Exponentially fitted two-step Runge-Kutta methods: construction and parameter selection

R. D'Ambrosio, E. Esposito, B. Paternoster

Universitá degli Studi di Salerno, Dipartimento di Matematica e Informatica, Fisciano (Sa) - Italy

Abstract

We derive exponentially fitted two-step Runge-Kutta methods for the numerical solution of $y' = f(x, y)$, specially tuned to the behaviour of the solution. Such methods have nonconstant coefficients which depend on a parameter to be suitably estimated. The construction of the methods is shown and a strategy of parameter selection is presented. Some numerical experiments are provided to confirm the theoretical expectations.

Keywords: Ordinary differential equations, Two-step Runge-Kutta methods, Exponential fitting, Parameter selection

1. Introduction

It is the purpose of this paper to introduce a family of exponentially-fitted numerical methods for the solution of ordinary differential equations (ODEs)

$$\begin{cases} y'(x) = f(x, y(x)), \quad x \in [x_0, X], \\ y(x_0) = y_0 \in \mathbb{R}^d, \end{cases} \tag{1.1}$$

where $f : [x_0, X] \times \mathbb{R}^d \to \mathbb{R}^d$ is a sufficiently smooth function ensuring that the corresponding problem is well posed. The class of methods we aim to consider is the family of two-step Runge-Kutta methods

$$\begin{cases} y_{n+1} = \theta y_{n-1} + (1-\theta) y_n + h \displaystyle\sum_{j=1}^{m} \Big(v_j f(Y_j^{[n-1]}) + w_j f(Y_j^{[n]}) \Big), \\ Y_i^{[n]} = u_i y_{n-1} + (1-u_i) y_n + h \displaystyle\sum_{j=1}^{m} \Big(a_{ij} f(Y_j^{[n-1]}) + b_{ij} f(Y_j^{[n]}) \Big), \end{cases} \tag{1.2}$$

with $i = 1, 2, \ldots, m$. In (1.2), y_n is an approximation of order p to $y(x_n)$, $x_n = x_0 + nh$, and $Y_i^{[n]}$ are approximations of order q to $y(x_{n-1} + c_i h)$, $i = 1, 2, \ldots, m$, where $y(x)$ is the solution to (1.1) and $c = [c_1, \ldots, c_m]^T$ is the abscissa vector.

TSRK methods (1.2) can be represented by the abscissa vector c and the table of their coefficients

$$\begin{array}{c|c|c} u & A & B \\ \hline \theta & v^T & w^T \end{array} = \begin{array}{c|cccc|cccc} u_1 & a_{11} & a_{12} & \cdots & a_{1m} & b_{11} & b_{12} & \cdots & b_{1m} \\ u_2 & a_{21} & a_{22} & \cdots & a_{2m} & b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ u_m & a_{m1} & a_{m2} & \cdots & a_{mm} & b_{m1} & b_{m2} & \cdots & b_{mm} \\ \hline \theta & v_1 & v_2 & \cdots & v_m & w_1 & w_2 & \cdots & w_m \end{array}.$$

The peculiarity of two-step Runge-Kutta methods (1.2) lies in their dependency on the stage derivatives at two consecutive step points: as a consequence, "we gain extra degrees of freedom associated with a two-step scheme

Email address: {rdambrosio, elesposito, beapat}@unisa.it (R. D'Ambrosio, E. Esposito, B. Paternoster)

without the need for extra function evaluations" (see [30]), because the function evaluations $f(Y_j^{[n-1]})$ are completely inherited from the previous step and, therefore, the computational cost of these formulae only depends on the structure of the matrix B. The achieved degrees of freedom can be used in order to improve the properties of existing one-step methods, especially in terms of order of convergence and stability.

Two-step Runge-Kutta methods (1.2), introduced by Jackiewicz and Tracogna [30], have been extensively investigated by several authors: there is a rich bibliography on TSRK methods (fully referenced in the monograph [33]) regarding, for instance, the derivation of general order conditions by means of Albrecht approach [30], Butcher trees [7] and B-series [25], the estimation of the local truncation error [2, 43], technical issues for a variable stepsize-variable order implementation of TSRK methods [2, 21, 31, 32, 43], A-, L- and algebraically stable TSRK methods for the numerical treatment of stiff problems [11, 12, 19, 21], derivation of continuous extensions for TSRK methods [3, 5, 31] and the development of the family of collocation based TSRK methods [12, 16, 20, 21, 22]. These methods have also been introduced in the context of Volterra integral equations [13, 15] and delay differential equations [1, 4]. However, up to now, nothing has been said concerning the development of TSRK methods based on functions other than algebraic polynomials.

Within the class of TSRK methods (1.2), we aim to introduce a family of formulae with nonconstant coefficients aiming to solve problems (1.1) whose solutions exhibit a prominent exponential behaviour. Classical numerical methods for ODEs may not be well-suited to follow a prominent exponential or oscillatory behaviour of the solutions, because a very small stepsize would be required, with corresponding deterioration of the numerical performances, especially in terms of efficiency. One of the possible ways to proceed in order to derive numerical methods particularly tuned to the behaviour of the solution can be realized by imposing that a numerical method exactly integrate (within the round-off error) problems of type (1.1) whose solution can be expressed as linear combination of functions other than polynomials: this is the spirit of the exponential fitting (EF) technique (see [9, 10, 18, 23, 27, 28, 36, 41, 45, 46] and the monograph [29] together with the references therein contained), where the *adapted* numerical method is developed in order to be exact on problems whose solution is linear combination of the following basis functions:

$$\{1, x, \ldots, x^K, \exp(\pm\mu x), x\exp(\pm\mu x), \ldots, x^P \exp(\pm\mu x)\},$$

where K and P are integer numbers. When $K = 0$ and μ is a complex value, the oscillatory case is also recovered.

The aim of this paper is the derivation of EF-based methods within the class (1.2), depending on the value of a parameter to be suitably determined. Such value would be known only if the analytic expression of the exact solution could be given in advance, which is in general an unrealistic requirement. However, an approximation to the unknown parameter can also be experimentally found: nevertheless, in many cases, even when it would be possible to approximate or measure the value of the parameter through suitable experiments, the derived value would anyway suffer from the presence of errors. If the value of the parameter is not determined with a sufficient level of accuracy, the performances of the corresponding EF-based numerical method would be subject to a relevant deterioration.

The estimation of the unknown parameter is, in general, a nontrivial problem. In fact, up to now, a rigorous theory for the exact computation of the parameter has not yet been developed, but several attempts have been done in the literature (see, for instance, [27, 29] and references therein contained) in order to provide an accurate estimation, generally based on the minimization of the leading term of the local discretization error. This is essentially the strategy we will follow in this paper in order to determine an approximation to the parameter, in such a way that the performances of the corresponding adapted methods are not dramatically compromised by the lack of knowledge of the exact value of the parameter.

The paper is organized as follows: in Section 2 we present the constructive technique of adapted TSRK methods; Section 3 approaches the problem to estimate the unknown parameter on which the coefficients of the methods depend, while in Section 4 we analyze the linear stability properties of the derived methods. Finally section 5 provides numerical tests confirming the theoretical expectations. Section 6 contains some conclusions and further developments of this research. The paper concludes with an appendix, containing a *MATHEMATICA* script for the generation of a family of adapted TSRK methods.

2. Derivation of the methods

This section is devoted to the presentation of the constructive technique leading to our class of special purpose TSRK formulae. Since we aim to obtain exponentially fitted TSRK methods, we adapt to our purposes the six-step procedure introduced by Ixaru and Vanden Berghe in [29]. This procedure provides a general way to derive EF formulae whose coefficients are expressed in a regularized way and, as a consequence, they do not suffer from numerical cancellation.

In agreement with this procedure, we first associate to the method (1.2) the following set of $m+1$ functional operators

$$\begin{aligned} \mathcal{L}[h,\mathbf{a}]y(x) &= y(x+h)-\theta y(x-h)-(1-\theta)y(x)-h\sum_{i=1}^{m}(v_i y'(x+(c_i-1)h)+w_i y'(x+c_i h)), && (2.3)\\ \mathcal{L}_i[h,\mathbf{b}]y(x) &= y(x+c_i h)-u_i y(x-h)-(1-u_i)y(x)-h\sum_{j=1}^{m}(a_{ij}y'(x+(c_j-1)h)+b_{ij}y'(x+c_j h)), && (2.4)\end{aligned}$$

$i=1,\dots,m$, where

$$\mathbf{a}=\begin{bmatrix}\theta & v^T & w^T\end{bmatrix},\qquad \mathbf{b}=\begin{bmatrix}u & A & B\end{bmatrix}.$$

Then, the constructive procedure consists in the following six steps.

- step (i) *Computation of the classical moments*. We apply the linear operators (2.3) and (2.4) to the monomials x^q, $q=0,1,\dots$, obtaining

$$\begin{aligned}\mathcal{L}[h,\mathbf{a}]x^q &= h^q L_q^*(\mathbf{a}),\\ \mathcal{L}_i[h,\mathbf{b}]x^q &= h^q L_{iq}^*(\mathbf{b}),\end{aligned}$$

where

$$\begin{aligned} L_q^*(\mathbf{a}) &= 1+(-1)^{q+1}\theta-q\sum_{i=1}^{m}(v_i(c_i-1)^{q-1}+w_i c_i^{q-1}),\ q=0,1,\dots, && (2.5)\\ L_{iq}^*(\mathbf{b}) &= c_i^q+(-1)^{q+1}u_i-q\sum_{j=1}^{m}(a_{ij}(c_j-1)^{q-1}+b_{ij}c_j^{q-1}),\ i=1,\dots,m,\ q=0,1,\dots, && (2.6)\end{aligned}$$

are the so-called *reduced classical moments* (compare [29]).

- step (ii) *Compatibility analysis*. We examine the linear systems

$$\begin{aligned} L_q^*(\mathbf{a}) &= 0,\ q=0,1,\dots,M'-1, && (2.7)\\ L_{iq}^*(\mathbf{b}) &= 0,\ i=1,\dots,m,\ q=0,1,\dots,M-1 && (2.8)\end{aligned}$$

to determine the maximal values of the integers M and M' such that the above systems are compatible. We define the following recursive relation

$$\begin{aligned} r^{(0)}(j) &= \frac{(-1)^j}{j},\\ r^{(k)}(j) &= r^{(k-1)}(j)-r^{(k-1)}(j-1)d_k, \end{aligned} \qquad (2.9)$$

for $k,j=1,2,\dots,2m$, where

$$d_k=\begin{cases} c_k-1, & k=1,2,\dots,m,\\ c_{k-m}, & k=m+1,\dots,2m.\end{cases} \qquad (2.10)$$

Then the following result holds.

Theorem 2.1. *Assume $M=M'=2m+2$, $c_i\neq c_j$, $c_i\neq c_j-1$ for $i\neq j$ and, moreover, that $r^{(k)}(j)\neq 0$, $k,j=1,2,\dots,2m$. Then the linear systems (2.7) and (2.8) admit an unique solution.*

Proof: In correspondence of the value $M = 2m+3$, the system (2.7) in the unknowns θ, v_i and w_i, $i = 1, 2, \dots, m$, takes the form

$$\begin{cases} L_0^*(\mathbf{a}) = 0, \\ L_1^*(\mathbf{a}) = 1 + \theta - \sum_{i=1}^{m}(v_i + w_i) = 0, \\ L_2^*(\mathbf{a}) = 1 - \theta - 2\sum_{i=1}^{m}(v_i(c_i - 1) + w_i c_i) = 0, \\ \vdots \\ L_{2m+1}^*(\mathbf{a}) = 1 + \theta - (2m+1)\sum_{i=1}^{m}(v_i(c_i - 1)^{2m} + w_i c_i^{2m}) = 0, \end{cases}$$

or, equivalently,

$$\begin{cases} \theta - \sum_{i=1}^{m}(v_i + w_i) = -1, \\ -\theta - 2\sum_{i=1}^{m}(v_i(c_i - 1) + w_i c_i) = -1, \\ \vdots \\ \theta - (2m+1)\sum_{i=1}^{m}(v_i(c_i - 1)^{2m} + w_i c_i^{2m}) = -1, \end{cases}$$

which is a linear system with coefficient matrix

$$H = \begin{bmatrix} 1 & -1 & \dots & -1 \\ -1 & -2d_1 & \dots & -2d_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ (-1)^{j-1} & -jd_1^{j-1} & \dots & -jd_{2m}^{j-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -(2m+1)d_1^{2m} & \dots & -(2m+1)d_{2m}^{2m} \end{bmatrix},$$

where

$$d_j = \begin{cases} c_j - 1, & 1 \le j \le m, \\ c_j, & m+1 \le j \le 2m. \end{cases}$$

In order to achieve the thesis, we need to prove that the matrix H is nonsingular by computing its determinant. Using the elementary properties of determinants (compare [34]), we obtain

$$\det H = (2m+1)! \det H'$$

where

$$H' = \begin{bmatrix} r^{(0)}(1) & -1 & \dots & -1 \\ r^{(0)}(2) & d_1 & \dots & d_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ r^{(0)}(j) & d_1^{j-1} & \dots & d_{2m}^{j-1} \\ \vdots & \vdots & \vdots & \vdots \\ r^{(0)}(2m+1) & d_1^{2m} & \dots & d_{2m}^{2m} \end{bmatrix}.$$

where $r^{(0)}(j) = (-1)^{j-1}$. H' is now clearly of Vandermonde type, except the first column: in order to compute its determinant, we can apply with suitable modifications the procedure introduced to compute the determinant of the Vandermonde matrix [35]. By means of $2m$ switches, we can put the first column of H' in the last position, obtaining the matrix

$$H'' = \begin{bmatrix} -1 & \dots & -1 & r^{(0)}(1) \\ d_1 & \dots & d_{2m} & r^{(0)}(2) \\ \vdots & \vdots & \vdots & \vdots \\ d_1^{j-1} & \dots & d_{2m}^{j-1} & r^{(0)}(j) \\ \vdots & \vdots & \vdots & \vdots \\ d_1^{2m} & \dots & d_{2m}^{2m} & r^{(0)}(2m+1) \end{bmatrix},$$

which satisfies $\det H'' = \det H'$. As a consequence, $\det H = (2m+1)!\det H''$. We now aim to compute the determinant of H''. We subtract d_1 times the first row from the other $2m$ rows, obtaining the matrix

$$\begin{bmatrix} -1 & -1 & \dots & -1 & r^{(0)}(1) \\ 0 & d_2 - d_1 & \dots & d_{2m} - d_1 & r^{(0)}(2) - r^{(0)}(1)d_1 \\ 0 & d_2^2 - d_1 d_2 & \dots & d_{2m}^2 - d_1 d_{2m} & r^{(0)}(3) - r^{(0)}(2)d_1 \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & d_2^{j-1} - d_1 d_2^{j-2} & \dots & d_{2m}^{j-1} - d_1 d_{2m}^{j-2} & r^{(0)}(j-1) - r^{(0)}(j-2)d_1 \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & d_2^{2m} - d_1 d_2^{2m-1} & \dots & d_{2m}^{2m} - d_1 d_{2m}^{2m-1} & r^{(0)}(2m) - r^{(0)}(2m-1)d_1 \end{bmatrix},$$

whose determinant is equal to the opposite of the determinant of the matrix

$$\begin{bmatrix} d_2 - d_1 & \dots & d_{2m} - d_1 & r^{(1)}(2) \\ d_2(d_2 - d_1) & \dots & d_{2m}(d_{2m} - d_1) & r^{(1)}(3) \\ \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & \\ d_2^{2m-1}(d_2 - d_1) & \dots & d_{2m}^{2m-1}(d_2 - d_1) & r^{(1)}(2m) \end{bmatrix},$$

where

$$r^{(1)}(j) = r^{(0)}(j) - r^{(0)}(j-1)d_1, \qquad j = 2, ...2m.$$

As a consequence, we obtain

$$\det H'' = -(d_2 - d_1)(d_3 - d_1)\dots(d_{2m} - d_1)\cdot\det\begin{bmatrix} 1 & \dots & 1 & r^{(1)}(2) \\ d_2 & \dots & d_{2m} & r^{(1)}(3) \\ \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & \\ d_2^{2m-1} & \dots & d_{2m}^{2m-1} & r^{(1)}(2m) \end{bmatrix},$$

and, iterating this process, we get

$$\det H'' = -\prod_{1\le i\le k\le 2m}(d_k - d_i) r^{(2m-1)}(2m),$$

where

$$r^{(2m-1)}(2m) = r^{(2m-2)}(2m) - r^{(2m-2)}(2m-1)d_{2m-1}.$$

According to the hypothesis, it is $r^{(2m-1)}(2m) \neq 0$ and, as a consequence, it follows that H'' is nonsingular and, finally, that H is nonsingular. By analogous arguments, we obtain that the system (2.8), $i = 1, 2, \dots, m$, assumes the form

$$\begin{cases} u_i - \sum_{j=1}^m (a_{ij} + b_{ij}) = -c_i, \\ -u_i - 2\sum_{j=1}^m (a_{ij}(c_j - 1) + b_{ij}c_j) = -c_i^2, \\ \vdots \\ u_i - (2m+1)\sum_{j=1}^m (a_{ij}(c_j-1)^{2m} + b_{ij}c_j^{2m}) = -c_i^{2m+1}, \end{cases}$$

and its coefficient matrix is the matrix H. Therefore, the same analysis as previous holds. □

- step (iii) *Computation of the G functions*. In order to derive EF methods, we need to compute the so-called *reduced (or starred) exponential moments* (see [29], p. 42), i.e.

$$E_0^*(\pm z, \mathbf{a}) = \exp(\pm\mu x)\mathcal{L}[h, \mathbf{a}]\exp(\pm\mu x), \tag{2.11}$$

$$E_{0i}^*(\pm z, \mathbf{b}) = \exp(\pm\mu x)\mathcal{L}_i[h, \mathbf{b}]\exp(\pm\mu x), \; i = 1, \dots, m. \tag{2.12}$$

Once the reduced exponential moments have been computed, we can derive the G functions, defined in the following way:

$$\begin{aligned}
G_i^+(Z,\mathbf{b}) &= \frac{1}{2}\Big(E_{0i}^*(z,\mathbf{b}) + E_{0i}^*(-z,\mathbf{b})\Big),\ i=1,\dots,m,\\
G_i^-(Z,\mathbf{b}) &= \frac{1}{2z}\Big(E_{0i}^*(z,\mathbf{b}) - E_{0i}^*(-z,\mathbf{b})\Big),\ i=1,\dots,m,\\
G^+(Z,\mathbf{a}) &= \frac{1}{2}\Big(E_0^*(z,\mathbf{a}) + E_0^*(-z,\mathbf{a})\Big),\\
G^-(Z,\mathbf{a}) &= \frac{1}{2z}\Big(E_0^*(z,\mathbf{a}) - E_0^*(-z,\mathbf{a})\Big),
\end{aligned}$$

where $Z = z^2$. In our case, the G functions take the following form

$$\begin{aligned}
G^+(Z,\mathbf{a}) &= \eta_{-1}(Z) - \theta\eta_{-1}(Z) - Z\sum_{i=1}^{m}\Big(v_i(c_i-1)\eta_0((c_i-1)^2Z) + w_i c_i\eta_0(c_i^2Z)\Big) - (1-\theta),\\
G^-(Z,\mathbf{a}) &= \eta_0(Z) + \theta\eta_0(Z) - \sum_{i=1}^{m}\Big(v_i\eta_{-1}((c_i-1)^2Z) + w_i\eta_{-1}(c_i^2Z)\Big),\\
G_i^+(Z,\mathbf{b}) &= \eta_{-1}(c_i^2Z) - u_i\eta_{-1}(Z) - (1-u_i) - Z\sum_{j=1}^{m}\Big(a_{ij}(c_j-1)\eta_0((c_j-1)^2Z) + b_{ij}c_j\eta_0(c_j^2Z)\Big),\\
G_i^-(Z,\mathbf{b}) &= c_i\eta_0(c_i^2Z) + u_i\eta_0(Z) - \sum_{j=1}^{m}\Big(a_{ij}\eta_{-1}((c_j-1)^2Z) + b_{ij}\eta_{-1}(c_j^2Z)\Big).
\end{aligned}$$

We observe that the above expressions depend on the functions $\eta_{-1}(Z)$ and $\eta_0(Z)$ (compare [28, 29]), which are defined as follows: when Z is real, they assume the form

$$\eta_{-1}(Z) = \begin{cases} \cos(|Z|^{1/2}) & \text{if } Z \le 0 \\ \cosh(Z^{1/2}) & \text{if } Z > 0 \end{cases}, \quad \eta_0(Z) = \begin{cases} \sin(|Z|^{1/2})/|Z|^{1/2} & \text{if } Z < 0 \\ 1 & \text{if } Z = 0 \\ \sinh(Z^{1/2})/Z^{1/2} & \text{if } Z > 0 \end{cases},$$

while, when Z is complex, they take the form

$$\eta_{-1}(Z) = \cos(iZ^{1/2}), \quad \eta_0(Z) = \begin{cases} \sin(iZ^{1/2})/Z^{1/2} & \text{if } Z \neq 0 \\ 1 & \text{if } Z = 0 \end{cases},$$

or, equivalently,

$$\eta_{-1}(Z) = \frac{1}{2}[\exp(Z^{1/2}) + \exp(-Z^{1/2})], \quad \eta_0(Z) = \begin{cases} \frac{1}{2Z^{1/2}}[\exp(Z^{1/2}) - \exp(-Z^{1/2})] & \text{if } Z \neq 0 \\ 1 & \text{if } Z = 0 \end{cases}.$$

We next compute the p-th derivatives $G^{\pm(p)}$ and $G_i^{\pm(p)}$, taking into account the formula for the p-th derivative of $\eta_k(Z)$ (see [29])

$$\eta_k^{(p)}(Z) = \frac{1}{2^p}\eta_{k+p}(Z),$$

and obtain

$$\begin{aligned}
G^{+(p)}(Z,\mathbf{a}) &= \frac{(1-\theta)}{2^p}\eta_{p-1}(Z) - \sum_{i=1}^{m}\left(v_i(c_i-1)\frac{d^p}{dZ^p}\Big(Z\eta_0((c_i-1)^2Z)\Big) + w_i c_i \frac{d^p}{dZ^p}\Big(Z\eta_0(c_i^2Z)\Big)\right),\\
G^{-(p)}(Z,\mathbf{a}) &= \frac{(1+\theta)}{2^p}\eta_{p}(Z) - \sum_{i=1}^{m}\left(v_i\frac{d^p}{dZ^p}\eta_{-1}((c_i-1)^2Z) + w_i \frac{d^p}{dZ^p}\eta_{-1}(c_i^2Z)\right),\\
G_i^{+(p)}(Z,\mathbf{b}) &= \frac{c_i^{2p}}{2^p}\eta_{p-1}(c_i^2Z) + \frac{u_i}{2^p}\eta_{p-1}(Z) - \sum_{j=1}^{m}\left(a_{ij}(c_j-1)\frac{d^p}{dZ^p}\Big(Z\eta_0((c_j-1)^2Z)\Big) + b_{ij}c_j \frac{d^p}{dZ^p}\Big(Z\eta_0(c_j-1)^2Z)\Big)\right),\\
G_i^{-(p)}(Z,\mathbf{b}) &= \frac{c_i^{2p+1}}{2^p}\eta_{p}(c_i^2Z) + \frac{u_i}{2^p}\eta_{p}(Z) - \sum_{j=1}^{m}\left(a_{ij}\frac{d^p}{dZ^p}\eta_{-1}((c_j-1)^2Z) + b_{ij} \frac{d^p}{dZ^p}\eta_{-1}(c_j^2Z)\right).
\end{aligned}$$

- step (iv) *Definition of the function basis*. We now consider the choice of the basis functions to take into account, i.e. we choose the set of functions annihilating the linear operators (2.3) and (2.4). As a consequence, the corresponding method exactly integrates all those problems whose solution is linear combination of the chosen basis functions. In general, the set of M functions is a collection of both powers and exponentials, i.e.

$$\{1, x, \ldots, x^K, \exp(\pm\mu x), x\exp(\pm\mu x), \ldots, x^P\exp(\pm\mu x)\}, \tag{2.13}$$

where K and P are integer numbers satisfying the relation

$$K + 2P = M - 3 = 2m - 1. \tag{2.14}$$

Let us next consider the set of M' functions

$$\{1, x, \ldots, x^{K'}, \exp(\pm\mu x), x\exp(\pm\mu x), \ldots, x^{P'}\exp(\pm\mu x)\}, \tag{2.15}$$

annihilating the operators $\mathcal{L}_i[h,\mathbf{b}]y(x)$, $i = 1, 2, \ldots, m$, and assume that $K' = K$ and $P' = P$, i.e. the external stages and the internal ones are exact for the same basis functions.

- step (v) *Determination of the coefficients*. After a suitable choice of K and P, we next solve the following algebraic systems:

$$\begin{aligned}
G_i^{\pm(p)}(Z,\mathbf{a}) &= 0,\ i = 1,\ldots,m,\ p = 0,...,P,\\
G^{\pm(p)}(Z,\mathbf{b}) &= 0,\ p = 0,...,P.
\end{aligned}$$

In the numerical experiments we will consider EF-based TSRK methods (1.2) with $m = 2$, $K = 1$ and $P = 1$, i.e. we choose the function basis

$$\{1, x, \exp(\pm\mu x), x\exp(\pm\mu x)\}. \tag{2.16}$$

The coefficients of the resulting methods are reported in the appendix.

- step (vi) *Error analysis*. According to the six-step procedure [29], the expression of the local truncation error for an EF method with respect to the basis of functions (2.13) takes the form

$$lte^{EF}(x) = (-1)^{P+1}h^M\frac{L^*_{K+1}(\mathbf{a}(Z))}{(K+1)!Z^{P+1}}D^{k+1}(D-\mu)^{P+1}y(x), \tag{2.17}$$

with K, P and M satisfying the condition (2.14).

For the sake of completeness, we remark that this expression of the local truncation error can be derived by using the approach of Coleman and Ixaru [9], who provided an adaptation of the theory by Ghizzetti and

Ossicini (1970) to the case of EF-based formulae. This approach consists in regarding the error associated to an EF-based formula as

$$E[y] = L[y](\xi) \int_{-h}^{h} \Phi(x)dx,$$

where $\xi \in (-h, h)$ and, in our case, $L[y] = D^{k+1}(D-\mu)^{P+1}y(x)$. We observe that the kernel $\Phi(x)$ is an even function in the null space of L. The analysis of the local error associated to the developed test case $K = P = 1$, introduced in step (v), is reported in the Appendix.

The local error analysis also provides a starting point for the estimation of the unknown parameter μ which is, in general, a nontrivial problem. In fact, up to now, a rigorous theory for the exact computation of the parameter μ has not yet been developed, but several attempts have been done in the literature in order to provide an accurate estimation (see [29] and references therein), generally based on the minimization of the leading term of the local discretization error. Our attempts to estimate the unknown parameter is described in the following section.

3. Parameter estimation

Step (vi) of the constructive strategy described in Section 2 provided us the expression of the local truncation error

$$lte^{EF}(x) = (-1)^{P+1}h^M \frac{L^*_{K+1}(\mathbf{a}(Z))}{(K+1)!Z^{P+1}} D^{k+1}(D-\mu)^{P+1}y(x). \tag{3.18}$$

We aim to estimate the value of the parameter μ that annihilates or minimizes the leading term of (3.18), by solving the equation

$$D^{K+1}(D-\mu_j)^{P+1}y(x_j) = 0, \tag{3.19}$$

where μ_j is an approximation to the unknown parameter μ in the point x_j of the grid. We observe that (3.19) is a nonlinear equation of degree $P+1$ in μ_j: if $P = 0$, (3.19) assumes the form

$$(D^{K+2} - \mu_j D^{K+1})y(x_j) = 0,$$

whose unique solution is

$$\mu_j = \frac{D^{K+2}y(x_j)}{D^{K+1}y(x_j)}, \tag{3.20}$$

while, for any $P \geq 1$, Equation (3.19) admits $P+1$ solutions among whom we aim to search for the best approximation of the unknown parameter. In order to determine such an appropriate and reliable estimation to the unknown parameter, we follow the lines drawn in [27] in the case of two-point boundary value problems.

In particular, we first analyze the solutions of (3.19) when the solution $y(x)$ belongs to the fitting space: for instance, we assume that

$$y(x) = x^q e^{\mu x}.$$

The following result holds

Theorem 3.1. *Assuming that $y(x) = x^q e^{\mu x}$ is solution of the problem (1.1), then $\nu = \mu$ is a root of multiplicity $P-q+1$ of (3.19).*

Proof: In correspondence of $y(x) = x^q e^{\mu x} = D^q_\mu e^{\mu x}$, Equation (3.19) assumes the form

$$D^q_\mu D^{K+1}_x (D_x - \nu)^{P+1} e^{\mu x} = 0,$$

or, equivalently,

$$D^{K+1}_x D^q_\mu (\mu - \nu)^{P+1} e^{\mu x} = 0. \tag{3.21}$$

We observe that $\nu = \mu$ is a root of multiplicity $P+1$ of (3.19). Therefore, after q differentiations, we obtain that $\nu = \mu$ is a root of multiplicity $P-q+1$ of (3.21). □

The above theorem can be interpreted as follows: by denoting the left hand side of (3.19) as $p^{(P)}(\mu_j)$, if the solution $y(x)$ to the problem (1.1) belongs to the fitting space, then by solving the nonlinear equations $p^{(P)}(\mu) = 0$, $p^{(P+1)}(\mu) = 0, \ldots, p^{(P-q+1)}(\mu) = 0$ we will obtain a constant solution overall the integration interval for each equation, which will next be chosen as approximation to the unknown parameter μ. On the contrary, if we obtain a nonconstant solution for the above equations, i.e. values of μ_j varying along the integration interval, we can next conclude that the function $y(t)$ does not belong to the fitting space and we will assume as approximation to the parameter μ the smallest modulus, in order to avoid inaccurate results due to numerical instability.

This approach for the estimation of the unknown parameter will next be applied to some test cases reported in Section 5.

4. Linear stability analysis

In this section we aim to carry out the linear stability analysis of the adapted formulae (1.2), by considering the linear scalar problem

$$y' = \lambda y,$$

where λ is a complex parameter such that $\mathrm{Re}(\lambda) < 0$. We recall that, for TSRK methods (1.2), the stability matrix assumes the form ([33])

$$\mathbf{M}(\omega, z) = \begin{bmatrix} 1-\theta+\omega w^T Q(\omega)(e-u) & \theta+\omega w^T Q(\omega)u & \omega(v^T+\omega w^T Q(\omega)A) \\ 1 & 0 & 0 \\ Q(\omega)(e-u) & Q(\omega)u & \omega Q(\omega)A \end{bmatrix} \in \mathbb{R}^{(m+2)\times(m+2)}, \tag{4.22}$$

where $\omega = \lambda h \in \mathbb{C}$ and $Q(\omega) = (I-\omega B)^{-1}$. This matrix is then used in order to determine the three-dimensional stability region (compare with [17]) in the parameter space $(\mathrm{Re}(\omega), \mathrm{Im}(\omega), z)$, thus extending a concept introduced in [10] for second order differential equations coherently with the following definition.

Definition 4.1. *The region of the three-dimensional* $(\mathrm{Re}(\omega), \mathrm{Im}(\omega), z)$ *space on which the inequality*

$$\rho(\mathbf{M}(\omega, z)) < 1, \tag{4.23}$$

is satisfied is called a region of stability Ω *for the method (1.2).*

Some examples of stability regions are reported in the appendix.

5. Numerical results

We now present some numerical experiments in order to test the performances of the derived methods on some differential problems (1.1). Such numerical evidence is based on the implementation of the following methods:

- EF5: exponentially-fitted TSRK method (1.2), which can be generated using the MATHEMATICA script, reported in the appendix. This method has order and stage order 5, and depends on the value of the parameter μ to be estimated;
- TSRK5: two-stage TSRK method (compare with [16]) with abscissa vector $c = [1/2, 1]^T$ and Butcher tableau

$$\begin{array}{c|c|c} u & A & B \\ \hline \theta & v^T & w^T \end{array} = \begin{array}{c|cc|cc} -\frac{11}{232} & -\frac{39}{464} & \frac{9}{29} & \frac{111}{464} & -\frac{3}{232} \\ \frac{1}{29} & \frac{4}{87} & \frac{4}{29} & \frac{20}{29} & \frac{14}{87} \\ \hline \frac{1}{29} & \frac{4}{87} & \frac{4}{29} & \frac{20}{29} & \frac{14}{87} \end{array},$$

  of order and stage order 5, with constant coefficient matrices.

As far as EF5 is concerned, with the aim to apply the strategy described in Section 3 for the selection of the parameter μ, we will treat Equation (3.19) not only by exactly computing the values of the derivatives appearing in such equation through the required evaluations of the f function, but also, for the sake of comparison, by approximating them through backward finite differences in the following way

$$y^{(r)}(x_{n+1}) \approx \frac{1}{h^r} \sum_{i=0}^{r} (-1)^i \binom{r}{i} y_{n-i}. \tag{5.24}$$

We observe that both methods possess the same order and stage order and, due to the equal number of stages, they have the same computational cost. We implement both methods in a fixed stepsize environment, with stepsize $h = 1/2^k$, with k positive integer number. Similarly as in [26], in order to reduce the influence of round-off errors, we have reformulated the implemented methods as follows

$$\begin{cases} Z_i^{[n]} = h \displaystyle\sum_{j=1}^{m} a_{ij} f(Z_j^{[n-1]} + \theta y_{n-1} + (1-\theta) y_n) + h \sum_{j=1}^{m} b_{ij} f(Z_j^{[n]} + \theta y_{n-1} + (1-\theta) y_n), \\ y_{n+1} = \theta y_{n-1} + (1-\theta) y_n + h \displaystyle\sum_{j=1}^{m} v_j f(Z^{[n-1]} + \theta y_{n-1} + (1-\theta) y_n) + h \sum_{j=1}^{m} w_j f(Z^{[n]} + \theta y_{n-1} + (1-\theta) y_n), \end{cases} \tag{5.25}$$

where

$$Z_i^{[n]} = Y_i^{[n]} - \theta y_{n-1} - (1-\theta) y_n,$$

$i = 1, 2, \ldots, m$, and solved the nonlinear system in $Z_i^{[n]}$ with Newton iterations.

We apply the above methods on the following problems:

- *Problem 1.* The Prothero-Robinson equation [39]

$$\begin{cases} y'(x) = \varepsilon\big(y(x) - F(x)\big) + F'(x), \quad x \in [1, 5], \\ y(x_0) = y_0, \end{cases}$$

  where $\mathrm{Re}(\varepsilon) < 0$ and $F(x)$ is a slowly varying function on the integration interval. In our experiments, we have considered $F(x) = xe^{-2x}$ and y_0 such that the exact solution is $y(x) = xe^{-2x}$. As observed by Hairer and Wanner [26] in the context of Runge-Kutta methods this equation provides much insight into the behaviour of numerical methods for stiff problems. This equation with $F(x) = e^{-2x}$, was also used by Butcher [6] to investigate order reduction for m-stage Runge-Kutta-Gauss methods of order $p = 2m$;

- *Problem 2.* The nonlinear system

$$\begin{cases} y_1'(x) = -y_1(x) + y_2(x)(1 - y_1(x) - y_2(x)), \quad x \in [1, 2], \\ y_2'(x) = y_1(x) - y_2(x)(1 - y_1(x)) - e^{-x}, \\ y_1(x_0) = \dfrac{1}{e}, \quad y_2(x_0) = 0, \end{cases}$$

  whose exact solution is $y(x) = [e^{-x}, 0]^T$. This problem provides the nonstiff version of the system considered in [24].

We observe that the exact solutions of both problems belong to our chosen fitting space (2.16) and, therefore, EF5 method is able to exactly integrate them, within the round-off error. Of course, in order to apply such method, an estimation to the parameter μ on which it depends is required: even if in the considered test cases the exact solution is given (and, therefore, the value of the parameter is known), we assume that the exact value of the parameter cannot be a priori recognized. In order to derive an estimate to such value, we apply the approach for the parameter selection reported in Section 3, with the aim to test its effectiveness.

Numerical results are collected in tables which report, for the EF5 method,

h	EF5					TSRK5
	$e_h^{EX}(T)$	$e_h^{DER}(T)$	$e_\mu^{DER}(T)$	$e_h^{DIFF}(T)$	$e_\mu^{DIFF}(T)$	$e_h^{TSRK5}(T)$
$1/2^4$	9.74e-13	1.69e-12	1.30e-8	4.07e-4	2.91e-1	3.01e-6
$1/2^5$	7.16e-16	1.19e-15	2.50e-10	9.94e-8	1.43e-1	3.71e-8
$1/2^6$	2.39e-16	2.39e-16	3.26e-12	5.31e-10	6.52e-2	7.51e-10

Table 1: Numerical results for Problem 1

- the value $e_h^{EX}(T)$ of the global error at the endpoint of integration, when the exact value of μ is used;
- the value $e_h^{DER}(T)$ of the global error at the endpoint of integration, when the estimation of μ is obtained by solving Equation (3.19) using the exact values of the involved derivatives, through the evaluation of the f function;
- the (relative) error $e_\mu^{DER}(T)$ associated to the estimated parameter by using exact derivatives in (3.19);
- the global error $e_h^{DIFF}(T)$ in the endpoint of integration, when the derivatives involved in Equation (3.19) are approximated by means of backward finite differences;
- the (relative) error $e_\mu^{DIFF}(T)$ associated to the estimation of the parameter by approximating the derivatives in (3.19) through backward finite differences.

For the TSRK5 method, the tables report the value $e_h^{TSRK5}(T)$ of the global error at the endpoint of integration.

Concerning Problem 1, taking into account the expression (6.28) of local truncation error associated to the EF5 method, the approach described in Section 3 suggests us to assume as objective functions to be annihilated at each step point the algebraic polynomials

$$\begin{aligned} p^{(1)}(\mu) &= D^2(D-\mu)^2 y(x)\Big|_{x=x_n} = (D^4 - 2\mu D^3 + \mu^2 D^2)y(x)\Big|_{x=x_n}, \\ p^{(2)}(\mu) &= D^2(D-\mu)^3 y(x)\Big|_{x=x_n} = (D^5 - 3\mu D^4 + 3\mu^2 D^3 - \mu^3 D^2)y(x)\Big|_{x=x_n}, \end{aligned} \tag{5.26}$$

in the unknown μ, assuming that x_n is the current step point. We denote as μ_n the selected value of the parameter at the step point x_n, which is the minimum root among the zeros of $p^{(1)}(\mu)$ and $p^{(2)}(\mu)$ derived by using the Matlab command `roots`. Table 1 shows the obtained results, associated to the value $\varepsilon = -10$.

The numerical evidence confirms that the method EF5 is able to exactly integrate this problem within round-off error, since its solution falls in the fitting space, and shows better accuracy with respect to the TSRK5 method. The approach described in Section 3 provides a reliable estimation to the parameter μ, which does not deteriorate the performances of the EF5 method. At this stage we want to remark that the knowledge of a reasonably accurate value of the parameter is necessary in order to avoid a dramatical deterioration of the performance of the method. To support this thesis, we focus on the errors reported in Table 1 obtained when the derivatives in (5.26) are approximated by finite differences. In this case, due to the low accuracy in the derivatives approximation, the corresponding inaccurate estimation to the unknown parameter causes a relevant deterioration to the performances of the EF5 method, which are even worse than the ones of the constant coefficient TSRK5 method. This also confirms the importance to reliably estimate the parameter on which the coefficients of an EF-based method depend on.

h	EF5					TSRK5
	$e_h^{EX}(T)$	$e_h^{DER}(T)$	$e_\mu^{DER}(T)$	$e_h^{DIFF}(T)$	$e_\mu^{DIFF}(T)$	$e_h^{TSRK5}(T)$
$1/2^3$	2.71e-14	2.72e-14	2.20e-5	5.39e-5	16.60	5.66e-9
$1/2^4$	8.69e-15	8.78e-15	1.37e-5	5.55e-5	32.42	1.86e-10
$1/2^5$	6.18e-16	6.03e-16	1.68e-5	3.63e-5	64.18	4.54e-12

Table 2: Numerical results for Problem 2

Regarding Problem 2, according to the approach described in Section 3, the objective functions to be annihilated at each step point are the algebraic polynomials

$$\begin{aligned} p^{(1)}(\mu) &= D^2(D-\mu)^2 y(x)\Big|_{x=x_n} = (D^4 - 2\mu D^3 + \mu^2 D^2) y(x)\Big|_{x=x_n}, \\ p^{(2)}(\mu) &= D^2(D-\mu)^3 y(x)\Big|_{x=x_n} = (D^5 - 3\mu D^4 + 3\mu^2 D^3 - \mu^3 D^2) y(x)\Big|_{x=x_n}, \\ p^{(3)}(\mu) &= D^2(D-\mu)^4 y(x)\Big|_{x=x_n} = (D^6 - 4D^5\mu + 6D^4\mu^2 - 4D^3\mu^3 + D^2\mu^4) y(x)\Big|_{x=x_n}. \end{aligned} \tag{5.27}$$

Table 2 shows the obtained results. The numerical evidence confirms the theoretical expectation: the method EF5 exactly integrates Problem 2 within round-off error. Also in this case the superiority of EF5 on TSRK5 is evident from the obtained results. Moreover, we observe that the usage of finite differences in replacement of the derivatives appearing in (5.27) causes a prominent worsen of the numerical performances, due to the inaccurate parameter selection. In fact, as it is evident from (5.24), such an approximation to the derivatives suffers from a severe numerical instability (compare with [40]).

We observe that a system equivalent to (5.27) is

$$D^k(D-\mu)^2 y(x)\Big|_{x=x_n} = 0, \quad k = 2, 3, 4,$$

which can be obtained by adding μ times the previous equation to each equation. The advantage here is a degree reduction of equations to be solved, and the possibility to solve with respect to μ by computing the nullspace $[1, 2\mu, \mu^2]^T$ of a 3×3 Wronskian, possibly using an SVD approach. This alternative strategy will be object of future analysis.

6. Conclusions and future works

We have developed a family of EF-based TSRK methods (1.2) for the numerical integration of initial value problems (1.1). We have particularly focused our attention on the computation of a reasonably accurate approximation to the unknown parameter on which the coefficients of the derived methods depend. The numerical evidence have revealed that the strategy presented in Section 3 for the estimation of the parameters allows to obtain an accuracy on the approximation to the solution of (1.1) which is comparable with the one we would have when the exact value of the parameter is known.

The strategy of parameter selection is essentially based on solving some nonlinear systems involving evaluations of the function f at the right hand side of (1.1) and its derivatives. The results of the implementation show that a reasonably good estimation to the parameter can be achieved only if the exact values of such derivatives are used, while the employ of approximate derivatives through finite differences would relevantly worsen the performances of the corresponding method. We are aware of the fact that a better accuracy in the parameter selection and, as a consequence, in the approximation to the solution of (1.1), requires an higher computational cost. Anyway, such cost is, according to our opinion, lower than the one of a TSRK method with constant coefficients able to provide the same accuracy of our EF-based solver: in fact, such constant coefficients method should have an higher order of convergence which is inevitably achieved by a larger number of stages, with consequent heightening of the dimension (and the cost) of the nonlinear system in the stages to be solved at each time step. The computational cost of the implementation is closely linked also to the choice of the parameter P: in fact, the more P is increased, the more the degree of the nonlinear equations to be solved at each step becomes higher. It is necessary, in our opinion, to create a reasonable balance between the exponential fitting parameter P and the computational effort of the solver: for instance, P can be chosen in such a way that the accuracy of the classical TSRK method with constant coefficients is remarkably improved with a comparable computational effort. This is the spirit of the choice of P we have made in this paper.

Further developments of this research will regard the introduction of adapted TSRK formulae based on more general function basis: in fact, this paper represents the first step in order to consider general function basis, following the spirit of *function fitting* (see, for instance, [37, 38]). We also aim to consider *revised* EF-formulae within the family of TSRK methods, extending the idea introduced in [18].

Acknowledgments

We express our gratitude to the anonymous referee for his valuable suggestions and comments.

Appendix

We report in this section a MATHEMATICA script for the generation of the coefficients of the EF5 method implemented in Section 5. The method belongs to the family of TSRK formulae (1.2) with $m = 2$ and it has been derived in order to exactly integrate each linear combination of the function basis

$$\{1, x, \exp(\pm\mu x), x\exp(\pm\mu x)\},$$

which is a subset of (2.13), corresponding to $K = P = 1$. EF5 method is characterized by the abscissa vector $c = [1/2, 3/4]^T$.

```
>> L[y[x]] := (y[x+h] - θ*y[x-h] - (1-θ)*y[x] - h*(v1*y'[x+(c1-1)*h]
           + v2*y'[x+(c2-1)*h] + w1*y'[x+c1*h] + w2*y'[x+c2*h]) /. x -> 0;
>> For[i = 1, i <= 2, Li[y[x]] = (y[x+ci*h] - ui*y[x-h] - (1-ui)*y[x]
           - h*(a_{i,1}*y'[x+(c1-1)*h] + a_{i,2}*y'[x+(c2-1)*h] + b_{i,1}*y'[x+c1*h]
           + b_{i,2}*y'[x+c2*h])) /. x -> 0; i++]
>> subs = {y[x_] -> x, y'[x_] -> 1};
>> eq1 = L[y[x]] /. subs;
>> eq2 = L1[y[x]] /. subs;
>> eq3 = L2[y[x]] /. subs;
>> {θ, u1, u2} = {θ, u1, u2} /.Together[Flatten[Solve[{eq1 == 0, eq2 == 0,
           eq3 == 0},{θ, u1, u2}]]];
>> h = z/mu;
>> subs1 = {y[x_] -> Exp[mu*x], y'[x_] -> mu*Exp[mu*x]};
>> subs2 = {y[x_] ->  Exp[-mu*x], y'[x_] -> -mu*Exp[-mu*x]};
>> Gp = FullSimplify[((L[y[x]] /. subs1) + (L[y[x]] /. subs2))/2];
>> Gm = FullSimplify[((L[y[x]] /. subs1) - (L[y[x]] /. subs2))/(2*z)];
>> Dp = FullSimplify[D[Gp, z]];
>> Dm = FullSimplify[D[Gm, z]];
>> {v1, v2, w1, w2} = {v1, v2, w1, w2} /. Together[Flatten[Solve[Gp == 0, Gm == 0,
           Dp == 0, Dm == 0, {v1, v2, w1, w2}]]]
>> Gp1 = FullSimplify[((L1[y[x]] /. subs1) + (L1[y[x]] /.subs2))/2];
>> Gp2 = FullSimplify[((L2[y[x]] /. subs1) + (L2[y[x]] /.subs2))/2];
>> Gm1 = FullSimplify[((L1[y[x]] /. subs1) - (L1[y[x]] /.subs2))/(2*z)];
>> Gm2 = FullSimplify[((L2[y[x]] /. subs1) - (L2[y[x]] /.subs2))/(2*z)];
>> Dp1 = FullSimplify[D[Gp1, z]];
>> Dp2 = FullSimplify[D[Gp2, z]];
>> Dm1 = FullSimplify[D[Gm1, z]];
```

```
>> Dm2 = FullSimplify[D[Gm2, z]];
>> {a1,1, a1,2, b1,1, b1,2, a2,1, a2,2, b2,1, b2,2} = {a1,1, a1,2, b1,1, b1,2, a2,1, a2,2, b2,1, b2,2} /.
        Together[Flatten[Solve[{Gp1 == 0, Gm1 == 0, Gp2 == 0, Gm2 == 0, Dp1 == 0,
        Dm1 == 0, Dp2 == 0, Dm2 == 0}, {a1,1, a1,2, b1,1, b1,2, a2,1, a2,2, b2,1, b2,2}]]]
```

The derived coefficients depend on the hyperbolic functions cosh and sinh. In order to convert them in terms of $\eta_k(Z)$-functions, the Mathematica package *formConv* can be used: it is described in [14] and can be downloaded from the web page

`http://cpc.cs.qub.ac.uk/summaries/AEFB_v1_0.html`

According to formula (3.18), the local error associated to this method is given by

$$lte^{EF}(x) = h^6 \frac{L_2^*(\mathbf{a}(Z))}{2Z^2} D^2(D-\mu)^2 y(x). \tag{6.28}$$

With the aim to derive the order of convergence of EF5 methods, we study the behaviour of classical order conditions of TSRK methods for z tending to 0. In particular we obtain that the stage order conditions

$$\frac{c^k}{k!} - \frac{(-1)^k}{k!}u - A\frac{(c-e)^{(k-1)}}{(k-1)!} - B\frac{c^{(k-1)}}{(k-1)!},$$

tend to 0 for k up to 5 and the same happens for the order conditions

$$\frac{1}{k!} - \frac{(-1)^k}{k!}\theta - v^T\frac{(c-e)^{(k-1)}}{(k-1)!} - w^T\frac{c^{(k-1)}}{(k-1)!}.$$

Therefore, the derived method has order and stage order both equal to 5. Concerning the linear stability properties of this method, we report in Figure 1 the stability region in the $(\mathrm{Re}\,\omega, \mathrm{Im}\,\omega, z)$-space and in Figure 2 the sections through the stability region by planes $z = -1$, $z = -2$, $z = -3$ and $z = -4$.

References

[1] Bartoszewski, Z.; Jackiewicz, Z. Stability analysis of two-step Runge-Kutta methods for delay differential equations, *Comput. Math. Appl.* 44 (2002), 83–93.

[2] Bartoszewski, Z.; Jackiewicz, Z. Nordsieck representation of two-step Runge-Kutta methods for ordinary differential equations, *Appl. Numer. Math.* 53 (2005), 149–163.

[3] Bartoszewski, Z.; Jackiewicz, Z. Derivation of continuous explicit two-step Runge-Kutta methods of order three, *J. Comput. Appl. Math.* 205 (2007), 764–776.

[4] Bartoszewski, Z.; Jackiewicz, Z. Construction of highly stable parallel two-step Runge-Kutta methods for delay differential equations, *J. Comput. Appl. Math.* 220 (2008), no 1-2, 257–270.

[5] Bartoszewski, Z.; Podhaisky, H.; Weiner, R. Construction of stiffly accurate two-step Runge-Kutta methods of order three and their continuous extensions using Nordsieck representation, Report No. 01, Martin-Luther-Universität Halle-Wittenberg, Fachbereich Mathematik und Informatik (2007).

[6] J.C. Butcher, The Numerical Analysis of Ordinary Differential Equations. Runge-Kutta and General Linear Methods, xvi+512 pp. John Wiley Sons, Chichester, New York (1987).

[7] Butcher, J. C.; Tracogna, S. Order conditions for two-step Runge-Kutta methods, *Appl. Numer. Math.* 24 (1997), 351–364.

[8] Chollom, J.; Jackiewicz, Z. Construction of two-step Runge-Kutta methods with large regions of absolute stability, *J. Comput. Appl. Math.* 157 (2003), 125–137.

[9] Coleman, J. P.; Ixaru, L.Gr. Truncation errors in exponential fitting for oscillatory problems, *SIAM J. of Numerical Analysis* 44 (2006), 1441-1465.

[10] Coleman, J. P.; Ixaru, L.Gr. *P*-stability and exponential-fitting methods for $y'' = f(x, y)$, *IMA J. Numer. Anal.* **16**, 2 (1996), 179–199.

[11] Conte, D.; D'Ambrosio, R.; Jackiewicz, Z., Two-step Runge-Kutta methods with quadratic stability functions, *J. Sci. Comput.* **44**, 2 (2010), 191–218.

[12] Conte, D.; D'Ambrosio, R.; Jackiewicz, Z.; Paternoster, B. Algebraically stable two-step collocation methods for ordinary differential equations, *submitted*.

[13] Conte, D.; D'Ambrosio, R.; Paternoster, B. Two–step diagonally–implicit collocation based methods for Volterra Integral Equations, in press on *Appl. Numer. Math.*

[14] Conte, D.; Esposito, E.; Paternoster, B.; Ixaru, L. Gr. Some new uses of the functions. Comput. Phys. Comm. 181 (2010), no. 1, 128–137.

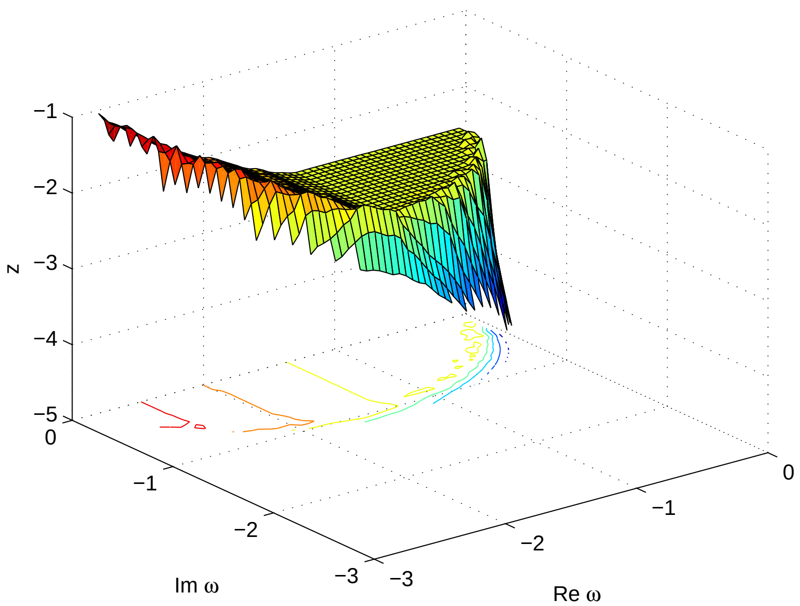

Figure 1: Stability region of the EF5 method in the (Re ω, Im ω, z)-space

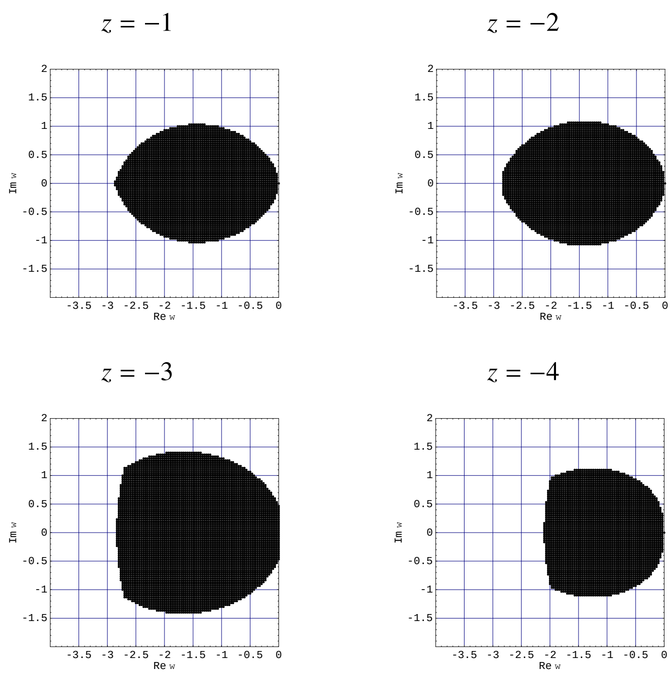

Figure 2: Projection of the stability region in the (Re ω, Im ω)-plane for the adapted TSRK method above reported.

[15] Conte, D.; Jackiewicz, Z.; Paternoster, B. Two-step almost collocation methods for Volterra integral equations, *Appl. Math. Comput.* 204 (2008), 839–853.
[16] D'Ambrosio, R.; Ferro, M.; Jackiewicz, Z.; Paternoster, B. Two-step almost collocation methods for ordinary differential equations, *Numer. Algorithms* 53 (2010), no. 2-3, 195-217.
[17] D'Ambrosio, R.; Ferro, M.; Paternoster, B. Trigonometrically fitted two-step hybrid methods for second order ordinary differential equations with one or two frequencies, *Math. Comput. Simulat.* 81 (5), 1068–1084 (2011).
[18] R. D'Ambrosio, L. Gr. Ixaru, B. Paternoster, Construction of the EF-based Runge-Kutta methods revisited, *Comput. Phys. Commun.* 182(2), 322–329 (2011).
[19] D'Ambrosio, R.; Izzo, G.; Jackiewicz, Z. Search for highly stable two-step Runge–Kutta methods for ordinary differential equations, in press on *Appl. Numer. Math*.
[20] D'Ambrosio, R.; Jackiewicz, Z. Continuous Two–Step Runge–Kutta Methods for Ordinary Differential Equations, *Numer. Algorithms* 54 (2), 169-193 (2010).
[21] D'Ambrosio, R.; Jackiewicz, Z. Construction and implementation of highly stable two-step continuous methods for stiff differential systems, *Math. Comput. Simulat.* 81(9),1707–1728 (2011).
[22] D'Ambrosio, R.; Paternoster, B. Two-step modified collocation methods with structured coefficient matrices, in press on *Appl. Numer. Math*.
[23] Franco, J. M. Exponentially fitted explicit Runge-Kutta-Nyström methods, *J. Comput. Appl. Math.* 167 (2004), no. 1, 1–19.
[24] Frank, J.E.; van der Houwen, P.J. Parallel Iteration of the Extended Backward Differentiation Formulas, *IMA J Numer Anal* 21 (2001), no. 1, 367–385.
[25] Hairer, E.; Wanner, G. Order conditions for general two-step Runge-Kutta methods, *SIAM J. Numer. Anal.* 34(1997), 2087-2089.
[26] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II – Stiff and Differential–Algebraic Problems, Springer Series in Computational Mathematics 14, Springer–Verlag, Berlin, 2002.
[27] Hollevoet, D.; Van Daele, M.; Vanden Berghe, G. The optimal exponentially-fitted Numerov method for solving two-point boundary value problems. J. Comput. Appl. Math. 230 (2009), no. 1, 260–269.
[28] Ixaru, L.Gr. Operations on oscillatory functions, *Comput. Phys. Comm.* 105 (1997), 1–19.
[29] Ixaru, L. Gr.; Vanden Berghe, G. *Exponential Fitting*, Kluwer Academic Publishers, Dordrecht, 2004.
[30] Jackiewicz, Z.; Tracogna, S. A general class of two-step Runge-Kutta methods for ordinary differential equations, *SIAM J. Numer. Anal.* 32 (1995), 1390–1427.
[31] Jackiewicz, Z.; Tracogna, S. Variable stepsize continuous two-step Runge-Kutta methods for ordinary differential equations, *Numer. Algorithms* 12 (1996), 347–368.
[32] Jackiewicz, Z.; Verner, J. H. Derivation and implementation of two-step Runge-Kutta pairs, *Japan J. Indust. Appl. Math.* 19 (2002), 227–248.
[33] Jackiewicz, Z. *General Linear Methods for Ordinary Differential Equations*, John Wiley & Sons, Hoboken, New Jersey, 2009.
[34] Lang, S. *Linear Algebra*, Springer, 1987.
[35] Martucci, S.; Paternoster, B. Vandermonde-type matrices in two step collocation methods for special second order Ordinary Differential Equations, in *Computational Science ICCS 2004*, ed. by M. Bubak em et al.; Lecture Notes in Comput. Sci. 3039, Part IV, Springer Verlag, 2004, 418-425.
[36] Konguetsof, A.; Simos, T. E. An exponentially-fitted and trigonometrically-fitted method for the numerical solution of periodic initial-value problems, *Comput. Math. Appl.* 45 (2003), no. 1-3, 547-554.
[37] Ozawa, K. A functional fitting Runge-Kutta method with variable coefficients, *Jpn. J. Ind. Appl. Math.* 18 (2001), no. 1, 107-130.
[38] Ozawa, K. A functional fitting Runge-Kutta-Nyström method with variable coefficients *Jpn. J. Ind. Appl. Math.* 19 (2002), no. 1, 55-85.
[39] A. Prothero and A. Robinson, On the stability and the accuracy of one-step methods for solving stiff systems of ordinary differential equations, *Math. Comput.*, 28, 145–162 (1974).
[40] Quarteroni, A.; Sacco, R.; Saleri, F. Numerical mathematics. Second edition. Texts in Applied Mathematics, 37. Springer-Verlag, Berlin, 2007. xviii+655 pp.
[41] Simos, T. E. An exponentially-fitted Runge-Kutta method for the numerical integration of initial-value problems with periodic or oscillating solutions, *Comput. Phys. Comm.* 115 (1), 1-8 (1998).
[42] Simos, T. E. Exponentially-fitted Runge-Kutta-Nyström method for the numerical solution of initial-value problems with oscillating solutions, *Appl. Math. Lett.* 15 (2002), no. 2, 217-225
[43] Tracogna, S. Implementation of two-step Runge-Kutta methods for ordinary differential equations, *J. Comput. Appl. Math.* 76 (1997), 113–136.
[44] Tracogna, S.; Welfert, B. Two-step Runge-Kutta: Theory and practice, *BIT* 40 (2000), 775–799.
[45] Vanden Berghe, G.; De Meyer, H.; Van Daele, M.; Van Hecke, T. Exponentially fitted Runge-Kutta methods, *Comput. Phys. Comm.* 123 (1999), 7-15.
[46] Vanden Berghe, G.; Van Daele, M. Exponentially-fitted Numerov methods, *J. Comput. Appl. Math.* 200 (2007), 140-153.
[47] Williams, P. S.; Simos, T. E. A new family of exponentially fitted methods, *Math. Comput. Modelling* 38 (2003), no. 5-6, 571-584.