$\Xi = A\Phi, \tag{29}$$

i.e. $\pi = 0$. Then the closed-loop system matrix will have a bunch of benign eigenvalues at $z = 0$ (i.e. the closed-loop interconnection has some dead-beat poles) and condition (*) becomes equivalent to

$$\rho(A\Phi - I) < 1. \tag{30}$$

With the choice of Ξ as in equation (29), the dynamic equation (21) can be rewritten as

$$\begin{bmatrix} x_{t+1} \\ u_{t+1} \end{bmatrix} = \begin{bmatrix} I \\ -A \end{bmatrix} \begin{bmatrix} I & \Phi \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} b$$

which, using $\nu_t := x_t + \Phi u_t$ in turn can be reduced to

$$\nu_{t+1} = (I - \Phi A)\nu_t + \Phi b \tag{31}$$

This is the general form of splitting algorithms discussed earlier for which standard stability results are available. In fact, let Φ be chosen such that the spectral radius $\rho(A\Phi - I) < 1$. Then (31) converges to the unique solution x_∞ of $Ax = b$, provided (30) holds.

Consequently, our design task has now simplified to choosing Φ such that condition (30) is satisfied. The obvious ideal choice is to select $\Phi = A^{-1}$ as this would yield a further set of dead-beat poles. However, we cannot really make that ideal choice of Φ, as the whole aim of this discussion is that A^{-1} is unavailable to us.

Even though A^{-1} may be unknown, in some situations we may have a guess of A^{-1} that is sufficiently accurate to satisfy inequality (30). In such situations, letting Φ equal this guess will give a closed-loop system that will converge linearly (at the rate of the slowest eigenvalue, that is the one closest to the unit circle) to the required solution x_∞. If no good guess of A^{-1} is available (as will be the case almost every time in practice), we can e.g. set $\Phi = \frac{(2-\epsilon)}{\overline{\sigma}(A)^2} A^*$ with $\epsilon = \frac{2}{1+\kappa(A)^2}$, as this choice will clearly satisfy condition (30). Though this is an appropriate choice, it suffers from a number of difficulties listed below:

1. Only *linear* rate of convergence for the sequence of residues $r_t = A\nu_t - b$ can be guaranteed and the rate depends on the slowest eigenvalue, that is the one closest to the unit circle.
2. All the eigenvalues of $A\Phi - I$ lie on the real line in the interval $[-1 + \epsilon, 1 - \epsilon]$. The factor $\epsilon = \frac{2}{1+\kappa(A)^2}$ was chosen so that the largest eigenvalue and the smallest eigenvalue are $A\Phi - I$ are equally away from the unit circle.
3. If the condition number $\kappa(A)$ of A is very large, then the smallest and the largest eigenvalues of $A\Phi - I$ become close to the unit circle, thus considerably slowing down convergence. Therefore, the algorithm works fast only for well-conditioned problems. This also suggests to use preconditioning techniques in order to improve overall convergence.

V. CONCLUSIONS

Based on elementary ideas from control theory, new iterative algorithms for solving linear systems of equations are introduced, that generalize the well known GMRES(m) solution method. The new algorithms have desirable properties, such as global convergence or more easily tunable control parameters. Note that choosing $\Xi = A\Phi$ in the feedback control scheme (21), as done in standard splitting method approaches, is only one possible choice. Finding control schemes that combine optimal performance with low computational complexity is an open research problem; our results can be regarded as first steps in this direction.

ACKNOWLEDGEMENTS

This work has been supported in parts by the German Research Foundation Grant DFG HE 1858/10-1 “KONNEW”, as well as by ARC Discovery-Projects Grant DP0342683 and National ICT Australia Ltd. National ICT Australia Ltd. is funded through the Australian Government’s *Backing Australia’s Ability* initiative, in part through the Australian Research Council.

REFERENCES

[1] M. Embree, “The tortoise and the hare: Restart gmres,” *SIAM Review*, vol. 45, pp. 1074–1109, 2003.

[2] S. Kashima, K. Ashida and Y. Yamamoto, “System theory for numerical analysis,” *preprint*, 2005.

[3] J. Jordan and U. Helmke, “Optimal control of iterative solution methods for linear systems of equations,” *Proc. Appl. Math. Mech.*, vol. 5, pp. 163–164, Mar. 2005.

[4] A. Greenbaum, *Iterative Methods for Solving Linear Systems*, ser. Frontiers in Applied Mathematics. SIAM, 1997, vol. 17.

[5] G. Opfer and G. Schober, “Richardson’s iteration for nonsymmetric matrices,” *Linear Algebra And Its Applications*, vol. 58, pp. 343–361, 1984.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore: John Hopkins University Press, 1989.

[7] R. M. Bitmead and M. Gevers, “Riccati difference and differential equations: convergence, monotonicity and stability,” in *The Riccati Equation*, W. Bittanti, Lamb, Ed. Springer-Verlag, 1991, pp. 263–291.

[8] P. Lancaster and R. Rodman, *The Algebraic Riccati Equation*. Oxford University Press, 1995.

[9] D. Hinrichsen and A. Pritchard, *Mathematical Systems Theory I*. Springer-Publ., 2005.