Linear Algebra and its Applications 443 (2014) 1–20

Contents lists available at ScienceDirect

Linear Algebra and its Applications

www.elsevier.com/locate/laa

On the stability of some algorithms for computing the action of the matrix exponential

Thomas M. Fischer

Department of Mathematics and Natural Sciences, University of Applied Sciences, D-64295 Darmstadt, Germany

ARTICLE INFO

Article history:
Received 3 September 2012
Accepted 5 November 2013
Available online 2 December 2013
Submitted by N.J. Higham

MSC:
65F60

Keywords:
Matrix exponential
One-step method
Rounding error analysis
Backward stability
Forward stability

ABSTRACT

This paper deals with the stability of some one-step methods for computing the action of a matrix exponential on a vector, i.e. $e^A b$, where A is an n-by-n matrix and b is a vector of dimension n. The methods are based on certain assumptions made for the case that A is a matrix having sufficiently small norm. If the norm of A is larger, a fixed step size is used to perform a scaling of A. An analysis of the roundoff error of the computed $e^A b$ shows that these methods are backward stable if A is Hermitian, and forward stable if, for example, A is skew-Hermitian or if A is real and essentially nonnegative and b is real and nonnegative. In addition, an upper bound on the forward error of the computed $e^A b$ is obtained for all A and b. This bound, which is expressed in terms of the condition number, is evaluated for several examples of the data. The results of this paper partly apply to the algorithm proposed by Al-Mohy and Higham (2011) for computing $e^A b$.

1. Introduction

The action of the exponential of a matrix $A \in \mathbb{C}^{n\times n}$ on a vector $b \in \mathbb{C}^n$, i.e. $e^A b$, is an operation, which plays a fundamental role in the solution of linear ordinary differential equations. It is well-known, for example, that the initial value problem

$$\frac{dx}{dt} = Ax, \qquad x(0) = b \tag{1.1}$$

has the solution $x(t) = e^{tA} b$ for $t \in \mathbb{R}$, which means that $e^A b$ solves (1.1) at $t = 1$. It has further been shown that some exponential integrator methods for the numerical solution of nonlinear initial value

E-mail address: thomas.fischer@h-da.de.

0024-3795/$ – see front matter
http://dx.doi.org/10.1016/j.laa.2013.11.003

problems can be based on this operation [4]. The computation of $e^A b$ can be regarded as a standard task in the field of matrix functions (see e.g. [16]).

A popular method for computing e^A is the scaling and squaring method (see [22,15] and the references therein). A version of this method is implemented in MATLAB. It therefore seems to be straightforward to evaluate $e^A b$ by computing e^A and multiplying the computed e^A into b. This approach is, however, not adequate if A is large and sparse or $e^{tA} b$ is needed for many values of t. In the former case, a computation of e^A should be avoided, since the exponential of a sparse matrix is generally full [22]. Krylov subspace techniques can be used in that case to reduce the $e^A b$ problem to a corresponding problem for a small dense matrix (see e.g. [12,17,10]). The algorithm, which has recently been proposed by Al-Mohy and Higham [4], is also suitable for computing $e^A b$ for general data A and b. Moreover, it allows an efficient evaluation of $e^{tA} b$ on grids of equally spaced values of t. The stability of this and related algorithms is the subject of the present analysis.

It was observed in numerical experiments that the mentioned algorithm by Al-Mohy and Higham as well as algorithms for computing e^A, for example the scaling and squaring method, behave in a forward stable manner for a great variety of the data (see [15] and [4] for the e^A and $e^A b$ problem, respectively). On the contrary, only little is known about the forward (or backward) stability of these algorithms in general. The most striking result is that of [5], where it was shown that the scaling and squaring method is forward stable for all matrices $A \in \mathbb{R}^{n\times n}$, which are normal or essentially nonnegative or for which a nonsingular diagonal matrix $D \in \mathbb{R}^{n\times n}$ exists so that $D^{-1}AD$ is essentially nonnegative. Note that a matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ is called *essentially nonnegative* if $a_{ij} \geqslant 0$ for $i \neq j$. In addition, bounds on the roundoff error of the computed e^A were obtained for all $A \in \mathbb{R}^{n\times n}$ [29,5]. However, these bounds were not related to the condition number of the e^A problem and are therefore not very informative.

The present paper deals with one-step methods of a certain type for computing $e^A b$. These methods use a fixed step size h to perform a scaling of A. More precisely, h is the reciprocal of a positive integer q, which is chosen so that hA has sufficiently small norm. Since $x_q = e^A b$ is the last element of the finite sequence $x_k = e^{hA} x_{k-1}$ for $k = 1, \ldots, q$, with $x_0 = b$, an approximation of the action of e^{hA} on a vector is needed for each step. The choice and algorithmic realization of this approximation determine the method. However, certain conditions, which ensure that the action of e^{hA} is always computed with high accuracy, are assumed to be satisfied. To some extent, the assumptions apply to the algorithm by Al-Mohy and Higham [4]. There the method of scaling allows larger step sizes than in the present case, provided that A is nonnormal. This makes the algorithm more efficient and often also more accurate.

Our aim is to study the effects of rounding errors on the accuracy of the computed result by using a backward and forward error analysis. It is shown that there are at least some classes of pairs (A, b), for which the algorithms of this paper are numerically stable:

If $A \in \mathbb{C}^{n\times n}$ is Hermitian and $b \in \mathbb{C}^n$ then the computed $e^A b$ has a normwise backward error, which is of the order $m(\epsilon)\epsilon$, where ϵ is the unit roundoff and $m(\epsilon)$ is a positive integer, which does not grow faster than a constant times $\log \frac{1}{\epsilon}$ as $\epsilon \to 0$.

If $A \in \mathbb{C}^{n\times n}$ is skew-Hermitian and $b \in \mathbb{C}^n$ or if $A \in \mathbb{R}^{n\times n}$ is essentially nonnegative and $b \in \mathbb{R}^n$ is a vector satisfying $b \geqslant 0$ or $b \leqslant 0$ then the computed $e^A b$ has a normwise forward error, which is of the order $m(\epsilon)\epsilon$ times the condition number of the $e^A b$ problem.

In addition, the normwise forward error of the computed $e^A b$ is estimated in terms of the condition number for all $A \in \mathbb{C}^{n\times n}$ and $b \in \mathbb{C}^n$. This estimate allows us to formulate a sufficient condition for the algorithms to behave in a forward stable manner. For many data A and b this condition is satisfied, but there are also cases where it is not.

The present analysis uses an approach, which is completely different from that of [5]. The classical concept of backward stability is weakened, if necessary, by enlarging the space of perturbations of A. It is shown that, for given $A \in \mathbb{C}^{n\times n}$ and $b \in \mathbb{C}^n$, the computed $e^A b$ can be interpreted as the exact solution of a perturbed initial value problem at $t = 1$, where this problem is obtained from (1.1) by replacing A and b by $A + E$ and $b + f$, respectively, with both E and $f \in \mathbb{C}^n$ having norm of order $m(\epsilon)\epsilon$. In general, the perturbation E of A is, however, a continuous function of $t \in [0, 1]$ with values in $\mathbb{C}^{n\times n}$ and not a constant matrix, as it is usually required. Backward stability in the strict sense is