



Article

A Fourth Order Symplectic and Conjugate-Symplectic Extension of the Midpoint and Trapezoidal Methods

Felice Iavernaro ¹ and Francesca Mazzia ^{2,*}

¹ Dipartimento di Matematica, Università degli studi di Bari Aldo Moro, 70125 Bari, Italy; felice.iavernaro@uniba.it

² Dipartimento di Informatica, Università degli studi di Bari Aldo Moro, 70125 Bari, Italy

* Correspondence: francesca.mazzia@uniba.it

Abstract: The paper presents fourth order Runge–Kutta methods derived from symmetric Hermite–Obreshkov schemes by suitably approximating the involved higher derivatives. In particular, starting from the multi-derivative extension of the midpoint method we have obtained a new symmetric implicit Runge–Kutta method of order four, for the numerical solution of first-order differential equations. The new method is symplectic and is suitable for the solution of both initial and boundary value Hamiltonian problems. Moreover, starting from the conjugate class of multi-derivative trapezoidal schemes, we have derived a new method that is conjugate to the new symplectic method.

Keywords: ordinary differential equations; Hamiltonian systems; multi-derivative methods; symplectic methods



Citation: Iavernaro, F.; Mazzia, F.; A Fourth Order Symplectic and Conjugate-Symplectic Extension of the Midpoint and Trapezoidal Methods. *Mathematics* **2021**, *9*, 1103. <https://doi.org/10.3390/math9101103>

Academic Editor: Higinio Ramos

Received: 14 April 2021

Accepted: 10 May 2021

Published: 13 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the present work, we will consider a suitable modification of multi-derivative one-step methods derived in [1] for the numerical solution of first order differential equations

$$y'(t) = f(y(t)), \quad t \in [t_0, T], \quad (1)$$

with sufficiently regular vector field $f : \mathbb{R}^m \rightarrow \mathbb{R}^m$, subject to initial conditions $y(t_0) = y_0$, or boundary conditions $g(y(t_0), y(T)) = 0$. The methods we are going to introduce are especially suited for the long time simulation of canonical Hamiltonian problems

$$y' = J\nabla H(y), \quad y(t_0) = y_0 \in \mathbb{R}^{2m}, \quad (2)$$

with

$$y = \begin{pmatrix} q \\ p \end{pmatrix}, \quad q, p \in \mathbb{R}^m, \quad J = \begin{pmatrix} O & I \\ -I & O \end{pmatrix}, \quad (3)$$

where q and p are the generalized coordinates and conjugate momenta, $H : \mathbb{R}^{2m} \rightarrow \mathbb{R}$ is the Hamiltonian function and I stands for the identity matrix of dimension m . The investigation of numerical methods for integrating differential equations such as (2) forms a branch of numerical analysis called *Geometric Integration*. Problem (2) admits the Hamiltonian function $H(y)$ as a first integral, namely $H(y(t)) = H(y_0)$ for $t \geq t_0$. It may admit additional constant of motions bringing important physical properties of the system that general-purpose codes would fail to reproduce in a long time simulation. Rather than focusing on the control of the accuracy in a given time interval of finite length, geometric integrators aim at reproducing the correct global behavior of the trajectory in the phase space, trying to retain the geometric features of the system itself. We refer the reader to the monographs [2–4] for the fundamental theory on the numerical treatment of conservative problems. Examples of the relevant role of geometric integration in several application areas may be found in the review papers [5–7] and references therein.

In [8,9] we analyzed two classes of one step symmetric Hermite–Obreshkov schemes with interesting symplectic properties and in [1] we focused on the symplectic properties of two families of multi-derivative high-order one-step methods which contains the well-known implicit midpoint and trapezoidal methods as seed formulae. Here, we consider a proper discretization of the Lie derivative appearing in these formulae, to define Runge–Kutta schemes with geometric properties. In the following, $y_1 = \Phi_h(y_0)$ denotes a generic one-step method of order $p > 0$ that provides the numerical solution of (2) with stepsize of integration $h > 0$. We recall a few definitions and properties which are relevant for our analysis. The one-step method Φ_h is called:

- *symplectic*, if its Jacobian matrix is symplectic, that is

$$\frac{\partial \Phi_h(y)}{\partial y}^\top J \frac{\partial \Phi_h(y)}{\partial y} = J, \quad \text{for all } y \in \mathbb{R}^{2m}. \tag{4}$$

- *conjugate-symplectic*, if it is topologically conjugate to a symplectic method $y_1 = \Psi_h(y_0)$, which means that a global change of coordinates $\chi_h(y) = y + O(h)$ exists such that

$$\Phi_h = \chi_h \circ \Psi_h \circ \chi_h^{-1}. \tag{5}$$

- *conjugate-symplectic up to order r*, if there exists a global change of coordinates $\chi_h(y) = y + O(h^p)$ such that (5) holds true, with the map Ψ_h satisfying

$$\frac{\partial \Psi_h(y)}{\partial y}^\top J \frac{\partial \Psi_h(y)}{\partial y} = J + O(h^{p+r+1}). \tag{6}$$

A symplectic method inherits relevant properties of the flow associated with a canonical Hamiltonian problem such as volume preservation of closed surfaces in the phase space under time evolution. For a detailed analysis of symplectic Runge–Kutta methods, see the monographs [2–4]. Further relevant features are the conservation of quadratic first integrals, and near conservation of the Hamiltonian function over exponentially long times ([4], page 366).

Conjugate-symplecticity leave these properties essentially unchanged (see [4] page 222 and [10]). In fact, from (5) we have

$$y_n = \Phi_h^n(y_0) = (\chi_h \circ \Psi_h \circ \chi_h^{-1})^n(y_0) = \chi_h \circ \Psi_h^n \circ \chi_h^{-1}(y_0).$$

The third property listed above is clearly a relaxation of conjugate-symplecticity. In this case, the method $\Phi_h(y)$ nearly conserves all quadratic first integrals and the Hamiltonian function over time intervals of length $O(h^{-r})$ (see [11]).

The starting point of our investigation is the class of Hermite–Obreshkov (HO) linear multistep methods [12].

$$\sum_{i=0}^k \alpha_i y_{n+i} = \sum_{j=1}^l h^j \sum_{i=0}^k \beta_{ji} y_{n+i}^{(j)}. \tag{7}$$

Here $y_{n+i}^{(j)}$ denotes an approximation to the j -th derivative of the solution $y(t)$ at t_{n+i} , with $t_{n+i} = t_n + ih$ and is defined as

$$y_n^{(j)} := D_{j-1} f(y_n), \quad j = 1, 2, \dots, l. \tag{8}$$

For a given integer $k \geq 0$ and $u \in \mathbb{R}^m$, $D_k(f(u))$ is the k -th order Lie derivative of the vector field f , defined as the k -th time derivative of $f(y(t))$ formally computed at $y(t) = u$, assuming that $y(t)$ satisfies the differential equation in (1) ($D_0 = I$ is the identity operator). We have used the subscript to define this operator to avoid confusion with the same order classical derivative operator denoted by D^k . Of course, the two operators take the same

values when applied to the projection of the true solution $y(t)$ on the mesh points but, in general, they will differ. Recently, we have introduced and analyzed four different one step ($k = 1$) HO methods:

- *Euler–Maclaurin methods*: higher derivative collocation methods deriving their name from the well-known Euler–Maclaurin integration formula. In [9] it has been shown that these integrators are conjugate-symplectic up to order $p + 2$.
- *BS Hermite–Obreshkov methods*: based on a special subclass of Hermite–Obreshkov methods which admit a continuous spline extension [8], these formulae are a multi-derivative generalization of the BS linear multistep methods derived in [13] and generalized in the field of quasi-interpolation in [14–16]. In [8] it has been shown that the BS Hermite–Obreshkov methods are conjugate-symplectic up to order $p + 2$.
- *multi-derivative midpoint and trapezoidal methods*: generalizations of the classical midpoint and trapezoidal methods, these formulae are derived by a combination of the implicit Taylor and the explicit Taylor expansion up to a given order [1]. The multi-derivative midpoint (MDMP) and trapezoidal (MDTR) methods are conjugate-symplectic up to order $p + 2$.

The analysis of these classes of multi-derivative methods has been also motivated by the possibility of computing the derivative efficiently, by exploiting the Infinity Computer arithmetic as described in [17–19]. In fact, the analytical computation of the j -th derivative $y^{(j)}$ involves a tensor of order j , which evidently heavily affects the computational cost associated with the implementation of the method. In this respect, the use of the Infinity Computer is able to accurately evaluate $y^{(j)}(z)$ without requiring its explicit expression in terms of the derivatives of f . In this paper, we consider the more natural approach that uses finite differences to approximate the Lie derivatives appearing in a given multi-derivative formula.

Since a certain degree of freedom is allowed in the choice of the derivative discretization stepsize, it turns out that the final full discretized formula may exhibit more favorable geometric properties with respect to the original one. This is the case for two special methods in the classes we are going to introduce: they form a pair of symplectic and conjugate-symplectic Runge–Kutta integrators that originate from the midpoint method and its conjugate-symplectic counterpart, namely the trapezoidal methods. To the best of our knowledge, no couple of symplectic and conjugate-symplectic Runge–Kutta methods of order higher than two has been devised up to now, so their existence constitutes the core result of the paper.

In addition, we introduce and analyze a new technique for solving the nonlinear systems emerging from the implementation of the methods. It consists of a block-diagonal variant of the simplified Newton scheme which requires, at each integration step, a single Jacobian evaluation of the vector field and a single LU factorization of a matrix having the same size of the continuous problem. Moreover, the diagonal structure of the linear systems involved in the iterative procedure, makes it suitable for a parallel implementation. A comparison of the new integrators with other existing symplectic integrators in terms of their efficiency is a delicate question and will not be addressed here.

The paper is organized as follows. In Section 2 we illustrate the multi-derivative fourth-order extensions of the midpoint and trapezoidal methods, while their modifications obtained by approximating the Lie derivatives are introduced in Sections 3 and 4 respectively. Section 5 analyzes the above-mentioned nonlinear systems solver needed to advance the solution in time. Some numerical illustrations are presented in Section 6. Finally, Section 7 contains some concluding remarks.

2. MDMP and MDTR Methods

The multi-derivative generalization of the midpoint (MP) and trapezoidal (TR) methods we are interested in is easily obtained via a standard Taylor approach by exploiting the property that such schemes may be viewed as composition of the Implicit (IE) and Explicit Euler (EE) methods, in direct and reverse order, applied on half the integration time-step:

MP = EE ◦ IE:

$$y_{n+1} = y_n + hf\left(\frac{y_n + y_{n+1}}{2}\right) \iff \begin{cases} y_{n+1/2} = y_n + \frac{h}{2}f(y_{n+1/2}), \\ y_{n+1} = y_{n+1/2} + \frac{h}{2}f(y_{n+1/2}), \end{cases}$$

TR = IE ◦ EE:

$$y_{n+1} = y_n + \frac{h}{2}(f(y_n) + f(y_{n+1})) \iff \begin{cases} y_{n+1/2} = y_n + \frac{h}{2}f(y_n), \\ y_{n+1} = y_{n+1/2} + \frac{h}{2}f(y_{n+1}). \end{cases}$$

We focus on the two conjugate classes of the MDMP and MDTR methods of order four. By denoting as ET4 (IT4) the explicit (implicit) Taylor method of order four we have that

MDMP4 = ET4 ◦ IT4:

$$y_{n+1} = y_n + hf(y_{n+1/2}) + \frac{h^3}{24}D_2f(y_{n+1/2}),$$

$$\iff \begin{cases} y_{n+1/2} = y_n + \frac{h}{2}f(y_{n+1/2}) - \frac{h^2}{8}D_1f(y_{n+1/2}) + \frac{h^3}{48}D_2f(y_{n+1/2}), \\ y_{n+1} = y_{n+1/2} + \frac{h}{2}f(y_{n+1/2}) + \frac{h^2}{8}D_1f(y_{n+1/2}) + \frac{h^3}{48}D_2f(y_{n+1/2}), \end{cases}$$

MDTR4 = IT4 ◦ ET4:

$$y_{n+1} = y_n + \frac{h}{2}(f(y_{n+1}) + f(y_n)) - \frac{h^2}{8}(D_1f(y_{n+1}) - D_1f(y_n)) + \frac{h^3}{48}(D_2f(y_{n+1}) + D_2f(y_n)),$$

$$\iff \begin{cases} y_{n+1/2} = y_n + \frac{h}{2}f(y_n) + \frac{h^2}{8}D_1f(y_n) + \frac{h^3}{48}D_2f(y_n), \\ y_{n+1} = y_{n+1/2} + \frac{h}{2}f(y_{n+1}) - \frac{h^2}{8}D_1f(y_{n+1}) + \frac{h^3}{48}D_2f(y_{n+1}). \end{cases}$$

Here we introduce and analyze two families of methods depending on a real parameter, which are derived by approximating the Lie derivative appearing in the formulae above by means of suitable difference schemes. These latter are defined with the aid of auxiliary local steps that will be exploited for this purpose. We observe that two Lie derivatives used in the MDMP4 and MDTR4 methods are multiplied respectively by h^2 and h^3 , so we shall approximate them by means of symmetric difference schemes of order at least two to preserve the symmetry and order properties of the original methods.

For the same reason, the formulae used to approximate the solution in the additional local steps should also be at least of order two. In particular, they take the form of implicit or explicit methods of order two so that the symmetry condition of the resulting method is preserved. In the next two sections we introduce these new fourth-order methods.

3. Approximated MDMP

In this section, we show two generalizations of the MDMP4 method. All the presented extensions are based on the computation of two local approximations of the solution in the two additional steps

$$t_{n+1/2-\alpha} = t_{n+1/2} - \alpha h \quad \text{and} \quad t_{n+1/2+\alpha} = t_{n+1/2} + \alpha h,$$

where α is a real positive parameter. In all the cases to approximate the first and second Lie derivatives we use the following standard finite differences schemes of order two:

$$\begin{aligned} \hat{D}_1f_{n+1/2} &= \frac{f(y_{n+1/2+\alpha}) - f(y_{n+1/2-\alpha})}{2\alpha h}, \\ \hat{D}_2f_{n+1/2} &= \frac{f(y_{n+1/2+\alpha}) - 2f(y_{n+1/2}) + f(y_{n+1/2-\alpha})}{\alpha^2 h^2}. \end{aligned} \tag{9}$$

3.1. Computation of the Additional Stages Using the Standard Explicit RK Method of Order 2

Let us approximate the value of $y(t)$ at $t_{n+1/2-\alpha}$ and $t_{n+1/2+\alpha}$ by using the explicit Runge–Kutta method of order 2 backward and forward, starting at $t_{n+1/2}$. The obtained values are used to compute the approximation of the derivatives using the central differences formulas in (9). The resulting method is

$$\begin{aligned}
 Y_{n+\frac{1}{2}-\alpha} &= y_{n+\frac{1}{2}} - h\alpha f_{n+\frac{1}{2}}, \\
 y_{n+\frac{1}{2}-\alpha} &= y_{n+\frac{1}{2}} - \frac{h\alpha}{2}(F_{n+\frac{1}{2}-\alpha} + f_{n+\frac{1}{2}}), \\
 Y_{n+\frac{1}{2}+\alpha} &= y_{n+\frac{1}{2}} + h\alpha f_{n+\frac{1}{2}}, \\
 y_{n+\frac{1}{2}+\alpha} &= y_{n+\frac{1}{2}} + \frac{h\alpha}{2}(F_{n+\frac{1}{2}+\alpha} + f_{n+\frac{1}{2}}), \\
 y_{n+\frac{1}{2}} &= y_n + \frac{h}{2}f_{n+\frac{1}{2}} - \frac{h^2}{8}\hat{D}_1f_{n+\frac{1}{2}} + \frac{h^3}{48}\hat{D}_2f_{n+\frac{1}{2}}, \\
 y_{n+1} &= y_n + hf_{n+\frac{1}{2}} + \frac{h^3}{24}\hat{D}^2f_{n+\frac{1}{2}}
 \end{aligned}$$

where $f_{n+\frac{1}{2}} = f(y_{n+\frac{1}{2}})$, $Y_{n+\frac{1}{2}\pm\alpha}$ are the stages of the two Runge–Kutta steps and $F_{n+\frac{1}{2}\pm\alpha} = f(Y_{n+\frac{1}{2}\pm\alpha})$. We call this method AMDMP4_RK2. Written as a Runge–Kutta scheme the method is described by the tableau

$$\begin{array}{c|c} c & A \\ \hline & b \end{array} \equiv \begin{array}{c|ccccc} 1/2 - \alpha & \frac{1}{16\alpha} + b_1 & -\frac{\alpha}{2} & -\frac{\alpha}{2} + b_3/2 & 0 & -\frac{1}{16\alpha} + b_5 \\ 1/2 - \alpha & \frac{1}{16\alpha} + b_1 & 0 & -\alpha + b_3/2 & 0 & -\frac{1}{16\alpha} + b_5 \\ 1/2 & \frac{1}{16\alpha} + b_1 & 0 & b_3/2 & 0 & -\frac{1}{16\alpha} + b_5 \\ 1/2 + \alpha & \frac{1}{16\alpha} + b_1 & 0 & \alpha + b_3/2 & 0 & -\frac{1}{16\alpha} + b_5 \\ 1/2 + \alpha & \frac{1}{16\alpha} + b_1 & 0 & \frac{\alpha}{2} + b_3/2 & \frac{\alpha}{2} & -\frac{1}{16\alpha} + b_5 \\ \hline & b_1 & 0 & b_3 & 0 & b_5 \end{array}$$

where $b_1 = b_5 = 1/(24\alpha^2)$ and $b_3 = 1 - 1/(12\alpha^2)$. The fourth-order conditions have been checked by exploiting the formulas in [12] (Table 2.2 p. 148). Applying the method to the scalar linear test problem $y' = \lambda y$, we obtain the recurrence relation $y_{n+1} = R(h\lambda)y_n$, where $R(z) = 1 + zb^T(I_5 - zA)^{-1}e$ is the stability function (I_5 denotes the identity matrix of size 5 and $e = (1, 1, 1, 1, 1)^T$). For a general introduction to the linear stability theory of Runge–Kutta schemes, we refer to [20] (chapter IV.3) and [21]. It turns out that, for AMDMP4_RK2 methods, the stability function actually does not depend on the parameter α . Setting, as usual, $q = h\lambda$, it takes the form

$$R(q) = \frac{P(q)}{Q(q)} = \frac{q^3 + 6q^2 + 24q + 48}{-q^3 + 6q^2 - 24q + 48}.$$

Since $P(q) = Q(-q)$, we get $|R(q)| = 1$ on the imaginary axis and since the poles of $R(q)$ have a positive real part, the maximum modulus principle shows that all the formulae in the family are precisely A-stable, that is its domain of absolute stability coincides with \mathbb{C}^- .

In comparison with the original MDMP method we see that the computational cost for the implementation of the new formulae decreases, because we just need to add the computation of the two explicit steps in the nonlinear iteration.

3.2. Approximation of the Derivative Using the Trapezoidal Scheme of Order 2

The second family of methods is derived by approximating the Lie derivatives with the aid of the backward and forward trapezoidal schemes. The resulting method is

$$\begin{aligned}
 y_{n+\frac{1}{2}-\alpha} &= y_{n+\frac{1}{2}} - \frac{h\alpha}{2}(f_{n+\frac{1}{2}-\alpha} + f_{n+\frac{1}{2}}) \\
 y_{n+\frac{1}{2}+\alpha} &= y_{n+\frac{1}{2}} + \frac{h\alpha}{2}(f_{n+\frac{1}{2}+\alpha} + f_{n+\frac{1}{2}}) \\
 y_{n+\frac{1}{2}} &= y_n + \frac{h}{2}f_{n+\frac{1}{2}} - \frac{h^2}{8}\hat{D}_1f_{n+\frac{1}{2}} + \frac{h^3}{48}\hat{D}_2f_{n+\frac{1}{2}}, \\
 y_{n+1} &= y_n + hf_{n+\frac{1}{2}} + \frac{h^3}{24}\hat{D}_2f_{n+\frac{1}{2}}
 \end{aligned}$$

Written as a Runge–Kutta scheme we have the following tableau

$1/2 - \alpha$	$-\frac{\alpha}{2} + \frac{1}{16\alpha} + \frac{1}{48\alpha^2}$	$-\frac{\alpha}{2} + \frac{1}{2} - \frac{1}{24\alpha^2}$	$\frac{1}{16\alpha} + \frac{1}{48\alpha^2}$	(10)
$1/2$	$\frac{1}{16\alpha} + \frac{1}{48\alpha^2}$	$\frac{1}{2} - \frac{1}{24\alpha^2}$	$-\frac{1}{16\alpha} + \frac{1}{48\alpha^2}$	
$1/2 + \alpha$	$\frac{1}{16\alpha} + \frac{1}{48\alpha^2}$	$-\frac{\alpha}{2} + \frac{1}{2} + \frac{1}{24\alpha^2}$	$\frac{\alpha}{2} + \frac{1}{16\alpha} + \frac{1}{48\alpha^2}$	
	$\frac{1}{24\alpha^2}$	$1 - \frac{1}{12\alpha^2}$	$\frac{1}{24\alpha^2}$	

and it is easy to check that its order is four. More interestingly, within this family we may discover a new fourth-order symplectic Runge–Kutta formula.

Theorem 1. *The Runge–Kutta scheme defined by the tableau (10) is A-stable for $\alpha < 1/\sqrt{6}$ and it is symplectic if we choose $\alpha = \sqrt{2}/4$.*

Proof. The stability function $R(q)$ is equal to the following rational function:

$$R(q) = \frac{P(q)}{Q(q)} = \frac{-(6\alpha^2 - 1)q^3 - (12\alpha^2 - 6)q^2 + 24q + 48}{(6\alpha^2 - 1)q^3 - (12\alpha^2 - 6)q^2 - 24q + 48}, \tag{11}$$

from which $P(q) = Q(-q)$ and hence $|R(q)| = 1$ on the imaginary axis. In order to impose that the poles of R lie in the right-half of the complex plane, we can apply the Routh–Hurwitz stability criterion to $Q(-q) = P(q)$. A direct computation then leads to the condition $\alpha < 1/\sqrt{6}$ for the roots of $P(q)$ to have negative real part. Consequently, for these values of α , the method is precisely A-stable. A sufficient condition for symplecticity (see [4] (Theorem 4.3) or [22]) is $b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, i, j = 1, 2, 3$. It turns out that this condition is satisfied only if we choose $\alpha = \sqrt{2}/4$. \square

The new symplectic RK method is

$1/2 - \frac{\sqrt{2}}{4}$	$\frac{1}{6}$	$\frac{1}{6} - \frac{\sqrt{2}}{8}$	$\frac{1}{6} - \frac{\sqrt{2}}{8}$	(12)
$1/2$	$\frac{1}{6} + \frac{\sqrt{2}}{8}$	$\frac{1}{6}$	$\frac{1}{6} - \frac{\sqrt{2}}{4}$	
$1/2 + \frac{\sqrt{2}}{4}$	$\frac{1}{6} + \frac{\sqrt{2}}{8}$	$\frac{1}{6} + \frac{\sqrt{2}}{8}$	$\frac{1}{6}$	
	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	

Symplectic Runge–Kutta schemes with three stages and order four are already known in the literature, see for example [23]. The method (12) has the special property of being defined as the composition of two formulae: this suggests that the method obtained by the reverse composition is conjugate to (12), and thus we get a couple of symplectic/conjugate-symplectic Runge–Kutta schemes that extend to the fourth-order the well-known couple formed by the midpoint and trapezoidal methods. The conjugate-symplectic method associated with (12) is introduced in the next section.

4. Approximated MDTR4 Methods

In this section, we devise two classes of methods obtained by approximating the Lie derivatives appearing in the MDTR4 formulae: each method in these families are conjugate to the corresponding one derived in Section 3. All the generalizations are based on the computation of two local approximations of the solution in two additional steps depending on a positive parameter α . This time, the new additional steps are

$$t_{n+1-\alpha} = t_{n+1} - \alpha h \quad \text{and} \quad t_{n+1+\alpha} = t_{n+1} + \alpha h.$$

To approximate the Lie derivatives, we use the same finite differences schemes of order two used in (9) for the AMDMP4 methods:

$$\hat{D}_1 f_{n+1} = \frac{f(y_{n+1+\alpha}) - f(y_{n+1-\alpha})}{2\alpha h} \tag{13}$$

$$\hat{D}_2 f_{n+1} = \frac{f(y_{n+1+\alpha}) - 2f(y_{n+1}) + f(y_{n+1-\alpha}))}{\alpha^2 h^2} \tag{14}$$

4.1. Approximation of the Derivative Using the Standard Explicit RK Method of Order 2

As in the previous section, the first family of methods is derived by employing the second order explicit Runge–Kutta methods to approximate the solution in the two extra abscissae $t_{n+1-\alpha}$ and $t_{n+1+\alpha}$. We get

$$\begin{aligned} Y_{n+1-\alpha} &= y_{n+1} - h\alpha f_{n+1}, \\ y_{n+1-\alpha} &= y_{n+1} - h\frac{\alpha}{2}(F_{n+1-\alpha} + f_{n+1}), \\ Y_{n+1+\alpha} &= y_{n+1} + h\alpha f_{n+1}, \\ y_{n+1+\alpha} &= y_{n+1} + h\frac{\alpha}{2}(F_{n+1+\alpha} + f_{n+1}), \\ y_{n+1} &= y_n + \frac{h}{2}(f_n + f_{n+1}) - \frac{h^2}{8}(\hat{D}_1 f_n - \hat{D}_1 f_{n+1}) \\ &\quad + \frac{h^3}{48}(\hat{D}_2 f_n + \hat{D}_2 f_{n+1}). \end{aligned} \tag{15}$$

This method, denoted by AMDTR4_RK2, is symmetric and has order four. Written as a Runge–Kutta scheme, it has $s = 10$ stages and the following tableau:

$-\alpha$	0	$-\frac{\alpha}{2}$	$-\frac{\alpha}{2}$	0	0	0	0	0	0	0
$-\alpha$	0	0	$-\alpha$	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
α	0	0	α	0	0	0	0	0	0	0
α	0	0	$\frac{\alpha}{2}$	$\frac{\alpha}{2}$	0	0	0	0	0	0
$1 - \alpha$	b_1	0	b_3	0	b_5	b_6	$-\frac{\alpha}{2}$	$-\frac{\alpha}{2} + b_8$	0	b_{10}
$1 - \alpha$	b_1	0	b_3	0	b_5	b_6	0	$-\alpha + b_8$	0	b_{10}
1	b_1	0	b_3	0	b_5	b_6	0	b_8	0	b_{10}
$1 + \alpha$	b_1	0	b_3	0	b_5	b_6	0	$\alpha + b_8$	0	b_{10}
$1 + \alpha$	b_1	0	b_3	0	b_5	b_6	0	$\frac{\alpha}{2} + b_8$	$\frac{\alpha}{2}$	b_{10}
	b_1	0	b_3	0	b_5	b_6	0	b_8	0	b_{10}

where the non null weights b_i are

$$b_1 = b_{10} = -\frac{1}{16\alpha} + \frac{1}{48\alpha^2}, \quad b_3 = b_8 = \frac{1}{2} - \frac{1}{24\alpha^2}, \quad b_5 = b_6 = \frac{1}{16\alpha} + \frac{1}{48\alpha^2}.$$

The coefficient matrix A has a block structure with many vanishing elements. The AMDTR4_RK2 scheme may be also cast as a parameterized implicit Runge–Kutta (PIRK) method having the general form

$$Y_i = v_i y_n + (1 - v_i) y_{n+1} + h \sum_{j=1}^s x_{i,j} f(t_n + c_j h, Y_j), \quad i = 1, \dots, s$$

$$y_{n+1} = y_n + h \sum_{j=1}^s b_j f(t_n + c_j h, Y_j),$$

and represented by the tableau

$$\begin{array}{c|c|c} c & v & X \\ \hline & & b^T \end{array}$$

with $c_i = v_i + \sum_{j=1}^s x_{i,j}$. Notice that a PIRK is equivalent to a RK with $A = X + vb^T$. Order results for this class of methods are reported in [24], where the special subclass of mono-implicit Runge–Kutta (MIRK) methods is analyzed. The MIRK class has been investigated by many authors since it has the interesting feature that the matrix X is strictly lower triangular. In this form, the method AMPTR_RK2 has the following tableau:

$$\begin{array}{c|c|cccccccccccc} -\alpha & 0 & 0 & -\frac{\alpha}{2} & -\frac{\alpha}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\alpha & 0 & 0 & 0 & -\alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha & 0 & 0 & 0 & \frac{\alpha}{2} & \frac{\alpha}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 - \alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{\alpha}{2} & -\frac{\alpha}{2} & 0 & 0 & 0 & 0 \\ 1 - \alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\alpha & 0 & 0 & 0 & 0 \\ 1 & 0 & b_1 & 0 & b_3 & 0 & b_5 & b_6 & 0 & b_8 & 0 & b_{10} & 0 & 0 \\ 1 + \alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 \\ 1 + \alpha & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{\alpha}{2} & \frac{\alpha}{2} & 0 & 0 & 0 \\ \hline & & b_1 & 0 & b_3 & 0 & b_5 & b_6 & 0 & b_8 & 0 & b_{10} & 0 & 0 \end{array}$$

Separating the block involving t_n with the one involving t_{n+1} leads to another representation of the AMDTR4_RK2 method as a one-step block method with $s = 5$ stages, namely

$$TZ_{n+1} = e_3 e_3^T Z_n + h e_3 d^T G_n + h B G_{n+1},$$

where (see (15))

$$Z_n = \begin{pmatrix} Y_{n-\alpha} \\ y_{n-\alpha} \\ y_n \\ Y_{n+\alpha} \\ y_{n+\alpha} \end{pmatrix}, \quad G_n = \begin{pmatrix} f(Y_{n-\alpha}) \\ f(y_{n-\alpha}) \\ f(y_n) \\ f(Y_{n+\alpha}) \\ f(y_{n+\alpha}) \end{pmatrix}, \quad d = \begin{pmatrix} b_1 \\ 0 \\ b_3 \\ 0 \\ b_5 \end{pmatrix}, \quad e_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$

$$T = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & -\alpha/2 & -\alpha/2 & 0 & 0 \\ 0 & 0 & -\alpha & 0 & 0 \\ b_6 & 0 & b_8 & 0 & b_{10} \\ 0 & 0 & \alpha & 0 & 0 \\ 0 & 0 & \alpha/2 & \alpha/2 & 0 \end{pmatrix},$$

with T, B, d and e_3 used as linear operators, to simplify the notation avoiding the explicit use of the Kronecker products. This latter expression, besides being more compact than the previous ones, leads to a better implementation in terms of computational efficiency.

Applied to the test equation $y' = \lambda y$, the AMDTR4_RK2 method defines the following recursion:

$$Y_{n+1} = (I - qT^{-1}B)^{-1}T^{-1}(e_3e_3^T + qe_3d^T)Y_n = G(q)Y_n$$

where $q = h\lambda$. The matrix $G(q)$ has four eigenvalues equal to zero and one equal to the following rational function:

$$R(q) = \frac{P(q)}{Q(q)} = \frac{q^3 + 6q^2 + 24q + 48}{-q^3 + 6q^2 - 24q + 48}$$

Again we see that $P(q) = Q(-q)$ and a direct computation shows that $R(q)$ is analytic in the left-half of the complex plane, so the method is precisely A-stable.

4.2. Approximation of the Derivative Using the Trapezoidal Scheme of Order 2

Approximating the solution in the additional points by the trapezoidal scheme yields

$$\begin{aligned} y_{n+1-\alpha} &= y_{n+1} - h\frac{\alpha}{2}(f_{n+1-\alpha} + f_{n+1}), \\ y_{n+1+\alpha} &= y_{n+1} + h\frac{\alpha}{2}(\frac{1}{3}f_{n+1} + f_{n+1+\alpha}), \\ y_{n+1} &= y_n + \frac{h}{2}(f_{n+1} + f_n) - \frac{h^2}{8}(\hat{D}_1f_n - \hat{D}_1f_{n+1}) + \frac{h^3}{48}(\hat{D}_2f_n + \hat{D}_2f_{n+1}). \end{aligned}$$

These formulae, denoted by AMDTR4_TR2, have order 4. Choosing $\alpha = \sqrt{2}/4$ we get a method which is conjugate to the corresponding method AMDMP4_TR2 defined at (12). In block form, this method assumes the following shape:

$$TZ_{n+1} = e_2e_2^T Z_n + he_2d^T G_n + hBG_{n+1},$$

where

$$\begin{aligned} Z_n &= \begin{pmatrix} y_{n-\alpha} \\ y_n \\ y_{n+\alpha} \end{pmatrix}, \quad G_n = \begin{pmatrix} f(y_{n-\alpha}) \\ f(y_n) \\ f(y_{n+\alpha}) \end{pmatrix}, \quad d = \begin{pmatrix} -\frac{1}{16\alpha} + \frac{1}{48\alpha^2} \\ \frac{1}{2} - \frac{1}{24\alpha^2} \\ \frac{1}{16\alpha} + \frac{1}{48\alpha^2} \end{pmatrix}, \quad e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \\ T &= \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} -\alpha/2 & -\alpha/2 & 0 \\ d_3 & d_2 & d_1 \\ 0 & \alpha/2 & \alpha/2 \end{pmatrix}, \end{aligned}$$

with T, B, d and e_2 used as linear operators.

Applied to the test equation $y' = \lambda y$ we have that the solution is defined by the following recursion:

$$Y_{n+1} = (I - qT^{-1}B)^{-1}T^{-1}(e_2e_2^T + qe_2d^T)Y_n = G(q)Y_n$$

where $q = h\lambda$. The matrix $G(q)$ has two vanishing eigenvalues and one eigenvalue equal to the same rational function defined at (11). From Theorem 1, it then follows that the formulae are precisely A-stable for $\alpha < 1/\sqrt{6}$.

5. Solution of the Nonlinear Systems

In order to optimize the computational cost associated with each method derived in the previous sections, we employ a suitable modified Newton scheme to solve the nonlinear systems emerging from their implementation. In particular, we approximate the Jacobian

matrix with a diagonal block-matrix with constant diagonal blocks. The derived iterative scheme for the AMDMP methods is the following

$$\left(I_s \otimes I - \frac{h}{\beta} (I_s \otimes J) \right) (Y_{n+1}^{r+1} - Y_{n+1}^r) = -(Y_{n+1}^r - e \otimes y_n) + h(A \otimes I)f(Y_{n+1}^r) \quad (16)$$

where $A \in \mathbb{R}^{s \times s}$ denotes the coefficient matrix of the method (s is the number of stages), $J \in \mathbb{R}^{m \times m}$ is the Jacobian matrix of the vector field f , evaluated at (t_n, y_n) , y_n is the solution computed at the previous step t_n , β is a positive parameter, while I_s and I stand for the identity matrices of size s and m respectively.

Theorem 2. *The nonlinear iteration scheme (16) applied to the AMDMP4_TR2 method for the solution of the test equations $y' = \lambda y$ is convergent for $\text{Re}(q) < 0$ if the spectral radius ρ of the matrix $\beta A - I$ is less than one. When $\rho > 1$, the method converges if*

$$|q| \in \left\{ \frac{\beta \cos \theta + \sqrt{\beta^2 (\cos^2 \theta + \rho^2 - 1)}}{\rho^2 - 1}, \quad -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2} \right\}$$

Proof. The scheme (16) applied to the test equation reduces to the following iteration:

$$Y_{n+1}^{r+1} = q \left(1 - \frac{q}{\beta}\right)^{-1} \left(A - \frac{1}{\beta} I\right) Y_{n+1}^r + \left(1 - \frac{q}{\beta}\right)^{-1} e \otimes y_n.$$

The eigenvalues of the iteration matrix are the eigenvalues of the matrix $\beta A - I$ scaled by $q/(\beta - q)$. It is straightforward to check that if $\rho < 1$ the method is convergent when $\text{Re}(q) < 0$. The region of convergence when $\rho > 1$ is computed by imposing that $\rho |q/(\beta - q)| < 1$. \square

Corollary 1. *The nonlinear iteration scheme (16) with $0 < \beta \leq 7$ applied to the AMDMP4_TR2 with $\alpha = \sqrt{2}/4$ for the solution of the test equations $y' = \lambda y$, converges if $\Re(\lambda) < 0$. The minimum value at ∞ is ≈ 0.5637 and is attained choosing $\beta \approx 4.6721$.*

Proof. The eigenvalues of the matrix $(\beta A - I)$ have absolute value less than one for $0 < \beta \leq 7$ with minimum value ≈ 0.5637 for $\beta \approx 4.6721$. \square

Observe that the iterative scheme (16) only requires one LU factorization of size m and the solution of the nonlinear systems could be easily performed in parallel. Depending on the structure of the problem, this is surely an interesting approach for solving nonlinear equations.

A similar argument may be applied to the solution of the nonlinear systems arising from the implementation of the AMDTR4_TR2 methods. The details are omitted for the sake of brevity.

6. Numerical Illustrations

In the present section, we compare the behavior of the newly-derived formulae with that of the original multi-derivative methods. These integrators have been applied to the well-known Kepler problem, a super-integrable Hamiltonian system that describes the motion of two bodies subject to Newton's law of gravitation (see, for example [25]). By setting the origin of the coordinate system on one of the two bodies, the Hamiltonian function takes the form

$$H(q, p) = \frac{1}{2}(p_1^2 + p_2^2) - \frac{1}{\sqrt{q_1^2 + q_2^2}}.$$

In particular, taking as initial conditions

$$q_1(0) = 1 - e, \quad q_2(0) = 0, \quad p_1(0) = 0, \quad p_2(0) = \sqrt{\frac{1+e}{1-e}},$$

the trajectory describes an ellipse with eccentricity e in the $q_1 - q_2$ plane and is periodic with period $T = 2\pi$. Besides the total energy H , further relevant first integrals are the angular momentum

$$M(q, p) = q_1 p_2 - q_2 p_1,$$

and the Lenz vector $A = (A_1, A_2, A_3)^\top$, whose components are

$$A_1(q, p) = p_2 M(q, p) - \frac{q_1}{\|q\|_2}, \quad A_2(q, p) = -p_1 M(q, p) - \frac{q_2}{\|q\|_2}, \quad A_3(q, p) = 0.$$

Of the four first integrals H, M, A_1 and A_2 , only three are independent so, for example, A_1 can be neglected.

Having set $e = 0.6$ and $h = T/200$, we integrated the problem over 10^3 periods and computed the error $\|y_n - y_0\|_1$ in the solution at specific times multiples of the period T , that is for $n = 200k$, with $k = 1, 2, \dots$. All computations have been carried out on an Intel i7 quad-core CPU with 16GB of RAM, running MATLAB R2020b.

Figures 1 and 2 report the results for the considered methods. On the top-left picture is the absolute error of the numerical solution; the top-right picture shows the error in the Hamiltonian function; the error in the angular momentum is drawn in the bottom-left picture, while the bottom-right picture concerns the error in the second component of the Lenz vector. To get more insights on the performance and conservation properties of the symplectic AMDMP_TR2 formula, in Figure 1 we also report the results for the Gauss method of order four. We observe that, to make the pictures more readable, the errors in the first integrals are computed at the midpoint of each period, that is at the points t_{n+100} .

As is expected from a symplectic or a conjugate-symplectic integrator, we can see a linear drift in the error $\|y_n - y_0\|_1$ as the time increases. The same linear growth is experienced in the Lenz invariant. The conjugate-symplectic AMDMP_TR2 scheme assures near conservation of the Hamiltonian function and angular momentum. This latter quadratic invariant is precisely conserved (up to machine precision) by the symplectic schemes.

In Tables 1 and 2, we compare the symplectic and conjugate-symplectic formulae with other methods in the same families, in terms of their ability in conserving the angular momentum. To this end, the value of $\alpha = \sqrt{2}/4$ that generates the symplectic and conjugate-symplectic schemes has been scaled by factors γ and $1/\gamma$, for a few values of $\gamma > 0$. We observe that when γ increases, the errors related to the decreasing values of α approach the value of the corresponding multi-derivative methods. As was expected, the angular momentum is exactly preserved by the symplectic scheme while the value attained at $t_{1/2} = t_0 + h/2$ is exactly preserved by the conjugate-symplectic one.

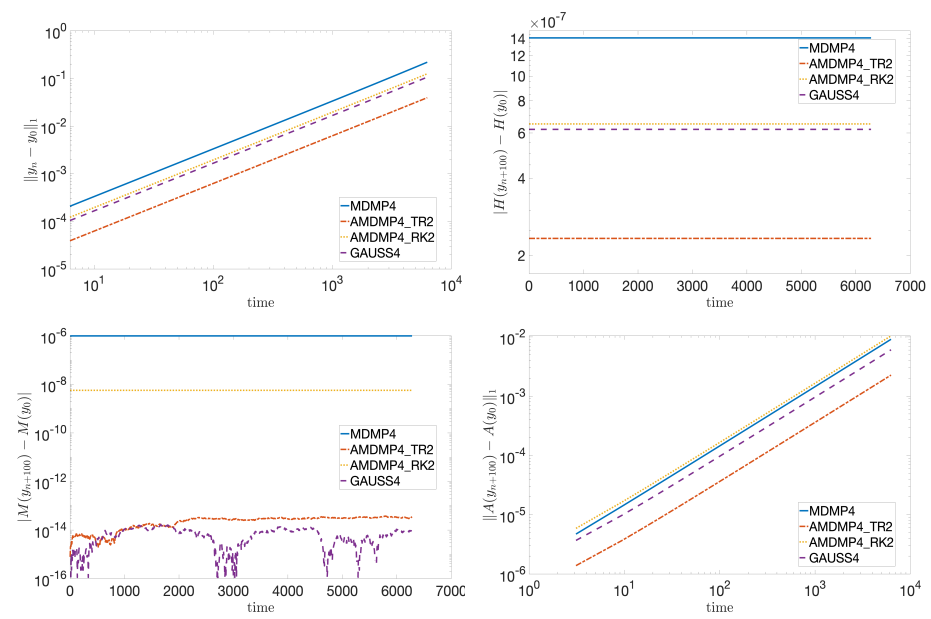


Figure 1. Results for the fourth-order MDMP (solid lines), Gauss (dashed lines), AMDMP_TR2 (dash-dotted lines) and AMDMP_R2 (dotted lines) methods applied to the Kepler problem computed at each period.

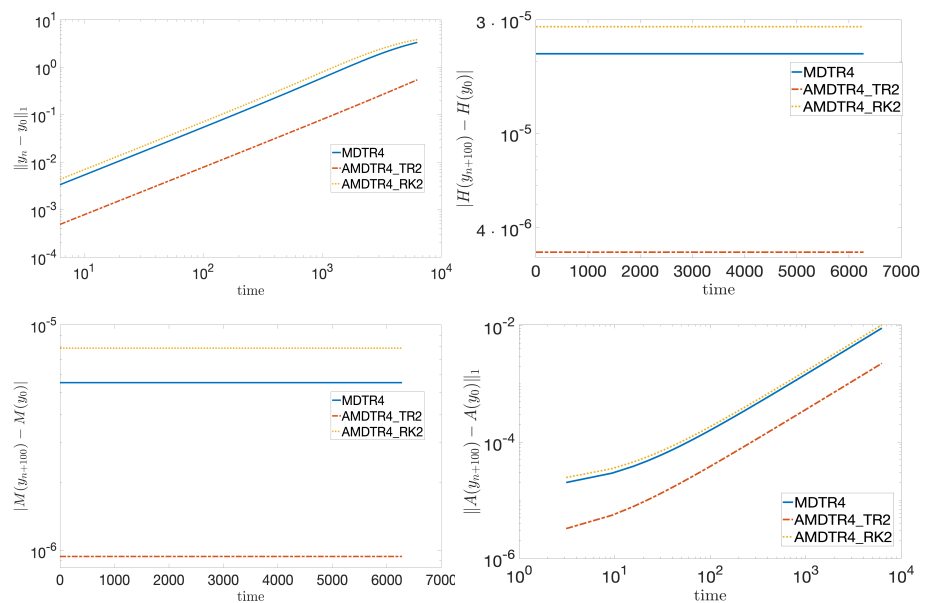


Figure 2. Results for the fourth-order MDTR (solid lines), Gauss (dashed lines), AMDTR_TR2 (dash-dotted lines) and AMDTR_R2 (dotted lines) methods applied to the Kepler problem computed at each period.

Table 1. Error in the angular momentum for the methods in the MDMP family.

AMDMP			
TR2		RK2	
α	$\frac{\sqrt{2}}{4\gamma}$	$\frac{\sqrt{2}\gamma}{4}$	$\frac{1}{2\gamma}$
γ			
1	5.32×10^{-15}	5.32×10^{-15}	3.60×10^{-7}
1.2	4.86×10^{-6}	6.97×10^{-6}	4.65×10^{-6}
1.4	7.81×10^{-6}	1.51×10^{-5}	7.67×10^{-6}
1.6	9.72×10^{-6}	2.45×10^{-5}	9.62×10^{-6}
1.8	1.10×10^{-5}	3.51×10^{-5}	1.09×10^{-5}
2.0	1.19×10^{-5}	4.68×10^{-5}	1.19×10^{-5}
3.0	1.42×10^{-5}	1.20×10^{-4}	1.41×10^{-5}
MDMP			
1.60×10^{-5}			

Table 2. Error in the angular momentum for the methods in the MDTR family, computed at the midpoints $t_{n+1/2}$. As reference value, we use the one computed at $t_{1/2}$.

AMDTR				AMDTR Midpoints			
TR2		RK2		TR2		RK2	
α	$\frac{\sqrt{2}}{4\gamma}$	$\frac{\sqrt{2}\gamma}{4}$	$\frac{1}{2\gamma}$	α	$\frac{\sqrt{2}}{4\gamma}$	$\frac{\sqrt{2}\gamma}{4}$	$\frac{1}{2\gamma}$
γ				γ			
1	1.55×10^{-5}	1.55×10^{-5}	1.32×10^{-4}	1	5.88×10^{-15}	5.88×10^{-15}	3.40×10^{-7}
1.2	1.89×10^{-5}	6.46×10^{-5}	6.34×10^{-5}	1.2	4.73×10^{-6}	6.78×10^{-6}	4.52×10^{-6}
1.4	3.96×10^{-5}	1.22×10^{-4}	2.13×10^{-5}	1.4	7.59×10^{-6}	1.47×10^{-5}	7.45×10^{-6}
1.6	5.31×10^{-5}	1.88×10^{-4}	6.87×10^{-6}	1.6	9.45×10^{-6}	2.38×10^{-5}	9.35×10^{-6}
1.8	6.23×10^{-5}	2.63×10^{-4}	2.55×10^{-5}	1.8	1.07×10^{-5}	3.41×10^{-5}	1.06×10^{-5}
2.0	6.90×10^{-5}	3.45×10^{-4}	3.91×10^{-5}	2.0	1.16×10^{-5}	4.55×10^{-5}	1.15×10^{-5}
3.0	8.47×10^{-5}	8.69×10^{-4}	7.14×10^{-5}	3.0	1.38×10^{-5}	1.17×10^{-4}	1.37×10^{-5}
MDTR				MDTR midpoints			
9.730×10^{-5}				1.55×10^{-5}			

To show the convergence properties of the diagonal nonlinear iteration (16) introduced in Section 5, we solved the problem over 10^2 periods with sepsize $h = T/N$, choosing $\beta = 4.6721$ (see Corollary 1). For comparison purposes, we also consider the use of the standard simplified Newton scheme defined by approximating the Jacobian matrix with $(I_3 \otimes I - hA \otimes J)$ (compare with (16)).

In Table 3 we show for the AMDMP4_TR2 method with $\alpha = \sqrt{2}/4$ the absolute error, the computed convergence rate and the mean number of iterations needed to reach the convergence up to machine precision for the two techniques. The scheme defined by the block-diagonal matrix requires around 1.7 more iterations to attain convergence, but the cost of the Jacobian factorization decreases from $27 m^3$ to m^3 . The execution times of the two algorithms for this problem are essentially equivalent, since the computation of the factorization is a built-in optimized function in Matlab so the differences do not consistently show up.

Table 3. Convergence rate and mean number of approximated Newton iterations for the AMDMP4_TR2 method with $\alpha = \sqrt{2}/4$.

N	Error	Order	Iterations Per Step (Simplified Newton)	Iterations Per Step (Scheme (16))
100	4.6981×10^{-2}		5.18	9.32
200	3.0275×10^{-3}	3.95	4.52	8.12
400	1.9059×10^{-4}	3.98	4.21	7.24
800	1.1933×10^{-5}	3.99	3.83	6.48

7. Conclusions

Starting from two fourth-order multi-derivative extensions of the midpoint and trapezoidal formulae, we have derived one-parameter families of standard Runge–Kutta methods through the approximation of the first and second-order Lie derivatives by means of suitable finite difference schemes. The resulting formulae retain the order and the symmetry properties of the original methods while, avoiding the use of Lie derivatives, their implementation turns out to be simplified. More interestingly, for a specific choice of the parameter, a symplectic/conjugate-symplectic pair of methods may be detected. This novel result generalizes to order four the well-known conjugacy relationship between the midpoint and the trapezoidal methods. Concerning the implementation of the methods, a block-diagonal version of the simplified Newton scheme has been employed. The iteration requires, at each step, a single Jacobian evaluation of the vector field and a LU factorization of a matrix having the same size of the continuous problem and, interestingly, may be executed in parallel.

Author Contributions: These authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the INdAM-GNCS 2020 Research Project “Numerical algorithms in optimization, ODEs, and applications” (the authors are members of the INdAM Research group GNCS).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Iavernaro, F.; Mazzia, F. On conjugate-symplecticity properties of a multi-derivative extension of the midpoint and trapezoidal methods. *Rend. Semin. Mat.* **2018**, *76*, 123–134.
- Feng, K.; Quin, M. *Symplectic Geometric Algorithms for Hamiltonian Systems*; Springer, Zhejiang Publishing United Group Zhejiang Science and Technology Publishing House: Zhejiang, China, 2010.
- Sanz-Serna, J.; Calvo, M. *Numerical Hamiltonian Problems*; Chapman & Hall: London, UK, 1994.
- Hairer, E.; Lubich, C.; Wanner, G. *Geometric Numerical Integration. Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd ed.; Springer: Berlin, Germany, 2006.
- Budd, C.J.; Piggott, M.D. Geometric Integration and its Applications. In *Handbook of Numerical Analysis*; North-Holland Publishing Co.: Amsterdam, The Netherlands, 2003; Volume 11, pp. 35–139.
- McLachlan, R.I. Perspectives on geometric numerical integration. *J. R. Soc. N. Z.* **2019**, *49*, 114–125. [[CrossRef](#)]
- Diele, F.; Marangi, C. Geometric numerical integration in ecological modelling. *Mathematics* **2020**, *8*, 25. [[CrossRef](#)]
- Mazzia, F.; Sestini, A. On a class of conjugate symplectic Hermite-Obreshkov one-step methods with continuous spline extension. *Axioms* **2018**, *7*, 58. [[CrossRef](#)]
- Iavernaro, F.; Mazzia, F.; Mukhametzhanov, M.; Sergeev, Y. Conjugate-symplecticity properties of Euler–Maclaurin methods and their implementation on the Infinity Computer. *Appl. Numer. Math.* **2020**, *155*, 58–72. [[CrossRef](#)]
- Hairer, E. Conjugate-symplecticity of linear multistep methods. *J. Comput. Math.* **2008**, *26*, 657–659.
- Hairer, E.; Zbinden, C.J. On conjugate symplecticity of B-series integrators. *IMA J. Numer. Anal.* **2013**, *33*, 57–79. [[CrossRef](#)]
- Hairer, E.; Nørsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations. I. Nonstiff Problems*, 2nd ed.; Springer Series in Computational Mathematics; Springer: Berlin, Germany, 1993.
- Mazzia, F.; Sestini, A.; Trigiante, D. The continuous extension of the B-spline linear multistep methods for BVPs on non-uniform meshes. *Appl. Numer. Math.* **2009**, *59*, 723–738. [[CrossRef](#)]
- Mazzia, F.; Sestini, A. The BS class of Hermite spline quasi-interpolants on nonuniform knot distributions. *BIT Numer. Math.* **2009**, *49*, 611–628. [[CrossRef](#)]

15. Mazzia, F.; Sestini, A. Quadrature formulas descending from BS Hermite spline quasi-interpolation. *J. Comput. Appl. Math.* **2012**, *236*, 4105–4118. [[CrossRef](#)]
16. Bracco, C.; Giannelli, C.; Mazzia, F.; Sestini, A. Bivariate hierarchical Hermite spline quasi-interpolation. *BIT Numer. Math.* **2016**, *56*, 1165–1188. [[CrossRef](#)]
17. Sergeyev, Y.; Mukhametzhanov, M.; Mazzia, F.; Iavernaro, F.; Amodio, P. Numerical methods for solving initial value problems on the infinity computer. *Int. J. Unconv. Comput.* **2016**, *12*, 3–23.
18. Iavernaro, F.; Mazzia, F. Symplecticity properties of Euler-Maclaurin methods. *AIP Conf. Proc.* **2018**, *1978*. [[CrossRef](#)]
19. Iavernaro, F.; Mazzia, F.; Mukhametzhanov, M.; Sergeyev, Y. Computation of higher order Lie derivatives on the Infinity Computer. *J. Comput. Appl. Math.* **2021**, *383*. [[CrossRef](#)]
20. Hairer, E.; Wanner, G. *Solving Ordinary Differential Equations II. Stiff and Differential Algebraic Problems*, 2nd ed.; Springer: Berlin, Germany, 1996.
21. Butcher, J. Runge-Kutta methods. *Scholarpedia* **2007**, *2*, 3147, revision #91735. [[CrossRef](#)]
22. Sanz-Serna, J. Runge-Kutta schemes for Hamiltonian systems. *BIT* **1988**, *28*, 877–883. [[CrossRef](#)]
23. Maeda, S. Certain types of Runge-Kutta-type formulas and reproduction of orbits of linear systems. *Electron. Commun. Jpn.* **1991**, *74*, 98–104. [[CrossRef](#)]
24. Burrage, K.; Chipman, F.; Muir, P. Order Results for Mono-Implicit Runge–Kutta Methods. *SIAM J. Numer. Anal.* **1994**, *31*, 876–891. [[CrossRef](#)]
25. Brugnano, L.; Iavernaro, F. *Line Integral Methods for Conservative Problems*; Monographs and Research Notes in Mathematics; CRC Press: Boca Raton, FL, USA, 2016.