A Parallel Numerical Scheme for Linear Differential Equations

Frédéric Hecht, Sidi-Mahmoud Kaber

A Parallel Numerical Scheme for Linear Differential Equations

F. Hecht, S.-M. Kaber
Laboratoire J.-L. Lions,
Sorbonne Université,
Université Paris Cité, CNRS,
LJLL, F-75005 Paris, France

July 18, 2023

Abstract

New approximations of the matrix φ functions are developed. These approximations are rational functions of a specific form allowing simple and accurate schemes for linear systems. Furthermore, these approximations are fully parallelizable. Several tests show the efficiency of the method and its good parallelization properties.

Keywords : matrix exponential, parallel computing, exponential integrators.

1 Introduction

The main objectives in scientific computing are accuarcy and saving of computational time. We are mostly interested here in the second objective. For a matrix $A \in \mathcal{M}_d(\mathbb{C})$, we propose to use rational approximations of its exponential $\exp(A)$ and some related functions to design an accurate and parallelizable numerical scheme to solve linear differential equations. In many problems, what is truly important is rather the action of $\exp(A)$ on a vector than the computation of the exponential itself. For any $v \in \mathbb{R}^d$, $\exp(A)v$ is approximated by $\mathscr{R}_{n,0}(A)v = \sum_{k=1}^{n} a_k v_k$ with $\mathscr{R}_{n,0}(A)$ the rational approximation of $\exp(A)$ defined in [5]: each vector v_k is the solution of the linear system $(A + \theta_k I)v_k = v$ where a_k and θ_k are fixed complex numbers depending only on n and not on A, nor on v. It is key to note that each vector v_k can be computed separately from the others by assigning to each node a linear system to solve. This is the essential property of the method. If we have at our disposal n processors, we could compute on each

processor one vector v_k independently from the others. Hence the cost of the computation of $\mathscr{R}_{n,0}(A)v$ is that of solving just one linear system $(A+\theta_k I)x = v$ if we neglect computational time to sum up the n vectors v_k. We are mainly interested in negative semidefinite symmetric matrices $A \in \mathcal{M}_d(\mathbb{R})$ stemming from the spatial discretisation of parabolic Partial Differential Equationsent and the associated evolution equations. It is a question of computing an approximation of the solution $u : I = [0,T] \to \mathbb{R}^d$ to the linear differential equation

$$u'(t) = Au(t) + f(t) \tag{1}$$

with a source term $f : I \to \mathbb{R}^d$ and a square matrix A that do not depend on the variable t. The initial condition is

$$u(0) = u_0 \in \mathbb{R}^d. \tag{2}$$

This equation plays a key role in many physical problems. Its solution is given by Duhamel's formula

$$u(t) = \exp(tA)u_0 + \int_0^t \exp((t-s)A)f(s)ds. \tag{3}$$

For polynomials f, the last integral is a linear combination of generalized moments of the exponential. In order to get a good approximation of $u(t)$, it is necessary not only to compute accurately the exponential, but also some of its moments. This is the heart of the exponential integrators schemes. For example, if the source term is linear $f = f_0 + tf_1$, the solution of problem (1)-(2) is

$$u(t) = \varphi_0(tA)u_0 + t\varphi_1(tA)f_0 + t^2\varphi_2(tA)f_1, \tag{4}$$

with the "φ functions" defined for $z \in \mathbb{C}$ by $\varphi_0(z) = \exp(z)$ and $\varphi_1(z) = \int_0^1 \exp(\theta z)\, d\theta = (\exp(z)-1)/z$, $\varphi_2(z) = \int_0^1 \theta \exp(\theta z)\, d\theta = (\exp(z)-1-z)/z^2$. The idea in exponential time schemes is to compute $u(t)$ using some approximations of the φ functions. If $\tilde{\varphi}_\ell$ denotes an approximation of φ_ℓ, we define an approximation of the solution by $u(t) \simeq \tilde{\varphi}_0(tA)u_0 + t\tilde{\varphi}_1(tA)f_0 + t^2\tilde{\varphi}_2(tA)f_1$. Therefore, we could compute accurately the solution of the problem if accurate approximations of the φ functions are available. For a constant source terme, relation (4) is sometimes written $u(t) = u_0 + t\varphi(tA)(f + Au_0)$ with φ denoting φ_1. This function φ has been fully studied in a number of papers. More generally, the φ functions, denoted φ_ℓ and defined in (5), play a major role in the design of numerical schemes for stiff problems. In [10], $\varphi(A)$ is computed as a bloc of the exponential of a larger matrix (twice the dimension of A) that looses nice propreties of the matrix A such as symmetry. In many references, the computation of $\exp(A)v$ with a unitary vector $v \in \mathbb{R}^d$ is approximated by $p_{m-1}(A)v$ with p_{m-1} a polynomial of degree less or equal to m. Since such an approximation belongs to the Krylov subspace $K_m(A,v) = \text{span}\{v, Av, \cdots, A^{m-1}v\}$, the use of the Arnoldi or Lanczos algorithm is necessary. Such algorithmes produce, at each iteration

m, an upper Hessenberg matrix H_m of size $m \times m$ and a matrix V_m of size $d \times m$ whose columns form an orthonoral basis of $K_m(A, v)$ such that $H_m = V_m^T A V_m$. At the end of the iterations (according to a stopping criterion), $\exp(A)v$ is approximated by $\alpha V_m \exp(H_m) e_1$, with e_1 the first vector of the canonical basis of $\mathbb{R}^m$ (consult [10]). The computation of $\exp(A)$ is then reduced to the computation of $\exp(H_m)$. Since m is small (this is the case in many examples reported), the computation of $\exp(H_m)$ could be done using the eigendecomposition of the matrix H_m. Other alternatives exist: $\exp(H_m) \simeq p(H_m)$ where p is the Hermite interpolant of the exponential function on the spectrum of H_m (taking into account the multiplicities of the eigenvalues), or p is the best uniform approximation of the exponential on an intervall that contains the spectrum of H_m, ...The method presented in [6] uses Krylov subspaces method combined with a recursive scheme allowing the computation of $\varphi_1(kz)$ ($k \in \mathbb{N}, z \in \mathbb{C}$) in terms of z and $\varphi_1(z)$, based on the observation that $\varphi_1(2z) = (exp(x) + 1)\varphi_1(x)/2$. In [7], $\varphi_1(A)$ is computed, for a symmetric matrix A, via an orthonormal reduction of A to a tridiagonal matrix T. Then, $\varphi_1(T)$ is approximated by $R(T)$ with R the best rational approximation of φ_1 on $]-\infty, 0]$. Function R is written in its partial fraction decomposition form. We also use the partial fraction decomposition in our approximation. In rational Krylov methods, $exp(H_m)v$ is approximatex by $q^{-1}(H_m)p(H_m)v$ wher p and q are polynomials. The poles of the approximation should not intersect the spectrum of H_m. See the analysis in [4], [3] among other references. In [12], rational approximations and contours integrals are used to compute general φ_ℓ. Theses approximations share a common property with the method developed in this paper: the use of the same set of poles to compute all the φ functions. As mentionned in [12], this enables to reduce the work dramatically. We refer to the survey papers [9] et [8] to understand the need of an efficient computation of $\varphi(A)v$ in the design of exponential integrators. We define in this paper new approximations of functions φ_ℓ and use theses approximations to define a parallel numerical scheme computing directly the solution at time t using an accurate approximation of (4). As an illustration, consider the linear source term case. We put forward the following approximation of the solution (4): $u(t) \simeq \mathscr{R}_{n,0}(tA)u_0 - t\mathscr{R}_{n,0}(tA)f_0 - t^2\mathscr{R}_{n,1}(tA)f_1$ with $\mathscr{R}_{n,\ell}$ the approximation of φ_ℓ defined by $\mathscr{R}_{n,\ell}(z) = \sum_{k=1}^{n} \frac{a_{k,\ell}}{z+\theta_k}$, and $a_{k,\ell} = \frac{a_{k,0}}{(-\theta_k)^\ell}$. In other words, $u(t)$ is aproximated by $\sum_{k=1}^{n} a_{k,0}\omega_k$, with ω_k the unique solution to the linear system $(tA+\theta_k I)\omega_k = u_0 - \frac{t}{\theta_k}f_0 + \frac{t^2}{\theta_k^2}f_1$. The idea is to carry out the computation of each v_k on a different processor. Indeed, each vector v_k could be computed independently from the others by solving the linear complex system $(A + \theta_k I)v_k = v$. Then, the n vectors are combined to form $\mathscr{R}_{n,\ell}(A)v$. The total computing time cost is just the cost of solving one linear system if we neglect the summation of scalars and vectors.

The paper is organized as follows. In section 2, we define rational approximations of functions φ_ℓ. The resulting numerical scheme is the subject of section 4. We present in section 5 several applications for both Ordinary Differential Equations (ODE) and

Partial Differential Equations (PDE). Finally, the last section is devoted to conclusion and possible extension of the numerical scheme followed by brief concluding remarks.

2 The φ functions

For polynomial f, the solution of (1)-(2) is explicitely known as a combination of the so-called φ functions defined for $z \in \mathbb{C}$ by $\varphi_0(z) = \exp(z)$ and for $\ell \geq 1$

$$\varphi_\ell(z) = \frac{\exp(z) - \exp_{\ell-1}(z)}{z^\ell} \tag{5}$$

with $\exp_k$ the finite Taylor series of the exponential of order k. The values at the origin of these functions and their derivatives are

$$\varphi_\ell(0) = \frac{1}{\ell!}, \quad \varphi'_\ell(0) = \frac{1}{(\ell+1)!} \quad (\ell \geq 0). \tag{6}$$

The following definitions of the entire functions φ_ℓ are equivalent to definition (5).

- Recurrence relation : $\varphi_0(z) = \exp(z)$ and for $\ell \geqq 0$,

$$\varphi_{\ell+1}(z) = \frac{\varphi_\ell(z) - \varphi_\ell(0)}{z}. \tag{7}$$

- Power series : for $\ell \geq 0$

$$\varphi_\ell(z) = \sum_{k \geq 0} \frac{z^k}{(k+\ell)!}. \tag{8}$$

There are also several other equivalent definitions: in terms of special functions (hypergeometric functions, Mittal-Leffler functions, ...), definition by real integrals or complex contours. For a square matrix A, we choose to define matrix $\varphi_\ell(A)$ using the power series definition (8)

$$\varphi_\ell(A) = \sum_{k \geq 0} \frac{A^k}{(k+\ell)!}.$$

Note that for any diagonal matrix A, $\varphi_\ell(A)$ is also diagonal and $(\varphi_\ell(A))_{p,p} = \varphi_\ell(A_{p,p})$. Note also that for all similar matrices B and $A = PBP^{-1}$, we have $\varphi_\ell(A) = P\varphi_\ell(B)P^{-1}$. There is a matrix recurrence relation for φ functions similar to the scalar relation (7):

$$A\varphi_{\ell+1}(A) = \varphi_\ell(A) - \varphi_\ell(0)I \qquad (\ell \geqq 0). \tag{9}$$

We conclude this section by giving the explicit expression of the solution of ODE (1) in the case of polynomial source terms. It involves functions φ_ℓ as stated earlier.

Lemma 1 *If the right-hand side in (1) is polynomial* $f(t) = \sum_{j=0}^{J} \frac{t^j}{j!} f_j$*, with constant vectors* $f_\ell \in \mathbb{R}^d$*, then the solution of (1)-(2) is*

$$u(t) = \exp(tA)u_0 + \sum_{j=0}^{J} t^{j+1}\varphi_{j+1}(tA)f_j. \tag{10}$$

`Proof.` Indeed $t \longmapsto u_j(t) = t^{j+1}\varphi_{j+1}(tA)f_j$ is a solution of the ODE with $f(t) = \frac{t^j}{j!}f_j$ and a null initial condition. ■

The idea now is to compute the solution using accurate approximations of functions φ_ℓ. As stated in the Introduction section, several methods exist to compute these functions using different mathematical tools. We recall in the next section some propreties of the rational approximation of φ_0 analysed in [5] and define new approximations of φ_ℓ. In what follows n denotes an even number and $(\theta^{(n)})_{i=1}^n$ are the n zeros of $\exp_n$, the truncated Taylor series of $\exp(z)$, $z \in \mathbb{C}$. These zeros are pairwise complex conjugate and none of which is a real number.

3 Approximation of the φ functions

The following rational approximations of $\exp(z)$ is straightforward:

$$\exp(z) = \varphi_0(z) \simeq \mathscr{R}_{n,0}(z) := \frac{1}{\exp_n(-z)} = \sum_{k=1}^{n} \frac{a_{k,0}}{z + \theta_k}, \qquad (\forall z \in \mathbb{C})$$

with complex numbers $a_{k,0}$ evaluated from the θ_k's. From now on, we consider only matrices $A \in \mathcal{M}_d(\mathbb{C})$ such that

$$\text{all matrices } A - \theta_k I \text{ are nonsingular,} \tag{H1}$$

meaning that their spectrum does not contain any root of any $\exp_n$. This is the case for Hermitian matrices and also for every matrix as long as n is large enough. We will make use of the following approximation fully analysed in [5].

$$\exp(A) = \varphi_0(A) \simeq \mathscr{R}_{n,0}(A) := \sum_{k=1}^{n} a_{k,0}(A + \theta_k I)^{-1}. \tag{11}$$

For diagonalisable matrices $A = PDP^{-1}$, there is a naturel bound for the approximation error

$$\|\exp(A) - \mathscr{R}_{n,0}(A)\|_2 \leq \kappa_2(P) \max_{\lambda \in Sp(A)} |\exp(\lambda) - \mathscr{R}_{n,0}(\lambda)|,$$

with $\kappa_2(P) = \|P\|_2 \, \|P^{-1}\|_2$ the condition number of the change-of-basis matrix P. For Hermitian matrices, the approximation error is precisely

$$\|\exp(A) - \mathscr{R}_{n,0}(A)\|_2 = \max_{\lambda \in Sp(A)} |\exp(\lambda) - \mathscr{R}_{n,0}(\lambda)|.$$

Thus, if the spectrum of matrix A is included in a region Γ of the real line where the approximation of the exponential is *very accurate*, *i.e.* there exists a sequence $(\varepsilon_n)_{n\geq 0}$ that goes fastly to 0 such that $\max_{z\in\Gamma} |\varphi_0(z) - \mathscr{R}_{n,0}(z)| \leq \varepsilon_n$, then $\mathscr{R}_{n,0}(A)$ is a *very accurate* approximation of $\exp(A)$ since for Hermitian matrices $\|\exp(A) - \mathscr{R}_{n,0}(A)\|_2 \leq \varepsilon_n$. Note that for non-Hermitian matrices, the condition number $\kappa_2(P)$ may be too large and destroys the accuracy. Convergence is indeed geometric for Hermitian matrices with spectrum in the negative real axis. If a part of the spectrum is positive, a shift is necessary to get a good approximation. See the end of this Section.

Scalar case. To define the approximations of φ_ℓ, we draw on definition (5) of the exponentional integrators.

Definition 1 *For integer ℓ $(1 \leq \ell \leq n+1)$, we define rational functions $\mathscr{R}_{n,\ell}$ by*

$$\mathscr{R}_{n,\ell}(z) = \frac{\mathscr{R}_{n,0}(z) - \exp_{\ell-1}(z)}{z^\ell}, \qquad z \in \mathbb{C}. \tag{12}$$

Indeed, Taylor series of $\mathscr{R}_{n,\ell}(z)$ shows that theses functions are defined on the whole complex plane (see (15) below). The values of functions $\mathscr{R}_{n,\ell}$ and their derivatives at the origin are (compare with (6))

$$\mathscr{R}_{n,\ell}(0) = \frac{1}{\ell!} \, (0 \leq \ell \leq n), \quad (\mathscr{R}_{n,\ell})'(0) = \frac{1}{(\ell+1)!} \, (0 \leq \ell \leq n-1). \tag{13}$$

Hence $\mathscr{R}_{n,\ell}$ coincide with φ_ℓ at the origin for $\ell = 0, \cdots, n$. Their derivatives coincide also for $\ell = 0, \cdots, n-1$. The error $\varphi_1(z) - \mathscr{R}_{n,1}(z)$ is displayed in Figure 1. The shape of the curves are very similar to the case $\ell = 0$ analysed in [5]. For x small enough ($|x| < 10^{-10}$ in this figure), we set $\varphi_\ell(x) = \varphi_\ell(0) = 1/(\ell!)$ to eliminate the instabilities, otherwise we use the following reccurence relation

$$z \, \mathscr{R}_{n,\ell+1}(z) = \mathscr{R}_{n,\ell}(z) - \mathscr{R}_{n,\ell}(0), \qquad (0 \leq \ell \leq n). \tag{14}$$

Rational functions $\mathscr{R}_{n,\ell}$ are indeed approximations of functions φ_ℓ as it is shown in next Proposition.

Proposition 1 *For $0 \leq \ell \leq n+1$, we have*

$$\mathscr{R}_{n,\ell}(z) = \varphi_\ell(z) + \sum_{k=n+1-\ell}^{+\infty} (\lambda_{n,\ell+k} - 1) \frac{z^k}{(k+\ell)!}, \tag{15}$$

with $\lambda_{n,k} = (\frac{1}{\exp_n(-x)})^{(k)}(0)$.

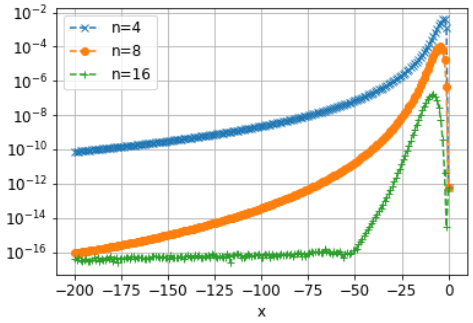

Figure 1: *Approximation of* φ_1 *by* $\mathscr{R}_{n,1}$ *on* $[-200, 0]$*: pointwise error* $|\mathscr{R}_{n,1}(x) - \varphi_1(x)|$ *for* $n = 4, 8, 16$.

`Proof.` For $\ell = 0$, see [5]. For $\ell \geq 1$, use definitions (5) and (12) of φ_ℓ and $\mathscr{R}_{n,\ell}$. ■

Thus, $\mathscr{R}_{n,\ell}$ is an approximation of function φ_ℓ whose accuracy decreases as ℓ increases: $\mathscr{R}_{n,0}$ is a n-order approximation of φ_0 in a neigherbohood of the origin in the sense that $\mathscr{R}_{n,0}(z) - \exp(z) = \mathcal{O}(z^{n+1})$ while $\mathscr{R}_{n,n}$ is only a first-order approximation of φ_n. Fortunately, it is not essential to insure high order approximation of all φ_ℓ since the usual numerical schemes use only φ_ℓ with small values of ℓ. For example, the forth-order exponential Rosenbrock verifier scheme make use of φ_ℓ for $\ell \leq 4$ only. We should keep in mind that only small values of ℓ are necessary for numerical applications. Let us now present a key observation: it turns out that the partial fraction decomposition of $\mathscr{R}_{n,\ell}$ and $\mathscr{R}_{n,0}$ are very similar.

Proposition 2 *The poles of* $\mathscr{R}_{n,\ell}$ *are those of* $\mathscr{R}_{n,0}$ *and the following decomposition holds*

$$\mathscr{R}_{n,\ell}(z) = \sum_{k=1}^{n} \frac{a_{k,\ell}}{z + \theta_k}, \qquad a_{k,\ell} = \frac{a_{k,0}}{(-\theta_k)^\ell}.$$

`Proof.` Indeed, the relation is true for $\ell = 0$ (see [5]). For $1 \leq \ell \leq n$, it results from (14). ■

As mentioned in [12], using a set of common poles for all φ_ℓ functions is an attractive option, in particular for the computation of the action of a linear combination of $\varphi_\ell(A)$ on a fixed vector. We illustrate this in Section 4. Our idea in the design of exponential integrators is to replace $\varphi_\ell(z)$ by $\mathscr{R}_{n,\ell}(z)$ in the explicit expression of the solution of the ODE. There are two situations where an approximation error can easily be derived: for z near the origin or z in the negative real half-axis.

Proposition 3 (Approximation error, scalar case) *Let* $0 \leq \ell \leq n$. *Near the origin, we have* $|\mathscr{R}_{n,\ell}(z) - \varphi_\ell(z)| = \mathcal{O}(z^{n+1-\ell})$ *and for negative* $x \leq -\varrho < 0$, $|\mathscr{R}_{n,\ell}(x) - \varphi_\ell(x)| \leq \frac{1}{2^n \varrho^\ell}$.

`Proof.` Combining (6), (7), (13), and (14), we get $z^\ell(\mathscr{R}_{n,\ell}(z) - \varphi_\ell(z)) = \mathscr{R}_{n,0}(z) - \exp(z)$. The results derive then from the approximation of $\exp(z)$ by $\mathscr{R}_{n,0}(z)$ (see [5] using results from [2]). Outside the disk $|z| \geq \varrho > 0$, we have $|\mathscr{R}_{n,\ell}(z) - \varphi_\ell(z)| \leq \frac{1}{\varrho^\ell} |\mathscr{R}_{n,0}(z) - \exp(z)|$. ■

Matrix case. Partial fraction decomposition of $\mathscr{R}_{n,\ell}$ is the basis of the parallel computation of $\mathscr{R}_{n,\ell}(A)$ and $\mathscr{R}_{n,\ell}(A)v$ for any vector v. For square matrix that satisfies the hypothesis (H1), we use Proposition 2 to define the following approximation

$$\varphi_\ell(A) \simeq \mathscr{R}_{n,\ell}(A) := \sum_{k=1}^{n} a_{k,\ell}(A + \theta_k I)^{-1}.$$

For diagonal matrix A, $\mathscr{R}_{n,\ell}(A)$ is a diagonal matrix too and $(\mathscr{R}_{n,\ell}(A))_{p,p} = \mathscr{R}_{n,\ell}(A_{p,p})$. For similar matrices A and $B = PAP^{-1}$, we have $\mathscr{R}_{n,\ell}(B) = P\mathscr{R}_{n,\ell}(A)P^{-1}$. There is a reccurence relation for $(\mathscr{R}_{n,\ell}(A))_\ell$ similar to reccurence relation (9) for $(\varphi_\ell(A))_\ell$. It follows from the decomposition in Proposition 2:

$$A\mathscr{R}_{n,\ell+1}(A) = \mathscr{R}_{n,\ell}(A) - \mathscr{R}_{n,\ell}(0)I, \qquad (0 \leq \ell \leq n). \tag{16}$$

We now see how to transfer the approximation error results from the scalar case in Proposition 3 to the matrix case. For any diagonalisable matrix $A = PDP^{-1}$, we have $\mathscr{R}_{n,\ell}(A) - \varphi_\ell(A) = P(\mathscr{R}_{n,\ell}(D) - \varphi_\ell(D))P^{-1}$. If in addition A is Hermitian, then

$$\|\mathscr{R}_{n,\ell}(A) - \varphi_\ell(A)\|_2 = \|\mathscr{R}_{n,\ell}(D) - \varphi_\ell(D)\|_2 = \max_{\lambda \in Sp(A)} |\mathscr{R}_{n,\ell}(\lambda) - \varphi_\ell(\lambda)|\,.$$

On the other hand, using recurrence relations (9), (16) and formulas (6), (13), we obtain for $0 \leq \ell \leq n$

$$D^\ell(\mathscr{R}_{n,\ell}(D) - \varphi_\ell(D)) = \mathscr{R}_{n,0}(D) - \varphi_0(D).$$

From theses relations, we deduce the following approximation result.

Proposition 4 (Approximation error, matrix case) *For nonsingular Hermitian matrices, we have*

$$\|\mathscr{R}_{n,\ell}(A) - \varphi_\ell(A)\|_2 = \max_{\lambda \in Sp(A)} \left| \frac{\mathscr{R}_{n,0}(\lambda) - \exp(\lambda)}{\lambda^\ell} \right|$$

for $0 \leq \ell \leq n$.

We are mainly interested in matrices A with negative spectrum (arising from space-discretization of parabolic equations). For nonsingular matrices with spectral abscissa $\alpha(A) := \max_{\lambda \in \mathrm{Sp}(A)} \mathrm{Re}(\lambda) < 0$, we have

$$\|\mathscr{R}_{n,\ell}(A) - \varphi_\ell(A)\|_2 \leq \frac{1}{2^n |\alpha(A)|^\ell}.$$

For singular matrices, the spectrum is splitted into two parts $\Lambda_1 = \{\lambda < -\varepsilon\}$ and $\Lambda_2 = \{-\varepsilon \leq \lambda \leq 0\}$ with $\varepsilon > 0$. On Λ_2, the error behaves like $C^{ste}\varepsilon^{n+1-2\ell}$ (see Proposition 3).

If a part of the spectrum of the Hermitian matrix A is positive, there is no guarantee that $\mathscr{R}_{n,0}(A)$ is an suitable approximation of $\exp(A)$. However the shifted matrix $A-cI$ with $c \geq \alpha(A)$ satisfies the hypothesis (H1) and should be well approximated. This leads to the following approximation: $\exp(A) \simeq e^c\,\mathscr{R}_{n,0}(A - cI)$.

4 The numerical scheme

We suppose in this section that the symmetric matrix A has a negative spectrum. The goal is to compute $\varphi_\ell(A)v$ for $v \in \mathbb{R}^d$ without explicitely forming the matrix $\varphi_\ell(A)$. , we propose the following approximation of $\varphi_\ell(A)v$

$$\varphi_\ell(A)v \simeq \mathscr{R}_{n,\ell}(A)v = \sum_{k=1}^{n} a_{k,\ell} v_k, \tag{17}$$

with vectors v_k defined by $(A + \theta_k I)v_k = v$. The idea is to carry out the computation of each v_k on a different processor. Indeed, each vector v_k could be computed independently from the others by solving the linear complex system $(A + \theta_k I)v_k = v$. Then, the n vectors are combined to form $\mathscr{R}_{n,\ell}(A)v$. Computing $\mathscr{R}_{n,\ell}(A)v$ this way is as expensive as solving a single linear system if we neglect the cost of umming up in (17). In practice However, the interconnections between processors are not always neglectible. To compute the action of matrix $\varphi_\ell(A)$ on a linear combination of vectors, we use the approximation

$$\varphi_\ell(A)(\sum_{p=1}^{P} \alpha_p v_p) \simeq \sum_{p=1}^{P} \alpha_p \mathscr{R}_{n,\ell}(A)v_p = \sum_{p=1}^{P} \alpha_p (\sum_{k=1}^{n} a_{k,\ell}) v_{p,k}.$$

Each vector $v_{p,k}$ is the solution to the linear system $(A + \theta_k I)v_{p,k} = v_p$. The $n \times P$ vectors are computed in parallel leading to a total computing time cost that is just the cost of solving one linear system if we neglect the summation of scalars and vectors. To compute the action of a linear combination of matrices $\varphi_\ell(A)$ on the same vector v, we make following approximation: for $X = \sum_{\ell=1}^{J} \alpha_\ell \varphi_\ell(A)$ and vector v,

$$\big[\sum_{\ell=1}^{J} \alpha_\ell \varphi_\ell(A)\big] v \simeq \sum_{k=1}^{n} \big(\sum_{\ell=1}^{J} \alpha_\ell a_{k,\ell}\big) v_k.$$

Here again, the total cost is essentially that of solving one linear system. Note that the sum delimited by parenthesis could be precomputed if necessary.

A parallel scheme. The aim is not to discretise the differential equation but to compute directly an approximation of the solution at the prescribed final time without passing through intermediate times, unlike usual numerical time integration schemes. It is not our purpose to study here the case of a general analytical source term $f(t) = \sum_{\ell=0}^{+\infty} \frac{t^j}{j!} f_j$ where the solution involves a infinite number of φ_ℓ. We only consider the polynomial case. Following Lemma 1, we define the approximation

$$u(t) \simeq \sum_{k=1}^{n} a_{k,0}\omega_k \tag{18}$$

with ω_k the unique solution to the linear system

$$(tA + \theta_k I)\omega_k = u_0 + \sum_{\ell=0}^{J} \left[\frac{t}{-\theta_k}\right]^{\ell+1} f_j. \tag{19}$$

Note that matrices $tA + \theta_k I$ are always nonsingular for any symmetric matrix A. Of course, each ω_k will be computed separately from the others leading to savings in computational time as illustrated by the numerical tests in the next section.

5 Applications

We consider only polynomial source terms. We are interested in the accuracy of the approximation of the solution at time t given by (18)-(19). The vector norm used to compute the errors is $\|w\|^2 = \frac{1}{d}\sum_{\ell=1}^{d} |w_i|^2$. We also investigate the computing time. We present three tests: the first one is an ODE with an affine source term. The two others are differential equations obtained from the space discretisation of a Partial Differential Equations (PDE). Namely, the heat equation. We consider firstly its one-dimensional version, then the two-dimensional one. All computations are done using tools from `Python`'s librairies and the parallelization is only simulated as we call "parallel computing time" the time used to compute one vector ω_k in (19) since each of these n vectors could be computed separately. It is important to realize that inceasing n increases the accuracy without increasing the computational time since the vectors ω_k are computed in parallel. This is true but only in exact arithmetic. A detailed analysis shows that it is not recommended to use large values of n unless computations are done using more than the usual double precision format. Hence we shall limit $n \simeq 30$. The problems arising for larger values of n are due to the accuracy of the computations of the poles θ_k, the partial fraction decomposition in floating-point arithmetic, ... See [5].

Test 1. Ordinary Differential Equations. Consider (1) with an affine source term $f(t) = f_0 + tf_1(t)$ and a tridiagonal matrix $A = -\frac{1}{h^2}\texttt{tridiag}(-1, 2, -1) \in \mathcal{M}_d(\mathbb{R})$,

with $h = 1/(d+1)$. Note that the spectrum of matrix tA is negative for all $t \geq 0$. The initial condition u_0 as well as the vectors f_0 and f_1 are all constant equal to $(1, \cdots, 1)^T \in \mathbb{R}^d$. The solution of this problem is $u(t) = \exp(tA)u_0 + t\varphi_1(tA)f_0 + t^2\varphi_2(tA)f_1 = \exp(tA)(u_0 - g_0 - g_2) + (g_0 + g_2 + tg_1)$ whith vectors g_0, g_1, and g_2 defined by $Ag_0 = -f_0$, $Ag_1 = -f_1$, and $Ag_2 = -g_1$. For t large enough, the solution behaves like $g_0 + g_2 + tg_1$. Hence, there is no steady state. For different times t corresponding to a transient state and the asymptotic one, we consider the approximation $u(t) \simeq \mathscr{R}_{n,0}(tA)u_0 - t\mathscr{R}_{n,0}(tA)f_0 - t^2\mathscr{R}_{n,1}(tA)f_1$ and display on Figure 2 the norm of the error as a function of the truncation level n and the dimension of the problem d. The exact solution is computed using function `expm` of `Python`'s librairy `Numpy`. As expected, accuracy increases with n. It is worth pointing out that the accuracy does not depend on the dimension d. Indeed, when the dimension increases, the spectrum of the matrix becomes more and more negative (containing large negative eigenvalues) which makes the approximation more effective, see Figure 1. We note that $n = 32$ is more than enough for pratical purposes.

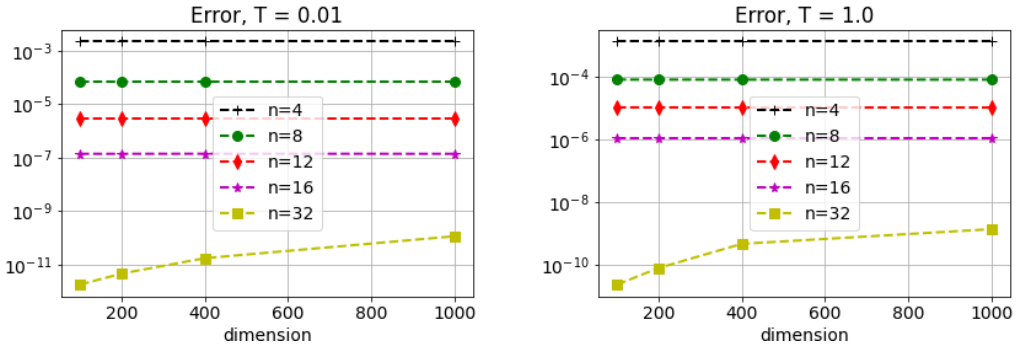

Figure 2: *Test 1. Accuracy of the approximation of the solution as a function of the truncation level n and the dimension of the problem d.*

Test 2. Space discretisation of a one-dimensional Partial Differential Equations. The problem is now to seek $v(t, x)$ solution of the one-dimensional heat equation

$$v_t(t, x) - v_{xx}(t, x) = f(x), \qquad t > 0,\ x \in \Omega =]0, \pi[,$$

with zero Dirichlet boundary conditions. Let $\psi_n(x) = \sin(nx)$ be the eigenfunctions and $\lambda_n = n^2$ the associated eigenvalues of the problem. For initial condition $v_0 = \psi_n$ and source term $f = \lambda_m\psi_m$, the solution is $v(t, x) = \exp(-\lambda_n t)\psi_n(x) + (1 - \exp(-\lambda_m t))\psi_m(x)$. It goes from ψ_n to ψ_m ($n \neq m$) as t goes from 0 to $+\infty$. Space discretisation by the usual second order centered Finite Difference scheme leads to the ODE

$$u_h'(t) = A_h u_h(t) + f_h$$

where A_h is the matrix of the previous test, $f_h = (f(x_j))_{j=1}^d$ with $x_j = jh$ and $h = 1/(d+1)$. The entries of the vector $u_h(t) \in \mathbb{R}^d$ are the unknowns of the problem: the approximations of $(u(t, x_\ell))_{\ell=1}^d$. We display on Figure 3 the error at different times corresponding to transient and asymptotic states. The error is computed with respect to the solution of the EDO. As expected again, accuracy increases with n. On the same Figure, we plot also the accuracy of the solution given by the ODE solver `solve_ivp` from the `Python`'s module `Scipy` using its default parameters. In the transient case, $n = 4, 8$ or 12 are not enough to supersede the solution given by `solve_ivp` but $n \geq 16$ is clearly better. In the asymptotic case, even $n = 4$ is enough to get the accuracy given by `solve_ivp`. In the next example, we compare our scheme with `solve_ivp` using several options: implicit solvers, control of the local error, ...

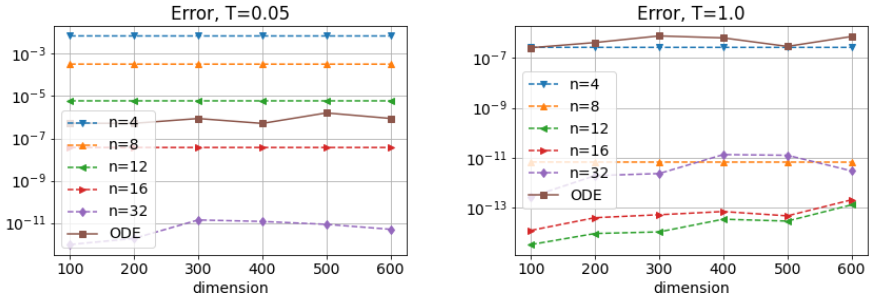

Figure 3: *Test 2. Accuracy of the approximation as a function of the truncation level n and the dimension of the problem d. The accuracy of the solution obtained by the ODE solver `solve_ivp` is marked with filled squares.*

Test 3. Space discretisation of a two-dimensional Partial Differential Equations. We consider the two-dimensional heat equation on $\Omega =]0, \pi[\times]0, \pi[$ with zero Dirichlet boundary conditions. The problem is now: find $v(t, x, y)$ solution of

$$v_t(t, x, y) - \Delta v(t, x, y) = f(x, y), \qquad t > 0,\ (x, y) \in \Omega.$$

The eignenvalues and associated eigenfunctions are $\lambda_{n,m} = n^2 + m^2$, $\psi_{n,m}(x, y) = \sin(nx)\sin(my)$. With $v_0 = \psi_{n,m}$ and $f = \lambda_{m,n}\psi_{m,n}$, the solution is

$$v(t, x, y) = \exp(-\lambda_{n,m}t)\psi_{n,m}(x, y) + (1 - \exp(-\lambda_{m,n}t))\psi_{m,n}(x, y).$$

It goes from $\psi_{n,m}$ to $\psi_{m,n}$ as t goes from 0 to $+\infty$. For the space discretisation, we use d internal points in each direction. The unknown is a vector in $\mathbb{R}^{d^2}$. The error is computed with respect to the solution of the ODE.

We use again `solve_ivp` for camparison, testing now different options. We compare our scheme to two accurate implicit solvers: *(i)* BDF solver (an implicit multi-step

variable-order method based on a backward differentiation formula), *(ii)* Radau solver (an implicit Runge-Kutta method of order 5). See the documentation [11] for a precise description of `solve_ivp`. We also play with the parameters `atol` and `rtol`, the absolute and relative tolerances. We fix in the simulation $d = 80$. We plot on Figure 4 work-precision diagrams (error in 2-norm as a function of the computational time) for BDF and Radau solvers with different tolerences parameters `atol`=`atol` = 10^{-k}, $k = 5, \cdots, 9$. As tolerences parameters decrease, the computing times for the solvers increase obviously. We do compare the schemes at two different times: a transient one $t = 0.01$ where the solution is moving from $\psi_{n,m}$ to $\psi_{m,n}$ and an asymptotic one where the solution is very close to $\psi_{m,n}$. To measure the gain in computational time, we define the speedup relative to one of the solvers as the ratio between the computational time for the solver and the computational time for our method. Let us recall again that parallelization is only simulated as we call "parallel computing time" the time used to compute one vector ω_k in (19) since each of these n vectors could be computed separately.

- $t = 0.01$. With truncation parameters $n = 8, 16, \cdots$, we obtain a high precision that is never achieved by the BDF solver. The speed up is almost the same for every value of the tolerence parameters. Our method runs about 25 times faster than BDF. The Radau solver is much accurate than the BDF solver but very time-consuming. With $n = 8, 16, \cdots$, we achieve its highest accuracy with a speedup is $\simeq 70$.

- $t = 1$. Starting from $n \geq 16$, we reach the accuracy of the BDF solver. The accuracy of the Radau solver is reached for $n \geq 20$. For the $n = 16$, the speedup with the BDF solver is $\simeq 30$. For the highest accuracy achieved by the BDF solver, the speedup is $\simeq 197$. Concerning the Radau solver, the speedup $\simeq 40$ for $n = 20$. To get the highest accuracy obtained with the solver, $n = 32$ is necessary. The speedup is then $\simeq 145$.

6 Conclusion

We have analysed a new method to compute the solution of linear Ordinary Differential Equations of the form $u'(t) = Au(t) + f(t)$ where the matrix A is hermitian and the function f is polynomial. Instead of discretising the ODE as done in usual numerical integration schemes, our method computes directly an approximation of the solution at the prescribed final time without passing through intermediate times. The solution at time t is approximated by $\sum_{k=1}^{n} a_{k,0}\omega_k$ where each vector ω_k could be computed separately from the others, by solving a linear system, leading to substantial savings in computational time. We tested succesfully our method on matrices arising from Finite

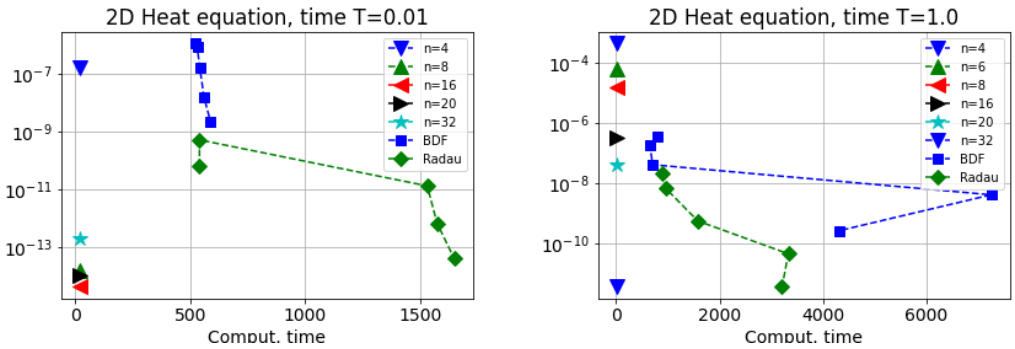

Figure 4: *Test 3. Work-precision diagrams: the* 2*-norm as a function of the computing time. For BDF and Radau solvers, several tolerances are used. For the rational approximation, several values of the truncation level are used.*

Difference approximations of the heat equation in one and two dimensions. In each case, our method has proved to be successful. We plan to apply now our method to nonlinear ODE of the form $u'(t) = Au(t) + f(u(t))$ that may result from a local linearisation around a state of interest. The symmetric matrix A is then a Jacobian matrix. In the analysis of exponential integrators schemes, the φ-functions are supposed to be known exactly which is not the case here. The analysis of our scheme is under unvestigation with a special emphasis on stability taking into account the fact that φ- functions are not computed exactly but only approximated with enough accuracy to define and efficient scheme, both in terms of speed and accuracy.

References

[1] A. H. Al-Mohy, N. J. Higham: *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM J. Sci. Comput., vol 33, No. 2, (2011).

[2] W.J. Cody, G. Meinardus, R.S. Varga, *Chebyshev Rational Approximations to* $\exp(-x)$ *in* $[0, +\infty)$ *and Applications to Heat-Conduction Problems*. Journal of Approximation Theory, vol. 2, (1969).

[3] T. Göckler, V. Grimm, *Uniform approximation of* φ*-functions in exponential integrators by a rational Krylov subspace method with simple poles*. SIAM J. Matrix Anal. Appl., vol. 35, No. 4, (2014).

[4] S. Güttel, *Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection*, GAMM-Mitt., vol. 36, No. 1, (2013).

[5] F. Hecht, S.M. Kaber, L. Perrin, A. Plagne, J. Salomon: *Parallel approximation of the exponential of Hermitian matrices*, https://hal.science/hal-03948509v2, (2022). Submitted.

[6] M. Hochbruck, Ch. Lubich, H. Selhofer: *Exponential Integrators for Large Systems of Differential Equations*, SIAM Journal on Scientific Computing, vol. 19, No 5, (1998).

[7] Y.Y. Lu: *Computing a matrix function for exponential integrators*, Journal of Computational and Applied Mathematics, vol. 161 (2003).

[8] M. Hochbruck, A. Ostermann: *Exponential integrators*, Acta Numerica, vol. 19, (2010).

[9] B. Minchev, W. Wright: *A review of exponential integrators for first order semilinear problems*, http://www.ii.uib.no/ borko/pub/N2-2005.pdf

[10] Y. Saad: *Analysis of some Krylov subspace approximations to the matrix exponential operator*, SIAM J. Numer. Anal., vol 29, No 1, (1992).

[11] *https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html*

[12] T. Schmelzer, L. N. Trefethen: *Evaluating matrix functions for exponential integrators via Carathéodory-Fejér approximation and contour integrals*, ETNA, vol. 29, (2007).