

Fifth Order Runge-Kutta-Nyström Methods with Complex Coefficients

M. Atakan Gürkan^{1,2, a)}

¹⁾*Sterrewacht Leiden*

Niels Bohrweg 2, 2333 CA, Leiden, the Netherlands

²⁾*Sabanci University, Faculty of Engineering and Natural Sciences*

Orhanlı-Tuzla, 34956, İstanbul, Turkey

(Dated: 1 March 2012)

We present fifth order Runge-Kutta-Nyström methods, where we allow the timestep coefficients to assume complex values. Among the methods with complex timesteps, we focus on the ones with the coefficients that have positive real parts. This property makes them suitable for problems where a negative coefficient is not acceptable. In addition, the leading order terms in the error expansion of these methods are purely imaginary, effectively increasing the order of the methods by one for real variables.

PACS numbers: 02.60.Jh,02.30.Hq,02.60.Cb,95.10.Ce

^{a)}Electronic mail: ato.gurkan@gmail.com

I. INTRODUCTION

Splitting methods¹ provide a number of advantages for studying the evolution of Hamiltonian systems. Not only are they simple to implement, but also they can be fine tuned to exploit the structure of the problem at hand^{2,3}. A large class of physical problems are described by the separable Hamiltonian

$$H(q, p) = T(p) + V(q), \quad (1)$$

where the kinetic energy $T(p) = \frac{1}{2}p^T M^{-1}p$ is quadratic in momenta, and the potential energy $V(q)$ is only a function of the coordinates. For these problems, Hamilton's equations

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \quad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i} \quad (2)$$

lead to the second order differential equation

$$\frac{d^2 q}{dt^2} = -M^{-1} \frac{\partial V(q)}{\partial q} = f(q). \quad (3)$$

This equation can be efficiently integrated using Runge–Kutta–Nyström (RKN) methods⁴.

RKN methods are particularly effective for orders higher than 4, since some of the terms in the error expansion vanish identically, thanks to the special structure in equation (3)^{5,6}. Zhu & Qin⁷ and Okunbor & Skeel⁸ found 5th order explicit canonical RKN methods with 5 stages, which is the minimum required by the order conditions. The stages in these methods have large (around unity) step size coefficients, some of which are negative. Large coefficients can lead to large factors for integrations errors⁶, while negative coefficients can make the method unacceptable for the underlying problem⁹. Starting from a splitting scheme, Chambers¹⁰ showed that these difficulties can be overcome by allowing the step size coefficients to be complex numbers, and found a third order splitting method with two stages. The coefficients for this method have small and positive real parts, leading to small errors. An interesting property of this method is that the leading order terms in the error expansion have purely imaginary coefficients. This makes the method effectively one order higher for real variables (see ref. 11, for other examples of such methods).

In this paper, we present multiple 5th order RKN methods with both real and complex step size coefficients. We put special emphasis on methods with coefficients that have small, positive real parts, and purely imaginary leading error terms.

II. 5TH ORDER CANONICAL RKN METHODS

For a system with Hamiltonian (1), we can define “velocity” as $\dot{q} = M^{-1}p$. Then an s -stage RKN method is given by^{5,8}

$$\begin{aligned} y_i &= q_n + c_i h \dot{q}_n + h^2 \sum_{j=1}^s a_{ij} f(y_j), \quad i = 1, 2, \dots, s, \\ q_{n+1} &= q_n + h \dot{q}_n + h^2 \sum_{i=1}^s b_i f(y_i), \quad \dot{q}_{n+1} = \dot{q}_n + h \sum_{i=1}^s B_i f(y_i), \end{aligned} \quad (4)$$

where $f(q)$ is defined in equation (3). This method is explicit, that is a step depends only on previous steps, if $a_{ij} = 0$ for $j \geq i$. An explicit s -stage RKN method without redundant stages is canonical if^{5,8}

$$b_i = B_i(1 - c_i), \quad 1 \leq i \leq s, \quad (5)$$

$$a_{ij} = B_j(c_i - c_j), \quad j < i. \quad (6)$$

For methods of order 5, the following order conditions must hold^{7,8}:

$$\begin{aligned} t_1 : \sum_i B_i &= 1, & t_6 : \sum_i \sum_{j < i} B_i B_j (c_i - c_j) &= \frac{1}{6}, \\ t_2 : \sum_i B_i c_i &= \frac{1}{2}, & t_7 : \sum_i \sum_{j < i} B_i B_j c_i (c_i - c_j) &= \frac{1}{8}, \\ t_3 : \sum_i B_i c_i^2 &= \frac{1}{3}, & t_8 : \sum_i \sum_{j < i} B_i B_j c_i^2 (c_i - c_j) &= \frac{1}{10}, \\ t_4 : \sum_i B_i c_i^3 &= \frac{1}{4}, & t_9 : \sum_i \sum_{j < i} B_i B_j c_i c_j (c_i - c_j) &= \frac{1}{30}, \\ t_5 : \sum_i B_i c_i^4 &= \frac{1}{5}, & t_{10} : \sum_i \sum_{j < i} \sum_{l < i} B_i B_j B_l (c_i - c_j)(c_i - c_l) &= \frac{1}{20}. \end{aligned} \quad (7)$$

By solving these equations, we can obtain an RKN method of order 5; note, however, that these equations by themselves do not give us the error behaviour.

III. SPLITTING SCHEME

Since we will be using a Hamiltonian splitting scheme for implementing our integrators and error analysis, we now briefly review the basics of this technique using Lie algebra, in a manner similar to the treatment of Yoshida¹².

First we write Hamilton's equations for a state $w(t) \equiv (p(t), q(t))$ as

$$\frac{dw}{dt} = \{w, H\} = \left(\sum_i \frac{\partial H}{\partial p_i} \frac{\partial}{\partial q_i} - \frac{\partial H}{\partial q_i} \frac{\partial}{\partial p_i} \right) w = \mathbb{H}w = (\mathbb{T} + \mathbb{V})w, \quad (8)$$

where \mathbb{H} , \mathbb{T} , \mathbb{V} are Lie operators⁹ corresponding to the Hamiltonian, the kinetic energy and the potential energy, respectively. The solution for $w(t)$ can then be written as

$$w(\tau) = e^{\tau \mathbb{H}} w(0), \quad (9)$$

and the exponential operator is approximated by

$$e^{\tau \mathbb{H}} \approx e^{\tau \mathbb{Z}} = \prod_i e^{\alpha_i \tau \mathbb{T}} e^{\beta_i \tau \mathbb{V}}. \quad (10)$$

The expansion for \mathbb{Z} in terms of \mathbb{T} and \mathbb{V} can be obtained by repeatedly applying the Baker-Campbell-Hausdorff (BCH) formula (for BCH expressions with minimum number of terms, see^{13,14}). The coefficients α_i , β_i are chosen such that $\tau \mathbb{Z} = \tau \mathbb{H} + O(\tau^{n+1})$, for a method of order n . Note that the expansion of \mathbb{Z} can be considerably simplified, since the commutator $[\mathbb{V}, [\mathbb{V}, [\mathbb{T}, \mathbb{V}]]]$ vanishes for the systems described by equation (3).

The evolution due to the exponential operators on the right hand side of equation (10) is given by simple displacements

$$e^{\alpha_i \tau \mathbb{T}}(q, p) = (q + \alpha_i \tau p, p) \quad \text{and} \quad e^{\beta_i \tau \mathbb{V}}(q, p) = (q, p + \beta_i \tau f(q)). \quad (11)$$

The relation between the coefficients of an RKN method and a splitting scheme is given by⁵:

$$\alpha_i = c_i - c_{i-1}, \quad \alpha_{s+1} = 1 - c_s, \quad \beta_i = B_i, \quad \beta_{s+1} = 0, \quad (12)$$

(where $1 \leq i \leq s$, $c_0 = 0$), leading to the following procedure for the method⁵:

$$\begin{aligned} y_0 &= q_n, \\ \dot{y}_0 &= \dot{q}_n, \\ \text{for } i &= 1, 2, \dots, s \\ y_i &= y_{i-1} + h(c_i - c_{i-1})\dot{y}_{i-1}, \\ \dot{y}_i &= \dot{y}_{i-1} + hB_i f(y_i), \\ q_{n+1} &= y_s + h(1 - c_s)\dot{y}_s \\ \dot{q}_{n+1} &= \dot{y}_s. \end{aligned}$$

This corresponds to a splitting scheme with the structure

$$e^{\tau\mathbb{H}} \approx e^{\tau\mathbb{Z}} = e^{\alpha_6\tau\mathbb{T}} e^{\beta_5\tau\mathbb{V}} e^{\alpha_5\tau\mathbb{T}} \dots e^{\beta_1\tau\mathbb{V}} e^{\alpha_1\tau\mathbb{T}}, \quad (13)$$

to which we will refer as **RKNA** scheme. Another method with the structure

$$e^{\tau\mathbb{H}} \approx e^{\tau\mathbb{Z}} = e^{\beta_6\tau\mathbb{V}} e^{\alpha_5\tau\mathbb{T}} e^{\beta_5\tau\mathbb{V}} \dots e^{\alpha_1\tau\mathbb{T}} e^{\beta_1\tau\mathbb{V}}, \quad (14)$$

to which we will refer as **RKNB** scheme, would have similar computational cost. However, because of the special structure of the RKN methods, the coefficients for the latter scheme would be entirely different⁶. To obtain the coefficients from the same order conditions, we rewrite equation (14) as

$$e^{\tau\mathbb{H}} \approx e^{\tau\mathbb{Z}} = e^{\alpha_7\tau\mathbb{T}} e^{\beta_6\tau\mathbb{V}} e^{\alpha_6\tau\mathbb{T}} e^{\beta_5\tau\mathbb{V}} \dots e^{\alpha_2\tau\mathbb{T}} e^{\beta_1\tau\mathbb{V}} e^{\alpha_1\tau\mathbb{T}}, \quad (15)$$

i.e., as a 6 stage method. Since $\alpha_7 = \alpha_1 = 0$, we have $c_1 = 0$ and $c_6 = 1$. Hence, the number of unknowns is again equal to the number of equations.

IV. CONSTRUCTING THE METHODS

We solved the order conditions, eq. (7), using **fsolve** routine of MAPLE, with **complex** optional keyword. We started from a large number of random initial guesses for B_i and c_i in the rectangle $-2 \leq \Re(z), \Im(z) \leq 2$. This yielded both of the previously known real solutions for **RKNA** scheme, along with their adjoints; as well as three real solutions for **RKNB** scheme that seems to have not been discovered before. In addition, we found a large number of complex solutions for both schemes.

Once the coefficients are calculated, we repeatedly applied BCH formula to calculate the leading order error terms as nested commutators of \mathbb{T} and \mathbb{V} . Because of the Jacobi identity and the simplification $[\mathbb{V}, [\mathbb{V}, [\mathbb{T}, \mathbb{V}]]] = 0$, not all commutators are independent. Also, to calculate the leading order error of the method, we are only interested in terms with up to 6 operators; so, we worked in a Philip Hall basis¹⁴ with 2 generators and nilpotency 7, leading to 23 terms. Denoting our generators by $X_1 = \mathbb{T}$ and $X_2 = \mathbb{V}$, we constructed a

“multiplication table”:

$$\begin{aligned}
 [X_1, X_2] &= X_3; [X_1, X_3] = X_4; [X_1, X_4] = X_6; [X_1, X_5] = X_7; \\
 [X_1, X_6] &= X_{12}; [X_1, X_7] = X_9 + X_{13}; [X_1, X_9] = X_{16}; \\
 [X_1, X_{10}] &= X_{11} + X_{17}; [X_1, X_{12}] = X_{19}; [X_1, X_{13}] = X_{16} + X_{20}; \\
 [X_1, X_{14}] &= -X_{11} - X_{17}; [X_2, X_3] = X_5; [X_2, X_4] = X_7; [X_2, X_6] = X_{13}; \\
 [X_2, X_7] &= -X_{10}; [X_2, X_9] = -X_{11} + X_{17}; [X_2, X_{12}] = X_{20}; \\
 [X_2, X_{13}] &= -3X_{17}; [X_3, X_4] = X_9; [X_3, X_5] = X_{10}; [X_3, X_6] = X_{16}; \\
 [X_3, X_7] &= X_{17}; [X_4, X_5] = X_{11}; \quad \text{with} \quad [X_i, X_j] = -[X_j, X_i].
 \end{aligned} \tag{16}$$

Apart from the commutators given in this table, all commutators of the basis operators vanish. Using this table, we wrote a `Python`¹⁵ program, using the `SymPy`¹⁶ package, to obtain the expansion for \mathbb{Z} up to and including 6 operator terms. This allowed us to validate the methods and calculate the coefficients for the leading order error.

For **RKNA** scheme, we found two previously published methods with real coefficients. For **RKNB** scheme, we found three methods with real coefficients that we did not find elsewhere in the literature. We give the coefficients for these methods in Table I.

We also found a large number of methods with complex coefficients. Among these, ten for **RKNA** scheme and five for **RKNB** scheme had coefficients with positive real parts and purely imaginary leading order errors. We give the coefficients for two methods with smallest error coefficients for each scheme in Table II.

The leading order errors are given by the coefficients in front of the six-term commutators in the expansion of \mathbb{Z} . These coefficients, for the methods presented, are given in Table III.

V. NUMERICAL EXPERIMENTS

As a simple validation, we first compare the behaviour of two of these methods (**RKNAC1** and **RKNBR1**) with other methods from the literature, for the gravitational two-body problem. We integrated the equations of motion of two equal point masses, on orbit around each other with eccentricity $e = 0.2$, for fifty orbital periods. To follow the orbit, we used a Gragg-Bulirsch-Stoer (GBS) integrator⁴. At each timestep, we also calculated the expected position and velocity for each of the methods we tested. The difference between the outcomes of GBS integrator and the method gives an estimate of the error made, as long as they are

TABLE I. Coefficients for real fifth order splitting schemes. The adjoint methods are also valid and can be obtained by reversing the order of the coefficients. The coefficients are not entirely independent since $\sum \alpha_i = \sum \beta_i = 1$.

	α_i	β_i
AR1	0.96172990014645096	0.39682804502722538
	-0.09525408032034999	-0.824377563589592
	-0.73942683539212613	0.2042028689314904
	0.62730935078241887	1.0021847152077973
	-0.52506178465602220	0.22116193442307898
	0.77070344943962849	—
AR2	0.69883375727545265	0.40090379269659899
	-0.49469565362085154	0.95997088013405985
	0.81641946634957295	0.0884951581272243
	-0.65762956677338285	1.2214390923487315
	-0.057841894299102682	-1.6708089233066146
	0.69491389106831146	—
BR1	0.54200976680171613	0.24566294009066009
	-0.04060817665564392	1.1433587581365421
	-0.87779698530109766	-1.3796706973507000
	0.86474236062251646	-0.019611260781217307
	0.51165303453250898	0.87087215441178844
	—	0.13938810549292669
BR2	0.42637413177222316	0.15102308452230116
	-0.82438794434938248	0.72768821316253478
	-0.63140077574154094	-0.26217627934521390
	0.38590710518893978	-0.044211509719803855
	1.6435074831297605	0.23596222045571453
	—	0.19171427092446728
BR3	1.0413749845202060	0.12696076271851077
	-0.61784769849171965	-1.4166626058695677
	0.62570540985789957	-0.62172666654176438
	-0.63446409452971410	0.69301448863793809
	0.58523139864332822	1.2079876026916669
	—	1.0104264183632164

TABLE II. Coefficients for a selection of complex fifth order splitting schemes, with purely imaginary leading order error terms. All the methods are skew-symmetric, i.e., reversing the order of the coefficients gives their complex conjugates. This property and $\sum_i \alpha_i = \sum_i \beta_i = 1$ can be used to calculate the missing coefficients. The adjoints of the methods, obtained by reversing the order of the coefficients, are also valid.

		\Re	\Im
AC1	α_1	0.087808410045663212	0.028523844251341822
	α_2	0.17916539354193987	-0.067857083007249973
	α_3	0.23302619641239692	-0.097952003128893425
	β_1	0.17526734338348050	0.057642040076250593
	β_2	0.18488007701471166	-0.19410647329733509
AC2	α_1	0.087634204536037057	0.028807372065269351
	α_2	0.18007104463252914	-0.068253589313355443
	α_3	0.23229475083143381	-0.097060961378624794
	β_1	0.17526840907207411	0.057614744130538702
	β_2	0.18487368019298416	-0.19412192275724959
BC1	β_1	0.093106790861751605	-0.026812950639104607
	β_2	0.14578332225686154	0.076033669531385746
	β_3	0.26110988688138685	0.10851236434561279
	α_1	0.15950063058390336	-0.060127448366782494
	α_2	0.19085044206705213	0.20369642527600502
BC2	β_1	0.10625796854753310	-0.037213537431233983
	β_2	0.35767992721948460	-0.022169204268009056
	β_3	0.036062104232982296	0.057072185585748646
	α_1	0.26934942679787788	-0.093675141997563700
	α_2	0.14580813747862993	0.49930185549019606

not dominated by the truncation errors arising from limited machine accuracy ($\sim 10^{-15}$). We then calculated 0.25 and 0.75 quantile and took their difference, to get the *interquartile range*. This is a robust statistic that gives a measure of the dispersion in data.

TABLE III. Coefficients of the terms in the expansion $\mathbb{H} - \mathbb{Z}$, for the methods given in Tables I and II.

	X_{11}	X_{16}	X_{17}	X_{19}	X_{20}	$(\sum_i X_i ^2)^{1/2}$
AR1	-3.1×10^{-3}	4.5×10^{-4}	5.6×10^{-3}	1.8×10^{-5}	-6.3×10^{-5}	6.44×10^{-3}
AR2	5.0×10^{-3}	-3.1×10^{-4}	-5.7×10^{-3}	1.5×10^{-5}	2.5×10^{-4}	7.58×10^{-3}
BR1	1.5×10^{-4}	6.2×10^{-5}	-4.1×10^{-4}	5.7×10^{-5}	1.6×10^{-4}	4.71×10^{-4}
BR2	-7.0×10^{-5}	9.7×10^{-4}	2.7×10^{-4}	5.2×10^{-4}	5.2×10^{-4}	1.25×10^{-3}
BR3	3.5×10^{-2}	1.9×10^{-4}	-3.5×10^{-2}	-3.0×10^{-5}	-1.0×10^{-4}	4.90×10^{-2}
AC1	$-i2.5 \times 10^{-6}$	$i7.3 \times 10^{-6}$	$-i5.0 \times 10^{-6}$	$i8.3 \times 10^{-7}$	$i4.2 \times 10^{-6}$	1.02×10^{-5}
AC2	$i3.0 \times 10^{-6}$	$-i7.9 \times 10^{-6}$	$i6.0 \times 10^{-6}$	$-i9.4 \times 10^{-7}$	$-i4.7 \times 10^{-6}$	1.15×10^{-5}
BC1	$i1.2 \times 10^{-5}$	$-i1.3 \times 10^{-5}$	$i8.4 \times 10^{-6}$	$i2.2 \times 10^{-7}$	$-i4.2 \times 10^{-6}$	1.96×10^{-5}
BC2	$-i4.1 \times 10^{-5}$	$i4.9 \times 10^{-5}$	$-i1.2 \times 10^{-4}$	$i7.7 \times 10^{-6}$	$i5.0 \times 10^{-5}$	1.49×10^{-4}

In figure 1, we plot the interquartile range of position error, for various step sizes and different methods. For implementing the method of Chambers¹⁰ and our methods, we chose to throw away the imaginary part of the positions and the velocities after each step. This destroys the symplecticness of the methods (or rather reduces the order for which the methods are symplectic) but leads to good error behaviour¹¹.

The comparison indicates that, for this problem, method of Chambers¹⁰ shows fourth order and method **AC1** shows sixth order behaviour, even though they are formally third and fifth order, respectively. Analogous results were obtained by Chambers¹⁰ for similar problems. It is interesting to see that the optimized 4th order method **RKNb6**⁶ outperforms a method of higher order, **RKNbr1**, down to machine accuracy level.

To make a more comprehensive and meaningful test, we also developed integrators for the gravitational N -body problem, based on various methods. We simulated the evolution of a Plummer sphere with 400 equal mass particles (see ref. 18 for a procedure for constructing a Plummer sphere). We followed the same procedure to estimate the errors and calculated the interquartile range for various methods and stepsizes. The dependence of errors on step size is given in figure 2.

Since complex arithmetic requires more operations than real arithmetic, we also made a comparison of CPU times for various methods. It was not possible to calculate CPU

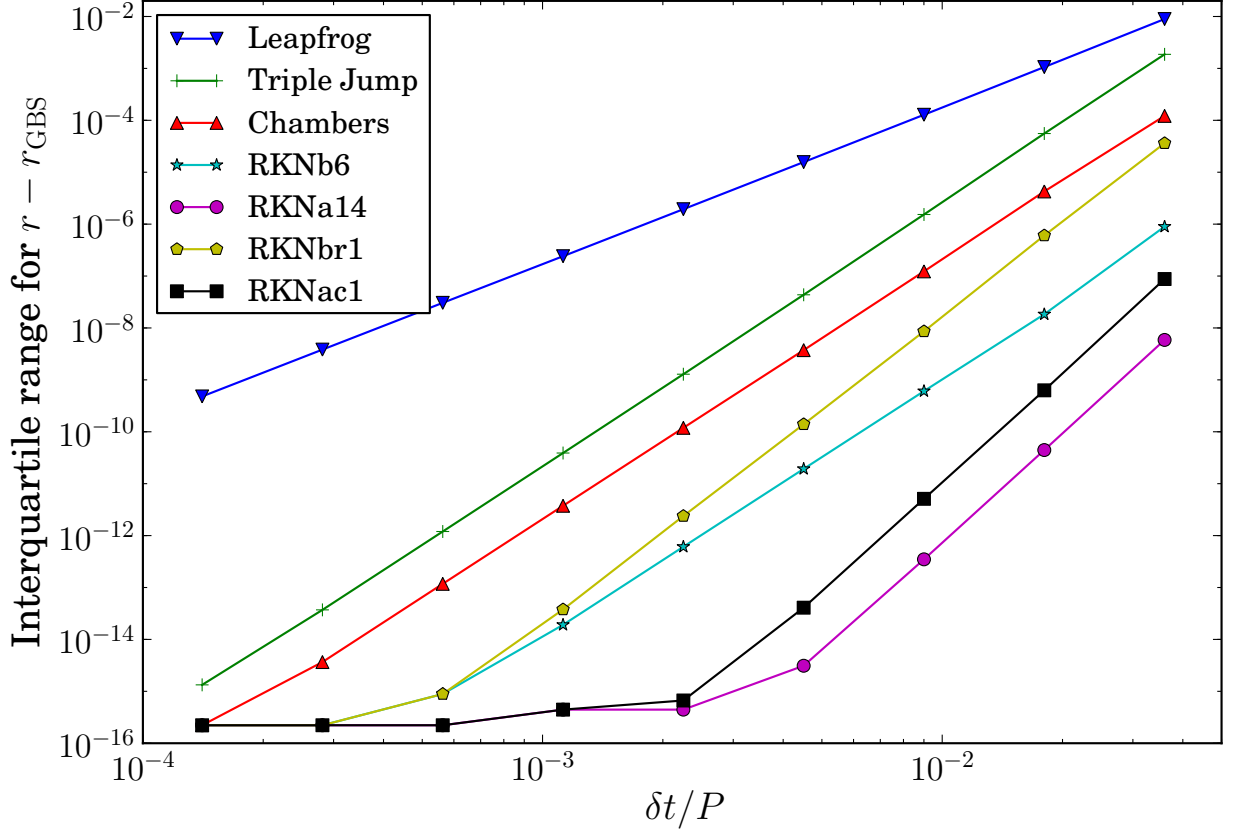


FIG. 1. Comparison of the behaviour of various methods with respect to changing step size, for gravitational two-body problem. x -axis the logarithm of the time step measured in orbital periods, y -axis is the logarithm of the interquartile range for position error. **Leapfrog** (down-triangles) is the standard second order Störmer-Verlet scheme, **Triple jump** (pluses) is the fourth order scheme of Yoshida¹², **Chambers** (up-triangles) is the complex third order scheme of Chambers¹⁰, **RKNb6** (stars) and **RKNa14** (circles) are optimized fourth and sixth order methods by Blanes & Moan^{6,17}, **RKNbr1** (pentagons) and **RKNac1** (squares) are two of the methods presented in this work.

times for two-body or 400-body problems accurately, since each step took too little time. Consequently, we setup a system of 10000 particles and integrated over a few steps. While calculating the CPU time, we subtracted off the time spent for the last substep for an integration step, since all the schemes we consider have the so called first-same-as-last property. For example, in **RKNA** schemes (equation 13) $e^{\alpha_1 \tau \mathbb{T}}$ substep can be combined with $e^{\alpha_6 \tau \mathbb{T}}$ substep of the next step. We present the CPU times per step for various methods in table IV.

The data here show that the integration time is proportional to the number of stages and

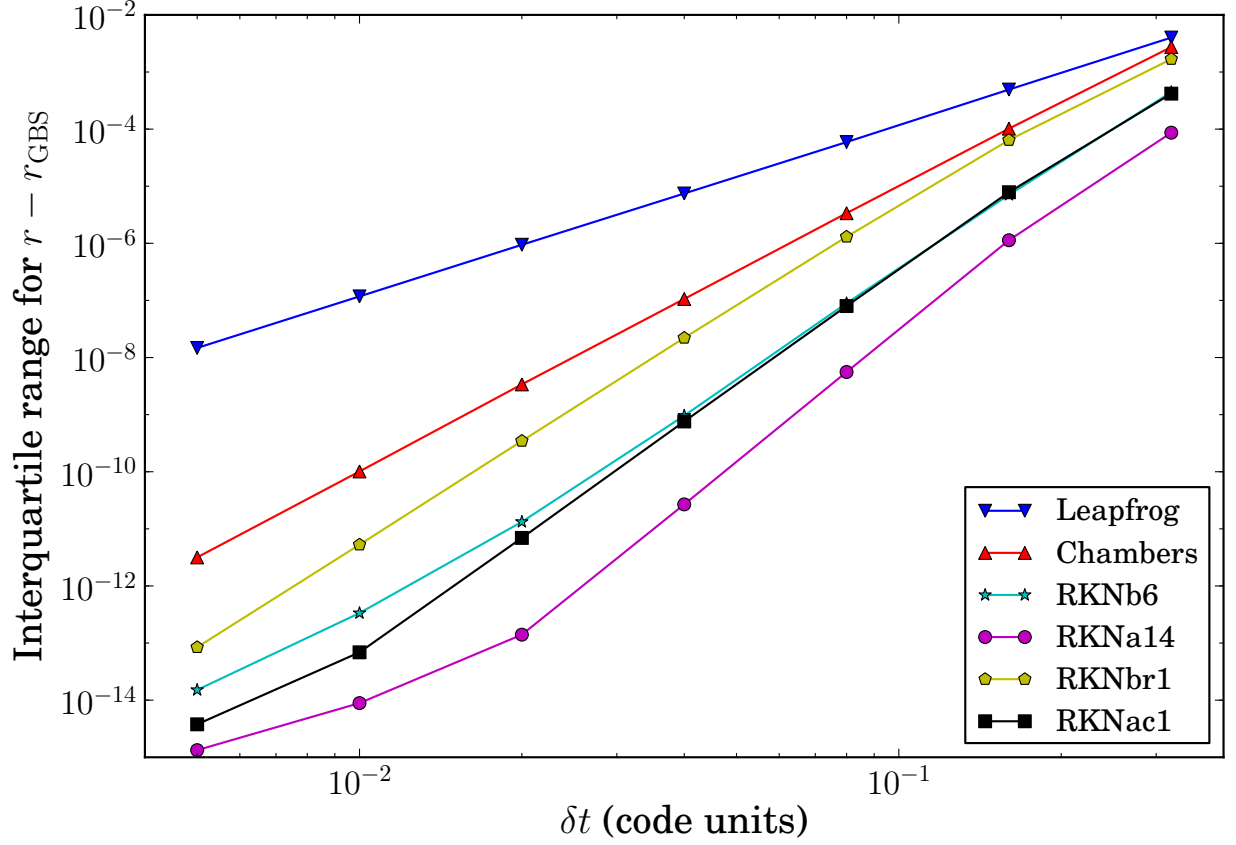


FIG. 2. Comparison of the behaviour of various methods with respect to changing step size, for gravitational 400-body problem. x -axis the logarithm of the time step in code units, y -axis is the logarithm of the interquartile range for position error. The symbols correspond to the same methods as figure 1, except **Triple Jump** is omitted, since it nearly coincides with **Chambers** for this problem.

TABLE IV. CPU times spent at each step for various methods. Values are normalized to Leapfrog method's CPU time.

Leapfrog	1
Chambers	12
Triple Jump	3
RKNb6	6
RKNa14	14
RKNbr1	5
RKNac1	28

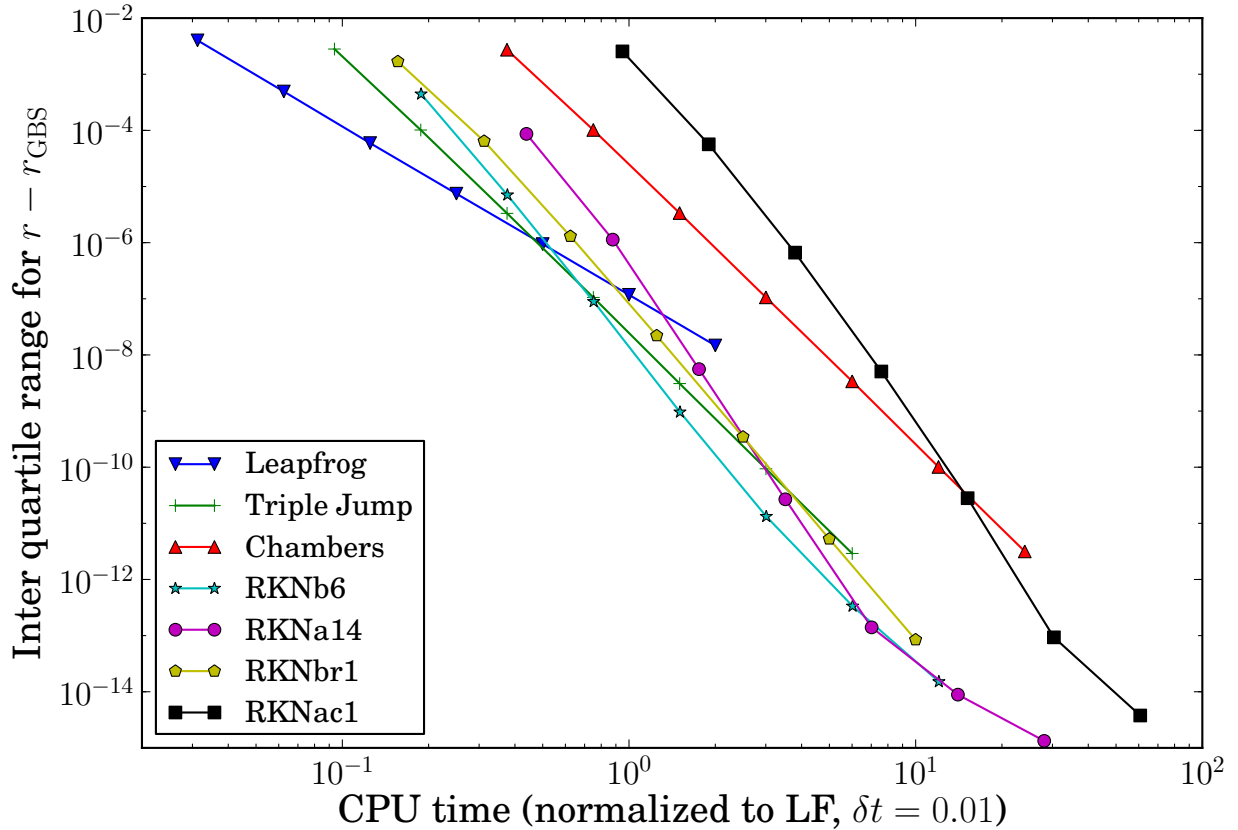


FIG. 3. CPU time spent vs. error for various methods applied to gravitational 400-body problem. x -axis the logarithm of the CPU time, normalized to time spent for Leapfrog method at time step $\delta t = 0.01$, y -axis is the logarithm of the interquartile range for position error. The symbols correspond to the same methods as figure 1.

using complex arithmetic increases the computation cost by about a factor of 6. This last factor surely depends on the problem, the implementation and the compiler. We present the error vs. CPU time, based on these findings in figure 3.

VI. DISCUSSION

In this work we constructed fifth order RKN methods with complex coefficients. Some of the methods we found have much smaller timesteps than the previously known methods, leading to smaller leading order errors. In addition, many of them have positive real coefficients, making them suitable for problems where negative real timesteps are not acceptable^{9,19,20}. Similar methods, satisfying this requirement, were also developed by Bandyuk et al.²¹. Other high order methods with complex coefficients with small positive

real parts were developed and analyzed by Hansen & Ostermann²² and Castella et al.²³; however these authors did not specifically study the RKN case, which allows considerable simplification for high order schemes.

For problems where complex arithmetic and/or positive real parts of the coefficients is a necessity, we expect these methods to be already competitive with lower order methods. However, the results of Blanes & Moan⁶ and Sofroniou & G. Spaletta²⁴ suggest that there is room for improvement by increasing the number of stages. We consider finding fully optimized methods with more stages beyond the scope of this paper; such an effort would require developing new software and would need spending considerable computational power. However, the methods found here can be improved by readily available tools. The idea is to turn a 5-stage method into a 6-stage one by adding an additional stage.

$$e^{\alpha_6 \tau \mathbb{T}} e^{\beta_5 \tau \mathbb{V}} e^{\alpha_5 \tau \mathbb{T}} \dots e^{\beta_1 \tau \mathbb{V}} e^{\alpha_1 \tau \mathbb{T}} \rightarrow e^{\beta'_7 \tau \mathbb{V}} e^{\alpha_6 \tau \mathbb{T}} e^{\beta'_6 \tau \mathbb{V}} \dots e^{\alpha_1 \tau \mathbb{T}} e^{\beta'_1 \tau \mathbb{V}}, \quad (17)$$

and set $\beta'_7 = \beta'_1 = 0$. We can then start in the vicinity of this solution and use MAPLE's minimization routines. Starting from solution AC1 in table II, we obtained the following skew-symmetric method:

$$\begin{aligned} \alpha_1 &= .101907705405177865 + i.130701756906677735, \\ \alpha_2 &= .218628781976265590 + i.0126440811480678494, \\ \alpha_3 &= .179463512618556560 - i.148112326926992222, \\ \beta'_1 &= .0489489561074426954 + i.0669384556781967844, \\ \beta'_2 &= .166479171860817010 + i.0764027877516731402, \\ \beta'_3 &= .192297943665939275 - i.0835834606213808479, \\ \beta'_4 &= .184547856731601789. \end{aligned} \quad (18)$$

This method has an error $(\sum_i |X_i|^2)^{1/2} = 2.6 \times 10^{-7}$, about two orders of magnitude smaller than the other methods (see table III).

Since we minimized the (imaginary) sixth order error terms, this optimization does not benefit the solutions of gravitational N -body problem. A possible venue for exploration is to increase the number of stages and minimize the (real part of) seventh order errors by using the extra variables. For this investigation, using a Lyndon basis would be more preferable, since the BCH expansion has much fewer terms in this basis¹⁴ and the consequences of

the simplification $[\mathbb{V}, [\mathbb{V}, [\mathbb{T}, \mathbb{V}]]] = 0$ is more straightforward. Here, we used a Philip Hall basis, since this allowed us to check our algebra by the “Lie Tools Package”²⁵ of Miguel Torres-Torriti.

The source code of the MAPLE, Python and C programs used in this work are freely available online: https://github.com/atakan/Complex_Coeff_RKN

ACKNOWLEDGMENTS

I thank Yuri Levin, Sergio Blanes, Ander Murua, Fernando Casas, Tevhide Altekin and Tolga Güver for various discussions and an anonymous referee for suggestions that improved this paper. This work has been supported by a Veni Fellowship from Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) and by the Turkish Scientific Research Council (TÜBİTAK) under Doktora Sonrası Geri Dönüş Programı (2232)

REFERENCES

- ¹S. Blanes, F. Casas, and A. Murua, “Splitting and composition methods in the numerical integration of differential equations,” *Bol. Soc. Esp. Mat. Apl.* **45**, 87–143 (2008).
- ²H. F. Levison and M. J. Duncan, “Symplectically Integrating Close Encounters with the Sun,” *Astron. J.* **120**, 2117–2123 (Oct. 2000).
- ³E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration* (Springer-Verlag, Berlin, 2006).
- ⁴E. Hairer, S. P. Nørsett, and G. Wanner, *Solving Ordinary Differential Equations: vol I* (Springer-Verlag, Berlin, 1993).
- ⁵Daniel I. Okunbor and Robert D. Skeel, “Explicit canonical methods for Hamiltonian systems,” *Math. Comp.*, 439–455(1992).
- ⁶S. Blanes and P. C. Moan, “Practical symplectic partitioned Runge–Kutta and Runge–Kutta–Nyström methods,” *J. Comput. Appl. Math.* **142**, 313–330 (2002), ISSN 0377-0427.
- ⁷Wen-Jie Zhu and Meng-Zhao Qin, “Order conditions of two kinds of canonical difference schemes,” *Comput. Math. Appl.* **25**, 61 – 74 (1993), ISSN 0898-1221, <http://www.sciencedirect.com/science/article/B6TYJ-46NX513-S8/2/0e47764b449efd0000b930a177fcc356>.

- ⁸Daniel I. Okunbor and Robert D. Skeel, “Canonical runge-kutta-nyström methods of orders five and six,” J. Comput. Appl. Math. **51**, 375 – 382 (1994), ISSN 0377-0427, <http://www.sciencedirect.com/science/article/B6TYH-45DB1WG-9/2/1f2b6aa752af8dbd4d126e60f3e3a9e6>.
- ⁹Siu A. Chin, “Symplectic integrators from composite operator factorizations,” Phys. Lett. A **226**, 344 – 348 (1997), ISSN 0375-9601, <http://www.sciencedirect.com/science/article/B6TVM-3SPD5CS-2W/2/976279ad9a7d1a4398e77b1ff79bcd58>.
- ¹⁰J. E. Chambers, “Symplectic Integrators with Complex Time Steps,” Astronom. J. **126**, 1119–1126 (Aug. 2003).
- ¹¹S. Blanes, F. Casas, and A. Murua, “Splitting methods with complex coefficients,” Bol. Soc. Esp. Mat. Apl., 47–61(2010), arXiv:1001.1549, <http://arxiv.org/abs/1001.1549>.
- ¹²H. Yoshida, “Recent Progress in the Theory and Application of Symplectic Integrators,” Celestial Mech. Dynam. Astronom. **56**, 27–43 (Mar. 1993).
- ¹³Sergio Blanes and Fernando Casas, “On the convergence and optimization of the Baker-Campbell-Hausdorff formula,” Linear Algebra and Appl. **378**, 135 – 158 (2004), ISSN 0024-3795.
- ¹⁴F. Casas and A. Murua, “An efficient algorithm for computing the Baker-Campbell-Hausdorff series and some of its applications,” J. Math. Phys. **50**, 033513 (Mar. 2009), arXiv:0810.2656 [math-ph].
- ¹⁵<http://www.python.org>.
- ¹⁶<http://code.google.com/p/sympy>.
- ¹⁷S. Blanes and P. C. Moan, “Splitting methods for non-autonomous differential equations,” J. Comput. Phys. **170**, 205–230 (2001).
- ¹⁸S. J. Aarseth, M. Henon, and R. Wielen, “A comparison of numerical methods for the study of star cluster dynamics,” Astronom. and Astrophys. **37**, 183–187 (Dec. 1974).
- ¹⁹Siu A. Chin, “Quadratic diffusion monte carlo algorithms for solving atomic many-body problems,” Phys. Rev. A **42**, 6991–7005 (Dec 1990).
- ²⁰Andr D. Bandrauk and Hai Shen, “Improved exponential split operator method for solving the time-dependent schrödinger equation,” Chem. Phys. Lett. **176**, 428 – 432 (1991), ISSN 0009-2614, <http://www.sciencedirect.com/science/article/B6TFN-44F1R9Y-PD/2/58055d1a9c471906acc9b47a237ac7d1>.
- ²¹Andr D. Bandrauk, Effat Dehghanian, and Huizhong Lu, “Complex integration steps in

decomposition of quantum exponential evolution operators,” *Chem. Phys. Lett.* **419**, 346 – 350 (2006), ISSN 0009-2614.

²²Eskil Hansen and Alexander Ostermann, “High order splitting methods for analytic semigroups exist,” *BIT* **49**, 527–542 (2009), ISSN 0006-3835, <http://dx.doi.org/10.1007/s10543-009-0236-x>.

²³F. Castella, P. Chartier, S. Descombes, and G. Vilmart, “Splitting methods with complex times for parabolic equations,” *BIT* **49**, 487–508 (2009), ISSN 0006-3835, <http://dx.doi.org/10.1007/s10543-009-0235-y>.

²⁴M. Sofroniou and G. Spaletta, “Derivation of Symmetric Composition Constants for Symmetric Integrators,” *Optim. Methods Softw.* **20**, 597 (2005).

²⁵<http://www.cim.mcgill.ca/~migueltt/ltp/ltp.html>.