Original Research / *Artículo Original*

Ciencias Matemáticas, Vol. 36, No. Único, 2022-2023, Pag. 13-19
Received (*Recibido*): 27-04-2024, Revised (*Revisado*): 21-06-2024
Accepted (*Aceptado*): 28-06-2024, Online (*En línea*): 18-07-2024

Matrix-free computation of linear combinations of phi-functions times vectors in exponential integrators *Cálculo sin matrix de combinación lineal de función phi por vector en integradores exponenciales*

Frank Sadan Naranjo-Noda[1]*, Juan Carlos Jiménez Sobrino [2]

Abstract Numerical computation of linear combinations of phi-functions times vectors arises as a major issue in a number of exponential-type integrators for large systems of initial value problems (IVPs). This short communication deals with such a computation in the case that evaluating and storing the involved Jacobian matrices is unfeasible. To this end, a matrix-free Krylov-Padé approximation is introduced, and its error determined. Numerical simulations are provided with Jacobian matrices and vector fields of IVPs resulting from the spatial discretization of known partial differential equations.

Keywords: phi-functions, iterative numerical method, matrix-free method, exponential integrator, high dimensional initial value problem.

Resumen *El cálculo numérico de combinaciones lineales de productos de funciones phi por vectores surge como un problema importante en la implementación de integradores exponenciales para Problemas de Valor Inicial (PVI) de grandes dimensiones. Esta comunicación corta trata con dicho cálculo cuando es inviable evaluar y almacenar las matrices Jacobianas involucradas. Para ello, se introduce una aproximación de Krylov-Padé apropiada y se determina su error. Se realizan simulaciones numéricas con matrices Jacobianas y campos vectoriales de PVI resultantes de la discretización espacial de ecuaciones de derivadas parciales conocidas.*

Palabras Clave: *funciones phi, método numérico iterativo, método sin evaluación de matriz, integrador exponencial, problemas de valor inicial de grandes dimensiones.*

Mathematics Subject Classification: 65F60, 65F10, 65L05.

[1]Departamento Matemática Interdisciplinaria, Instituto de Cibernética, Matemática y Física. La Habana, Cuba. Email: fsadannn@gmail.com, fsadannn@icimaf.cu
[2]Departamento Matemática Interdisciplinaria, Instituto de Cibernética, Matemática y Física. La Habana, Cuba. Email: juan_carlos_js@yahoo.com, jcarlos@icimaf.cu
***Corresponding Author (*Autor para Correspondencia*)**

Edited by (*Editado por*): Damian Valdés Santiago, Facultad de Matemática y Computación, Universidad de La Habana, Cuba.

Citation (*Citar como*): Naranjo-Noda, F.S.,& Jiménez Sobrino, J.C. (2024). Matrix-free computation of linear combinations of phi-functions times vectors in exponential integrators. *Ciencias Matemáticas*, 36(Único), 13–19. Available in https://revistas.uh.cu/rcm/article/view/9070

Introduction

In the context of exponential-type integrators for Initial Value Problems (IVPs):

$$\frac{dx}{dt} = f(x), \quad x(t_0) = x_0 \in \mathbb{R}^d, \quad t \in [t_0, T],$$

there exists a number of numerical schemes (see, e.g., [14, 4, 5]) that involve the computation of the linear combination of multiple integrals times vectors of the form:

$$\sum_{i=1}^{l} \phi_i(A,h)a_i, \quad with \ \ \phi_i(A,h) = \int_0^h e^{A(h-s)}s^{i-1}ds, \qquad (1)$$

being A an square matrix representing the Jacobian matrix $f_x(y)$ of the vector field f evaluated at $y \in \mathbb{R}^d$, a_i d-dimensional column vectors, $h > 0$ the integrator step size, and $l = 2, 3, \cdots$. The case $l = 1$ is excluded in (1) to avoid focusing on the computation of single products $\phi_1(A,h)a_1$, which has been the subject of a number of papers [13, 12, 9, 11]).

For small dimensional matrix A, there is a number of methods for computing the linear combination of phi-functions (1) (see, [6] for a recent review). Analogously, for large matrix A, various methods have also been considered (see, e.g., [13, 12, 10]). In all of them, the evaluation and saving of the matrix A is required.

This short note focuses on the computation of (1) in the case that evaluating and storing the matrix A is unfeasible. To this end, in the next section, (1) is rewritten as the action of the operator ϕ_1 over a vector to use the matrix-free Krylov-Padé approximation of [11] to single products $\phi_1(A,h)a_1$. The proposed method can be seemed as a matrix-free extension of the Krylov-Padé method considered in section 3.1 of [10] for computing (1) and, for this reason, in the last section the performance of both methods is compared in simulations.

Highlights

This work deals with the numerical computation of linear combinations of phi-functions times vectors in the case that evaluating and storing the involved Jacobian matrices is unfeasible. For this purpose, a matrix-free Krylov-Padé approximation is introduced, and its error determined. The result is relevant for efficient implementations of Jacobian-free exponential-type integrators for large systems of initial value problems.

1. Main result

By using Theorem 1 in [3], the linear combination (1) can be straightforward rewritten as [7, 4, 3]:

$$\sum_{i=1}^{l} \phi_i(A,h)a_i = Le^{hM}r, \tag{2}$$

with $L=[I_{d\times d}\ 0_{d\times l}]$, $r=[0_{1\times(d+(l-1))}\ 1]^\intercal$, and,

$$M=\begin{bmatrix} A & \overline{a}_l & \overline{a}_{l-1} & \cdots & \overline{a}_1 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix},$$

where $\overline{a}_i = a_i(i-1)!$, for all $i=1,\cdots,l$.

Let us suppose that there exist $\eta+1$-th continuously differentiable functions $g:\mathbb{R}^d\times\mathbb{R}^d\times\mathbb{R}_+\to\mathbb{R}^d$ that approximate the product $f_x(x)b$ with order η, such that:

$$||g(x,b;\delta)-f_x(x)b|| \le c\,||b||^{\eta+1}\,\delta^\eta, \tag{3}$$

where $b\in\mathbb{R}^d$ is a column vector, and c is a positive constant depending only on the norm of derivatives of f_x. Using this approximation for the matrix times vector Au_1 and taking into account that the product of the partitioned matrix M times a partitioned column vector $u=[u_1^\intercal,\cdots,u_{l+1}]^\intercal\in\mathbb{R}^{d+l}$ is given by,

$$\begin{bmatrix} A & \overline{a}_l & \overline{a}_{l-1} & \cdots & \overline{a}_1 \\ 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ \vdots & \vdots & \vdots & \ddots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_{l+1} \end{bmatrix} = \begin{bmatrix} Au_1+\overline{a}_l u_2+\cdots+\overline{a}_1 u_{l+1} \\ u_3 \\ \vdots \\ u_{l+1} \\ 0 \end{bmatrix}\in\mathbb{R}^{d+l},$$

we can replace the classical Arnoldi algorithm [1] by the following extension of the classical matrix-free Arnoldi algorithm of [2].

Algorithm 1: Matrix-free Arnoldi algorithm to compose the orthonormal basis $\{\widehat{v}_1,\widehat{v}_2,\cdots,\widehat{v}_{\mathfrak{m}}\}$ of the $\mathfrak{m}$-th Krylov subspace $\widehat{\mathscr{K}_{\mathfrak{m}}}(\tau M(x),b;\delta)$, with partitioned matrix M defined as in (2).

Input: function $g:\mathbb{R}^d\times\mathbb{R}^d\times\mathbb{R}\to\mathbb{R}^d$ defined as in (3), $b\in\mathbb{R}^{d+l}$, x, $\overline{a}_1,\overline{a}_2,\cdots,\overline{a}_l\in\mathbb{R}^d$, $\tau,\delta>0$, and Krylov dimension $\mathfrak{m}$.

Output: $\widehat{V}_{\mathfrak{m}}=[\widehat{v}_1,\widehat{v}_2,\cdots,\widehat{v}_{\mathfrak{m}}]\in\mathbb{R}^{(d+l)\times\mathfrak{m}}$, upper Hessenberg matrix $\widehat{H}^*_{\mathfrak{m}}=(\widehat{\mathfrak{h}}^*_{ij})\in\mathbb{R}^{\mathfrak{m}\times\mathfrak{m}}$, $\widehat{v}_{\mathfrak{m}+1}\in\mathbb{R}^{d+l}$, $\widehat{\mathfrak{h}}^*_{\mathfrak{m}+1,\mathfrak{m}}$.

$\widehat{v}_1 = b/\|b\|_2$
for $j=1,2,\cdots,\mathfrak{m}$ **do**
  $q_j =$
$$\begin{bmatrix} g(x,\tau[\widehat{v}_j^{[1]},\ldots,\widehat{v}_j^{[d]}]^\intercal;\delta)+\tau\overline{a}_l\widehat{v}_j^{[d+1]}+\cdots+\tau\overline{a}_1\widehat{v}_j^{[d+l]} \\ \tau\widehat{v}_j^{[d+2]} \\ \vdots \\ \tau\widehat{v}_j^{[d+l]} \\ 0 \end{bmatrix}$$
  $\widehat{w}_j = q_j$
  for $i=1,2,\cdots,j$ **do**
    $\widehat{\mathfrak{h}}^*_{ij} = \langle q_j,\widehat{v}_i\rangle$
    $\widehat{w}_j = \widehat{w}_j - \widehat{\mathfrak{h}}^*_{ij}\widehat{v}_i$
  end
  $\widehat{\mathfrak{h}}^*_{j+1,j} = \|\widehat{w}_j\|_2$
  $\widehat{v}_{j+1} = \widehat{w}_j/\widehat{\mathfrak{h}}^*_{j+1,j}$
end
$\widehat{v}_j^{[k]}$ denotes the $k-$th element of the vector $\widehat{v}_j$, and $\langle\cdot,\cdot\rangle$ the scalar product of two vectors.

Let the matrices $\widehat{V}_{\mathfrak{m}}\in\mathbb{R}^{(d+l)\times\mathfrak{m}}$ and $\widehat{H}^*_{\mathfrak{m}}\in\mathbb{R}^{\mathfrak{m}\times\mathfrak{m}}$, the vector $\widehat{v}_{\mathfrak{m}+1}\in\mathbb{R}^{d+l}$, and the positive number $\widehat{\mathfrak{h}}^*_{\mathfrak{m}+1,\mathfrak{m}}$ be outputs of the matrix-free Arnoldi Algorithm 1 for the $\mathfrak{m}-$th Krylov

subspace $\widehat{\mathscr{K}_{\mathfrak{m}}}(h^{\beta}M(x),b;\delta)$, where $M(x)$ is the partitioned matrix defined in (2) with $A=f_x(x)$ the Jacobian matrix of the vectorial function $f(x)$, $x\in\mathbb{R}^d$, $b\in\mathbb{R}^{d+l}$, $h,\delta>0$ and $\beta\geq 0$. The matrix-free $(\mathfrak{m},\mathfrak{p},\mathfrak{q},k)$-Krylov-Padé approximation of $\phi_1(M,h)b$ is then defined by [11]:

$$\begin{aligned}\widehat{K}^{\mathfrak{p},\mathfrak{q}}_{\mathfrak{m},k}(h,M,b;\eta,\delta,\beta) &= ||b||_2\,\widehat{V}_{\mathfrak{m}}\,[P_h]_{11}e_1 \\ &+ ||b||_2\,\widehat{\mathfrak{h}}_{\mathfrak{m}+1,\mathfrak{m}}e_{\mathfrak{m}}^{\intercal}\,[P_h]_{12}\widehat{v}_{\mathfrak{m}+1},\end{aligned} \tag{4}$$

where P_h denotes the $(\mathfrak{p},\mathfrak{q})$-Padé approximation with scaling and squaring k for the matrix exponential $\mathrm{e}^{h\overline{H}}$,

$$\overline{H}=\begin{bmatrix}\widehat{H}_{\mathfrak{m}} & e_1 & 0_{\mathfrak{m}\times 1} & 0_{\mathfrak{m}\times 1}\\ 0_{1\times\mathfrak{m}} & 0 & 1 & 0\\ 0_{1\times\mathfrak{m}} & 0 & 0 & 1\\ 0_{1\times\mathfrak{m}} & 0 & 0 & 0\end{bmatrix},$$

$\widehat{H}_{\mathfrak{m}}=\widehat{H}^*_{\mathfrak{m}}/h^{\beta}$, $\widehat{\mathfrak{h}}_{\mathfrak{m}+1,\mathfrak{m}}=\widehat{\mathfrak{h}}^*_{\mathfrak{m}+1,\mathfrak{m}}/h^{\beta}$, k is the smallest natural number such that $\left\|2^{-k}h\overline{H}\right\|_{\infty}\leq\frac{1}{2}$, and e_i the i-th canonical vector of $\mathbb{R}^{\mathfrak{m}}$.

Thus, for evaluating the linear combinations of phi-functions times vectors (1), we introduce the following main result.

Theorem 1 *Let A be an $d\times d$ matrix representing the Jacobian matrix $f_x(x)$ of the vector field f evaluated at $x\in\mathbb{R}^d$, a_i is d-dimensional vectors, and $h>0$. Then,*

$$\sum_{i=1}^{l}\phi_i(A,h)a_i=L\phi_1(M,h)Mr, \tag{5}$$

where the matrices L, M and the vector r are defined as in (2). Moreover,

$$\begin{aligned}&\left\|\sum_{i=1}^{l}\phi_i(A,h)a_i-L\widehat{K}^{\mathfrak{p},\mathfrak{q}}_{\mathfrak{m},k}(h,M,Mr;\eta,\delta,\beta)\right\|_2\\ &\qquad\leq \mathfrak{c}_0h^{\min\{\mathfrak{m}+2,\mathfrak{p}+\mathfrak{q}+1\}}+\mathfrak{c}_1h^{\beta\eta+2}\delta^{\eta},\end{aligned} \tag{6}$$

where $\widehat{K}^{\mathfrak{p},\mathfrak{q}}_{\mathfrak{m},k}(h,M,Mr;\eta,\delta,\beta)$ is the Matrix-free $(\mathfrak{m},\mathfrak{p},\mathfrak{q},k)$-Krylov-Padé approximation of $\phi_1(M,h)Mr$ defined in (4), $\mathfrak{c}_0$ and $\mathfrak{c}_1$ are positive constants, and $Mr=\begin{bmatrix}\overline{a}_1^{\intercal} & 0_{1\times l}\end{bmatrix}^{\intercal}$.

Proof Taking into account that $Lr=0$, the first statement results from the fact that $e^{Mt}r$ and $r+\phi_1(M,t)Mr$ represent the unique solution x of the linear initial value problem:

$$dx/dt=Mx,\quad x(0)=r,\quad \text{for}\quad t\geq 0.$$

Since $\widehat{K}^{\mathfrak{p},\mathfrak{q}}_{\mathfrak{m},k}(h,M,Mr;\eta,\delta,\beta)$ approximates $\phi_1(M,h)Mr$, the inequality (6) straightforward arises from the statement (5) and Theorem 4.2 in [11] for the error of such approximation. ■

When the matrix-free Krylov-Padé approximation in (6) is used to compute the linear combination (2) in an exponential integrator, the above results is essential to determine the convergence rate of the resulting numerical scheme as the step size h goes to zero (see, e.g., Theorems 4.2 and 5.1 in [11] for integrators computing (2) with $l=1$). Furthermore, at each integration step of the numerical scheme, the Krylov dimension $\mathfrak{m}$, the order $(\mathfrak{p},\mathfrak{q})$ and scaling k of the Padé approximation in (6) can be suitable estimated as in section 4.1 of [11].

As mentioned in the introduction, the new approximation (4) can be considered as a matrix-free extension of the Krylov-Padé approximation $K^{\mathfrak{p},\mathfrak{q}}_{\mathfrak{m},k}(h,M,Mr)$ defined in [9] for $\phi_1(M,h)Mr$ in the case that the matrix M can be evaluated and stored in memory. In such a case, when computing $\phi_1(M,h)Mr$ in (5), Theorem 3.1 in [9] yields:

$$\left\|\sum_{i=1}^{l}\phi_i(A,h)\mathbf{a}_i-LK^{\mathfrak{p},\mathfrak{q}}_{\mathfrak{m},k}(h,M,Mr)\right\|_2\leq \mathfrak{c}_0h^{\min\{\mathfrak{m}+2,\mathfrak{p}+\mathfrak{q}+1\}}, \tag{7}$$

bound that coincides with the first term of the bound (6). In the above inequality, the approximation $K^{\mathfrak{p},\mathfrak{q}}_{\mathfrak{m},k}(h,M,Mr)$ – used in [10] to compute (5) – has the two terms of (4) but, with matrices $\widehat{V}_{\mathfrak{m}}$ and $\widehat{H}^*_{\mathfrak{m}}$, vector $\widehat{v}_{\mathfrak{m}+1}$, and positive number $\widehat{\mathfrak{h}}^*_{\mathfrak{m}+1,\mathfrak{m}}$ outputs of the classical Arnoldi algorithm [1].

2. Numerical simulations

In this section, we consider the computation of the linear combination (2) with Jacobian matrix f_x and vector field f of the IVPs resulting from the spatial discretization of two known partial differential equations. The following Jacobian matrices and discretized equations are taken from [14].

Example 1 *$2N\times 2N$ Jacobian matrix:*

$$f_x(x)=\begin{bmatrix}diag(2u\cdot v-4) & diag(u\cdot u)\\ diag(3-2u\cdot v) & -diag(u\cdot u)\end{bmatrix}+\frac{\alpha}{(\Delta z)^2}\begin{bmatrix}K & 0\\ 0 & K\end{bmatrix},$$

with

$$x=\begin{bmatrix}u\\ v\end{bmatrix}\in\mathbb{R}^{2N},\quad K=\begin{bmatrix}-2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1\\ & & & & 1 & -2\end{bmatrix}_{N\times N},$$

corresponding to the vector field f of the $2N$-dimensional discretized Brusselator equation:

$$\begin{aligned}\frac{du_i}{dt} &= 1+u_i^2v_i-4u_i+\frac{\alpha}{(\Delta z)^2}(u_{i-1}-2u_i+u_{i+1})\\ \frac{dv_i}{dt} &= 3u_i-u_i^2v_i+\frac{\alpha}{(\Delta z)^2}(v_{i-1}-2v_i+v_{i+1})\end{aligned}$$

with $\alpha=1/50$, $u_i(0)=1+\sin(2\pi z_i)$, $v_i(0)=3$, $z_i=i/(N+1)$, $\Delta z=1/(N+1)$, $i=1,2,\cdots,N$, and $N=800$. In the expression for $f_x(x)$, $diag(\cdot)$ is the operation of constructing a diagonal matrix from a vector, and

$$u\cdot v=[u_1v_1,\cdots,u_Nv_N]^{\intercal}.$$

Example 2 *Tridiagonal $N \times N$ Jacobian matrix:*

$$f_x(x) = \frac{1}{2\Delta z}\begin{bmatrix} 0 & -x_2 & & & & \\ x_1 & 0 & -x_3 & & & \\ & x_2 & 0 & -x_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & x_{N-2} & 0 & -x_N \\ & & & & x_{N-1} & 0 \end{bmatrix} + \frac{\nu K}{(\Delta z)^2},$$

corresponding to the vector field f of the N-dimensional discretized Burgers equation

$$\frac{dx_i}{dt} = -\frac{x_{i+1}^2 - x_{i-1}^2}{4\Delta z} + \frac{\nu}{(\Delta z)^2}(x_{i-1} - 2x_i + x_{i+1}),$$

with K as in Example 1, $\nu = 0.0003$, $x_i(0) = \sin^2(3\pi z_i)(1 - z_i)^{3/2}$, $z_i = i/(N+1)$, $\Delta z = 1/(N+1)$, $i = 1, 2, \cdots, N$, and $N = 2000$.

Matlab codes *JF1-Phi* and *JF2-Phi* implement the matrix-Free Kylov-Padé approximation of (6) with $\beta = 0$ and, respectively, with the finite differences of first and second order:

$$g(x,b;\delta) = \frac{f(x+\delta b) - f(x)}{\delta}, \text{ and}$$
$$g(x,b;\delta) = \frac{f(x+\delta b) - f(x-\delta b)}{2\delta},$$

inside of Algorithm 1, where $\delta = \frac{\sqrt{(1+||x||_2)\varepsilon_{mach}}}{\varepsilon_{mach} + ||b||_2}$ as suggested in [8], being ε_{mach} the spacing of the floating point number 1. Matlab code *Phi* implements the Kylov-Padé approximation $K_{\mathfrak{m},k}^{\mathfrak{p},\mathfrak{q}}(h,M,Mr)$ of [9] for $\phi_1(M,h)Mr$ (the same code used in section 3.1 of [10] for computing (2) with evaluation of the matrix A).

For the two examples, the matrix and vectors defining M in (2) are set as $A = f_x(x(0))$, $a_1 = a_2 = f(x(0))$, $a_3 = 2a_1$ and $a_4 = 6a_1$ in concordance with the number of terms l of the linear combination (2) used in each example.

With the aim of comparison, the 'exact' matrix e^{hM} in (2) for each example and the matrices $e^{\tau\overline{H}}$ in the codes *JF1-Phi*, *Phi* and *JF2-Phi* are computed with the same Matlab function *expm*. In the three codes, the value of the Krylov dimension $\mathfrak{m}$ is determined automatically depending on the specified relative *rtol* and absolute *atol* tolerances, as indicated in [10] and [9] for these two approximations. Euclidean norm quantifies the difference between the 'exact' vector $Le^{hM}r$ and its approximations.

Top row of Figures 1-2 presents, for each example, the log-log plots of the error (*error*) versus relative tolerance (*rtol*) in the computation of (2) via the approximations *JF1-Phi*, *JF2-Phi* and *Phi* with relative and absolute tolerances $rtol = 10^{-j}$ and $atol = 0.1\ rtol$, with $j = 1, 2, \cdots, 6$. Bottom row of these figures presents the plots of the estimated Krylov dimension $\mathfrak{m}$ versus the $log(rtol)$ corresponding to the approximations in the top row of the figures.

Observe that, a major difference among these matrix-free and the matrix-exact approximations is the threshold for the errors of the first ones as the tolerance decreases. As expected, when the tolerance *rtol* decreases, the Krylov dimension $\mathfrak{m}$ for the three approximations increases and, consequently, their errors also decreases. However, for increasing values of $\mathfrak{m}$, the fixed value of the second right hand term in (6) dominates the decreasing values of first term, which explains the thresholds for the errors of the matrix-free approximations *JF1-Phi* and *JF2-Phi* in Figures 1-2. Contrary, as predicted by (7), the error of the matrix-exact approximation *Phi* in these figures always decreases when $\mathfrak{m}$ increases.

In general, for the same tolerance *rtol*, the matrix-free approximations *JF1-Phi* and *JF2-Phi* work with lower value of $\mathfrak{m}$, which explains their higher errors. Note also that, in correspondence with the second term of the bound (6), the accuracy of the approximation with second order finite difference *JF2-Phi* is slightly higher than that of the approximation with first order finite difference *JF1-Phi* only for the greater values of h (top left plot in the figures). Finally, from left to right, the top plots of Figures 1-2 shows an increasing of the accuracy of the three approximations as h decreases, which corresponds to error estimates (6) and (7) for these approximations. Naturally, the accuracy of the two matrix-free approximations is lower than that of the approximation evaluating the exact Jacobian matrix.

Conclusions

A matrix-free Krylov-Padé approximation was introduced for computing linear combinations of phi-functions times vectors in the case that evaluating and storing the involved matrix is unfeasible. The numerical simulations corroborated the main implications of the error analysis for such matrix-free approximation, that is, threshold for the errors when the Krylov-subspace dimension increases, lower error of the approximation with second order finite difference, and lower accuracy than the Krylov-Padé approximation with exact matrix.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary information

The Matlab code used in this paper could be consulted in https://github.com/fsadannn/PhiXv.

References

[1] Arnoldi, W.E.: *The principle of minimized iterations in the solution of the matrix eigenvalue problem.*

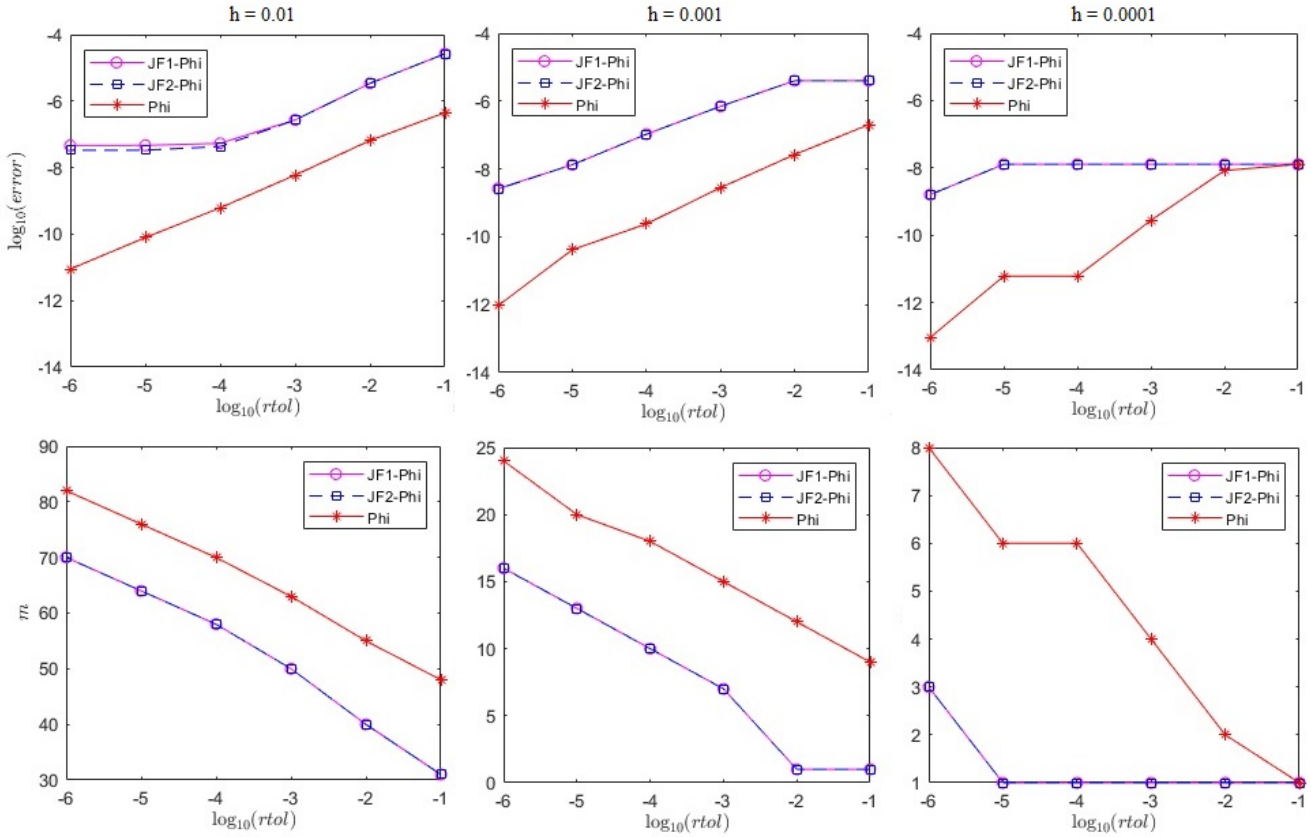

Figure 1. Top: Log-log plots of error (*error*) versus relative tolerance (*rtol*) in the computation of $\phi_1(f_x,h)a_1+\phi_2(f_x,h)a_2+\phi_3(f_x,h)a_3+\phi_4(f_x,h)a_4$ for the equation of Example 1 via the approximations *JF1-Phi*, *JF2-Phi* and *Phi* with $rtol=10^{-j}$ and $j=1,2,\cdots,6$. From left to right, with $h=0.01,\ 0.001,\ 0.0001$. Bottom: plots of Krylov dimension $\mathfrak{m}$ versus $log(rtol)$ corresponding to the approximations in the top of the figure [*Arriba: Gráficos logarítmicos del error (error) versus la tolerancia relativa (rtol) en el cálculo de* $\phi_1(f_x,h)a_1+\phi_2(f_x,h)a_2+\phi_3(f_x,h)a_3+\phi_4(f_x,h)a_4$ *para la ecuación del Ejemplo 1 mediante las aproximaciones JF1-Phi, JF2-Phi y Phi con* $rtol=10^{-j}$ *y* $j=1,2,\cdots,6$. *De izquierda a derecha, con* $h=0,01,\ 0,001,\ 0,0001$. *Abajo: gráficos de la dimensión de Krylov* $\mathfrak{m}$ *versus el* $log(rtol)$ *correspondientes a las aproximaciones de la parte superior de la figura*].

Quarterly of Applied Mathematics, 9:17–29, 1951. https://www.ams.org/journals/qam/1951-09-01/S0033-569X-1951-42792-9/S0033-569X-1951-42792-9.pdf.

[2] Brown, P.N.: *A local convergence theory for combined inexact-newton/finite-difference projection methods*. SIAM Journal on Numerical Analysis, 24:407–434, 1987. https://epubs.siam.org/doi/abs/10.1137/0724031.

[3] Carbonell, F., J.C. Jimenez, and L. Pedroso: *Computing multiple integrals involving matrix exponentials*. Journal of Computational and Applied Mathematics, 213:300–305, 2008. https://www.sciencedirect.com/science/article/pii/S0377042707000283.

[4] Cruz, H. de la, R.J. Biscay, F. Carbonell, T. Ozaki, and J.C. Jimenez: *A higher order local linearization method for solving ordinary differential equations*. Applied Mathematics and Computation, 185:197–212, 2007. https://www.sciencedirect.com/science/article/abs/pii/S0096300306008514.

[5] Hochbruck, M. and A. Ostermann: *Exponential multistep methods of Adams-type*. BIT Numerical Mathematics, 51:889–908, 2011. https://link.springer.com/article/10.1007/s10543-011-0332-6.

[6] Jimenez, J.C., H. de la Cruz, and P. De Maio: *Efficient computation of phi-functions in exponential integrators*. Journal of Computational and Applied Mathematics, 374:112758, 2020. https://www.sciencedirect.com/science/article/abs/pii/S0377042720300492.

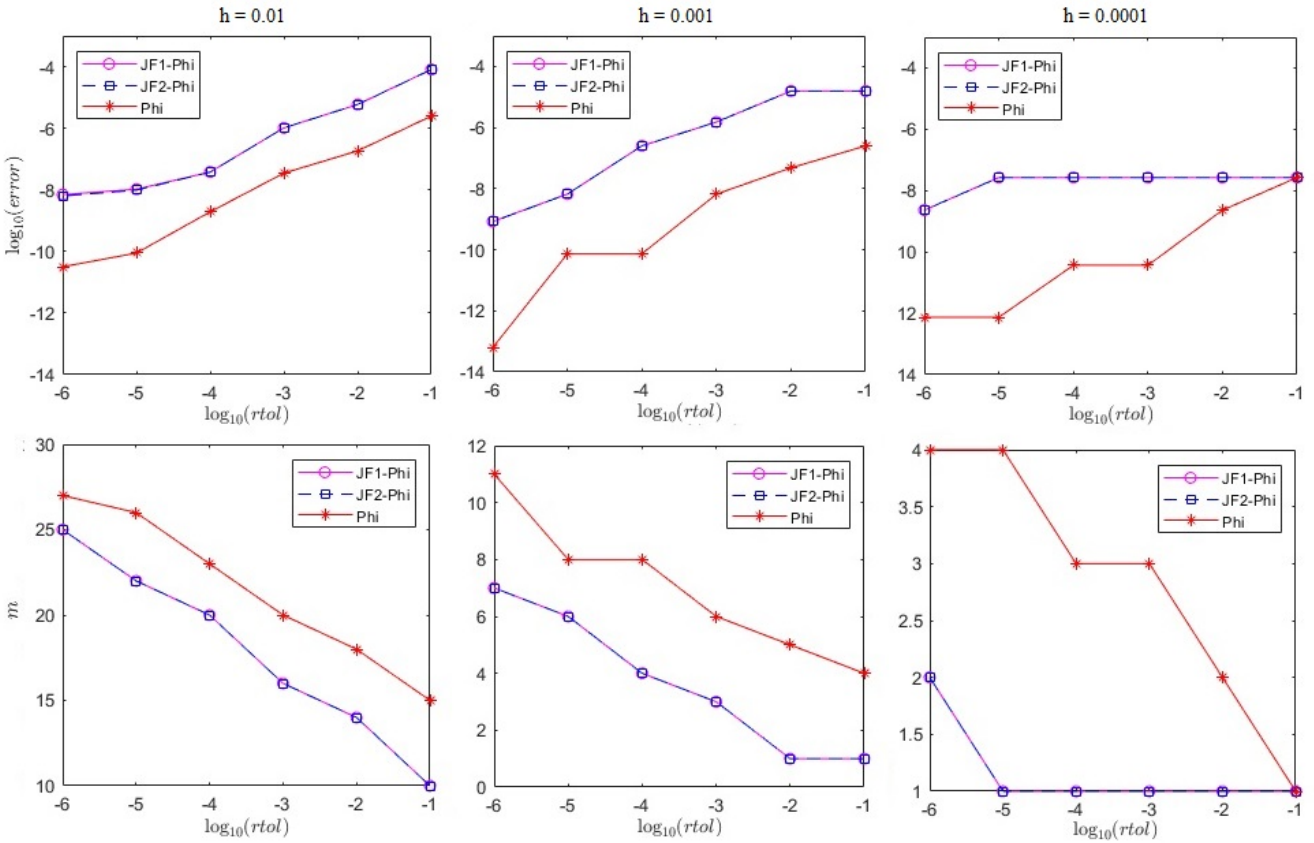

Figure 2. Top: Log-log plots of error (*error*) versus relative tolerance (*rtol*) in the computation of $\phi_1(f_x,h)a_1+\phi_2(f_x,h)a_3+\phi_3(f_x,h)a_3$ for the equation of Example 2 via the approximations *JF1-Phi*, *JF2-Phi* and *Phi* with $rtol = 10^{-j}$ and $j = 1,2,\cdots,6$. From left to right, with $h = 0.01,\ 0.001,\ 0.0001$. Bottom: plots of Krylov dimension $\mathfrak{m}$ versus $log(rtol)$ corresponding to the approximations in the top of the figure [*Arriba: Gráficas logarítmicas del error (error) versus tolerancia relativa (rtol) en el cálculo de* $\phi_1(f_x,h)a_1+\phi_2(f_x,h)a_3+\phi_3(f_x,h)a_3$ *para la ecuación del Ejemplo 2 mediante las aproximaciones JF1-Phi, JF2-Phi y Phi con* $rtol = 10^{-j}$ *y* $j = 1,2,\cdots,6$. *De izquierda a derecha, con* $h = 0,01,\ 0,001,\ 0,0001$. *Abajo: gráficas de la dimensión de Krylov* $\mathfrak{m}$ *versus el* $log(rtol)$ *correspondientes a las aproximaciones de la parte superior de la figura*].

[7] Jimenez, J.C., L. Pedroso, F. Carbonell, and V. Hernadez: *Local linearization method for numerical integration of delay differential equations*. SIAM Journal of Numerical Analysis, 44:2584–2609, 2006. https://epubs.siam.org/doi/abs/10.1137/040607356.

[8] Knoll, D.A. and Keyes D.E.: *Jacobian-free Newton-Krylov methods: a survey of approaches and applications*. Journal of Computational Physics, 193:357–397, 2004. https://www.sciencedirect.com/science/article/abs/pii/S0021999103004340.

[9] Naranjo-Noda, F.S. and J.C. Jimenez: *Locally linearized runge-kutta method of dormand and prince for large systems of initial value problems*. Journal of Computational Physics, 426:109946, 2021. https://www.sciencedirect.com/science/article/abs/pii/S0021999120307208.

[10] Naranjo-Noda, F.S. and J.C. Jimenez: *Computing high dimensional multiple integrals involving matrix exponentials*. Journal of Computational and Applied Mathematics, 421:114844, 2023. https://www.sciencedirect.com/science/article/abs/pii/S0377042722004423.

[11] Naranjo-Noda, F.S. and J.C. Jimenez: *Jacobian-free high order local linearization methods for large systems of initial value problems*. Applied Numerical Mathematics, 187:158–175, 2023. https://www.sciencedirect.com/science/article/abs/pii/S0168927423000405.

[12] Niesen, J. and Wright WM. Algorithm 919: *Algorithm 919: A Krylov subspace algorithm for evaluating the φ-functions appearing in exponential integrators*. ACM Transactions on Mathematical Software (TOMS), 38:1–9, 2012. https://dl.acm.org/doi/abs/10.1145/2168773.2168781.

[13] Sidje, R.B.: *Expokit: A software package for computing matrix exponentials*. ACM Transactions on Mathematical Software (TOMS), 24:130–156, 1998. https://dl.acm.org/doi/abs/10.1145/285861.285868.

[14] Tokman, M.: *Efficient integration of large stiff systems of ODEs with exponential propagation iterative (EPI) methods*. Journal of Computational Physics, 213(2):748–776, 2006. https://www.sciencedirect.com/science/article/abs/pii/S0021999105004158.