Applied Mathematical Modelling 000 (2015) 1–8

Contents lists available at ScienceDirect

Applied Mathematical Modelling

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

Solving engineering models using hyperbolic matrix functions[☆]

Emilio Defez[a,*], Jorge Sastre[b], Javier Ibáñez[c], Jesús Peinado[c]

[a] Instituto de Matemática Multidisciplinar, Universitat Politècnica de València, Camino de Vera s/n, Valencia, 46022 Spain
[b] Instituto de Telecomunicaciones y Aplicaciones Multimedia, Universitat Politècnica de València, Camino de Vera s/n, Valencia, 46022 Spain
[c] Instituto de Instrumentación para Imagen Molecular, Universitat Politècnica de València, Camino de Vera s/n, Valencia, 46022 Spain

A R T I C L E I N F O

Article history:
Received 19 December 2013
Revised 31 July 2015
Accepted 23 September 2015
Available online xxx

Keywords:
Hermite matrix polynomial
Hyperbolic matrix functions
Series expansion

A B S T R A C T

In this paper a method for computing hyperbolic matrix functions based on Hermite matrix polynomial expansions is outlined. Hermite series truncation together with Paterson–Stockmeyer method allow to compute the hyperbolic matrix cosine efficiently. A theoretical estimate for the optimal value of its parameters is obtained. An efficient and highly-accurate Hermite algorithm and a MATLAB implementation have been developed. The MATLAB implementation has been compared with the MATLAB function `funm` on matrices of different dimensions, obtaining lower execution time and higher accuracy in most cases. To do this we used an NVIDIA Tesla K20 GPGPU card, the CUDA environment and MATLAB. With this implementation we get much better performance for large scale problems.

1. Introduction

Coupled partial differential problems are frequently found in many different fields of technology and science: elastic and inelastic contact problems of solids, biochemistry, magnetohydrodynamic flows, cardiology, diffusion problems, etc., see for example [1–4] and references therein. In particular, coupled hyperbolic systems appear in microwave heating processes [5] and optics [6] for instance.

In [7], an exact solution of coupled hyperbolic systems of the form

$$\left.\begin{array}{r} u_{tt}(x,t) - Au_{xx}(x,t) = 0, 0 < x < 1\ ,\ t > 0, \\ u(0,t) + B_1 u_x(0,t) = 0,\ t > 0, \\ A_2 u(1,t) + B_2 u_x(1,t) = 0,\ t > 0, \\ u(x,0) = f(x),\ 0 \le x \le 1, \\ u_t(x,0) = g(x),\ 0 \le x \le 1, \end{array}\right\} \tag{1.1}$$

where A, B_1, A_2, B_2 are $r \times r$ complex matrices, and u, f, g are r-vector valued functions, was constructed in terms of a series which used hyperbolic cosine and sine of a matrix, respectively defined by

$$\cosh(Ay) = \frac{e^{Ay} + e^{-Ay}}{2}, \sinh(Ay) = \frac{e^{Ay} - e^{-Ay}}{2}, A \in \mathbb{C}^{r \times r}. \tag{1.2}$$

☆ This work has been supported by Spanish Ministerio de Educación TIN2014-59294-P.
* Corresponding author. Tel.: +34963877663.
E-mail addresses: edefez@imm.upv.es (E. Defez), jorsasma@iteam.upv.es (J. Sastre), jjibanez@dsic.upv.es (J. Ibáñez), jpeinado@dsic.upv.es (J. Peinado).

http://dx.doi.org/10.1016/j.apm.2015.09.050

Functions of a matrix are used in many areas of science and arise in numerous applications in engineering [8]. In particular, matrix exponential e^A and matrix functions sine and cosine have been those that have received the most attention, see [9–14] for example, not only for its computational difficulties but also for its importance in solving differential systems of first and second order.

For the numerical solution of problem (1.1), analytic-numerical approximations are most suitably obtained by truncation of the exact series solution given in [7]. Thus, we need to compute approximations of both matrix functions hyperbolic cosine and sine with good accuracy and efficiency. It is well known that this computation can be reduced to the computation of the cosine of a matrix, due to the identities

$$\cosh(A) = \cos(iA), \quad \sinh(A) = i\cos\left(A - \frac{i\pi}{2}I\right),$$

but this approach has the disadvantage, however, to require complex arithmetic even though the matrix A is real, which is usually the case in applications. Furthermore, it is possible to reduce the computation of $\sinh(A)$ to the computation of $\cosh(A)$ by the relation

$$\sinh(A) = -i\cosh\left(-A - \frac{\pi}{2}iI\right), \tag{1.3}$$

but obviously, formula (1.3) also requires complex arithmetic although matrix A is real, which contributes substantially to the computational overhead. Direct calculation through the exponential matrix using (1.2) is also costly. A method based on Hermite series of the hyperbolic matrix cosine was recently presented in the *17th European Conference on Mathematics for Industry 2012* (Lund, Sweden) and also published in [15]. In this paper a bound of the error was given, however, an algorithm for its calculation was not discussed. In the *Mathematical Modelling in Engineering* & *Human Behaviour 2013* Conference(Valencia, Spain) [16] this same idea was applied to compute the sine and hyperbolic cosine of a matrix, finding different bounds, more accurate than those obtained in [15]. An algorithm based on the recurrence relation of three terms was developed, which allowed the simultaneous calculation of both functions. The proceedings of that conference were published as a book chapter [17] and the summaries were published in [16,18].

In this work, we propose an algorithm to compute both matrix functions, $\sinh(A)$ and $\cosh(A)$, simultaneously, avoiding complex arithmetic whenever possible, following a similar approach to that used in [14] for computing approximations of the matrix cosine. The proposed method uses Hermite matrix polynomial expansions of both matrix functions in order to provide a very accurate and competitive method for computing them. For this purpose, firstly we have to determinate error bounds of the approximation error of both functions using a series of Hermite matrix polynomials, bounds different from and more accurate than the bounds presented in [15,17]. Then, we have developed an algorithm based on Paterson–Stockmeyer method [19], instead of the algorithm based on the recurrence relation of three terms, in which the corresponding scaling of the matrix and the order of the approximation have been determined from the bound expression. To analyze the accuracy of the proposed method, implementations have been developed in MATLAB . We have also performed numerous tests to compare this algorithm with other state-of-the-art algorithms (`funm` MATLAB function). Finally, high performance implementations have been developed on large dimension matrix problems by using GPGPUs cards.

This work is organized as follows. Sections 2 and 3 summarize previous results of Hermite matrix polynomials and include a Hermite series expansion of the matrix hyperbolic cosine and sine with the respectively error bounds. An algorithm for the proposed method is given in Section 4. Section 5 deals with several numerical tests in order to investigate the accuracy of the proposed method and the method behavior for large scale problems. Both of them were done with MATLAB. When evaluating the performance of the method for large scale problems, we also did a GPGPU (General Purpose Graphics Processing Unit) implementation. The idea is to compare the CPU vs GPGPU implementations.

In this paper, $\lceil x \rceil$ rounds to the nearest integer greater than or equal to x. The matrices I_r and $\theta_{r\times r}$ in $\mathbb{C}^{r\times r}$ denote the matrix identity and the null matrix of order r, respectively. We will use subordinate matrix norms $\| A \|$, $A \in \mathbb{C}^{r\times r}$, and $\| A \|_1$ denotes the usual 1-norm. If $\mathcal{A}(k,n)$ is a matrix in $\mathbb{C}^{r\times r}$ for $n \geq 0$, $k \geq 0$, then [20]:

$$\sum_{n\geq 0}\sum_{k\geq 0}\mathcal{A}(k,n) = \sum_{n\geq 0}\sum_{k=0}^{n}\mathcal{A}(k,n-k). \tag{1.4}$$

2. Hermite matrix polynomial series expansions of hyperbolic matrix cosine: error bound

The following properties of Hermite matrix polynomials which have been established in [13,20,21] will be used in this paper. From (3.4) of [21, p. 25] the nth Hermite matrix polynomial is defined by

$$H_n\left(x, \frac{1}{2}A^2\right) = n!\sum_{k=0}^{\left[\frac{n}{2}\right]}\frac{(-1)^k (xA)^{n-2k}}{k!(n-2k)!}, \tag{2.1}$$

for an arbitrary matrix A in $\mathbb{C}^{r\times r}$. From [13], we obtain

$$\cosh(Ay) = e^{\frac{1}{\lambda^2}}\sum_{n\geq 0}\frac{1}{\lambda^{2n}(2n)!}H_{2n}\left(y\lambda, \frac{1}{2}A^2\right), \ \lambda \in \mathbb{R}^+. \tag{2.2}$$

Please cite this article as: E. Defez et al., Solving engineering models using hyperbolic matrix functions, Applied Mathematical Modelling (2015), http://dx.doi.org/10.1016/j.apm.2015.09.050