Bollettino
Unione Matematica Italiana

Marina Popolizio

Numerical Approximation of Matrix Functions for Fractional Differential Equations

Bollettino dell'Unione Matematica Italiana, Serie 9, Vol. **6** (2013), n.3, p. 793–815.

Unione Matematica Italiana

<http://www.bdim.eu/item?id=BUMI_2013_9_6_3_793_0>

Bollettino dell'Unione Matematica Italiana, Unione Matematica Italiana, 2013.

Bollettino U. M. I.
(9) **VI** (2013), 793-815

Numerical Approximation of Matrix Functions for Fractional Differential Equations

MARINA POPOLIZIO

Sunto. – *In questo lavoro si presentano delle connessioni rilevanti tra le funzioni di matrice e la soluzione di equazioni differenziali di ordine frazionario. Questo nesso è stato notato solo recentemente ed ora riscuote notevole interesse. Si presenta qui una rassegna dei fondamenti del calcolo frazionario e della teoria dell'approssimazione di funzioni di matrice; si mostrano inoltre i contributi che, insieme ad i miei coautori, abbiamo recentemente elaborato su questo argomento* [13, 14, 15, 16, 32].

Abstract. – *In this paper relevant insights are given on the connection between matrix functions and the solution of differential equations of fractional order. This nexus only recently has been disclosed and is gaining weight in the current research. We present here a review on the basics of fractional calculus and matrix function approximations, together with the main results my coauthors and me have given to the subject in the recent works* [13, 14, 15, 16, 32].

1. – Introduction

Functions of matrices deserve a fundamental role in several branches of applied sciences as biology, physics, engineering and many others.

From a theoretical point of view, the first one to address the problem of matrix functions was Cayley who, in his paper *A Memoir on the Theory of Matrices* dated back to 1858, treated the square root of a matrix of dimension 2×2 and 3×3. The formal definition of matrix function dates back to 1883 and is due to Sylvester. Matrix functions appear in several important applications and this makes their numerical approximation a fundamental topic in the research community; many different approaches have been proposed during the last years; we cite, among the others, [27, 28, 39, 30, 24, 36, 5, 6, 37, 38, 29, 31, 32] and the recent book by Higham [19] completely devoted to this subject.

The most common matrix function is certainly the exponential and it is also the most studied one from the computational point of view. The applications involving this matrix are several and very common: among the most important we cite its role in the solution of systems of ordinary differential equations

(ODEs) of the form

$$\frac{dy(t)}{dt} = Ay(t), \quad y(0) = y_0, \tag{1}$$

whose solution is

$$y(t) = \exp(tA)y_0.$$

It is thus clear that the matrix exponential has a key role in the numerical solution of systems of differential equations; however, in this field other matrix functions are very important, as we will discuss in Section 3 when introducing methods for differential equations of fractional order.

Systems of ODEs like (1) derive from real life models or, more often, are the result of the process of discretization of partial differential equations (PDEs) by means of the *method of lines*, that consists in discretizing only the spatial derivatives to reduce the problem to a system of differential equations just in the time variable.

This is actually the source of some of the difficulties one faces in the numerical approximation of matrix functions. Indeed, in this context the dimension of the matrix argument A is usually large, while the classical numerical methods only work for small dimension problems. To find effective numerical methods for large dimension problems is thus compelling, with efficiency intended both for the computational costs and for the memory requirements.

2. – Fractional calculus

The origin of fractional calculus is far and, probably, this history dates back to 1695, when Leibnitz and L'Hopital started to intellectualize about the meaning of derivative of order $1/2$. The theoretical analysis of this topic has received a great attention and important results have been collected over the years; this development is due also to the interest that the subject receives from many fields, such as control theory, finance, physics and many others. A meaningful example of the importance of fractional calculus is the class of *non-newtonian* fluids, as asphalt, rubbers and polymers, whose dynamics are adequately described only through models involving fractional derivatives.

For a deep analysis of the subject and for a wide list of references we refer to the cornerstone books [34, 22] and, for the more recent developments, to [8, 9, 10, 11, 15, 18].

A differential equation of *fractional order* (FDE) is of the form

$$D_0^{\alpha} y(t) = g(t, y(t)) \tag{2}$$

where $\alpha > 0$ is the fractional order. Different definitions hold for the fractional

derivative $D_0^\alpha y(t)$; however, for the applications, usually the Caputo's definition, given in 1967, is preferred

$$D_0^\alpha y(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\tau)^{m-\alpha-1} \left(\frac{d^m}{d\tau^m} y(\tau) \right) d\tau, \ m = \lceil \alpha \rceil.$$

Indeed, with this definition, (2) can be combined with initial conditions of the form

$$y^{(k)}(0) = y_{0,k}, \quad k = 0, \dots, m-1$$

which, as in the case of ODEs, have a clear physical meaning.

Substantial differences with the ordinary derivatives arise from the definition of D_0^α: indeed, it involves an integral over the whole interval $[0, T]$ (for this reason the origin is pointed out in the symbol of fractional derivative operator). This is a very interesting feature of the fractional derivatives and a strong point of it: indeed, this integral representation works well in describing all those phenomena "with memory", for example in finance, psychology, rheology and so on.

However the numerical computation of the fractional derivatives is clearly more difficult for this feature; indeed there is no possibility to extend the classical methods for ODEs which substantially work in a step-by-step way.

For linear FDEs the solution can be easily recovered by means of the *variation of constants formula*. Let consider the solution of the following system of linear FDEs

$$\begin{cases} D_0^\alpha y(t) + Ay(t) = g(t) \\ y(0) = y_0 \end{cases} \qquad 0 < \alpha < 1. \tag{3}$$

The Laplace transform applied to every term gives

$$s^\alpha Y(s) - s^{\alpha-1} y_0 + AY(s) = G(s)$$

so, if I denotes the identity matrix of the same dimension of A,

$$Y(s) = s^{\alpha-1}(s^\alpha I + A)^{-1} y_0 + (s^\alpha I + A)^{-1} G(s).$$

The exact solution in the temporal domain can then be written as

$$y(t) = e_{\alpha,1}(t; A) y_0 + \int_0^t e_{\alpha,\alpha}(t-s; A) g(s) ds \tag{4}$$

with $e_{\alpha,\beta}(t; \lambda)$ denoting the inverse of the Laplace transform $\dfrac{s^{\alpha-\beta}}{s^\alpha + \lambda}$ for scalar λ's.

Throughout the paper the extension to the matrix case is intended to be well defined, for the matrices and the functions we deal with.

The variation of constants formula can be generalized to the case of semilinear FDEs of the form

$$\begin{cases} D_0^{\alpha} y(t) + Ay(t) = g(y(t)) \\ y(0) = y_0 \end{cases} \qquad 0 < \alpha < 1 \tag{5}$$

and its use allows us to express the solution as

$$y(t) = e_{\alpha,1}(t;A)y_0 + \int_0^t e_{\alpha,\alpha}(t-s;A)g(y(s))ds. \tag{6}$$

It is easy to verify that (5) is a generalization to FDEs of the variation of constants formula for semilinear ODEs like

$$y'(t) + Ay(t) = g(y(t)), \;\; y(0) = y_0$$

whose solution is

$$y(t) = e^{-tA}y_0 + \int_0^t e^{(s-t)A}g(y(s))ds. \tag{7}$$

The expressions for the solutions of these differential equations clearly show that the key role of the exponential function in the ODEs' solution is exactly the same that the $e_{\alpha,\beta}$ functions have for FDEs.

In particular, the numerical approximation of these problems undiscovers a more deep relationship between the $e_{\alpha,\beta}$ functions and the φ-functions, derived from the exponential, which are fundamental for some classes of numerical solvers, we will discuss later on.

To get a simple numerical scheme for (7) we fix a discretization step h and we interpolate the term $g(y(t))$ in the known value $g(y_n)$, so that

$$y_{n+1} = e^{-hA}y_n + h\varphi_1(-hA)g(y_n), \quad \varphi_1(z) = \frac{e^z - 1}{z};$$

this method goes under the name *explicit exponential Euler*.

More accurate methods can be defined by expressing the solution as a linear combination of terms like $\varphi_j(A)$ with φ_j defined by the relation

$$\varphi_0(z) = \exp(z), \;\; \varphi_k(0) = \frac{1}{k!}, \;\; \varphi_{k+1}(z) = \frac{\varphi_k(z) - \varphi_k(0)}{z}. \tag{8}$$

Methods of this form belong to the class of *Exponential Integrators*, and we will discuss them in Section 3.

The functions $e_{\alpha,\beta}$ in (4) and (6) can also be defined by the classical Mittag-Leffler (ML) functions

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\alpha k + \beta)} \tag{9}$$

for $z, \alpha, \beta \in \mathbb{C}$ with $\Re(\alpha), \Re(\beta) > 0$ and Γ denoting the Euler Gamma function. It holds, indeed, that [34]

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

The analogy between $E_{\alpha,\beta}$ and the exponential is evident; moreover, it is immediate to verify that these functions coincide for $\alpha = \beta = 1$. In practice a strict relationship binds the ML functions with the φ-functions, as shown in the following proposition:

PROPOSITION 2.1 ([14]). – *Let $k \in \mathbb{N}$ and $z \in \mathbb{C}$. Then $E_{1,k+1}(z) = \varphi_k(z)$.*

The ML functions thus represent an extension of φ-functions to the non-integer case; moreover they have similar properties for $\beta \geq 1$; indeed, [14]

$$e_{\alpha,\alpha+\beta}(t;\lambda) = \frac{e_{\alpha,\beta}(t;\lambda) - e_{\alpha,\beta}(t;0)}{-\lambda}, \quad e_{\alpha,\beta}(t;0) = \frac{t^{\beta-1}}{\Gamma(\beta)},$$

which generalize (8).

Moreover, as shown in [34], for any $\gamma \in \mathbb{R}$

$$D_0^{\gamma} e_{\alpha,\beta}(t;\lambda) = e_{\alpha,\beta-\gamma}(t;\lambda).$$

3. – Generalized Exponential Integrators

The numerical solution of *stiff* differential equations is a tough problem to treat. In particular, we consider systems of ODEs of the form

$$\frac{d}{dt}y(t) + Ay(t) = g(t, y(t)), \quad y(0) = y_0 \tag{10}$$

with y and g vector valued functions and A a real matrix which represents the *stiff* part of the problem.

The term *Exponential Integrators* is commonly used to define a class of numerical methods to solve problems of this kind, whose origin probably dates back to the beginning of the 1960's with Certain and Lawson; in the following we will denote them with ETD, which is the acronym for "Exponential Time Differencing" methods, which is another name to define them.

The recent paper by Hochbruck and Ostermann [21] is an excellent reference for a deep analysis of the subject, with a rich description of the development these methods have had in the past. It is important to stress that their development is actually strictly connected with that of the numerical methods to approximate matrix functions since, as their name suggests, these methods are based on the exponential function and on the φ-functions defined in (8), for matrix arguments

which, in the applications, have very large dimensions. The lack of numerical techniques for this aspect has, for long time, discouraged the use of these integrators. Recently, the important development of effective techniques for the matrix functions approximation gave new light to these methods, which are now very popular and commonly employed in the applications.

Explicit methods are not adequate to solve stiff problems for the severe restriction on the integration step that the problem nature imposes. On the other hand, implicit methods are sometimes too tough to be applied since they entail the solution of a non-linear system at each integration step. Exponential integrators, instead, turn out to be very effective since they solve exactly the linear part in (10), or at least with a very high accuracy, while explicit schemes are applied to solve the non-linear term, without particular restriction on the integration step.

3.1 – *Linear FDEs with constant source term*

In [14] we addressed the numerical solution of linear systems of FDEs of the form (3) with constant source term $g(t) = b$. It is known that the solution of this problem can be found by means of (4) [34] and can be written as

$$y(t) = y_0 + t^{\alpha} E_{\alpha,\alpha+1}(-t^{\alpha}A)\big(-Ay_0 + b\big). \tag{11}$$

In [14] an experimental comparison has been proposed among different techniques to approximate (11), the difference being in the numerical approach to approximate the term $E_{\alpha,\alpha+1}(-t^{\alpha}A)\big(-Ay_0 + b\big)$, as we will describe in Section 4.

3.2 – *Semilinear case*

The exponential integrators for ODEs are essentially grounded on the *Variation of Constants formula* (7).

To numerically evaluate it we fix a mesh-grid of equispaced points $t_n = nh$ on the integration interval; thus (7) can be written as

$$y(t_{n+1}) = e^{-(t_{n+1}-t_n)A} y(t_n) + \int_{t_n}^{t_{n+1}} e^{-(t_{n+1}-s)A} g(y(s))ds; \tag{12}$$

the ETDs' strategy lies on the polynomial approximation of the non-linear term g; then the resulting integral is exactly solved. The simplest method in this class is the *explicit exponential Euler* as described in Section 2.

The technique used to evaluate the integral in (12) distinguishes the methods of the ETD family, as for example the exponential multistep methods, the exponential Runge-Kutta methods and so on.

The introduction of the φ-functions provides the opportunity to define ETD methods which are more accurate; the solution to (10) can indeed be expressed as

$$y(t_{n+1}) = e^{-hA}y(t_n) + \sum_{k=0}^{\infty} h^{k+1}\varphi_{k+1}\big(-hA\big)g^{(k)}(y(t_n)); \tag{13}$$

thus, for example, the exponential Adams-Bashforth and Adams-Moulton methods derive when the series in (13) is truncated and the derivatives are approximated by means of finite differences.

For the semilinear FDE (5) an analogous approach can be considered: so, (6) can be written as

$$y(t_n) = e_{\alpha,1}(t_n; A)y_0 + \sum_{j=0}^{n-1} \int_{t_j}^{t_{j+1}} e_{\alpha,\alpha}(t_n - s; A)g(y(s))ds.$$

To get a naïve version of the ETD methods one can use a polynomial interpolant for the term $g(y(s))$ in each subinterval $[t_j, t_{j+1}]$, and then solve the resulting integrals. The resultant methods were named *Generalized Exponential Time Differencing* (GETD) in [14] for the plain analogy with the corresponding exponential integrators for ODEs.

The simplest choice would be to approximate $g(y(s))$ in each subinterval $[t_j, t_{j+1}]$ by the constant value $g(y_j)$. To explicitely evaluate the coefficients of the corresponding scheme we make use of the formula

$$e_{\alpha,\beta}(t; \lambda) = h^{\beta-1}e_{\alpha,\beta}\left(\frac{t}{h}; h^{\alpha}\lambda\right), \quad \forall h > 0$$

and of the following result:

THEOREM 3.1 [14]. – *Let* $a < b \leq t$, $\Re(\alpha) > 0$, $\beta > 0$. *Then*

$$\int_a^b e_{\alpha,\beta}(t - s; A)ds = e_{\alpha,\beta+1}(t - a; A) - e_{\alpha,\beta+1}(t - b; A).$$

Then the resulting 1-*step GETD* method can be written as

$$y_n = e_{\alpha,1}(n; h^{\alpha}A)y_0 + h^{\alpha}\sum_{j=0}^{n-1}\big(e_{\alpha,\alpha+1}(n; h^{\alpha}A) - e_{\alpha,\alpha+1}(n-1; h^{\alpha}A)\big)g(y_j). \tag{14}$$

To get more accurate schemes we can simply interpolate $g(y(s))$ with higher degree polynomials. For quadratic polynomials, for example, the explicit expression for the coefficients of the 2-*step GETD* method has been derived in [14], together with an analysis of its main properties.

The introduction of these GETD methods in [14] is grounded also on rigorous analysis on accuracy and stability properties. We just cite an important result about the order of convergence of the two methods 1-*step GETD* and 2-*step GETD*:

THEOREM 3.2 [14]. – *If the source term g is smooth, then the* 1-step GETD *method has order* 1 *and the* 2-step GETD *has order* $1+\alpha$.

4. – Approximation of $f(A)$

All the numerical methods of exponential type described so far for the solution of ODEs or FDEs are essentially based on matrix functions, whose numerical approximation turns out to be crucial for the effectiveness of the method itself. This aspect fully motivates the analysis we are going to present.

For small dimension problems, that is, when the matrix argument A has a small dimension n, more than one strategy is often at disposal; for the exponential, for example, about twenty approaches are described in [27, 28].

More complicate is the situation for medium n and far more for large n, which is the most common case in the solvers described in Section 2. When speaking of systems of differential equations, indeed, the matrix A usually comes from a discretization process, thus to have so large dimensions to make impracticable any strategy of decomposition or similar, since they require a computational cost proportional to n^3, which is prohibitive for large n [17]. Thus, in these situations, it turns out to be more convenient to work with some approximation of the function f.

4.1 – *Rational Approximations*

Among the possible approximations of a given function f the rational ones show, under suitable hypotheses, interesting features which can be well exploited when the numerical approximation is in play.

A rational function is defined as the ratio of two polynomials, without common factors, of degree μ and ν not necessarily coinciding. The set of rational functions is thus defined as

$$R_{\mu,\nu} = \left\{ r : r(z) = \left(\sum_{k=0}^{\mu} a_k z^k \right) \Big/ \left(\sum_{k=0}^{\nu} b_k z^k \right) \right\}.$$

An usual problem in approximation theory is, given a continuous function f and two values μ and ν, to find a rational function $r^* \in R_{\mu,\nu}$ which minimizes the approximation error $\|f - r^*\|_\infty$ with respect to all other functions in the

set $R_{\mu,\nu}$. This *best* approximation always exists and it is called the *Chebyshev* approximation.

For the exponential function the Chebyshev rational approximation has been largely studied [4, 40, 26]. Particularly useful is the expression of this approximation in the case in which the numerator and the denominator have the same degree, say ν. Indeed, it can be written as

$$\mathcal{R}(x) = \sum_{k=0}^{\nu} \frac{g_k}{x - \xi_k} \tag{15}$$

whose terms g_k and ξ_k are the poles and the residues, respectively, of $\mathcal{R}$.

In [4], for the exponential function, the poles and the residues have been tabled for several values of ν; moreover, the error analysis for these approximations have been largely studied [33, 4, 26], with very interesting results showing an exponential decay for the error at the rate $9.28903^{-\nu}$, which thus ensures a good accuracy also for moderate values of ν.

In [36, 37, 38] the numerical approximation of $\exp(A)$ has been addressed with techniques based on the Chebyshev rational approximations.

4.2 – *Carathéodory-Fejér approximation*

To find the poles and the residues of the Chebyshev approximation of a generic function, as for example the ML one (9), is a tough problem, also because uniqueness is not guaranteed for the best approximation. This problem has been a core issue in approximation theory and, in 1917, the *Carathéodory-Fejér* (CF) theory was presented: this is an appealing approach since, other than theoretical results, it offers a practical way to determine a rational approximation of fixed degree for any given smooth function. The key idea of this approach is to look for the rational approximation in a different set, namely

$$\tilde{R}_{\mu,\nu} = \left\{ r : r(z) = \left(\sum_{k=-\infty}^{\mu} a_k z^k \right) \Big/ \left(\sum_{k=0}^{\nu} b_k z^k \right) \right\}.$$

The goal is to determine the function $r^{cf} \in \tilde{R}_{\mu,\nu}$ which minimizes the error $\|f - r^{cf}\|_\infty$ with respect to all other functions in the set $\tilde{R}_{\mu,\nu}$. Then the CF approximation is r^{cf} without the terms of negative degree.

From a theoretical point of view the CF approximations are particularly interesting for the good quality they guarantee. Moreover, for smooth functions, Trefethen [40, 41] has justified their use by showing that the error is very similar to the one of the Chebyshev approximation. For the exponential, for example, r^{cf} and r^*, both of degree ν, differ for a factor $\mathcal{O}(56^{-\nu})$ [26].

From a practical point of view the CF approximants are interesting since they are easy to determine and they can be expressed in the form (15). Recently an

elegant and effective strategy has been poposed to calculate the poles and the residues for generic functions f [42], by means of the SVD decomposition of the Hankel matrix defined by the coefficients of the Fourier series of f. In Section 6 we will present the results of numerical techniques based on this strategy.

In [14] we followed the approach in [42] to find the CF approximation of the ML function in order to approximate the terms $E_{\alpha,\beta}(A)v$ in (11).

The error analysis has highlighted the good accuracy these approximations reach even for moderate degrees, with a behavior for the error similar to the equioscillation, typical feature of the best approximations.

In Section 6 we will present some of the results in [14] on the accuracy of these approximations.

5. – Problems for the computation of $f(A)v$

The examples described so far stress the importance to approximate vectors of the form $f(A)v$, rather than to approximate just the matrix $f(A)$. To take into account this slight difference is fundamental from the computational point of view. To give a flavor of the difference between the two problems, we cite the trivial example of linear systems of the form $Ax = b$: in this case, even though the solution is, theoretically, $A^{-1}b$, no one can think to numerically compute it in this way, for the inaccuracies the matrix inversion can generate.

We thus analyze in depth the approximation of vectors like $f(A)v$, taking into account the approximation results described so far for f.

5.1 – $f(A)v$ *for* f *in partial fraction expansion*

Assume we know a rational approximation for f of the form (15), then it is possible to approximate $f(A)v$ by solving v linear systems with coefficient matrices of the form $A - \xi_j I$; indeed, if the argument is a matrix, (15) becomes equivalent to

$$\mathcal{R}(A)v = \sum_{j=0}^{v} g_j (A - \xi_j I)^{-1} v. \tag{16}$$

When the matrix A is not too large and/or ill-conditioned, then any numerical method can solve the linear systems in (16). A matrix A with large dimension leads to a completely different scenario, since iterative methods are compelling. Moreover, when the convergence of these methods is too slow a kind of preconditioning is needed. In [13] we compared different strategies to compute (16), some of them involving also the Krylov subspace methods, as we will define in Section 5.2. In [38], instead, we found a strict relationship between the *Shift-and-Invert* technique, we are going to describe, and the preconditioning of systems in (16).

An additional difficulty is related to the nature of the poles ξ_j: indeed, when these poles are complex, complex linear systems need to be solved. In this case, not only the execution time is far larger than that of real systems of the same dimension, but even more complicate is the problem of their preconditioning: this subject is not yet well understood and important contributes are on the way. For shifted linear systems, as in (16), that is, with coefficient matrices of the form $A + \xi I$, for a real A and a complex ξ, we cite, among the others [1, 2, 3].

5.2 – *Shift-and-Invert method*

The Krylov subspace methods are largely used for the computation of vectors like $f(A)v$ for their efficiency and for their good accuracy and convergence properties. Their use has been widely analyzed in the context of the exponential function [20, 39, 24, 25].

They aim to approximate $f(A)v$ in Krylov subspaces defined as

$$\mathcal{K}_m(A, v) \equiv span\{v, Av, \ldots, A^{m-1}v\}.$$

To build an orthonormal basis for this space the starting vector is $v_1 = v/\|v\|$; then at each step $i = 2, \ldots,$ a new vector v_i is defined, by applying the orthonormalization Gram-Schmidt process to Av_{i-1} with respect to the previously computed vectors. This technique defines the matrices $V_m = [v_1, \ldots, v_m]$ and H_m of dimension $m \times m$ whose entries $h_{i,j}$ are the orthonormalization coefficients. Through these matrices the following approximation holds

$$f(A)v \approx \|v\| V_m f(H_m) e_1$$

where e_1 denotes the first column of the identity matrix of dimension $m \times m$.

The potency of these techniques is that usually a small dimension m is sufficient to get a sufficiently accurate approximation; thus some classical method usually works to compute $f(H_m)$.

Anyway, as shown by Hochbruck and Lubich [20], the convergence of the Krylov subspace methods is connected to the spectral properties of the matrix A. This feature makes the approach particularly slow for matrices A with large spectrum, as it often happens when the matrix derives from the discretization of differential operators. The slow convergence motivated the investigation of acceleration techniques. Among them we put our attention on the *Shift-and-Invert* (SI) technique proposed by Moret and Novati [30] and afterwards by van den Eshof e Hochbruck [43]. This technique is grounded on the Krylov subspaces $\mathcal{K}_m((I + \gamma A)^{-1}, v)$ for a suitable shift parameter γ. In this way, if V_m and H_m denote the matrices for $\mathcal{K}_m((I + \gamma A)^{-1}, v)$, the approximation becomes

$$f(A)v \approx \|v\| V_m f(B_m) e_1,$$

with B_m defined by the equation

$$(I + \gamma B_m)H_m = I.$$

Several applications show the superiority of this approach with respect to the standard Krylov subspace methods. It is indeed evident that in lots of applications the SI method needs a space dimension m much smaller than the one required by the standard approach. However, for this approach the shift parameter γ plays a fundamental role. In the paper [38] we got the interesting result to fix this parameter *a priori*, independently on the matrix A. To get this result a different perspective was crucial: indeed, it was necessary to set the SI method in the framework of rational approximations for the exponential function. Indeed, if a rational approximation like (16) is used for $\exp(A)v$, then in [38] we demonstrated that the SI method corresponds to precondition all the systems in (16) by means of the same preconditioner for all poles, which depends on γ.

The problem to approximate $f(A)v$ reduces then to approximate $f(B_m)$, with B_m in general much smaller than A. Various approaches can then be applied to the last approximation: if, for example, $f = \exp$, then the Matlab code `expm` can be used: it is based on the classical *scaling-and-squaring* method combined with a Padé rational approximation [33]. For the tests in [37, 38], for example, this choice was used. In general, given a rational approximation to f with a partial fraction expansion like (16), one can think to compute $f(B_m)e_1$ by solving the linear systems in (16) by a direct solver, as the Matlab `backslash`, for example. Or, in Matlab environment, also the command `funm` can be used when the first derivative of f is at disposal; the code computes the Schur decomposition of the matrix and then evaluates the function f for the triangular factor by means of the Parlett algorithm (for details see [17] and references therein).

Podlubny and Kacenak [35] gave a Matlab code for the ML function with scalar arguments; it allows one to approximate the ML function with any desired accuracy and, for our numerical tests, we combined it with the `funm` command.

Recently Moret and Novati [31] studied the convergence of the SI method when applied to the computation of the ML for $0 < \alpha < 1$ and $\beta = 1$: an interesting result is that the convergence accelerates when α decreases, exactly the inverse of what happens for the standard Krylov subspace methods.

6. – Numerical tests

6.1 – *Comparison of numerical methods for linear FDEs*

In [13] we addressed the solution of systems of linear FDEs with constant source terms (3). In practice systems like (3) often derive from PDEs with fractional time derivatives and space derivatives of integer or fractional order. If

only the space derivatives are discretized, then these PDEs reduce to systems of FDEs in the time variable.

We report here these model problems:

Example 1: the fractional Fokker-Planck equation

The Fokker-Planck equation is commonly used in the statistical physics to describe the time evolution of a test particle subject to an external field; its solution is the probability $u(x,t)$ of the particle to be at the time t in a given position x. In presence of an highly non-homogeneous medium the anomalous diffusion is adequately described only by FDEs. In particular, according to the formulation given by [7], the Fokker-Planck equation in one dimension can be written as

$$\begin{aligned} &D_0^{\alpha}u(x,t) = \frac{F''(x)}{\eta}u(x,t) + \frac{F'(x)}{\eta}\frac{\partial}{\partial x}u(x,t) + ku(x,t), && x\in(a,b),\ t\in[0,T],\\ &u(x,0) = 0.1, && x\in[a,b],\\ &u(a,t) = 0.1, u(b,t) = 0.1, && t\geq 0, \end{aligned} \tag{17}$$

with $F(x) = \cos x - 6x$, $\eta = 6$ and $k = 2$.

Given an equispaced grid $x_j = a + jh$ on $[a,b]$ with step-size h, the discretization of the first and second order spatial derivatives by means of central differences leads to the system (3) with a tridiagonal matrix A. Thus, taking into account the initial and boundary conditions, the solution of (17) is equivalent to solve the systems of FDEs (3).

Example 2: A space-time fractional diffusion equation

We consider the generic space-time fractional diffusion equation

$$\frac{\partial^{\alpha}u(x,t)}{\partial t^{\alpha}} = \frac{1}{2}x^2\frac{\partial^{\beta}u(x,t)}{\partial x^{\beta}}, \quad 0<\alpha<1<\beta\leq 2, x\in[0,10], t\in[0,T]$$

subject to the initial conditions $u(x,0) = 0.1$ and $u_t(x,0) = 0.1$. When the fractional space derivatives are suitably discretized one gets a system like (3) with the matrix A in Hessenberg form.

Numerical methods:

Once the equation tests have been reduced to systems of FDEs, we compute their solution by different approaches to compare their accuracy and efficiency. We describe the methods under investigation, taking into account that the environment used is Matlab, and thus, when necessary, we directly refer to the specific command used;

- **CF + PFE**: Carathéodory-Fejér (CF) approximation of degree $v = 6$ for $E_{\alpha,\alpha+1}$ evaluated by means of its partial fraction expansion (PFE) as in (16); the `backslash` is used for the linear systems.

- **SI**+`funm`: Shift-and-Invert method combined with `funm` to evaluate $E_{\alpha,\alpha+1}(B_k)$. The shift parameter is $\gamma = 0.05$, as in [31]. Krylov spaces are built by using a direct approach every time vectors of the form $(A+\gamma I)^{-1}v_j$ are required.
  We stop the iteration for the smallest dimension m such that the distance between y_m and y_{m-1} is smaller than 10^{-7}.

- **SI + CF + PFE**: Shift-and-Invert method with CF approximation and PFE to evaluate $E_{\alpha,\alpha+1}(B_k)e_1$. We proceed as in **SI** + `funm`, the difference being that at each step for $E_{\alpha,\alpha+1}(B_k)e_1$ the expression (16) is used, as for **CF** + PFE.

- **PTR**: *product trapezoidal rule*. It is an implicit product-integration rule based on first degree polynomial interpolants; this formula combines a good accuracy and a reasonable computational cost; it is among the most used methods in the fractional calculus community [9, 23]. We use a time-step $h = 10^{-3}$ since it requires an execution time comparable to that of the other methods.

For the `mlf` code we use a tolerance 10^{-9} since it offers a good compromise between accuracy and speed. Moreover the Cuthill-McKee ordering (Matlab `symrcm`) is used to improve the matrix approaches under investigation. The reported elapsed time includes every part of the methods: the reordering, the computation of the CF coefficients and so on.

The accuracy of the methods

We report the results in [13] on the accuracy of the methods described above. Since the true solutions of our test equations are not explicitly known, we consider as reference solution the one obtained by the PTR approach with $h = 10^{-5}$. This is a very tiny value for the problems under investigation and it requires the computation of 100.000 steps to get the solution at $T = 1$, and about 90 minutes of CPU time (on the PC used for all other tests). It would be thus impracticable to consider smaller steps.

In Tables 1 and 2 we report the errors with respect to the reference solution of the methods under investigation.

The value n refers to the nodes number for the discretization of the space derivatives (thus the matrix A has dimension $n \times n$).

One of the most interesting aspects that Tables 1 and 2 put to light is that the approaches based on matrix functions always reach a high accuracy, comparable to the one of the classical PTR method. Moreover, this last approach is quite always the most expensive and this gap enlarges, according to other numerical tests in [13], when the dimension of the problem or the integration interval gets larger.

TABLE 1. – Comparison for accuracy and execution time (in parenthesis) to evaluate the solution of Example 1 for $T = 1$.

n	α	CF+PFE	SI + Schur	SI + CF+PFE	PTR $h = 10^{-3}$
50	0.5	6.3e-11 (1.29)	1.9e-10 (1.03)	8.5e-09 (1.32)	6.7e-08 (1.16)
	0.8	1.0e-09 (0.59)	8.0e-10 (0.48)	1.3e-09 (0.61)	9.9e-09 (1.10)
100	0.5	8.7e-11 (1.29)	1.1e-09 (1.01)	6.6e-09 (1.32)	9.5e-08 (2.35)
	0.8	1.4e-09 (0.59)	4.4e-10 (0.50)	3.8e-09 (0.61)	1.4e-08 (2.35)

TABLE 2. – Comparison for accuracy and execution time (in parenthesis) to evaluate the solution of Example 2 for $T = 1$.

n	α	β	CF+PFE	SI + Schur	SI + CF+PFE	PTR $h = 10^{-3}$
50	0.5	1.5	2.4e-10 (1.28)	2.4e-09 (1.60)	1.1e-08 (1.33)	1.3e-07 (1.12)
	0.6	1.8	2.7e-11 (1.13)	1.6e-10 (0.98)	8.5e-09 (1.16)	2.7e-08 (1.23)
100	0.5	1.5	3.8e-10 (1.29)	7.8e-09 (2.34)	9.3e-08 (1.35)	1.9e-07 (2.66)
	0.6	1.8	3.6e-11 (1.13)	2.1e-09 (1.09)	2.8e-08 (1.17)	3.8e-08 (2.70)

Efficiency of the methods

Another important factor to compare methods is the efficiency in terms of execution time. To this purpose in [13] test problems of large dimensions have been considered, without considering the PTR method which is not adequate to treat similar situations that, in practice, are the most common in the applications.

Tables 3 and 4 report the execution time and, for methods based on Krylov subspaces, we report in parenthesis the dimension of the space the solution belongs to.

TABLE 3. – Comparison of execution time and iterates (in parenthesis) to evaluate the solution of Example 1 for $T = 1$.

n	α	CF+PFE time	SI+funm time (iterations)	SI+CF+PFE time (iterations)
1000	0.5	1.315	**1.092** (20)	1.398 (19)
	0.8	0.603	**0.543** (18)	0.660 (17)
2000	0.5	1.305	**1.092** (20)	1.455 (19)
	0.8	0.611	**0.607** (18)	0.758 (17)
3000	0.5	**1.325**	1.335 (20)	1.760 (19)
	0.8	**0.635**	0.822 (18)	0.949 (17)

TABLE 4. – Comparison of execution time and iterates (in parenthesis) to evaluate the solution of Example 1 for $T = 1$.

n	α	β	CF+PFE time (avg its.)	SI+funm time (iterations)	SI+CF+PFE time (iterations)
1000	0.5	1.5	**1.880**	3.908 (31)	2.600 (29)
	0.6	1.8	**1.728**	2.194 (23)	2.115 (20)
2000	0.5	1.5	**3.721**	7.449 (31)	6.105 (29)
	0.6	1.8	**3.524**	5.141 (23)	4.978 (21)
3000	0.5	1.5	**7.131**	14.206 (31)	13.029 (30)
	0.6	1.8	**6.757**	10.594 (23)	10.189 (21)

The interesting result is that the execution time of these methods remains quite small; they are indeed effective also for very large problems, whose resolution would be not affordable by the classical methods.

6.2 – *Tests for semilinear FDEs*

The introduction of GETD methods in [14] was supported by several numerical tests aiming to validate the theoretical findings; we report here some of them, and we refer to [14] for details and further examinations.

The test equation is

$$\frac{\partial^\alpha}{\partial t^\alpha}u(t,x) + a\frac{\partial^2}{\partial x^2}u(t,x) = cu(t,x)\frac{\partial}{\partial x}u(t,x) + f(t,x), \quad x \in [0,1], \quad t \geq 0 \tag{18}$$

with $u(t,0) = u(t,1) = 0$, $u(0,x) = x(1-x)$ and

$$f(t,x) = -\big(2a + ce_{\alpha,1}(t;\psi)(2x^3 - 3x^2 + x) + \psi x(1-x)\big)e_{\alpha,1}(t;\psi).$$

The exact solution is $u(t,x) = e_{\alpha,1}(t;\psi)x(1-x)$.

Finite differences are used to discretize the spatial derivatives for a equispaced mesh-grid on $[0,1]$ with mesh-size $\Delta x = 1/(n+1)$; thus the system of semilinear FDEs to solve is

$$D_0^\alpha U(t) + \frac{a}{\Delta x^2}AU(t) = \frac{c}{2\Delta x}G(U(t)) + F(t),$$

with $U(t) = \big(u(t,x_1), u(t,x_2), \ldots, u(t,x_n)\big)^T$,

$$A = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 \end{pmatrix}, \quad g(U) = \begin{pmatrix} U_2U_1 \\ U_3U_2 - U_2U_1 \\ U_4U_3 - U_3U_2 \\ \vdots \\ -U_nU_{n-1} \end{pmatrix}$$

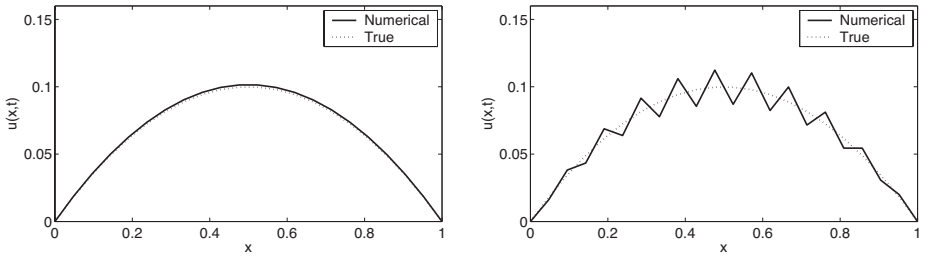

Fig. 1. – Comparison between 1-step GETD (left) and 1-step fractional Adams-Bashfort (right) for the fractional PDE (18).

and $F(t) = \big(f(t,x_1), f(t,x_2), \ldots, f(t,x_n)\big)^T$. For our numerical tests we used $\alpha = 0.7$, $a = -1$, $c = 1$ and $\psi = 1$ with $n = 20$ grid points.

The first plot in Figure 1 represents the exact solution $u(1,x)$ and the numerical solution for $t = 1$ evaluated by the 1-step GETD method (14) with step $h = 0.05$. It is interesting to notice that this quite large step leads to a numerical solution which is very similar to the exact one. The second plot in Figure 1 refers to an explicit product-integration method of the first order, commonly used for FDEs, with step-size $h = 3.685 \times 10^{-5}$. Some noticeable oscillations appear in this case, even though the remarkably smaller step-size has been used, thus confirming the difficulties described in Section 3 when explicit methods are used to solve stiff problems. The value h has been deliberately chosen close to the border of the stability region in order to highlight the difficulties of using explicit methods of traditional type.

7. – The restarted shift-and-invert Krylov method

In some situations even the convergence rate of the SI method, introduced in Section 4, can result slow. To overcome this difficulty, with related problems of memory storage and high computational costs, in [32] we proposed a variant of SI with *restart*. The basic idea is to fix a priori the largest dimension the Krylov subspace can reach to define suitable approximations by working only on spaces of the fixed dimension.

The theory developed in [32] applies to sectorial matrices, that is, for matrices $A \in \mathbb{C}^{N\times N}$ such that for every vector $u \in \mathbb{C}^N$

$$\mathrm{Re}\langle u, Au\rangle \geq 0.$$

Let $\rho(A)$ and $\sigma(A)$ denote the resolvent and the spectrum of A, respectively. Let apply the SI method defined in Section 4 to approximate $f(A)v$ with f as in (16)

and $Z = (I + \gamma A)^{-1}$. Then

$$f(A)v \approx y_m = \sum_{k=1}^{\nu} g_k w_{k,m}, \;\; w_{k,m} = \gamma V_m H_m (\mu_k H_m + I)^{-1} e_1, \;\; \mu_k = \gamma \xi_k - 1$$

with the assumption that $-\mu_k^{-1} \in \rho(H_m)$ for any $k = 1, 2, \ldots, \nu$.

The approximation error for $m \geq 2$, as shown in [29], can be expressed as

$$y - y_m = \sum_{k=1}^{\nu} g_k \eta_{k,m} (I + \gamma A)(\xi_k I + A)^{-1} v_{m+1} \tag{19}$$

with

$$\eta_{k,m} = h_{m+1,m} e_m^T (\mu_k H_m + I)^{-1} e_1.$$

It is thus possible to update y_m by approximating this error to get

$$\widetilde{y_m} = y_m + \gamma \sum_{k=1}^{\nu} g_k \eta_{k,m} v_{m+1};$$

the error now is

$$y - \widetilde{y_m} = -\sum_{k=1}^{\nu} g_k \eta_{k,m} \mu_k (\xi_k I + A)^{-1} v_{m+1}. \tag{20}$$

The restart process is grounded on the fact that both the errors (19) and (20) have partial fraction expansions like (16). In practice, the steps to follow in this procedure are the following:

- Find $y_m \in \mathcal{K}_m(Z, v)$;
- Update it $y^{(1)} \equiv y_m + \gamma \sum_{k=1}^{\nu} g_k \eta_{mk} v_{m+1}$
- Approximate the error $e^{(1)} \equiv -\sum_{k=1}^{\nu} g_k^{(1)} (\xi_k I + A)^{-1} v_{m+1}$ in $\mathcal{K}_m(Z, v_{m+1})$
- Update the approximation to y

$$y^{(2)} \;=\; y^{(1)} - e_m^{(1)} - \gamma \sum_{k=1}^{\nu} g_k^{(1)} \eta_{mk}^{(2)} v_{m+1}^{(2)}$$

which gives an error

$$e^{(2)} = y - y^{(2)} = \sum_{k=1}^{\nu} g_k^{(1)} \eta_{mk}^{(2)} \mu_k (\xi_k I + A)^{-1} v_{m+1}^{(2)}$$

- Approximate

$$e^{(2)} \approx e_m^{(2)} \in \mathcal{K}_m(Z, v_{m+1}^{(2)})$$

and so on.

For the approximation $\widetilde{y_m}$ the useful *a posteriori* error estimate holds [32]

$$\|y - \widetilde{y_m}\| \leq \sum_{k=1}^{v} |g_k \eta_{k,m}| \left\| \mu_k (\xi_k I + A)^{-1} v_{m+1} \right\|$$

which can work as a stopping criterion in the method's implementation with

$$\mu_k (\xi_k I + A)^{-1} v_{m+1} \cong \gamma (v_{m+1} - V_{m+1} (\mu_k H_{m+1} + I)^{-1} e_{m+1}). \tag{21}$$

7.1 – *Application to the Mittag-Leffler function*

In [32] we applied this restart version of SI to the numerical approximation of the ML function for $0 < \alpha < 1$ and $\beta = 1$, A and v real and $\sigma(A) \subset [a, b]$.

To approximate the ML we make use of the quadrature formulas proposed by Weideman e Trefethen [44]; in particular, we consider a parabolic contour

$$z(u) = d(iu + 1)^2, \ d > 0, \ -\infty < u < +\infty$$

which, according to the presentation in ([44], Sec. 3), leads to the approximation

$$E_\alpha(-t^\alpha w) \approx \frac{h}{2\pi i} \sum_{k=-L}^{L} e^{z(u_k)t} z'(u_k) z(u_k)^{\alpha-1} \left(z(u_k)^\alpha + w \right)^{-1}, \quad \text{for } t > 0, L \in \mathbb{N} \tag{22}$$

with $u_k = kh$, for $k = -L, ..., L$. When working in an interval like $[t_0, t_1]$ then

$$h = \frac{\sqrt{8R + 1}}{L} \quad \text{and} \quad d = \frac{\pi L}{4 t_1 \sqrt{8R + 1}}$$

are optimal parameters chosen as to guarantee, for $R = \dfrac{t_1}{t_0}$ and L fixed, an error like $E_L = \mathcal{O}\left(\exp\left(-\dfrac{2\pi L}{\sqrt{8R + 1}} \right) \right)$ for any $t \in [t_0, t_1]$. The formula (22) can be extended also to matrix arguments with a spectrum close to the real axis [44].

Recently an extension of this approach to the more general case $\beta \neq 1$ has been carried over [16] and, by following a similar error analysis, the optimal parameters for the parabolic contour and other coefficients have been detected to reach any accuracy.

For $E_\alpha(-t^\alpha A)v$ formula (22) becomes

$$y_L(t) = \frac{h}{2\pi i} \sum_{k=-L}^{L} e^{z(u_k)t} z'(u_k) z(u_k)^{\alpha-1} \left(z(u_k)^\alpha I + A \right)^{-1} v. \tag{23}$$

It is thus immediate to apply SI to approximate this sum. Moreover, since A and v are real, for the properties of the parabola z and the chosen nodes, formula (23) can be stated in such a way to reduce the addends from $2L + 1$ to $L + 1$.

A deep analysis has been carried out in [32] to fix the shift parameter γ and the conclusion was that it can be useful to use

$$\gamma = d^{-\alpha} \tag{24}$$

which, although not optimal in general, gives good results in many applications and it is independent on the problem. The numerical results presented in [32] show the similarity between the SI convergence with this γ and the one for the optimal γ obtained by applying the a priori error estimates as deduced in [32].

The strategy proposed by Weideman e Trefethen [44] applies to the computation of vectors like $f(A)v$ when f derives from the inversion of the Laplace transform of suitable functions, while the matrix A is chosen such that the linear systems in (23) can be easily solved by direct methods. For our tests the matrix A has the features outlined in [44]: in particular, we consider A as the discretization of the differential operator

$$D = -\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} + 10x\frac{d}{dx},$$

on the square $(0,1) \times (0,1)$, with homogeneous Dirichlet conditions. We use finite differences with constant step $h = \dfrac{1}{n+1}$ in both directions. The resulting matrix A has dimension $N = n^2$, is sparse and has real eigenvalues. The vector v derives from the discretization of the function $v(x,y) = x(1-x)y(1-y)$.

Many applications require the evaluation of $E_\alpha(-t^\alpha A)v$ for several values of t; let assume for example we need this vector for 40 distinct values of t's in the interval $[10^{-3}, 1]$. In practice, we partition this interval in 4 subintervals of the same length. Indeed, by taking into account the error estimate cited above, to reach an accuracy $\mathcal{O}(10^{-7})$, one would consider $L = 230$ terms in (16) when working in $[10^{-3}, 1]$, while only $L = 18$ if the mentioned partition is considered.

Motivated by the results of some numerical tests, we use the γ in (24) for an "intermediate" subinterval, that is, we use the γ for the second subinterval since it (experimentally) turns out to be effective also on the other subintervals [32]. Using just one γ is useful when building the Krylov subspace with a direct method; indeed, the factorization of $A + \gamma I$ can be done just once and reused any time a vector of the form $(A + \gamma I)^{-1} v_j$ is needed. This is the strategy adopted for our tests.

We compare the execution time to compute (23) for three different approaches: when solving the linear systems with the Matlab `backslash` ("Direct"), for SI, for SI with restart for a fixed dimension m ("SI(m)"). For $\alpha = 0.5$ we fix $m = 5$ while for $\alpha = 1$ the fixed dimension is $m = 10$. We impose an accuracy 10^{-9} on the error estimate (21).

TABLE 5. – Comparison of execution time, in seconds, to evaluate (23) in 40 points in $[10^{-3}, 1]$ by solving the linear systems with a direct approach, for SI with restart length m and for standard SI; the used values are $m = 5$ for $\alpha = 0.5$ and $m = 10$ for $\alpha = 1$.

α	Direct	SI(m)	SI
0.5	250.939	**0.268**	0.322
1	285.699	**0.269**	2.433

Results in Table 5 highlight the sharp difference between the "Direct" approach and the other two; the "SI(m)" method is always the fastest one, with an important difference with respect to "SI(m)" especially for $\alpha = 0.5$.

In these numerical tests it was interesting to notice that, in the "Direct" approach, the solution of the complex linear systems was actually the most expensive part, in practice the 99.7% of the total cost, thus confirming the difficulties discussed in Section 4.

REFERENCES

[1] BENZI M. and BERTACCINI D., *Block Preconditioning of Real-Valued Iterative Algorithms for Complex Linear Systems*, IMA Journal of Numerical Analysis, **28**, 598-618 (2008).

[2] BERTACCINI D., *Efficient preconditioning for sequences of parametric complex symmetric linear systems*, Electronic Transactions on Numerical Analysis, **18**, 49-64 (2004).

[3] BERTACCINI D. and POPOLIZIO M., *Adaptive updating techniques for the approximation of functions of large matrices*, preprint (2012).

[4] CODY W. J., MEINARDUS G. and VARGA R. S., *Chebyshev rational approximations to e^{-x} in $[0, +\infty)$ and applications to heat-conduction problems*, J. Approximation Theory, **2**, 50-65 (1969).

[5] DEL BUONO N., LOPEZ L. and PELUSO R., *Computation of the exponential of large sparse skew-symmetric matrices*, SIAM J. Sci. Comput., **27 (1)**, 278-293 (2005).

[6] DEL BUONO N., LOPEZ L. and POLITI T., *Computation of functions of Hamiltonian and skew-symmetric matrices*, Math. Comput. Simulation, **79 (4)**, 1284-1297 (2008).

[7] DENG W., *Numerical algorithm for the time fractional Fokker-Planck equation*, Journal of Computational Physics, **227**, 1510-1522 (2007).

[8] DIETHELM K., *The analysis of fractional differential equations,* Lecture Notes in Mathematics, Springer-Verlag, Berlin *(2010).*

[9] GARRAPPA R., *On linear stability of predictor-corrector algorithms for fractional differential equations*, International Journal of Computer Mathematics, **87 (10)**, 2281-2290 (2010).

[10] GARRAPPA R., *On some generalizations of the implicit Euler method for discontinuous fractional differential equations*, Mathematics and Computers in Simulation, , **95** (2014), 213-228.

[11] GARRAPPA R., *Stability-preserving high-order methods for multiterm fractional differential equations*, International Journal of Bifurcation and Chaos, **22 (4)** (2012), 1-13.

[12] Garrappa R., *A family of Adams exponential integrators for fractional linear systems*, Computers and Mathematics with Applications, (2013), in print, doi: http://dx.doi.org/10.1016/j.camwa.2013.01.022
[13] Garrappa R. and Popolizio M., *On the use of matrix functions for fractional partial differential equations*, Math. Comput. Simulation, **81 (5)**, (2011), 1045-1056.
[14] Garrappa R. and Popolizio M., *Generalized exponential time differencing methods for fractional order problems*, Comput. Math. Appl., **62(3)** (2011), 876-890
[15] Garrappa R. and Popolizio M., *On accurate product integration rules for linear fractional differential equations*, J. Comput. Appl. Math., **235 (5)** (2011), 1085-1097.
[16] Garrappa R. and Popolizio M., *Evaluation of generalized Mittag-Leffler functions on the real line*, Adv. Comput. Math., **39 (1)** (2013), 205-225.
[17] Golub G.H. and Van Loan C.F., *Matrix Computations*, Johns Hopkins Studies in the Mathematical Sciences (1996).
[18] Gorenflo R. and Mainardi F., *Some recent advances in theory and simulation of fractional diffusion processes.*, J. Comput. Appl. Math., **229 (2)** (2009), 400-415.
[19] Higham N. J., *Functions of Matrices: Theory and Computation*, SIAM (2008).
[20] Hochbruck M. and Lubich C., *On Krylov subspace approximations to the matrix exponential operator*, SIAM Journal on Numerical Analysis, **34** (1987), 1911-1925.
[21] Hochbruck M. and Ostermann A., *Exponential integrators*, Acta Numerica, **19** (2010), 209-286.
[22] Kilbas A. A., Srivastava H. M. and Trujillo J. J., *Theory and applications of fractional differential equations*, vol. 204 of North-Holland Mathematics Studies, Elsevier Science B.V., Amsterdam (2006).
[23] Linz P., *Analytical and numerical methods for Volterra equations*, SIAM Studies in Applied Mathematics, Philadelphia, PA (1985).
[24] Lopez L. and Simoncini V., *Analysis of projection methods for rational function approximation to the matrix exponential*, SIAM J. Numer. Anal., **44 (2)** (2006), 613-635.
[25] Lopez L. and Simoncini V., *Preserving geometric properties of the exponential matrix by block Krylov subspace methods*, BIT, **46 (4)** (2006), 813-830.
[26] Magnus A. P., *Asymptotics and super asymptotics for best rational approximation error norms to the exponential function (the "1/9" problem) by the Carathéodory-Fejér method*, Nonlinear numerical methods and rational approximation, **296 (II)** (1994), 173-185.
[27] Moler C. and Van Loan C., *Nineteen dubious ways to compute the exponential of a matrix*, SIAM Review, **20 (4)** (1978), 801-836.
[28] Moler C. and Van Loan C., *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later*, SIAM Review, **45 (1)** (2003), 3-49.
[29] Moret I., *Rational Lanczos approximations to the matrix square root and related functions*, Numerical Linear Algebra with Applications, **16** (2009), 431-445.
[30] Moret I. and Novati P., *RD-Rational Approximations of the Matrix Exponential*, BIT, Numerical Mathematics, **44** (2004), 595-615.
[31] Moret I. and Novati P., *On the convergence of Krylov subspace methods for matrix Mittag-Leffler functions*, SIAM Journal on Numerical Analysis, **49** (2011), 2144-2164
[32] Moret I. and Popolizio M., *The restarted shift-and-invert Krylov method for matrix functions*, Numerical Linear Algebra with Applications, **21** (2014), 68-80.
[33] Petrushev P.P. and Popov V.A., *Rational approximation of real function*, Cambridge University Press, Cambridge (1987).
[34] Podlubny I., *Fractional differential equations, Mathematics in Science and Engineering,* Academic Press Inc., San Diego, CA (1999).

[35] PODLUBNY I. and KACENAK M., *Matlab implementation of the Mittag-Leffler function*, available online: http://www.mathworks.com (2005).
[36] POLITI T. and POPOLIZIO M., *Schur Decomposition Methods for the Computation of Rational Matrix Functions*, Computational science-ICCS 2006. Part IV, Springer, **3994**, 708-715 (2006).
[37] POPOLIZIO M., *Tecniche di accelerazione per approssimare l'esponenziale di matrice*, La Matematica nella Società e nella Cultura, Rivista dell'Unione Matematica Italiana, Serie I, Vol. II, Agosto (2009), 275-278.
[38] POPOLIZIO M. and SIMONCINI V., *Acceleration Techniques for Approximating the Matrix Exponential Operator*, SIAM J. Matrix Analysis and Appl., **30** (2008), 657-683.
[39] SAAD Y., *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM Journal on Numerical Analysis, **29** (1992), 209-228.
[40] TREFETHEN L. N., *Rational Chebyshev approximation on the unit disk*, Numer. Math., **37 (2)** (1981), 297-320
[41] TREFETHEN L. N., *Circularity of the error curve and sharpness of the CF method in complex Chebyshev approximation*, SIAM J. Numer. Anal., **20 (6)** (1983), 1258-1263.
[42] TREFETHEN L. N., WEIDEMAN J. A. C. and SCHMELZER T., *Talbot quadratures and rational approximations*, BIT, **46 (3)** (2006), 653-670.
[43] VAN DEN ESHOF J. and HOCHBRUCK M., *Preconditioning Lanczos approximations to the matrix exponential*, SIAM Journal on Scientific Computing, **27** (2006), 1438-1457.
[44] WEIDEMAN J. A. C. and TREFETHEN L. N., *Parabolic and hyperbolic contours for computing the Bromwich integral*, Mathematics of Computation, **78** (2007), 1341-1358.

Marina Popolizio
Dipartimento di Matenatica e Fisica "Ennio De Giorgi"
Università del Salento, Via per Arnesano, 73100 Lecce
e-mail: marina.popolizio@unisalento.it

Received December 8, 2012 and in revised form December 8, 2013