A family of Adams exponential integrators for fractional linear systems[☆]

Roberto Garrappa

Università degli Studi di Bari, Dipartimento di Matematica
Via E. Orabona n.4, 70125 Bari, Italy

Abstract

The numerical solution of linear time–invariant systems of fractional order is investigated. We construct a family of exponential integrators of Adams type possessing good convergence and stability properties. The methods are devised in order to keep at a suitable level the computational effort necessary to solve problems of large size. Numerical experiments are provided to validate the theoretical results; the effectiveness of the proposed approach is tested and compared to some other classical methods.

Keywords: fractional differential equations, numerical methods, exponential integrators, Adams methods, Mittag–Leffler function
2000 MSC: 34A08, 65L06, 33E12, 65F60

1. Introduction

In several applications there are frequently involved linear time–invariant (LTI) systems of fractional order in the form

$$\begin{aligned} \frac{d^\alpha}{dt^\alpha} x(t) &= Ax(t) + bu(t) \\ y(t) &= c^T x(t) \end{aligned} \quad , \tag{1}$$

[☆]This work is supported by the project "High-performance robust fractional controllers for applications in industry and mechatronics", which is part of the national project PRIN 2009F4NZJP "Non integer order systems in modelling and control", funded by the Italian Ministry of University and Research (MIUR).

Email address: garrappa@dm.uniba.it, roberto.garrappa@uniba.it (Roberto Garrappa)

where α is any real positive number, $A \in \mathbb{R}^{N\times N}$, $b \in \mathbb{R}^{N\times N_i}$ and $c \in \mathbb{R}^{N\times N_o}$ are constant and $x(t)$, $u(t)$ and $y(t)$ are vector–valued functions mapping the interval $[0, +\infty)$ respectively into $\mathbb{R}^N$, $\mathbb{R}^{Ni}$ and $\mathbb{R}^{No}$.

Throughout the paper, the fractional derivative, with respect to the origin, is introduced according to the Caputo's definition

$$\frac{d^\alpha}{dt^\alpha}x(t) = \frac{1}{\Gamma(\lceil\alpha\rceil - \alpha)}\int_0^t \frac{x^{(\lceil\alpha\rceil)}(\tau)}{(t-\tau)^{\alpha+1-\lceil\alpha\rceil}}d\tau,$$

where $\Gamma(\cdot)$ is the Euler gamma function, $\lceil\alpha\rceil$ is the smallest integer such that $\alpha < \lceil\alpha\rceil$ and $x^{(\lceil\alpha\rceil)}$ denotes the standard derivative of integer order $\lceil\alpha\rceil$. This approach is the most natural in real–life applications since it allows to couple system (1) with classical initial conditions of Cauchy type

$$x^{(\nu)}(0) = x_{0,\nu}, \quad \nu = 0, \ldots, \lceil\alpha\rceil - 1.$$

We refer, for instance, to [1, 2, 3] for basic materials on fractional calculus.

Systems of this kind are frequently encountered in control theory; in this case $x(t)$ represents the state or pseudo-state variables and $u(t)$ and $y(t)$ denote respectively the input and output of the system. We are interested in problems with an highly internal complexity in which not only N is very large but it is also several orders of magnitude larger than N_i and N_o. The case $N_i = N_o = 1$ is also of practical interest. For instance, models with a very high degree of complexity have been proposed in the fields of electrical networks, power systems and smart grids [4]. Very often problems of this kind naturally originate from models described by partial differential equations (PDEs) with fractional–time derivatives after the discretization of 2D and 3D spatial derivatives by some suitable techniques (usually finite elements or finite differences); in these instances the number of internal states can vary from 10^3 to 10^6 or even 10^9.

The numerical solution of systems of fractional differential equations (FDEs), especially of very large size, is usually very demanding. Indeed, derivative operators of non–integer order are non–local operators and, consequently, the time–step integration involves, at each step, the use of all the previously evaluated values, with an increasing computational effort as the integration process moves forward. Furthermore, with *stiff* problems some restrictive bounds apply to the step–size in order to avoid errors for propagating very quickly.

Because of the combined effects of long–term memory and use of small step–sizes, the computationa can become prohibitive for practical purposes.

By chance, in most applications the main interest is not devoted to study (and compute) the evolution of the state $x(t)$, but rather to measure the response of the output $y(t)$ under the effects of a certain excitation $u(t)$. Especially with systems in which N_i and N_o are reasonably smaller than N, our ideas is that some advantages can be taken by directly evaluating the output $y(t)$ from the input $u(t)$ without explicitly calculating $x(t)$.

In this paper we introduce a new family of methods of Adams type specifically designed for numerically solving fractional LTI systems in order to achieve the following two main goals: *1)* providing highly accurate solutions with stable behavior without imposing severe restrictions on the step–size even in the presence of stiff problems; *3)* keeping the computational effort at a reasonable level by performing most of the computation in a space of size proportional to N_i and N_o instead of N.

The framework in which we will be operating is the exponential integrators. This is a class of numerical methods which has been widely used and investigated for ordinary differential equations (ODEs); their use within fractional calculus is however a rather unexplored area. To achieve the above mentioned goals, exponential integrators will be combined with Krylov subspace methods and rational approximations.

This paper is organized as follows: in Section 2 we review some basic facts about exponential integrators for ODEs and Adams methods for FDEs. Moreover we introduce a variation of constant formula for FDEs in order to describe in details, in Section 3, the way in which exponential integrators of Adams type are generalized to FDEs; convergence properties are also studied. In Section 4 we address the problem of evaluating, in an efficient way, the convolution weights (which are special functions with matrix arguments) and we discuss some important issues related to the treatment of large size problems. Finally, in Section 5 we present some numerical experiments to validate the theoretical findings and show the effectiveness of the proposed approach by means of some comparisons with other existing methods.

2. Preliminaries

For the sake of clarity and completeness, we briefly review some basic material which will be used later in the paper. We will include in this section just the essential notions necessary for the subsequent analysis and we will provide some references for the related topics.

2.1. Exponential integrators for ODEs

Exponential integrators are a class of powerful methods specifically designed for solving semilinear ODEs; basically, the linear term is separated and solved by a matrix exponential and a time–stepping technique is applied to the non linear term. The key tool in this process is the classical variation–of–constant formula for ODEs.

The main advantage is that once the linear term (which usually is stiff) has been solved exactly, the restrictions on the step–size due to stability requirements are greatly relaxed. We do not discuss this technique in more details since the way in which exponential integrators are devised will be clear when, in the next section, we will face with their generalization to FDEs; we just refer to the review in [5] and to [6, 7, 8] for specific results on exponential integrators of Adams type for ODEs.

2.2. Adams methods for FDEs

A classical approach for the time simulation of the FDE in (1) starts from the integral representation of the true solution

$$x(t) = \sum_{\nu=0}^{\lceil\alpha\rceil-1} \frac{t^\nu}{\nu!} x_{0,\nu} + \int_0^t \frac{(t-\tau)^{\alpha-1}}{\Gamma(\alpha)} (Ax(\tau) + bu(\tau)) d\tau.$$

On a uniform grid–mesh with constant spacing $h > 0$, the vector field $Ax(t) + bu(t)$ is approximated by a suitable piecewise polynomial and the resulting integrals are exactly evaluated. This technique, which is a generalization of classical Adams multistep methods for ODEs, usually goes under the name of Product Integration (PI) rules [9].

Unfortunately, this approach suffers from the same stability limitations of Adams methods for ODEs: for instance, it has been proved in [10] that for $\alpha = \frac{1}{2}$ stability is unconditionally guaranteed only for methods based on polynomials of degree not greater than 2 (see also [11]).

Because of these limitations, methods grounded on high order polynomials are usually not suitable for numerical computation and the rather low–order PI trapezoidal rule (based on first degree polynomials) is one of the most popular.

Based on the work of [12], some other generalizations of Adams multistep methods to FDEs have been investigated, for instance, in [13] and [14].

2.3. True solution and a generalized Mittag–Leffler function

To generalize exponential integrators to fractional problems, we preliminarily derive a variation–of–constant formula for problem (1). By rewriting (1) in the Laplace domain

$$s^\alpha X(s) - \sum_{\nu=0}^{\lceil\alpha\rceil-1} s^{\alpha-\nu-1}x_{0,\nu} = AX(s) + bU(s), \quad Y(s) = c^T X(s),$$

where $X(s)$, $U(s)$ and $Y(s)$ are the Laplace transforms of $x(t)$, $u(t)$ and $y(t)$ respectively, and solving with respect to $X(s)$, we obtain

$$Y(s) = c^T \sum_{\nu=0}^{\lceil\alpha\rceil-1} (s^\alpha I - A)^{-1}x_{0,\nu} + c^T(s^\alpha I - A)^{-1}bU(s).$$

By turning back to the temporal domain, the output $y(t)$ can hence be written as

$$y(t) = c^T \sum_{\nu=0}^{\lceil\alpha\rceil-1} e_{\alpha,\nu+1}(t; -A)x_{0,\nu} + I_\alpha(t), \tag{2}$$

where

$$I_\alpha(t) = \int_0^t c^T e_{\alpha,\alpha}(t-\tau; -A)bu(\tau)d\tau \tag{3}$$

and $e_{\alpha,\beta}(t; z)$ is the inverse of the Laplace transform of $s^{\alpha-\beta}/(s^\alpha + z)$.

The function $e_{\alpha,\beta}(t; z)$ is a generalization of the Mittag–Leffler (ML) function and plays an important role in fractional calculus; it has been investigated, for instance, in [15, 16] and fully discussed in [2]. It is possible to write $e_{\alpha,\beta}(t; z)$ in terms of the ML function with two parameters

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)}$$

according to

$$e_{\alpha,\beta}(t; z) = t^{\beta-1}E_{\alpha,\beta}(-t^\alpha z). \tag{4}$$

For $\alpha = \beta = 1$ it is easy to note that $e_{\alpha,\beta}(t; z)$ is the exponential e^{-tz}.

The extension of $e_{\alpha,\beta}(t; z)$ to matrix arguments to be used in (2) and (3) is quite straightforward; it can be done by means of the formula for the inversion of the Laplace transform

$$e_{\alpha,\beta}(t; A) = \frac{1}{2\pi i}\int_{Br} e^{st}s^{\alpha-\beta}(s^\alpha I + A)^{-1}ds,$$

where I stands for the identity matrix and Br is the Bromwich contour lying to the right of the region of analyticity of the integrand.

In the following we will make use of the following property of the function $e_{\alpha,\beta}(t;z)$ which holds also for matrix arguments; the proof can be easily carried out by working in the Laplace domain.

Lemma 1. *Let $a < t$, $\alpha, \beta > 0$ and $r \in \mathbb{R}$ such that $r > -1$. Then*

$$\frac{1}{\Gamma(r+1)} \int_a^t e_{\alpha,\beta}(t-s;z)(s-a)^r ds = e_{\alpha,\beta+r+1}(t-a;z).$$

By using the series representation of $E_{\alpha,\beta}(z)$ it is possible, for any $h > 0$, to provide the following scaling

$$e_{\alpha,\beta}(t;\lambda) = h^{\beta-1} e_{\alpha,\beta}(t/h; h^\alpha \lambda), \tag{5}$$

previously introduced in [17] and which will come in useful later.

3. Exponential integrators of Adams type for FDEs

A family of exponential integrators for fractional problems can be derived in a similar way as for Adams multistep methods. Given a uniform partition $t_n = nh$, $h > 0$, on the interval of integration $[0, T]$, the function $u(t)$ in (3) is approximated, in each interval $[t_n, t_{n+1}]$, by the polynomial of degree $k-1$

$$r_n(t_n + \theta h) = \sum_{\ell=0}^{k-1} (-1)^\ell \binom{-\theta+1}{\ell} \nabla^\ell u_{n+1},$$

where $u_n = u(t_n)$, $\nabla^\ell u_n$ are the classical backward differences of order ℓ

$$\begin{aligned} &\nabla^0 u_n = u_n \\ &\nabla^\ell u_n = \nabla^{\ell-1} u_n - \nabla^{\ell-1} u_{n-1}, \quad \ell = 1, 2, \dots \end{aligned}$$

and the binomial coefficients are defined in the usual way

$$\binom{-\theta+1}{\ell} = \begin{cases} 1, & \ell = 0 \\ \frac{(-\theta+1)(-\theta)\cdots(-\theta-\ell+2)}{\ell!}, & \ell \geq 1. \end{cases}$$

Clearly, $r_n(\tau)$ is the interpolant polynomial for $u(t)$ on $t_{n-k}, \dots, t_n, t_{n+1}$. When $k \geq 2$, to provide suitable starting values the same polynomial r_{k-2} can be used to approximate the input u in the first $k-1$ subintervals $[t_j, t_{j+1}]$, $j = 0, 1, \dots, k-2$.

By rewriting the integral (3) at $t = t_n$ in a piecewise way and adopting in each subinterval the change of variable $\tau = t_j + \theta h$, one can get

$$I_\alpha(t_n) = c^T h \sum_{j=0}^{n-1} \int_0^1 e_{\alpha,\alpha}(t_n - t_j - \theta h; -A) b u(t_j + \theta h) d\theta.$$

Now, the polynomial replacement $u(t_j + \theta h) \approx r_j(t_j + \theta h)$, together with (5), allows to express the approximation $y_n \approx y(t_n)$ of the output of the LTI system by means of the following k–step method

$$\begin{aligned} y_n = c^T & \sum_{\nu=0}^{\lceil\alpha\rceil-1} e_{\alpha,\nu+1}(t_n; -A) x_{0,\nu} + \\ & h^\alpha \sum_{\ell=0}^{k-1} \left(\sum_{j=0}^{k-2} \varphi_{\alpha,\ell}(n-j; -h^\alpha A) \nabla^\ell u_{k-1} + \sum_{j=k-1}^{n-1} \varphi_{\alpha,\ell}(n-j; -h^\alpha A) \nabla^\ell u_{j+1} \right), \end{aligned} \tag{6}$$

where $\varphi_{\alpha,\ell}(n; A) = c^T \hat{e}_\alpha^{(\ell)}(n; A) b$ and

$$\hat{e}_\alpha^{(\ell)}(n; A) = (-1)^\ell \int_0^1 \binom{-\theta + 1}{\ell} e_{\alpha,\alpha}(n - \theta; A) d\theta.$$

Remark 1. The term k–step is adopted for (6) by analogy with classical multistep methods and refers, in some way, more to the degree of the polynomial rather than to the actual number of time–steps. Indeed, due to the convolution structure of (6), all the past memory is involved in the computation of each approximation y_n. Memory preservation is a typical feature of fractional order problems which makes the implementation of methods for FDEs more demanding with respect to ODEs. For our integrators we can see the memory preservation as strictly related to the lack of the semigroup property of the ML function; indeed, unless $\alpha = \beta = 1$, it is $e_{\alpha,\beta}(t + s; \lambda) \neq e_{\alpha,\beta}(t; \lambda) e_{\alpha,\beta}(s; \lambda)$.

To achieve a compact representation of the weights $\varphi_{\alpha,\ell}(n; -h^\alpha A)$ we adopt the following notational convention: given an ℓ–degree polynomial $P_\ell(x) = p_0 + p_1 x + \ldots + p_\ell x^\ell$, the associated ML function $e_\alpha^{[P_\ell]}(t; z)$, for $t \in \mathbb{R}^+$ and z any (possibly matrix) argument, is defined as

$$e_\alpha^{[P_\ell]}(t; z) = \sum_{k=0}^{\ell} p_k e_{\alpha,\alpha+k+1}(t; z).$$

Proposition 2. *The weights $\varphi_{\alpha,\ell}(n;-h^\alpha A)$ of the convolution quadrature (6) are given by*

$$\varphi_{\alpha,\ell}(n;-h^\alpha A) = c^T(e_\alpha^{[P_\ell]}(n;-h^\alpha A) - e_\alpha^{[Q_\ell]}(n-1;-h^\alpha A))b$$

with $P_\ell(x)$ and $Q_\ell(x)$ some suitable ℓ–degree polynomials.

PROOF. We first split the integral in the definition of $\hat{e}_\alpha^{(\ell)}(n;A)$ as $\int_0^1 = \int_0^n - \int_1^n$ and hence, after observing that the binomial coefficient $\binom{-\theta+1}{\ell}$ is a polynomial in θ, the repeated application of Lemma 1 allows to conclude the proof. □.

It is an elementary task to verify that the first few polynomials $P_\ell(x)$ and $Q_\ell(x)$ of Proposition 2 are those listed in Table 1.

ℓ	$P_\ell(x)$	$Q_\ell(x)$
0	1	1
1	$x-1$	x
2	$x^2-\frac{1}{2}x$	$x^2+\frac{1}{2}x$
3	$x^3-\frac{1}{6}x$	$x^3+x^2+\frac{1}{3}x$
4	$x^4+\frac{1}{2}x^3-\frac{1}{12}x^2-\frac{1}{12}x$	$x^4+\frac{3}{2}x^3+\frac{11}{12}x^2+\frac{1}{4}x$

Table 1: First polynomials $P_\ell(x)$ and $Q_\ell(x)$ for $\ell = 0,1,\ldots,4$ of Proposition 2.

Remark 2. Methods (6) generalize to FDEs the exponential integrators of Adams type for ODEs investigated in [7]. Despite methods for ODEs, which are based on the exponential and some related functions, the counterparts for FDEs involve ML functions. However, for convenience, we will continue to use the term exponential also to denote integrators (6), although this could appear somewhat inappropriate in this context.

One of the main advantages of (6) is that the linear term $Ax(t)$ of the LTI system (1) is solved in a nearly exact way (we will address this issue in the next section) by the generalized ML functions $e_{\alpha,\beta}$ in the convolution weights $\varphi_{\alpha,\ell}(n;-h^\alpha A)$; in this way the adverse effects of possibly stiff matrices A are substantially discharged, thus providing a family of methods which do not suffer from restrictions on the step–size due to stability requirements.

We highlight that the behavior of convolution quadratures (6) strictly depends on the degree of the polynomials used to approximate the function

$u(t)$. To make an error analysis and study convergence properties we first introduce the following auxiliary result.

Lemma 3. *Let $f(x) = (1-x)^\beta h(x)$, where $\beta > -1$ and $h(x)$ is continuous in $[0,1]$. The general trapezoidal quadrature rule*

$$R^{[n,\delta]} f = \frac{1}{n} \sum_{j=1}^{n} f\Big(\frac{2j-1+\delta}{2n}\Big), \quad |\delta| < 1,$$

approximates the integral $If = \int_0^1 f(t)dt$ with an error

$$E^{[n,\delta]} f = R^{[n,\delta]} f - If = -\frac{\delta}{2} h(0) n^{-1} + \mathcal{O}(n^{-\beta-1}).$$

PROOF. This result is obtained from [18, Eq. (7.3)] after applying formula (10.02.03.0027.01) in [19].

To lighten the notation, we will neglect in the remainder of this section the contribution of the constant vectors b and c. It is clear that this simplification does not affect in any way the convergence properties of the method under investigation and the generalization to the standard case is possible in a straightforward way.

Theorem 4. *Let $u \in \mathcal{C}^k([0,T])$. The error $E_n = y(t_n) - y_n$ of the k–step Adams exponential integrator (6) is given by*

$$E_n = C_k h^k e_{\alpha,\alpha+1}(t_n; -A) u^{(k)}(\eta) + \mathcal{O}(h^{k+\alpha})$$

for $\eta \in (0, t_n)$ and C_k a constant which does not depend on h and n.

PROOF. We preliminary observe that

$$E_n = \int_{t_0}^{t_{k-1}} e_{\alpha,\alpha}(t_n - \tau; -A) R_{k-2}(\tau) d\tau + \sum_{j=k-1}^{n-1} \int_{t_j}^{t_{j+1}} e_{\alpha,\alpha}(t_n - \tau; -A) R_j(\tau) d\tau,$$

where the first summand vanishes for $k = 1$. $R_j(\tau) = u(\tau) - r_j(\tau)$ is the remainder of the polynomial interpolation at $\tau \in [t_j, t_{j+1}]$ and can be written as

$$R_j(\tau) = u(\tau) - r_j(\tau) = \frac{\omega_{j,k-1}(\tau)}{k!} u^{(k)}(t_{j+1} - \xi_j(\tau) h), \quad \xi_j(\tau) \in [0,1],$$

with $\omega_{j,k-1}(\tau) = (\tau - t_{j+1})(\tau - t_j)\cdots(\tau - t_{j-k+2})$. For the first summand (when $k \geq 2$), $R_{k-2}(\tau)$ is evaluated in $[0, t_{k-1}]$ and hence $\xi_{k-2}(\tau) \in [0, k-1]$. We can write now

$$\begin{aligned} E_n &= \sum_{j=0}^{k-2} \int_{t_j}^{t_{j+1}} e_{\alpha,\alpha}(t_n - \tau; -A) \frac{\omega_{k-2,k-1}(\tau)}{k!} u^{(k)}(t_{k-1} - \xi_{k-2}(\tau)h) d\tau + \\ &\sum_{j=k-1}^{n-1} \int_{t_j}^{t_{j+1}} e_{\alpha,\alpha}(t_n - \tau; -A) \frac{\omega_{j,k-1}(\tau)}{k!} u^{(k)}(t_{j+1} - \xi_j(\tau)h) d\tau \end{aligned}$$

and, by means of the change of variable $\tau = t_j + \theta h$, the use of the scaling (5) together with the Taylor's formula, we obtain

$$E_n = h^{k+\alpha} \sum_{j=0}^{n-1} \int_0^1 e_{\alpha,\alpha}(n-j-\theta; -h^\alpha A) \frac{\omega_{k-1}(\theta)}{k!} u^{(k)}(t_j + \theta h) d\theta + \mathcal{O}(h^{k+1+\alpha})$$

with $\omega_{k-1}(\theta) = (\theta - 1)\theta \cdots (\theta + k - 2)$. Denote now for shortness the function $\bar{u}(s) = u^{(k)}(sh)$ and, after using the series expansion defining the ML function

$$e_{\alpha,\alpha}(n - j - \theta; -h^\alpha A) = \sum_{\ell=0}^{\infty} \frac{(n-j-\theta)^{\alpha(\ell+1)-1}(-1)^\ell h^{\alpha\ell} A^\ell}{\Gamma(\alpha\ell + \alpha)},$$

we are able to write the error of the method as

$$E_n = \frac{h^{k+\alpha}}{k!} \sum_{\ell=0}^{\infty} \frac{(-1)^\ell h^{\alpha\ell} A^\ell}{\Gamma(\alpha\ell + \alpha)} \int_0^1 \omega_{k-1}(\theta) \sum_{j=0}^{n-1} (n-j-\theta)^{\alpha(\ell+1)-1} \bar{u}(j+\theta) d\theta + \mathcal{O}(h^{k+1+\alpha})$$

Consider the function $f(t) = (1-t)^\beta \bar{u}(nt)$ and its integral in $[0, 1]$

$$If = \int_0^1 f(t) dt = \frac{1}{n^{\beta+1}} \int_0^n (n - \tau)^\beta \bar{u}(\tau) d\tau$$

which can be approximated by the general trapezoidal quadrature rule $R^{[n,\delta]} f$ of Lemma 3

$$R^{[n,\delta]} f = \frac{1}{n^{\beta+1}} \sum_{j=0}^{n-1} (n - j - \theta)^\beta \bar{u}(j + \theta)$$

for $\delta = 2\theta - 1$. By the expansion of the error in Lemma 3 it is immediate to see that

$$\sum_{j=0}^{n-1} (n - j - \theta)^\beta \bar{u}(j + \theta) = \int_0^n (n - \tau)^\beta \bar{u}(\tau) d\tau + n^\beta \Big(\frac{1}{2} - \theta\Big) \bar{u}(0) + C_\beta,$$

for C_β a constant depending on β and on the value of $\bar{u}(0)$ but not on n. We can use this result in the expansion of the error E_n in order to write

$$\begin{aligned} y(t_n) - y_n &= \frac{h^{k+\alpha}}{k!} \sum_{\ell=0}^{\infty} \frac{(-1)^\ell h^{\alpha\ell} A^\ell}{\Gamma(\alpha\ell+\alpha)} \bar{C}_k \int_0^n (n-\tau)^{\alpha(\ell+1)-1} \bar{u}(\tau) d\tau \\ &+ \frac{h^{k+\alpha}}{k!} \sum_{\ell=0}^{\infty} \frac{(-1)^\ell h^{\alpha\ell} A^\ell}{\Gamma(\alpha\ell+\alpha)} n^{\alpha(\ell+1)-1} \bar{u}(0) \tilde{C}_k + \mathcal{O}(h^{k+\alpha}) \end{aligned}$$

where for shortness we put

$$\bar{C}_k = \int_0^1 \omega_{k-1}(\theta) d\theta, \quad \tilde{C}_k = \int_0^1 \omega_{k-1}(\theta) \Big(\frac{1}{2} - \theta\Big) d\theta.$$

By means of the mean value theorem we obtain for $\eta \in (0, n)$

$$\begin{aligned} y(t_n) - y_n &= \frac{h^k}{k!} \bar{C}_k \sum_{\ell=0}^{\infty} \frac{(-1)^\ell h^{\alpha\ell+\alpha} n^{\alpha\ell+\alpha} A^\ell}{\Gamma(\alpha\ell+\alpha+1)} \bar{u}(\eta) \\ &+ \frac{h^{k+\alpha}}{k!} \sum_{\ell=0}^{\infty} \frac{(-1)^\ell h^{\alpha\ell} A^\ell}{\Gamma(\alpha\ell+\alpha)} n^{\alpha(\ell+1)-1} \bar{u}(0) \tilde{C}_k + \mathcal{O}(h^{k+\alpha}) \end{aligned}$$

from which it is immediate to see that

$$y(t_n) - y_n = \frac{h^k}{k!} \bar{C}_k e_{\alpha,\alpha+1}(t_n; -A) \bar{u}(\eta) + \frac{h^{k+1}}{k!} \tilde{C}_k e_{\alpha,\alpha}(t_n; -A) \bar{u}(0) + \mathcal{O}(h^{k+\alpha})$$

which concludes the proof after putting $C_k = \bar{C}_k / k!$. □

Remark 3. Formulas (6) can be applied, with simple modifications, also to semilinear problems in which the input depends on the state variables, i.e. $u(t) = u(t, y(t))$. However, in this case the convergence properties stated in Theorem 4 are no longer valid since the smoothness hypothesis can not be guaranteed on the whole interval of integration (see the analysis in [20]). We think to deepen this issue in a forthcoming paper.

4. Evaluation of the convolution weights

The main computational challenge in the implementation of the integrators described in the previous section is the evaluation of the weights $\varphi_{\alpha,\ell}(n; -h^\alpha A)$ involving the computation of generalized ML functions with matrix arguments.

In the past most of the attention has been devoted to the numerical approximation of exponential and related functions; see, for instance, the classical book [21] or some more specific results in [22, 23, 24, 25, 26]. Only recently, an increasing interest is focused also upon the approximation of ML functions with matrix arguments [27, 28, 29, 30].

Since a direct evaluation of this special function is not reliable due to the slow convergence of the series in $E_{\alpha,\beta}(z)$, more effective techniques, for instance methods based on the numerical inversion of the Laplace transform, need to be used.

4.1. Numerical quadratures with rational approximations

Given a matrix A, which for simplicity we assume possessing the spectrum on the negative real line, by putting $z = sj$, we can rewrite the integral representation of $e_{\alpha,\beta}(j; -h^\alpha A)$ as

$$e_{\alpha,\beta}(j; -h^\alpha A) = \frac{j^{-1}}{2\pi i} \int_{\mathcal{C}} e^z \frac{z^{\alpha-\beta}}{j^{\alpha-\beta}} \left(\frac{z^\alpha}{j^\alpha} I - h^\alpha A \right)^{-1} dz, \tag{7}$$

where $\mathcal{C}$ is a suitable deformation of the Bromwich line.

To numerically evaluate the integral (7), as proposed in [31] and already successfully employed for FDEs in [32], we replace the exponential function e^z in (7) by a $(K-1, K)$ rational approximation $\mathcal{R}_K(z) = \sum_{k=1}^{K} r_k/(z - z_k)$; after assuming the contour $\mathcal{C}$ winding in the negative sense once around each pole z_k, we make use of the Cauchy's integral formula to provide an approximation of $e_{\alpha,\beta}(j; -h^\alpha A)$ in the form

$$e^{[K]}_{\alpha,\beta}(j; -h^\alpha A) = -j^{-1} \sum_{k=1}^{K} r_k \frac{z_k^{\alpha-\beta}}{j^{\alpha-\beta}} \left(\frac{z_k^\alpha}{j^\alpha} I - h^\alpha A \right)^{-1}.$$

For the computation of weights $\varphi_{\alpha,\ell}(j; -h^\alpha A)$ an immediate approach would be the repeated application of the above approximation for several values of β. This approach anyway would be highly inefficient since it involves a large amount of computation with matrices of large size N. It is preferable to move the largest possible part of the computation to a scalar level and confine, at the smallest possible level, the number of operations involving large matrices.

By applying Proposition 2, it is straightforward to see that the corresponding approximations for $\varphi_{\alpha,\ell}(j; -h^\alpha A)$ are hence given by

$$\varphi^{[K]}_{\alpha,\ell}(j; -h^\alpha A) \;=\; \sum_{k=1}^{K} \frac{r_k}{z_k} Q_\ell\left(\sigma_{k,j-1}\right) c^T y_{k,j-1}$$

$$-\sum_{k=1}^{K} \frac{r_k}{z_k} P_\ell\left(\sigma_{k,j}\right) c^T y_{k,j}, \tag{8}$$

where $\sigma_{k,j} = j/z_k$, P_ℓ and Q_ℓ are the polynomials introduced in Proposition 2 (and explicitly given below it) and each vector $y_{k,j}$ is solution of the system

$$\left(\frac{1}{\sigma_{k,j}^{\alpha}} I - h^\alpha A\right) y = b. \tag{9}$$

The accuracy of the approximation provided by (8) strictly depends on the type and on the degree of the approximation $\mathcal{R}_K(z)$ used for e^z. It is quite natural to focus on the best rational approximation which exponentially decays at a rate proportional to 9.28903^{-K} (see [33]) and hence allows to evaluate weights $\varphi_{\alpha,\ell}(n; -h^\alpha A)$, with an accuracy very close to the precision machine, by a reasonable degree K.

From the practical point of view, a quasi–best rational approximation, which differs from the best only for a factor of $\mathcal{O}(56^{-K})$, can be evaluated, in a very fast and efficient way, by means of the *Carathéodory–Fejér* approach discussed in [34] and [31].

If well–organized, the computation of weights $\varphi_{\alpha,\ell}(n; -h^\alpha A)$ can be quite cheap. Indeed, poles z_k and residues r_k in the rational approximation $\mathcal{R}_K(z)$ appear in complex conjugate pairs (e.g., see Figure 1); under the assumption that A and b are real, the solution of the system (9) for any complex $\sigma_{k,j}$ is the complex conjugate of the solution of the corresponding system with the complex conjugate $\overline{\sigma_{k,j}}$ replacing $\sigma_{k,j}$. Moreover, the polynomials of Proposition 2 have real coefficients; hence it is $P_\ell(\overline{\sigma_{k,j}}) = \overline{P_\ell(\sigma_{k,j})}$ and $Q_\ell(\overline{\sigma_{k,j}}) = \overline{Q_\ell(\sigma_{k,j})}$.

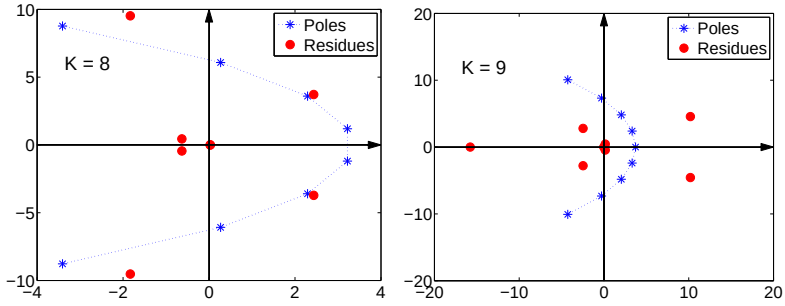

Figure 1: Location of poles and residues of the *Carathéodory–Fejér* rational approximations $\mathcal{R}_8(z)$ and $\mathcal{R}_9(z)$ in the complex plane

Thus, after sorting the poles z_k by their imaginary part and writing

$$\gamma_k = \frac{r_k}{z_k}\left(Q_\ell\left(\sigma_{k,j-1}\right) c^T y_{k,j-1} - P_\ell\left(\sigma_{k,j}\right) c^T y_{k,j}\right),$$

it is straightforward to see that

$$\gamma_k = \overline{\gamma_{K-k+1}}, \quad k = 1, \ldots, m_k, \quad m_k = \left\lfloor \frac{K}{2} \right\rfloor,$$

where γ_{m_k+1} is real for odd K. Thus, from (8) it is possible to evaluate

$$\varphi_{\alpha,\ell}^{[K]}(j; -h^\alpha A) = 2 \sum_{k=1}^{\left\lfloor \frac{K}{2} \right\rfloor} \Re(\gamma_k) + \gamma_{m_k+1},$$

where the last term γ_{m_k+1} drops out when K is even. The above expression has the remarkable advantage of reducing the number of operations in complex arithmetic and the number of coefficients γ_k to evaluate. Altogether the computation is nearly halved.

4.2. Solution of linear subsystems

The solution of the (possibly long) sequence of shifted systems (9) can be evaluated, without any difficulty, by means of standard techniques when the size of the problem is reasonably small. With problems whose size N is large the solution of several systems (9) can be computationally high demanding and hence more sophisticated techniques are necessary.

Recently, the use of Krylov subspace techniques has been actively investigated for problems of this type. Roughly speaking, these are iterative methods for solving systems of algebraic equations

$$Ax = b, \tag{10}$$

in which, at each iteration m, an approximation x_m of the solution is searched in the Krylov subspace

$$\mathcal{K}_m(A, b) = span\Big(b, Ab, A^2 b, \ldots, A^{m-1} b\Big).$$

Since the Krylov basis vectors $A^j b$ tend to align to the dominant eigenvector of A, these vectors are not suitable for practical computation. To avoid instability phenomena an alternative and orthonormal basis of $\mathcal{K}_m(A, b)$ is constructed by means of the Arnoldi algorithm.

Starting from the normalized vector $v_1 = b/\|b\|_2$, the Arnoldi algorithm builds new basis vectors v_{j+1} as linear combination of Av_j and the previous vectors $v_1, \dots v_j$; hence, a standard Gram–Schmidt procedure

$$\hat{v}_{j+1} = Av_j - \sum_{k=1}^{j} h_{k,j} v_k, \quad v_{j+1} = \frac{\hat{v}_{j+1}}{h_{j+1,j}}$$

performs the orthnormalization (coefficients $h_{k,j}$ are obtained by imposing the normality conditions $\hat{v}_{j+1}^T v_k = 0$, $k = 1, \dots, j$, and $h_{j+1,j} = \|\hat{v}_{j+1}\|_2$).

When A is symmetric, the less expensive Lanczos algorithm can be applied. For a complete discussion of these topics we refer to [35].

By collecting the new basis vectors v_j, $j = 1, \dots, m$, in the matrix $V_m \in \mathbb{R}^{N \times m}$, the Arnoldi algorithm after m steps provides the well–known relationships

$$\begin{aligned} AV_m &= V_m H_m + h_{m+1,m} v_{m+1} e_m^T \\ V_m^T A V_m &= H_m, \end{aligned} \tag{11}$$

where H_m is the upper Hessenberg matrix $H_m = (h_{k,j})$ and e_i is the ith unit vector in $\mathbb{R}^m$. Moreover, the orthonormal nature of v_j leads to

$$V_m^T V_m = I. \tag{12}$$

The way in which the approximation x_m of the solution of (10) is chosen in $\mathcal{K}_m(A, b)$ characterizes a particular Krylov subspace method.

For instance, in the Full Orthogonalization Method, the basis vectors in V_m are first used to build a generic vector in $\mathcal{K}_m(A, b)$, say $V_m y$, where $y \in \mathbb{R}^m$, and hence in the resulting system $AV_m y = b$ the approximation $AV_m \approx V_m H_m$ obtained from (11) is replaced. Thus, after multiplying by V_m^T both sides of $V_m H_m y = b$ and using (12) together to $V_m^T r = \|b\|_2 e_1$, we obtain the resulting system of reduced size $H_m y = \|b\|_2 e_1$. The approximated solution $x_m \approx A^{-1} b$ is therefore obtained as

$$x_m = V_m H_m^{-1} \|b\|_2 e_1.$$

The algorithm can be terminated when the Krylov sequence is exhausted and hence the resulting subspace is invariant under the adding of other Krylov vectors. In practice, some stopping criteria testing the residual $b - Ax_m$ of (10) is applied.

Some other examples of Krylov subspace methods are the Generalized Minimum Residual, the Quasi Minimal Residual, the Conjugate Gradient

and so forth. When b is a collection of row vectors ($N_i > 1$ in our system), some modified versions of the above algorithm, usually named as band Krylov methods, are used to deal with systems having multiple right-hand sides. It is beyond the scope of this paper to discuss in detail these methods; we just refer again to the book of [35] or the thorough review article [36].

The success of Krylov subspace methods is mainly due to the fact that they involve only matrix–vector multiplications and do not require matrix factorizations and/or inversions. Therefore they are suitable for problems of large size and take advantage of the possible sparsity of the matrix A (this frequently arises in systems in which each state–variable is connected with only few other state–variables).

Krylov subspace methods are particularly appealing for solving the sequence of shifted systems (9) because of the shift–invariance property of Krylov subspaces; indeed, since

$$\mathcal{K}_m(A, b) = \mathcal{K}_m(\sigma I + A, b), \quad \forall \sigma \in \mathbb{C},$$

the approximations for the solution of (9) can be evaluated for all shift parameters $\sigma_{k,j}$ by generating only one approximation subspace. In this way the weights in the exponential integrator (6) can be evaluated with a reasonable effort also for problems of remarkable size.

5. Numerical experiments

In this section we conduct some numerical experiments with the aim of verifying the theoretical findings and testing the effectiveness of the proposed schemes.

For comparison, we also apply a classical PI rule of Adams type; since the order barrier discussed in Subsection 2.2, the implicit trapezoidal PI rule, i.e. the one based on first order polynomials, is the most effective for stiff problems.

All the experiments are carried out by using Matlab ver. 7.9. on a computer equipped with an Intel dual-core CPU E5400 at 2.70GHz and running the Windows XP Operating System.

For the first test we consider the system coming form the prototype time–fractional PDE defined on the unit square $\Omega = [0, 1] \times [0, 1]$ according to

$$\frac{\partial^\alpha}{\partial t^\alpha} U(t, x, y) = \rho \left(\frac{\partial^2}{\partial x^2} U(t, x, y) + \frac{\partial^2}{\partial y^2} U(t, x, y) \right) \tag{13}$$

and subject to the boundary condition of Dirichlet type $U(t, x, y) = u(t)$, $(x, y) \in \partial\Omega$, where $u(t) = t^4/24 - 1$, and to the initial condition $U(0, x, y) =$

$g(x,y)$, $(x,y) \in \Omega$, where $g(x,y) = 1 + xy(1-x)(1-y)$. The problem has been suitably chosen as to provide a easily evaluable theoretical reference solution.

After the discretization of the spatial derivatives by central differences on a fixed grid of step–size $\bar{h} = 1/(\bar{N}+1)$, we are left with a system (1) of size $N = \bar{N}^2$, where $A = \rho(I \otimes E + E \otimes I)/\bar{h}$, $E = diag(-1, 2, -1)$ and $\otimes$ is the Kronecker product. The state variable $x(t)$ is obtained after reordering, in a natural way, the original variables $U(t, x_i, y_j)$; the input and initial vectors b and x_0 come in similar way after taking into account the boundary and initial conditions and the vector c is built so as to measure the output in the middle of the domain.

The parameters used in the experiments are $\rho = -2$ and $\bar{N} = 30$. The size of the corresponding system (1) is therefore $N = 900$ and, due to the symmetric nature of A, the shifted–systems (9) are projected by the Lanczos algorithm with a stopping criterion based on the residual of the seed system $Ay = b$; hence, the Matlab's backslash operator \ is used to solve the reduced linear systems.

The results for these experiments are presented, respectively for $\alpha = 0.5$ and $\alpha = 0.8$, in Tables 2 and 3 where the absolute errors $E(h)$, under decreasing step–sizes h, and the estimated order of convergence (EOC), obtained as $\log_2(E(h)/E(h/2))$, are reported. Each k–step method (6) is indicated as k–ExpInt.

	Trap. PI		1-ExpInt		2-ExpInt		3-ExpInt		4-ExpInt	
h	Error	EOC	Error	EOC	Error	EOC	Error	EOC	Error	EOC
1/8	1.98(−3)		5.05(−2)		1.99(−3)		1.60(−4)		5.07(−5)	
1/16	5.11(−4)	1.955	2.64(−2)	0.934	5.15(−4)	1.952	2.04(−5)	2.970	1.88(−6)	4.754
1/32	1.30(−4)	1.972	1.37(−2)	0.948	1.32(−4)	1.968	2.60(−6)	2.974	7.92(−8)	4.569
1/64	3.30(−5)	1.983	7.04(−3)	0.960	3.34(−5)	1.978	3.30(−7)	2.978	4.66(−9)	4.088
1/128	8.29(−6)	1.992	3.59(−3)	0.970	8.44(−6)	1.985	4.25(−8)	2.957	1.16(−9)	1.999

Table 2: Errors and estimated order of convergence (EOC) for $\alpha = 0.5$

The experimental estimates for the convergence order seem to match in an excellent way with the theoretical predictions stated in Theorem 4. Because of the errors related to the rational approximation in (7) and to use of Krylov subspace methods for solving (9), the overall error remains above a threshold of about 10^{-10} (see the last column in Tables 2 and 3). We think that for systems of large size this is however a satisfactory result.

Except for the 1–step method, which converges with order 1 as $h \to 0$, the other k–step exponential integrators show an accuracy equal or higher

	Trap. PI		1-ExpInt		2-ExpInt		3-ExpInt		4-ExpInt	
h	Error	EOC	Error	EOC	Error	EOC	Error	EOC	Error	EOC
1/8	1.89(−3)		4.83(−2)		1.90(−3)		1.68(−4)		8.69(−5)	
1/16	4.77(−4)	1.991	2.41(−2)	1.001	4.78(−4)	1.990	2.08(−5)	3.009	3.08(−6)	4.820
1/32	1.20(−4)	1.995	1.21(−2)	0.997	1.20(−4)	1.995	2.61(−6)	2.993	1.21(−7)	4.669
1/64	2.99(−5)	1.998	6.06(−3)	0.997	3.01(−5)	1.997	3.30(−7)	2.988	6.81(−9)	4.150
1/128	7.49(−6)	1.999	3.03(−3)	0.997	7.53(−6)	1.998	4.28(−8)	2.945	1.87(−9)	1.863

Table 3: Errors and estimated order of convergence (EOC) for $\alpha = 0.8$

with respect to the trapezoidal PI rule.

A more interesting comparison concerns with the effectiveness of the proposed methods. To this purpose we represent in Figure 2 the errors as function of the CPU time. The execution times are expressed in seconds and reliable measurements are achieved after repeating each test a large number of times and hence reporting the mean value of the execution times.

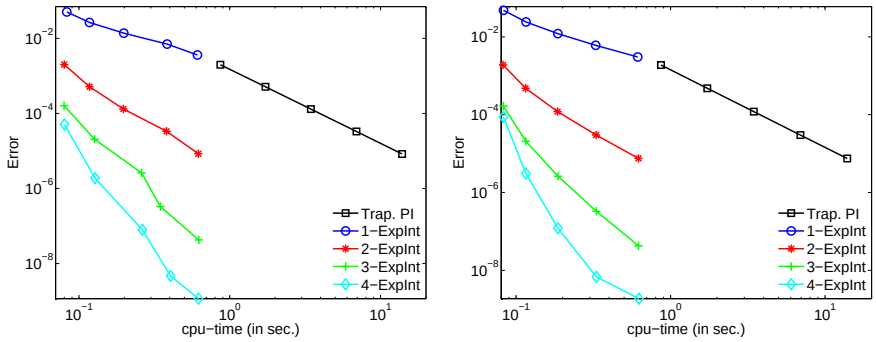

Figure 2: Error versus CPU time for the classical trapezoidal PI rule and the exponential integrators for $\alpha = 0.5$ (left) and $\alpha = 0.8$ (right) for the test problem of size $N = 900$

As we can easily observe, not only exponential integrators allow to obtain higher accuracy, but they are also able to operate in a shorter time. Although trapezoidal PI and the 2–step exponential methods provide the same accuracy, the computational effort required by the latter method is much smaller. Interestingly, exponential integrators of higher order increases in a considerable way the accuracy of the solution but do not affect significantly the computational time.

The next experiment concerns with a more sophisticated problem involved in a real–life application. We consider the system (1) deriving from the 2D model of an optical filter device tunable by thermal means and de-

scribed in [37]. For the simulation we make use of the matrices and vectors available at the website `http://simulation.uni-freiburg.de`. The 2D model is obtained after spatial discretization, by finite elements, of fractional PDEs and the resulting problem has an inner dimension of $N = 1668$. The extension to the fractional order of the original integer–order model allows to take into account the effects of anomalous heat conductivity in the thermal means.

The lack of symmetry in the resulting coefficient matrix forces us to use the more expensive Arnoldi algorithm to project the shifted–systems (9). Moreover, the different structure of the spectrum of the matrix implies a larger size of the resulting Krylov subspaces, thus to increase the overall computational effort.

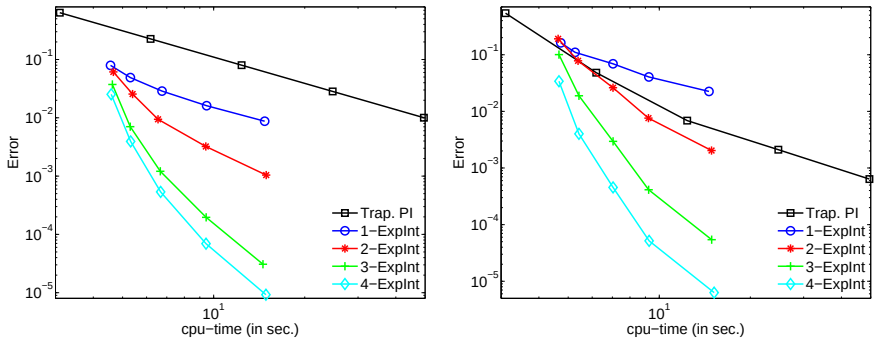

Figure 3: Error versus CPU time for the classical trapezoidal PI rule and the exponential integrators for $\alpha = 0.5$ (left) and $\alpha = 0.8$ (right) for the test problem of size $N = 1668$

Because of the characteristics of this problem, the comparison with the PI rule is less impressive. However, also in this case (see Figure 3) the numerical simulations emphasize the possibility of reaching high accuracy at a reasonable cost when exponential integrators are used.

References

[1] K. Diethelm, The analysis of fractional differential equations, Vol. 2004 of Lecture Notes in Mathematics, Springer-Verlag, Berlin, 2010.

[2] F. Mainardi, Fractional calculus and waves in linear viscoelasticity, Imperial College Press, London, 2010.

[3] I. Podlubny, Fractional differential equations, Vol. 198 of Mathematics in Science and Engineering, Academic Press Inc., San Diego, CA, 1999.

[4] O. Enacheanu, D. Riu, N. Retiére, P. Enciu, Identification of fractional order models for electrical networks, in: IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on, 2006, pp. 5392–5396.

[5] M. Hochbruck, A. Ostermann, Exponential integrators, Acta Numer. 19 (2010) 209–286.

[6] M. P. Calvo, C. Palencia, A class of explicit multistep exponential integrators for semilinear problems, Numer. Math. 102 (3) (2006) 367–381.

[7] M. Hochbruck, A. Ostermann, Exponential multistep methods of Adams-type, BIT 51 (4) (2011) 889–908.

[8] S. P. Nørsett, An A-stable modification of the Adams-Bashforth methods, in: Conf. on Numerical Solution of Differential Equations (Dundee, 1969), Springer, Berlin, 1969, pp. 214–219.

[9] A. Young, Approximate product-integration, Proc. Roy. Soc. London Ser. A. 224 (1954) 552–561.

[10] C. Lubich, A stability analysis of convolution quadratures for Abel-Volterra integral equations, IMA J. Numer. Anal. 6 (1) (1986) 87–101.

[11] R. Garrappa, On linear stability of predictor–corrector algorithms for fractional differential equations, Int. J. Comput. Math. 87 (10) (2010) 2281–2290.

[12] C. Lubich, Discretized fractional calculus, SIAM J. Math. Anal. 17 (3) (1986) 704–719.

[13] L. Galeone, R. Garrappa, Fractional Adams-Moulton methods, Math. Comput. Simulation 79 (4) (2008) 1358–1367.

[14] R. Garrappa, On some explicit Adams multistep methods for fractional differential equations, J. Comput. Appl. Math. 229 (2) (2009) 392–399.

[15] F. Mainardi, R. Gorenflo, Fractional oscillations and Mittag–Leffler functions, Tech. Rep. A-14/96, Freie Universitaet Berlin, Serie A Mathematik (1996).

[16] F. Mainardi, R. Gorenflo, On Mittag-Leffler-type functions in fractional evolution processes, J. Comput. Appl. Math. 118 (1-2) (2000) 283–299.

[17] R. Garrappa, M. Popolizio, Generalized exponential time differencing methods for fractional order problems, Comput. Math. Appl. 62 (3) (2011) 876–890.

[18] J. N. Lyness, B. W. Ninham, Numerical quadrature and asymptotic expansions, Math. Comp. 21 (1967) 162–178.

[19] Wolfram Research Inc., Zeta functions and polylogarithms (1998–2012). URL `http://functions.wolfram.com`

[20] C. Lubich, Runge-Kutta theory for Volterra and Abel integral equations of the second kind, Math. Comp. 41 (163) (1983) 87–102.

[21] N. J. Higham, Functions of matrices, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008.

[22] L. Lopez, V. Simoncini, Preserving geometric properties of the exponential matrix by block Krylov subspace methods, BIT 46 (4) (2006) 813–830.

[23] L. Lopez, V. Simoncini, Analysis of projection methods for rational function approximation to the matrix exponential, SIAM J. Numer. Anal. 44 (2) (2006) 613–635 (electronic).

[24] N. Del Buono, L. Lopez, A survey on methods for computing matrix exponentials in numerical schemes for ODEs, in: Computational science—ICCS 2003. Part II, Vol. 2658 of Lecture Notes in Comput. Sci., Springer, Berlin, 2003, pp. 111–120.

[25] F. Diele, I. Moret, S. Ragni, Error estimates for polynomial Krylov approximations to matrix functions, SIAM J. Matrix Anal. Appl. 30 (4) (2008/09) 1546–1565.

[26] M. Popolizio, V. Simoncini, Acceleration techniques for approximating the matrix exponential operator, SIAM J. Matrix Anal. Appl. 30 (2) (2008) 657–683.

[27] R. Garrappa, M. Popolizio, On the use of matrix functions for fractional partial differential equations, Math. Comput. Simulation 81 (5) (2011) 1045–1056.

[28] R. Garrappa, M. Popolizio, Evaluation of generalized Mittag–Leffler functions on the real line, Advances in Computational Mathematics (2012) 1–21doi:10.1007/s10444-012-9274-z.

[29] I. Moret, P. Novati, On the convergence of Krylov subspace methods for matrix Mittag–Leffler functions, SIAM J. Numer. Anal. 49 (5) (2011) 2144–2164.

[30] I. Moret, M. Popolizio, The restarted shift–and–invert Krylov method for matrix functions, submitted (2011).

[31] L. N. Trefethen, J. A. C. Weideman, T. Schmelzer, Talbot quadratures and rational approximations, BIT 46 (3) (2006) 653–670.

[32] R. Garrappa, M. Popolizio, On accurate product integration rules for linear fractional differential equations, J. Comput. Appl. Math. 235 (5) (2011) 1085–1097.

[33] P. Petrushev, V. Popov, Rational approximation of real function, Cambridge University Press, Cambridge, 1987.

[34] L. N. Trefethen, M. H. Gutknecht, The Carathéodory-Fejér method for real rational approximation, SIAM J. Numer. Anal. 20 (2) (1983) 420–436.

[35] Y. Saad, Iterative methods for sparse linear systems, 2nd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[36] V. Simoncini, D. B. Szyld, Recent computational developments in Krylov subspace methods for linear systems, Numerical Linear Algebra with Applications 14 (1) (2007) 1–59.

[37] T. Bechtold, E. B. Rudnyi, J. G. Korvink, Fast simulation of Electro–Thermal MEMS, Springer–Verlag, Berlin, 2010.