Computers and Mathematics with Applications 72 (2016) 76–91

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Parallel matrix function evaluation via initial value ODE modeling

Jean-Paul Chehab [a,*], Madalina Petcu [b,c]

[a] LAMFA, UMR 7352 CNRS, Université de Picardie Jules Verne, 33 rue Saint Leu, 80039 Amiens, France

[b] UMR 7348 CNRS, Laboratoire de Mathématiques et Applications, Université de Poitiers, Boulevard Marie et Pierre Curie, Téléport 2, BP30179, 86962 Futuroscope Chasseneuil Cedex, France

[c] Institute of Mathematics of the Romanian Academy, Bucharest, Romania

ARTICLE INFO

Article history:
Received 4 June 2015
Received in revised form 20 March 2016
Accepted 23 April 2016
Available online 17 May 2016

Keywords:
Functions of matrices
ODEs
Parareal methods

ABSTRACT

The purpose of this article is to propose ODE based approaches for the numerical evaluation of matrix functions $f(A)$, a question of major interest in the numerical linear algebra. For that, we model $f(A)$ as the solution at a finite time T of a time dependent equation. We use parallel algorithms, such as the parareal method, on the time interval $[0, T]$ in order to solve the obtained evolution equation. When $f(A)$ is reached as a stable steady state, it can be computed by combining parareal algorithms and optimal control techniques. Numerical illustrations are given.

1. Introduction

The efficient numerical computation of matrix functions, as well as the solution of matrix polynomial equations (such as Riccati's), is currently an important topic in numerical linear algebra this kind of problem arises in a number of situations such as the approximation of nonlocal operators or of infinitesimal generators as well as in computational control, we refer to [1,2] for the computation of pth-roots of matrices, to [3] for the solution of rational matrix equations and to the book of Nick Higham [4] for a general presentation.

Let A be a matrix, to evaluate $f(A)$ or $f(A)b$ for a given vector b, a number of strategies have been developed: they often use an application of approximation theory techniques to a complex representation of f, defining rational approximation methods, let us cite the recent works of Frommer et al. [5] on rational Krylov methods and the references therein.

The use of efficient parallel computations of parabolic PDEs has been proposed in e.g. in [6] where the generator $\exp(tA)$ is decomposed as a sum of independent polar terms, and more recently, by Gander and Güttel [7] with the Paraexp algorithm combining Duhamel's principle and additive identities. The evaluation of $\exp(A)$ is an old problem, one can find a nice survey of methods in [8]. However, in a number of situations, the function f is not known so the tools of approximation theory cannot be applied.

In a more general way, differential equations can be used to identify, then to compute, the matrix function $f(A)$, as a state of the solution; it can be a solution at finite time as well as a stable steady state, when good stability properties are present.

* Corresponding author.
E-mail addresses: Jean-Paul.Chehab@u-picardie.fr (J.-P. Chehab), Madalina.Petcu@math.univ-poitiers.fr (M. Petcu).

http://dx.doi.org/10.1016/j.camwa.2016.04.036
0898-1221/

To compute $f(A)$ as the solution of an ODE at finite time T, $T = 1$ for simplicity, a homotopy method can be displayed as follows: let us consider the matrix differential equation

$$\begin{aligned} &\frac{dX}{dt} = \mathcal{F}(X(t)), \quad t \in (0,1), \\ &X(0) = X_0. \end{aligned} \tag{1}$$

We look for $\mathcal{F}$ such that $X(1) = f(A)$. We introduce the homotopy path $\mathcal{A}(t) = X_0 + t(A - X_0)$, we set $X(t) = f(\mathcal{A}(t))$; X_0 is chosen in such a way $f(X_0)$ is easy to compute. We have:

$$\begin{aligned} &\frac{dX}{dt} = \frac{d\mathcal{A}}{dt} f'(f^{-1}(X(t))), \quad t \in (0,1), \\ &X(0) = f(X_0). \end{aligned} \tag{2}$$

A numerical approximation of $f(A) = X(1)$ can be obtained by using any time marching scheme. For example, taking $X_0 = I$, where I is the identity matrix and $\mathcal{F}(X) = X(I - A)X$ we have $X(1) = A^{-1}$, so Forward Euler method as well as Runge Kutta method can be applied to build inverse preconditioners of A, see [9].

However, in a number of situations, it can be difficult to model $f(A)$ as the solution of the ODE at finite time and it is interesting to link $f(A)$ to a stable steady state, see [9,10]. Defining $X = f(A)$ as an asymptotically stable root of $\mathcal{F}$, say $\mathcal{F}(X) = 0 \iff X = f(A)$ and assuming that the eigenvalues of the differential of $\mathcal{F}$ at $f(A)$, $D\mathcal{F}(A)$, have a strictly negative real part, we can consider the differential equation

$$\begin{aligned} &\frac{dX}{dt} = \mathcal{F}(X(t)), \quad t > 0, \\ &X(0) = f(X_0). \end{aligned} \tag{3}$$

This modeling allows the computation of sparse approximations to $f(A)$ by defining an associate sparse matrix flow, see [9].

Hence, if a differential system possesses $\mathcal{F}(A)$ as an asymptotical steady state, it is possible to compute numerically $\mathcal{F}(A)$ by applying an explicit time marching scheme. The efficiency of the method depends both on the dynamical properties of the differential system and on the numerical scheme in time.

Parallel methods in time have been introduced and developed as a mean to solve time dependent problems using parallel computing, they apply to the computation of the solution of an ODE at a given finite time. These methods were especially devoted to the computation of $f(A)b$, where b is a given vector in $\mathbb{R}^n$. Let us cite the recent Paraexp and Rational Krylov methods [7,11], to name but a few. However, it must be noticed that these methods are direct but not well adapted to the computation of $f(A)$, furthermore they assume to have a knowledge of some rational expansions of $f(z)$, $z \in \mathbb{C}$ which is not always the case. The Parareal Method (PM) is iterative and it can be seen as a multi-steps shooting scheme, where each step can be solved in parallel. Firstly designed for parabolic-like problems, PM has been successfully adapted to second order evolutive PDE, see [12–14] but little attention was given for their application in numerical linear algebra and the main purpose of this article is to show that interesting issues can be derived for evaluating $f(A)$.

In this article, we consider a framework in which the parallel computation of functions of matrices can be applied, we focus on PM for which we propose adaptations and implementations for computing numerical approximations of matrix functions. The article is organized as follows: in Section 2 we recall some parallel in time methods including the classical parareal algorithm which we apply to solve Eq. (4) in order to compute the matrix function $f(A)$ of a matrix A. We numerically illustrate the algorithm by approximating the inverse and the exponential of a matrix. In Section 3, we propose a version of the modified parareal algorithm seen as a multiple shooting method as described in [14], adapted here to the matricial computations. Using this algorithm, we compute the cosine of a matrix and we compare the results obtained with this algorithm to the case when the cosine is obtained using the classical parareal algorithm. We can notice that the modified parareal algorithm produces an acceleration for the convergence. In Section 4 we consider the case when the matrix function is found as the stable steady state of an ordinary differential equation. We propose some methods allowing to accelerate the convergence in time, in order to efficiently compute the steady state. Section 5 presents a particular acceleration procedure in order to converge to the steady state. The idea is to obtain the matrix function as a solution of an optimal control problem and to apply a parareal in time method in order to solve the control problem, as proposed by [15].

All the numerical computations presented here have been made with Matlab.

2. Parallel algorithms applied to the computation of matrix functions

Before focusing on the application of the parareal method to the numerical evaluation of functions of matrices, we recall briefly some special parallel algorithms in time for solving numerically the equation:

$$\begin{aligned} &\frac{du}{dt} = Au + g, \\ &u(0) = u_0, \end{aligned} \tag{4}$$

and where one needs to be able to evaluate $\exp(tA)b$ for a certain given vector b.