Applied Numerical Mathematics 151 (2020) 13–26

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Efficient implementation of the ARKN and ERKN integrators for multi-frequency oscillatory systems with multiple time scales [☆]

Wei Shi [a], Xinyuan Wu [b,c], Kai Liu [d,*]

[a] School of Physical and Mathematical Sciences, Nanjing Tech University, Nanjing 211816, PR China
[b] Department of Mathematics, Nanjing University, State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing 210093, PR China
[c] School of Mathematical Sciences, Qufu Normal University, Qufu 273165, Shandong, PR China
[d] College of Applied Mathematics, Nanjing University of Finance & Economics, Nanjing 210023, PR China

ARTICLE INFO

Article history:
Received 24 July 2019
Received in revised form 21 October 2019
Accepted 14 December 2019
Available online 19 December 2019

Keywords:
Numerical solutions of ODEs
Nonlinear oscillators
Multiple time scales
ARKN integrators
ERKN integrators
Matrix-valued functions
Stability analysis

ABSTRACT

It is known that, a notable feature of both the multi-frequency and multidimensional ARKN (Runge-Kutta-Nyström methods adapted to oscillatory system) and ERKN (extended Runge-Kutta-Nyström) integrators when they are applied to multi-frequency and multi-dimensional oscillatory system $q'' + Mq = f(t, q)$ with multiple time scales is that they exactly integrate the multi-frequency oscillatory homogeneous system $q'' + Mq = \mathbf{0}$. With regard to the efficient implementation issues of the integrators, it is significant to calculate efficiently the matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$ which are involved in the two kinds of integrators, where $V = h^2M$ and h is a stepsize. In this paper, we pay attention to efficient implementation issues of the multi-frequency and multidimensional ARKN and ERKN integrators which are closely related to the calculations of $\phi_0(V)$ and $\phi_1(V)$. Using the properties of $\phi_0(V)$ and $\phi_1(V)$ and their relations, we present an efficient algorithm to calculate the two matrix-valued functions at lower cost. Two illuminating numerical examples are accompanied and the numerical results show the remarkable efficiency of the algorithm. We also give an essential stability analysis for ARKN and ERKN integrators on the basis of the different approximations to $\phi_0(V)$ and $\phi_1(V)$ which gains an insight into the importance of the calculations of $\phi_0(V)$ and $\phi_1(V)$.

1. Introduction

In this work, we focus our attention on the efficient implementation issues of numerical integrators for the multi-frequency and multidimensional oscillatory system of the form

[☆] The research was supported in part by the Natural Science Foundation of China under Grant 11701271 and 11671200, by the Specialized Research Foundation for the Doctoral Program of Higher Education under Grant 20130091110041, by the 985 Project at Nanjing University under Grant 9112020301, by A Project Founded by the Priority Academic Program Development of Jiangsu Higher Education Institutions, by the Natural Science Foundation of Jiangsu Province under Grant BK20150934, and by the Natural Science Foundation of the Jiangsu Higher Education Institutions under Grant 16KJB110010.

[*] Corresponding author.
E-mail addresses: shuier628@163.com (W. Shi), xywu@nju.edu.cn (X. Wu), laukai520@163.com (K. Liu).

https://doi.org/10.1016/j.apnum.2019.12.014
0168-9274/© 2019 IMACS. Published by Elsevier B.V. All rights reserved.

$$\begin{cases} q''(t)+Mq(t)=f\big(t,q(t)\big), t\in[t_0,T],\\ q\big(t_0\big)=q_0, q'(t_0)=q_0', \end{cases} \tag{1}$$

where $M\in R^{m\times m}$ is a positive semi-definite matrix that implicitly preserves the main frequencies of the problem and $q:R\to R^m$, $f:R\times R^m\to R^m$, $q_0\in R^m, q_0'\in R^m$. This kind of problem usually arises in various fields of science and technology, such as applied mathematics, mechanics, physics, astronomy, molecular biology and engineering. A typical example can be obtained from the semidiscretization of wave equations based on the method of lines. Accordingly, there has been an enormous advance in dealing with the multi-frequency oscillatory system (1) and some useful approaches to constructing Runge-Kutta-Nyström (RKN)-type integrators have been proposed. We refer the reader to [3,5–7,9,14,15,17] for example. Recently, X. Wu et al. [15] formulated the general multi-frequency and multidimensional ARKN methods (Runge-Kutta-Nyström (RKN) methods adapted to system (1)) and derived the corresponding order conditions based on the B-series theory. Some novel multi-frequency and multidimensional ARKN methods were proposed in [14]. An advantage of ARKN methods is that their updates take into account the special oscillatory structures of the system (1) brought by the linear term Mq, therefore, they can naturally integrate the multi-frequency oscillatory homogeneous system $q''(t)+Mq(t)=\mathbf{0}$ exactly. Furthermore, X. Wu et al. [16] proposed a standard form of the multi-frequency and multidimensional ERKN methods (extended RKN methods) for the multi-frequency oscillatory system (1) and derived the corresponding order conditions based on the extended Nyström tree theory [16,18]. Compared with ARKN methods, the ERKN methods not only revise the updates, but also the internal stages of classical RKN methods. Moreover, the global error analysis for ERKN methods is made in a very recent paper by Wang et al. [13].

For both the multi-frequency and multidimensional ARKN and ERKN integrators, authors have defined the following matrix-valued functions

$$\phi_l(K):=\sum_{k=0}^{\infty}\frac{(-1)^kK^k}{(2k+l)!},\ \ l=0,1,\ldots, \tag{2}$$

which are used in the two kinds of numerical integrators. It should be noted that the series expansion of $\phi_0(V)$ and $\phi_1(V)$ is very similar with those of the cosine and sine functions. We will explain the relationship between functions ϕ_0,ϕ_1 and the cosine and sine functions later. Since the series

$$\sum_{k=0}^{\infty}\frac{x^k}{(2k+l)!}$$

has the radius of convergence $r=+\infty$, it is clear that

$$\|\phi_l(K)\|\leq\sum_{k=0}^{\infty}\frac{\|K\|^k}{(2k+l)!}.$$

To apply the integrators to the multi-frequency and multidimensional oscillatory system (1), the efficient evaluations of these matrix-valued functions are required. Although they may be replaced by truncating series expansions of the matrix-valued functions (see [2]), this will result in the shortcoming of lower accuracy and efficiency. What's more, it is natural to raise a question, that is, how to choose the degrees of the truncated series expansions. Clearly, in order to achieve and show the high efficiency of ARKN and ERKN methods completely, we have to approximate these matrix-valued functions as accurate and efficient as possible. On the other hand, the functions $\phi_l(K)$ $(l=0,1,\cdots)$ have the following recurrence relations

$$\phi_{j+2}(K)=K^{-1}\left(\frac{I}{j!}-\phi_j(K)\right),\ j=0,1,\cdots$$

if K is invertible. Therefore, the efficient approximations of the matrix-valued functions $\phi_0(K)$ and $\phi_1(K)$ are the most important. This motivates the further analysis of the implementation issues for both the multi-frequency and multidimensional ARKN and the ERKN integrators when they are applied to the multi-frequency oscillatory equations (1) in the present paper.

In the special and important case where K is a symmetric and positive semi-definite matrix, we have the decomposition of K as follows

$$K=P^TW^2P=\Omega_0^2\ \text{with}\ \Omega_0=P^TWP,$$

where P is an orthogonal matrix and W is a diagonal matrix with nonnegative diagonal entries which are the square roots of the eigenvalues of K. We then have

$$\phi_0(K):=\sum_{k=0}^{\infty}\frac{(-1)^kK^k}{(2k)!}=\cos(\Omega_0)=P^T\cos(W)P$$

and

$$\begin{aligned}\phi_1(K) :=& \sum_{k=0}^{\infty} \frac{(-1)^k K^k}{(2k+1)!} = \Omega_0^{-1} \sin(\Omega_0) = (P^T W P)^{-1} \sin(P^T W P)\\ =& P^T W^{-1} P P^T \sin(W) P = P^T W^{-1} \sin(W) P.\end{aligned}$$

An important fact is that, the diagonalization is not always feasible, especially in high-dimensional problems. Moreover, extra errors may be produced if the matrix-decomposition is used.

We note that, for this kind of oscillatory second-order initial value problem (1), M is not necessarily symmetric nor diagonal. Consequently, the computations of the matrix-valued functions $\phi_l(V)$ are required to be independent of matrix decompositions, where $V = h^2 M$ (here h is a stepsize). However, it turns out that they are closely related with the matrix-valued *sine* and *cosine* functions, whose numerical computations have received relative little attention so far.

The remainder of this paper is organized as follows. In Section 2, we recall the basic ideas of ARKN and ERKN methods for multi-frequency and multidimensional second-order oscillatory problems and give two illuminating examples including an oscillatory Hamiltonian system. An efficient algorithm on approximating functions $\phi_0(V)$ and $\phi_1(V)$ without using matrix decompositions is proposed in Section 3. In Section 4, we propose a strategy to choose the degrees of the approximations in the algorithm given in Section 3, so that the computational cost is minimized subject to achieving a desired truncation error. Numerical experiments are carried out and the numerical results demonstrate the remarkable efficiency of the algorithm with this strategy. In Section 5, we commence the important analysis of stability properties for both the ARKN and ERKN methods based on the different approximations to $\phi_0(V)$ and $\phi_1(V)$. The last section is devoted to the conclusive comments.

2. ARKN and ERKN integrators for multi-frequency and multidimensional oscillatory second-order differential systems

In this section, we first introduce the basic definitions of multi-frequency and multidimensional ARKN and ERKN methods and then, by applying them to solving two wave equations, we demonstrate the importance of calculating the matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$ which are involved in the two kinds of integrators for the multi-frequency oscillatory systems (1) with multiple time scales.

2.1. Basic definitions of multi-frequency and multidimensional ARKN and ERKN methods

By the definition of the matrix-valued functions $\phi_l(V)$, first, taking advantage of the special structure of (1) brought by the linear term Mq and revising the updates of classical RKN methods obtains the following s-stage multidimensional ARKN methods [15] for the multi-frequency oscillatory system (1).

Definition 2.1. An s-stage multi-frequency and multidimensional ARKN integrator for the oscillatory system (1) is defined to be

$$\begin{cases} Q_i = q_n + c_i h q'_n + h^2 \sum\limits_{j=1}^{s} \bar{a}_{ij}\big(f(t_n + c_j h, Q_j) - MQ_j\big), \qquad i = 1, \dots, s,\\ q_{n+1} = \phi_0(V) q_n + h\phi_1(V) q'_n + h^2 \sum\limits_{i=1}^{s} \bar{b}_i(V) f(t_n + c_i h, Q_i),\\ q'_{n+1} = -hM\phi_1(V) q_n + \phi_0(V) q'_n + h \sum\limits_{i=1}^{s} b_i(V) f(t_n + c_i h, Q_i), \end{cases} \tag{3}$$

where h is the time stepsize, $\bar{a}_{ij} \in \mathbb{R}, i, j = 1, \cdots, s$, $b_i(V)$ and $\bar{b}_i(V)$, $i = 1, \dots, s$ are matrix-valued functions of $V = h^2 M$.

Then, taking advantage of the special structure brought by Mq of (1) and revising not only the updates but also the internal stages of classical RKN methods leads to the so-called multi-frequency and multidimensional ERKN integrators [16].

Definition 2.2. An s-stage multi-frequency and multidimensional ERKN integrator for the oscillatory system (1) is defined to be

$$\begin{cases} Q_i = \phi_0(c_i^2 V) q_n + c_i \phi_1(c_i^2 V) h q'_n + h^2 \sum\limits_{j=1}^{s} \bar{a}_{ij}(V) f(t_n + c_j h, Q_j), \ i = 1, \dots, s,\\ q_{n+1} = \phi_0(V) q_n + h\phi_1(V) q'_n + h^2 \sum\limits_{i=1}^{s} \bar{b}_i(V) f(t_n + c_i h, Q_i),\\ q'_{n+1} = -hM\phi_1(V) q_n + \phi_0(V) q'_n + h \sum\limits_{i=1}^{s} b_i(V) f(t_n + c_i h, Q_i), \end{cases} \tag{4}$$

where $b_i(V)$, $\bar{b}_i(V)$, $i = 1, \dots, s$, and $\bar{a}_{ij}(V)$, $i, j = 1, \dots, s$ are matrix-valued functions of $V = h^2 M$.

2.2. Illuminating experiments

In this subsection, by solving two nonlinear wave equations, we briefly show the importance of computing the matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$ which are involved in the multi-frequency and multidimensional ARKN and ERKN methods.

For the implementation of an ARKN or ERKN method of order p, in this section, following Franco's suggestions in [2], we use polynomials $P_0(V)$ and $P_1(V)$ (the truncated series expansions of $\phi_0(V)$ and $\phi_1(V)$) in place of the functions $\phi_0(V)$ and $\phi_1(V)$, respectively, where $P_0(V)$ and $P_1(V)$ satisfy

$$\phi_0(V) = P_0(V) + \mathcal{O}(h^{r+1}), \quad \phi_1(V) = P_1(V) + \mathcal{O}(h^r) \tag{5}$$

with $V = h^2M$ and $r \geq p$. With these truncations, the resulting ARKN scheme or ERKN scheme also has algebraic order p and each is of only order r for the multi-frequency and multidimensional oscillatory homogeneous system $q''(t) + Mq(t) = \mathbf{0}$. Based on this fact, the numerical methods which we choose for comparisons are as follows:

- ARKN3s4-8: the three-stage explicit ARKN method of order 4 given in [14] with $\phi_0(V)$ and $\phi_1(V)$ approximated by $P_0(V)$ and $P_1(V)$, respectively, where $r = 8$ in (5).
- ARKN3s4-12: the three-stage explicit ARKN method of order 4 given in [14] with $\phi_0(V)$ and $\phi_1(V)$ approximated by $P_0(V)$ and $P_1(V)$, respectively, where $r = 12$ in (5).
- ERKN3s3-8: the three-stage explicit ERKN method of order 3 given in [16] (W2ERKN3s3) with $\phi_0(V)$ and $\phi_1(V)$ approximated by $P_0(V)$ and $P_1(V)$, respectively, where $r = 8$ in (5).
- ERKN3s3-12: the three-stage explicit ERKN method of order 3 given in [16] (W2ERKN3s3) with $\phi_0(V)$ and $\phi_1(V)$ approximated by $P_0(V)$ and $P_1(V)$, respectively, where $r = 12$ in (5).
- SRKN3s4: the symplectic three-stage RKN scheme of order 4 given in [7].

Problem 1. Consider the nonlinear wave equation [2]

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - a(x)\dfrac{\partial^2 u}{\partial x^2} + 92u = f(t,x,u), \quad 0 < x < 1, t > 0. \\ u(0,t) = 0, \quad u(1,t) = 0, \quad u(x,0) = a(x), \quad u_t(x,0) = 0 \end{cases}$$

with

$$f(t,x,u) = u^5 - a^2(x)u^3 + \frac{a^5(x)}{4}\sin^2(20t)\cos(10t), \quad a(x) = 4x(1-x).$$

The exact solution is given by

$$u(x,t) = a(x)\cos(10t)$$

and this oscillatory problem represents a vibrating string with angular speed 10. We carry out a semi-discretization on the spatial variable by using second-order centered differences, which is exactly an approximation of the spatial part and then the semi-discrete second-order ordinary differential equations in time is given by

$$\begin{cases} \dfrac{\mathrm{d}^2 u_i}{\mathrm{d}t^2} - a(x_i)\dfrac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + 92u_i = f(t, x_i, u_i), \quad 0 < t \leq t_{\text{end}}, \\ u_i(0) = a(x_i), \quad u_i'(0) = 0, \quad i = 1, \ldots, L-1, \\ u_0(0) = u_L(0) = 0, \end{cases}$$

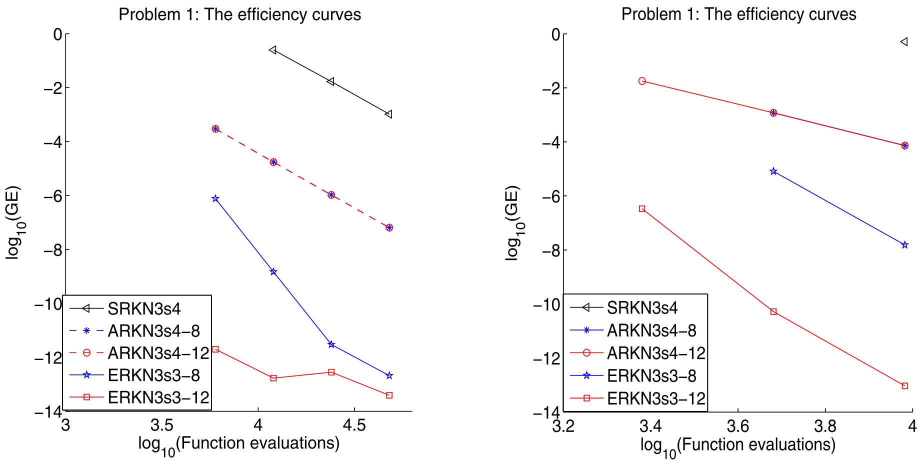
where $\Delta x = 1/L$ is the spatial mesh step and $x_i = i\Delta x$. This system has the form

$$\begin{cases} \dfrac{d^2U}{dt^2} + MU = F(t,U), \quad 0 < t \leq t_{end}. \\ U(0) = (a(x_1), \ldots, a(x_{L-1}))^T, \quad U'(0) = \mathbf{0}, \end{cases} \tag{6}$$

where $U(t) = (u_1(t), \ldots, u_{L-1}(t))^T$ with $u_i(t) = u(x_i,t), \ i = 1, \ldots, L-1$,

$$M = \frac{1}{\Delta x^2}\begin{pmatrix} 2a(x_1) & -a(x_1) & & & \\ -a(x_2) & 2a(x_2) & -a(x_2) & & \\ & \ddots & \ddots & \ddots & \\ & & -a(x_{L-2}) & 2a(x_{L-2}) & -a(x_{L-2}) \\ & & & -a(x_{L-1}) & 2a(x_{L-1}) \end{pmatrix} + 92I_{L-1},$$

and



(i) (ii)

Fig. 1. Results for the problem 1 based on the truncated series expansions of $\phi_0(V)$ and $\phi_1(V)$. The logarithm of the maximum global error (GE) over the integration interval against the number of function evaluations. (i): Stepsizes are $h = 1/(5 \times 2^j)$, $j = 2, 3, 4, 5$. (ii): Stepsizes are $h = 1/(2^j)$, $j = 2, 3, 4, 5$.

$$F(t, U) = \big(f(t, x_1, u_1), \ldots, f(t, x_{L-1}, u_{L-1})\big)^T.$$

It is noted that M is not a symmetric matrix and the system (6) is not a Hamiltonian system. However, (6) is a multi-frequency and multidimensional oscillatory system. The system is integrated on the interval $t \in [0, 100]$ with $L = 20$ and the numerical results are shown in Fig. 1.

We note that the orders of both the ARKN methods and the ERKN methods used here are no more than 4, therefore, theoretically, the choice of $r = 8$ and $r = 12$ in the experiments for both the ARKN and the ERKN methods are sufficient and thus, the numerical behaviour of both ARKN and ERKN methods with $r = 8$ and $r = 12$ should be similar to each other. Fig. 1 presents the errors of the different methods at $t_{end} = 100$ versus the computational effort.

If $f(q)$ in (1) is the negative gradient of a real-valued function and M is a symmetric and positive semi-definite matrix, then (1) is in fact identical to an oscillatory Hamiltonian system. This type of system exhibits a pronounced oscillatory or highly-oscillatory behavior and has received much attention in the last few years to explore more appropriate mathematical methodology and numerical approach. Therefore, we present an oscillatory Hamiltonian system as the second example.

Problem 2. Consider the nonlinear Klein-Gordon equation [10]

$$\begin{cases} \dfrac{\partial^2 u}{\partial t^2} - \dfrac{\partial^2 u}{\partial x^2} + u + u^3 = 0, \quad 0 < x < L, t > 0. \\ u(x, 0) = A\left(1 + \cos(\frac{2\pi}{L}x)\right), \\ u_t(x, 0) = 0, \\ u(0, t) = u(L, t) \end{cases}$$

with $L = 1.28, \ \ A = 0.4$.

By the same semi-discretization on the spatial variable as Problem 1, we obtain the corresponding semi-discrete second-order ordinary differential equations in time

$$\begin{cases} \dfrac{\mathrm{d}^2 u_i}{\mathrm{d}t^2} - \dfrac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} + u_i + u_i^3 = 0, \quad 0 < t \le t_{\mathrm{end}}, \\ u_i(0) = 0.4\left(1 + \cos(\frac{2\pi i}{d})\right), \quad u_i'(0) = 0, \quad i = 1, \ldots, d, \\ u_0(0) = u_d(0), \end{cases}$$

where $\Delta x = L/d$ is the spatial mesh step and $x_i = i\Delta x$. This system has the form

$$\begin{cases} \dfrac{d^2 U}{dt^2} + MU = F(U), \quad 0 < t \le t_{end}. \\ U(0) = \left(0.4\left(1 + \cos(\frac{2\pi i}{d})\right)\right)_{i=1}^{d}, \quad U'(0) = \mathbf{0}, \end{cases} \tag{7}$$

where $U(t) = (u_1(t), \ldots, u_d(t))^T$ with $u_i(t) = u(x_i, t), \ i = 1, \ldots, d,$

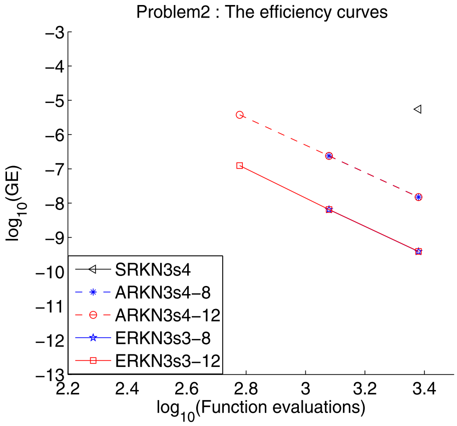


Fig. 2. Results for Problem 2 based on the truncated series expansions of $\phi_0(V)$ and $\phi_1(V)$. The logarithm of the maximum global error (GE) over the integration interval against the number of function evaluations. Stepsizes are $h = 3/(15 \times 2^j)$, where $j = 1, 2, 3, 4$.

$$M = \frac{1}{\Delta x^2}\begin{pmatrix} 2 & -1 & & & -1 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{pmatrix} + I_d,$$

and $F(U) = \left(-u_1^3(t), \dots, -u_d^3(t)\right)^T$.

For Problem 2, M is a symmetric matrix and the system (7) is a Hamiltonian system with

$$F(U) = -\nabla \frac{1}{4}\left(u_1^4(t) + u_2^4(t) + \cdots + u_d^4(t)\right).$$

We integrate the problem on the interval $[0, 10]$ with $d = 64$ and the results are shown in Fig. 2.

It can be observed from the results of the numerical experiments that the computational accuracy of functions $\phi_0(V)$ and $\phi_1(V)$ makes a notable impact on the efficiency of the numerical methods. Especially, Fig. 1 (ii) for Problem 1 shows that for a larger stepsize $h = 1/8$, both the methods ARKN3s4-12 and ERKN3s3-12 work well, however, the errors of both the methods ARKN3s4-8 and ERKN3s3-8 are too large to plot. This phenomenon also occurred in Fig. 2 for Problem 2. This fact transpires that the effect of implementation for an ARKN or ERKN integrator has some relation to the accuracy of evaluations of $\phi_0(V)$ and $\phi_1(V)$. Therefore, it is the key to designing an efficient algorithm for computing efficiently the matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$. Meanwhile, the stability analysis of the schemes based on the approximations to $\phi_0(V)$ and $\phi_1(V)$ is also significant and we will discuss this key point in detail in Section 5.

3. Efficient computation of matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$

In this section, we begin with some properties of the matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$ and their relations. We then present an algorithm for simultaneously computing $\phi_0(V)$ and $\phi_1(V)$ at lower cost. Throughout this paper, the vector norm and the matrix norm are denoted by $\|\cdot\|$ and we use the ∞-norm.

Proposition 3.1. *For a given matrix V, the matrix-valued functions*

$$\phi_0(V) := \sum_{k=0}^{\infty} \frac{(-1)^k V^k}{(2k)!}$$

and

$$\phi_1(V) := \sum_{k=0}^{\infty} \frac{(-1)^k V^k}{(2k+1)!}$$

defined by (2) *with $l = 0, 1$ satisfy the following properties:*

(i)

$$\phi_0^2(V) + V\phi_1^2(V) = I, \tag{8}$$

(ii)

$$\phi_0(4V) = 2\phi_0^2(V) - I, \tag{9}$$

(iii)

$$\phi_1(4V) = \phi_0(V)\phi_1(V), \tag{10}$$

where I is the identity matrix sharing the dimension with V.

Proof. Computing $\phi_0^2(V)$ and $\phi_1^2(V)$, respectively, we obtain

$$\begin{aligned}\phi_0^2(V) &= \sum_{k=0}^{\infty}\sum_{j=0}^{\infty}\frac{(-1)^{k+j}V^{k+j}}{(2k)!(2j)!} = I + \sum_{m=1}^{\infty}\sum_{k+j=m}\frac{(-1)^m V^m}{(2k)!(2j)!} \\ &= I + \sum_{m=1}^{\infty}\sum_{k=0}^{m}\frac{(-1)^m V^m}{(2k)!(2m-2k)!} = I + \sum_{m=1}^{\infty}\sum_{k=0}^{m}\frac{(-1)^m V^m}{(2m)!}\binom{2m}{2k} \\ &= I + \sum_{m=1}^{\infty} 2^{2m-1}\frac{(-1)^m V^m}{(2m)!}\end{aligned} \tag{11}$$

and

$$\begin{aligned}\phi_1^2(V) &= \sum_{k=0}^{\infty}\sum_{j=0}^{\infty}\frac{(-1)^{k+j}V^{k+j}}{(2k+1)!(2j+1)!} = \sum_{m=1}^{\infty}\sum_{k+j+1=m}\frac{(-1)^{m-1}V^{m-1}}{(2k+1)!(2j+1)!} \\ &= \sum_{m=1}^{\infty}\sum_{k=0}^{m-1}\frac{(-1)^{m-1}V^{m-1}}{(2k+1)!(2m-2k-1)!} = \sum_{m=1}^{\infty}\sum_{k=0}^{m-1}\frac{(-1)^{m-1}V^{m-1}}{(2m)!}\binom{2m}{2k+1} \\ &= \sum_{m=1}^{\infty} 2^{2m-1}\frac{(-1)^{m-1}V^{m-1}}{(2m)!} = -\sum_{m=1}^{\infty} 2^{2m-1}\frac{(-1)^{m}V^{m-1}}{(2m)!}.\end{aligned}$$

Therefore, it is easy to see that *(i)* is true.

Then it follows from (11) that

$$\begin{aligned}2\phi_0^2(V) - I &= 2I + \sum_{m=1}^{\infty} 2^{2m}\frac{(-1)^m V^m}{(2m)!} - I = I + \sum_{m=1}^{\infty}\frac{(-1)^m (4V)^m}{(2m)!} \\ &= \sum_{m=0}^{\infty}\frac{(-1)^m (4V)^m}{(2m)!} = \phi_0(4V),\end{aligned}$$

which proves *(ii)*.

The proof of *(iii)* is given as follows:

$$\begin{aligned}\phi_0(V)\phi_1(V) &= \sum_{k=0}^{\infty}\sum_{j=0}^{\infty}\frac{(-1)^{k+j}V^{k+j}}{(2k)!(2j+1)!} = I + \sum_{m=1}^{\infty}\sum_{k+j=m}\frac{(-1)^m V^m}{(2k)!(2j+1)!} \\ &= I + \sum_{m=1}^{\infty}\sum_{k=0}^{m}\frac{(-1)^m V^m}{(2k)!(2m+1-2k)!} = I + \sum_{m=1}^{\infty}\sum_{k=0}^{m}\frac{(-1)^m V^m}{(2m)!}\binom{2m+1}{2k} \\ &= I + \sum_{m=1}^{\infty} 2^{2m}\frac{(-1)^m V^m}{(2m+1)!} = \sum_{m=0}^{\infty}\frac{(-1)^m (4V)^m}{(2m+1)!} = \phi_1(4V). \quad \square\end{aligned}$$

We are now in a position to present the efficient algorithm for computing the functions $\phi_0(V)$ and $\phi_1(V)$.

Following the strategy in the algorithm of Serbin and Blalock [11], we define the matrix $B = 4^{-N}V$, where the value N is chosen to make the norm of B small enough. Therefore, we can obtain good approximations to $\phi_0(B)$ and $\phi_1(B)$ via truncating the infinite series at the r-th and s-th term, respectively. Then, using the relations (9) and (10), we can compute $\phi_0(V) = \phi_0(4^N B)$ and $\phi_1(V) = \phi_1(4^N B)$. In what follows we present the analysis of our algorithm in detail. First of all we calculate:

$$\begin{aligned}
&\phi_0(B) \approx I - \frac{B}{2!} + \frac{B^2}{4!} - \frac{B^3}{6!} + \ldots + (-1)^r \frac{B^r}{(2r)!} \equiv C_0, \\
&\phi_0(4B) = 2\phi_0^2(B) - I = 2C_0^2 - I \equiv C_1, \\
&\phi_0(4^i B) = 2\phi_0^2(4^{i-1}B) - I = 2C_{i-1}^2 - I \equiv C_i, \\
&\phi_1(B) \approx I - \frac{B}{3!} + \frac{B^2}{5!} - \frac{B^3}{7!} + \ldots + (-1)^s \frac{B^s}{(2s+1)!} \equiv S_0, \\
&\phi_1(4B) = \phi_0(B)\phi_1(B) = C_0 S_0 \equiv S_1, \\
&\phi_1(4^i B) = \phi_0(4^{i-1}B)\phi_1(4^{i-1}B) = C_{i-1}S_{i-1} \equiv S_i, \\
&i.e., \\
&C_i = 2C_{i-1}^2 - I \quad \text{and} \quad S_i = C_{i-1}S_{i-1}.
\end{aligned}$$

By the N-th step, we can obtain $\phi_0(V) = \phi_0(4^N B) = C_N$ and $\phi_1(V) = \phi_1(4^N B) = S_N$. This algorithm now can be summarized by Algorithm 3.1 as follows.

Algorithm 3.1. *Given a matrix V and a parameter $\alpha > 0$, this algorithm approximates the matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$. Here, we use r and s represent the degrees of truncated polynomials in Taylor expansions of $\phi_0(B)$ and $\phi_1(B)$, respectively.*

1. *Choose N such that $B = 4^{-N}V$ satisfies $\|B\| < \alpha$.*
2. *Calculate C_0 with*

   $$C_0 = I - \frac{B}{2!} + \frac{B^2}{4!} - \frac{B^3}{6!} + \ldots + (-1)^r \frac{B^r}{(2r)!},$$

   the approximation to $\phi_0(B)$.
3. *Calculate S_0 with*

   $$S_0 = I - \frac{B}{3!} + \frac{B^2}{5!} - \frac{B^3}{7!} + \ldots + (-1)^s \frac{B^s}{(2s+1)!},$$

   the approximation to $\phi_1(B)$.
4. *for $i = 1, \ldots, N$ do steps 5 and 6.*
5. *Set $C_i = 2C_{i-1}^2 - I$.*
6. *Set $S_i = C_{i-1}S_{i-1}$.*
7. *Set $\phi_0(V) = C_N$, $\phi_1(V) = S_N$.*
8. *STOP.*

Remark 3.1. Since the matrix M in the oscillatory system (1) is not necessarily symmetric, nor is V, thus, there may not exist a square root of V. The Algorithm 3.1 is designed based on the relations of matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$, but not on the relations of matrix cosine and sine functions. Consequently, even if matrix M is symmetric, we never compute the square root of M in our Algorithm 3.1.

Remark 3.2. Assume that the initial approximation to the $\phi_0(B)$ from the truncation of Taylor expansion is $C_0 = \phi_0(B + \Delta B)$, where $B = 4^{-N}V$. Applying step 5 of Algorithm 3.1 N times gives

$$\begin{aligned}
&C_1 = 2C_0^2 - I = \phi_0(4B + 4\Delta B), \\
&C_2 = 2C_1^2 - I = \phi_0(4^2 B + 4^2 \Delta B), \\
&\qquad\qquad \vdots \\
&C_N = 2C_N^2 - I = \phi_0(4^N B + 4^N \Delta B).
\end{aligned} \tag{12}$$

Therefore we have $\phi_0(V) \approx C_N = \phi_0(V + \Delta V)$ with $\Delta V = 4^N \Delta B$ and

$$\frac{\|\Delta V\|}{\|V\|} = \frac{\|4^N \Delta B\|}{\|4^N B\|} = \frac{\|\Delta B\|}{\|B\|}.$$

It shows that there is no growth in the relative backward error during the step 5. Similar arguments can be performed for step 6. Hence by choosing the parameters r and s so that $\frac{\|\Delta B\|}{\|B\|} \leqslant u$, we achieve an overall backward error bounded by u, where u is the machine epsilon. The discussion here is an analogue of Lemma 2.1 in [1].

It should be pointed out that in the computations, the matrix-valued functions C_0 and S_0 are evaluated by using the Paterson-Stockmeyer method, which is a potentially greater reduction in cost over Horner's method (see [4], Sec. 11.2.4). The main idea behind the algorithm is to apply Horner's rule to a matrix polynomial $p(A) = a_0 I + a_1 A + \ldots + a_q A^q$ regarded as a polynomial in A^s where $s \approx \sqrt{q}$. For example, if $q = 9$, then

$$p(A) = A^3[A^3[a_9 A^3 + (a_8 A^2 + a_7 A + a_6 I)] + (a_5 A^2 + a_4 A + a_3 I)] + (a_2 A^2 + a_1 A + a_0 I).$$

Hence, the cost of evaluating this approximation is 8 if we use Horner's method, dropping to 4 with the use of the Paterson-Stockmeyer method.

4. Proper choices of *r*, *s* and *N*

Following the approach in paper [8], we denote T_r and R_s the polynomial in $\phi_0(B)$ of degree r and $\phi_1(B)$ of degree s, respectively (we use Taylor expansions with the aid of the Paterson-Stockmeyer method in this paper). The truncation errors of $\phi_0(B)$ and $\phi_1(B)$ have the following forms:

$$\phi_0(B) - T_r(B) = \sum_{i=r+1}^{\infty} (-1)^i c_{2i} B^i,$$

and

$$\phi_1(B) - R_s(B) = \sum_{i=s+1}^{\infty} (-1)^i d_{2i+1} B^i,$$

where $c_{2i} = \dfrac{1}{(2i)!}$ and $d_{2i+1} = \dfrac{1}{(2i+1)!}$.

Denote $\theta = \|B\|$ and we then have

$$\|\phi_0(B) - T_r(B)\| \le \sum_{i=r+1}^{\infty} |c_{2i}|\theta^i, \tag{13}$$

and

$$\|\phi_1(B) - R_s(B)\| \le \sum_{i=s+1}^{\infty} |d_{2i+1}|\theta^i. \tag{14}$$

It is clearly that

$$\|\phi_0(B)\| \ge 1 - \frac{\|B\|}{2!} - \frac{\|B\|^2}{4!} - \frac{\|B\|^3}{6!} - \ldots = 1 - \left(\cosh(\|B\|^{\frac{1}{2}}) - 1\right) = 2 - \cosh(\theta^{\frac{1}{2}}),$$

and

$$\|\phi_1(B)\| \ge 1 - \frac{\|B\|}{3!} - \frac{\|B\|^2}{5!} - \frac{\|B\|^3}{7!} - \ldots = 1 - \left(\frac{\sinh(\|B\|^{\frac{1}{2}})}{\|B\|^{\frac{1}{2}}} - 1\right) = 2 - \frac{\sinh(\theta^{\frac{1}{2}})}{\theta^{\frac{1}{2}}}.$$

Combining these two bounds with (13) and (14), respectively, we conclude that

$$\frac{\|\phi_0(B) - T_r(B)\|}{\|\phi_0(B)\|} \le \frac{\sum\limits_{i=r+1}^{\infty} |c_{2i}|\theta^i}{2 - \cosh(\theta^{\frac{1}{2}})}, \text{ for } \theta < (\cosh^{-1}(2))^2 \approx 1.734, \tag{15}$$

and

$$\frac{\|\phi_1(B) - R_s(B)\|}{\|\phi_1(B)\|} \le \frac{\sum\limits_{i=s+1}^{\infty} |d_{2i+1}|\theta^i}{2 - \frac{\sinh(\theta^{\frac{1}{2}})}{\theta^{\frac{1}{2}}}}, \text{ for } 2 - \frac{\sinh(\theta^{\frac{1}{2}})}{\theta^{\frac{1}{2}}} > 0, \text{ i.e., } \theta < 4.741. \tag{16}$$

We now consider the approximation of function $\phi_0(B)$, define θ_r to be the largest value of θ such that the relative error bound in (15) does not exceed $u = 2^{-53} \approx 1.1 \times 10^{-16}$, the unit roundoff in IEEE double precision arithmetic.

Firstly, we note from Table 1 that as r increases, θ_r rapidly approaches $(\cosh^{-1}(2))^2$ and then for larger r, the differences are at most 10^{-5}.

Table 1
Maximum value θ_r of θ such that the relative error bound in (15) does not exceed $u = 2^{-53}$ and π_r the number of matrix multiplication required to evaluate T_r.

r	1	2	3	4	5	6	7	8	9	10	11	12
θ_r	5.2e-8	4.3e-5	1.5e-3	0.013	0.061	0.189	0.453	0.895	1.467	1.729	1.734	1.734
π_r	0	1	2	2	3	3	4	4	4	5	5	5

Table 2
Maximum value γ_s of θ such that the relative error bound in (14) does not exceed $u = 2^{-53}$ and π_s, the number of matrix multiplication required to evaluate R_s.

s	1	2	3	4	5	6	7	8	9	10	11	12
γ_s	1.2e-7	8.2e-5	2.5e-3	0.021	0.094	0.281	0.658	1.297	2.247	3.473	4.558	4.739
π_s	0	1	2	2	3	3	4	4	4	5	5	5

Secondly, it can be observed from Table 1 that T_3, T_5, $T_7(T_8)$, and $T_{10}(T_{11})$ are evaluated at the same cost as the more accurate results T_4, T_6, T_9, and T_{12}, respectively, accordingly, we can remove $r = 3, 5, 7, 8, 10, 11$ from the original consideration.

Likewise, for the computation of $\phi_1(V)$, define γ_s to be the largest value of θ such that the relative error bound in (16) does not exceed $u = 2^{-53} \approx 1.1 \times 10^{-16}$. Then we have the results in Table 2. Similarly, we can also remove $s = 3, 5, 7, 8, 10, 11$ from the original consideration and only the evaluations for $s = 1, 2, 4, 6, 9, 12$ in Table 2 are needed.

If $\theta = \|V\| \le \theta_r$, for some $r = 1, 2, 4, 6, 9, 12$ and $\theta = \|V\| \le \gamma_s$, for some $s = 1, 2, 4, 6, 9, 12$, then T_r and R_s with the corresponding smallest r and s are the approximations to $\phi_0(V)$ and $\phi_1(V)$, respectively. This also includes the case of $V = B$ which corresponds to the case of $N = 0$. Otherwise, we have to divide V by 4^N, where N is chosen minimally so that $\left\|4^{-N}V\right\| \le \min\{\theta_r, \gamma_s\}$ for some r and s. It should be noted that $r = 12$ and $r = 9$ are the only possibilities according to the reasons as follows.

(i) If for some N, s.t., $\theta_9 \le \left\|4^{-N}V\right\| \le \theta_{12}$, then nothing would be gained by a further scaling by 1/4 since $\theta_6 < \theta_9/4$.
(ii) As the reason stated in (i), for the cost of evaluating γ_s, only $s = 9$ and $s = 12$ should be considered. Furthermore, comparing Table 1 with Table 2, we can observe that $\theta_9 < \gamma_9$ and $\theta_{12} < \gamma_{12}$. Therefore, the N derived according to θ_9 and θ_{12} can also meet the requirements for obtaining R_s.

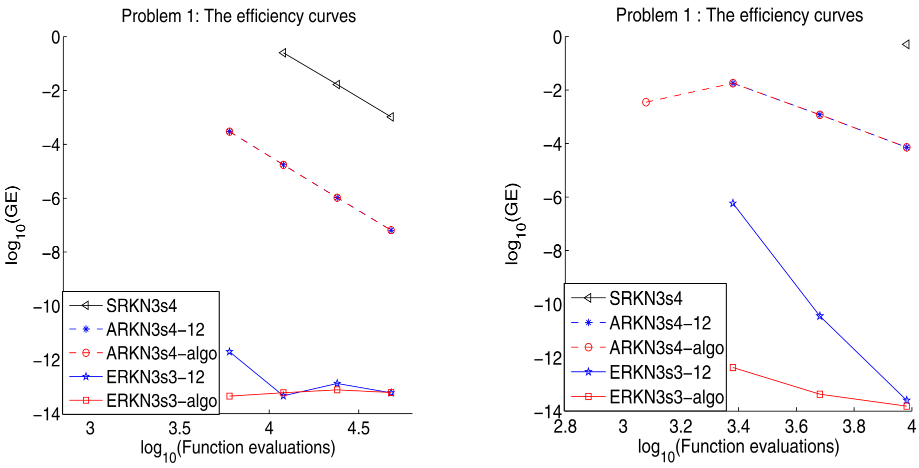
The strategy can minimize the number of double-angle steps. Then, the corresponding T_r and R_s are the approximations to $\phi_0(B)$ and $\phi_1(B)$, respectively.

Thus, the algorithm with minimizing the number of double-angle steps derived in this paper can be described by Algorithm 4.1 as follows.

Algorithm 4.1. *Given a matrix V and θ_r, γ_s, $r, s \in \{1, 2, 4, 6, 9, 12\}$ given in Table 1 and Table 2, this algorithm approximates the matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$.*

1. *Choose the minimum N such that $B = 4^{-N}V$ satisfies $\|B\| < \theta_r$, for some $r \in \{1, 2, 4, 6, 9, 12\}$ and $\|B\| < \gamma_s$, for some $s \in \{1, 2, 4, 6, 9, 12\}$.*
2. *Calculate C_0, where*
$$C_0 = I - \frac{B}{2!} + \frac{B^2}{4!} - \frac{B^3}{6!} + \ldots + (-1)^r \frac{B^r}{(2r)!}$$
is the approximation to $\phi_0(B)$.
3. *Calculate S_0, where*
$$S_0 = I - \frac{B}{3!} + \frac{B^2}{5!} - \frac{B^3}{7!} + \ldots + (-1)^s \frac{B^s}{(2s+1)!}$$
is the approximation to $\phi_1(B)$.
4. *for $i = 1, \ldots, N$ do steps 5 and 6.*
5. *Set $C_i = 2C_{i-1}^2 - I$.*
6. *Set $S_i = C_{i-1}S_{i-1}$.*
7. *Set $\phi_0(V) = C_N$, and Set $\phi_1(V) = S_N$.*
8. *STOP.*

We use Algorithm 4.1 to compute the values of $\phi_0(V)$ and $\phi_1(V)$ which are involved in the multi-frequency and multidimensional ARKN and ERKN integrators. With these values, we apply the methods ARKN3s4 and ERKN3s3 to the two



(i) (ii)

Fig. 3. Results for Problem 1 from Algorithm 4.1. The logarithm of the maximum global error (GE) over the integration interval against the number of function evaluations. (i): Stepsizes are $h = 1/(5 \times 2^j)$, $j = 2, 3, 4, 5$. (ii): Stepsizes are $h = 1/(2^j)$, $j = 2, 3, 4, 5$.

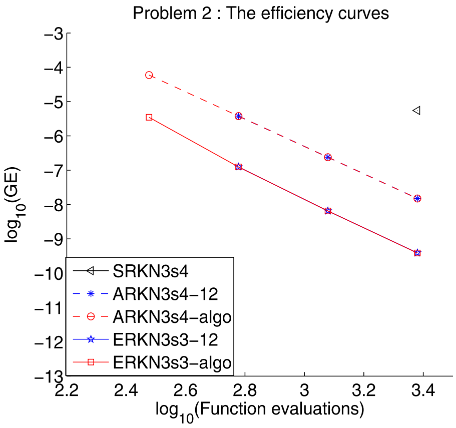


Fig. 4. Results for Problem 2 from Algorithm 4.1. The logarithm of the maximum global error (GE) over the integration interval against the number of function evaluations. Stepsizes are $h = 3/(15 \times 2^j)$, where $j = 1, 2, 3, 4$.

wave equations which are considered in the previous section. We denote the two integrators using the algorithm for the approximations to $\phi_0(V)$ and $\phi_1(V)$ by ARKN3s4-algo and ERKN3s3-algo, respectively, and compare them with ARKN3s4-12 and ERKN3s3-12, respectively. The results are shown in Fig. 3 and Fig. 4.

Comparing Fig. 1 with Fig. 3, it can be observed that, for ARKN method of order 4 and ERKN method of order 3, although the choice of $q = 8$ and $q = 12$ should be sufficient, the accuracy of matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$ is still important. For example, for stepsize $h = 1/4$, Fig. 1 (ii) shows that for all the methods used in Section 2, the errors are too large to plot. However, Fig. 3 (ii) demonstrates that for the same stepsizes, the method ARKN3s4-algo performs well. Especially, for comparatively larger stepsizes (it is very important to use larger stepsizes for long-term integration of oscillatory problems), the new algorithms behave much better than the compared conventional methods based on the truncated series expansions of $\phi_0(V)$ and $\phi_1(V)$. The same result also can be observed from Fig. 4. This phenomenon also motivates the new stability analysis of the ARKN and ERKN methods in the next section.

5. Stability analysis of ARKN and ERKN methods on the basis of approximations to $\phi_0(V)$ and $\phi_1(V)$

In this section, we propose a new approach to analyzing the stability of the methods ARKN3s4-8, ARKN3s4-12, ARKN3s4-algo, ERKN3s3-8, ERKN3s3-12 and ERKN3s3-algo.

In the case of classical RKN methods, the stability properties are analyzed usually by applying the methods to the second-order homogeneous linear test model $q'' = -\lambda^2 q$, $\lambda > 0$. Inspired by P.J. Van der Houwen et al.'s approach to the analysis of stability in [12], we consider the reformed linear test equation

$$q'' + \omega^2 q = -\varepsilon q, \tag{17}$$

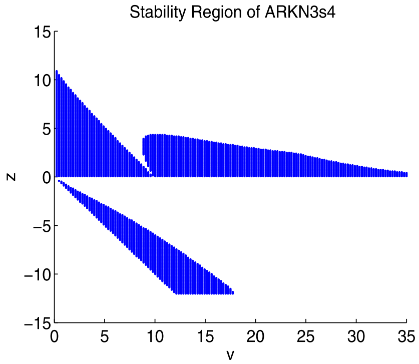


Fig. 5. Stability regions for the method ARKN3s4.

where $\varepsilon = \lambda^2 - \omega^2$ represents the error of the estimate ω of the principal frequency λ, $\omega^2 + \varepsilon > 0$. Applying ARKN methods (3) to (17) yields

$$\begin{cases} Q = eq_n + hcq'_n - (V+z)\bar{A}Q, \quad z = \varepsilon h^2, \quad V = h^2\omega^2, \\ q_{n+1} = \phi_0(V)q_n + \phi_1(V)hq'_n - z\bar{b}^T(V)Q, \\ hq'_{n+1} = -V\phi_1(V)q_n + \phi_0(V)hq'_n - zb^T(V)Q. \end{cases} \tag{18}$$

Therefore, the numerical solution gives the recursion

$$\begin{pmatrix} q_{n+1} \\ hq'_{n+1} \end{pmatrix} = R(V,z)\begin{pmatrix} q_n \\ hq'_n \end{pmatrix},$$

where the stability matrix is given by

$$R(V,z) = \begin{pmatrix} \phi_0(V) - z\bar{b}^T(V)N^{-1}e & \phi_1(V) - z\bar{b}^T(V)N^{-1}c \\ -V\phi_1(V) - zb^T(V)N^{-1}e & \phi_0(V) - zb^T(V)N^{-1}c \end{pmatrix}, \quad N = I + (V+z)\bar{A}. \tag{19}$$

The phase and stability behavior of the numerical solution depends on the eigenvalues or the spectrum of the numerical advancing matrix $R = R(V,z)$. Thus we have to consider the characteristic equation

$$\xi^2 - \mathrm{tr}(R)\xi + \det(R) = 0,$$

where $\mathrm{tr}(R)$ and $\det(R)$ are the trace and the determinant of R, respectively. The stability property of an ARKN method is characterized by the spectral radius $\rho(R)$.

Definition 5.1. By the region of stability for ARKN methods we mean the region $R_s = \{(V,z)|V > 0, \ \rho(R) < 1\}$ in the (V,z)-plane.

For the one-dimensional test equation (17), the stability region of the methods ARKN3s4 used in this paper without approximating the functions $\phi_0(V)$ and $\phi_1(V)$ is depicted in Fig. 5.

In numerical implementation, we use $\hat{\phi}_0(V)$ and $\hat{\phi}_1(V)$ instead of exact $\phi_0(V)$ and $\phi_1(V)$, respectively in the stability matrix (19) and obtain

$$\hat{R}(V,z) = \begin{pmatrix} \hat{\phi}_0(V) - z\bar{b}^T(V)N^{-1}e & \hat{\phi}_1(V) - z\bar{b}^T(V)N^{-1}c \\ -V\hat{\phi}_1(V) - zb^T(V)N^{-1}e & \hat{\phi}_0(V) - zb^T(V)N^{-1}c \end{pmatrix}, \quad N = I + (V+z)\bar{A}. \tag{20}$$

Here, $\hat{\phi}_0(V)$ and $\hat{\phi}_1(V)$ are the numerical approximations of $\phi_0(V)$ and $\phi_1(V)$. The corresponding stability region is $\hat{R}_s = \{(V,z)|V > 0, \ \rho(\hat{R}) < 1\}$ in the (V,z)-plane.

The stability region of the methods ARKN3s4-8, ARKN3s4-12 and ARKN3s4-algo based on the stability matrix (20) are depicted in Fig. 6. It is shown that the stability region of the method ARKN3s4-algo is the most close to the exact stability region which is depicted in Fig. 5.

Applying the multi-frequency and multidimensional ERKN method (4) to the equation (17) yields

$$\begin{cases} Q = \phi_0(c^2V)q_n + c\phi_1(c^2V)hq'_n - z\bar{A}(V)Q, \qquad z = \varepsilon h^2, \qquad V = h^2\omega^2, \\ q_{n+1} = \phi_0(V)q_n + \phi_1(V)hq'_n - z\bar{b}^T(V)Q, \\ hq'_{n+1} = -V\phi_1(V)q_n + \phi_0(V)hq'_n - zb^T(V)Q. \end{cases}$$

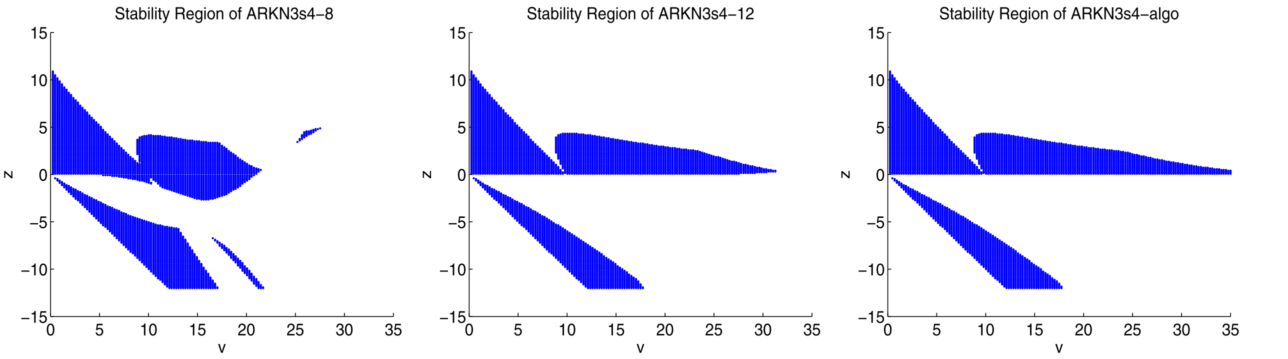


Fig. 6. Stability regions for the methods ARKN3s4-8 (left), ARKN3s4-12 (middle), and ARKN3s4-algo (right).

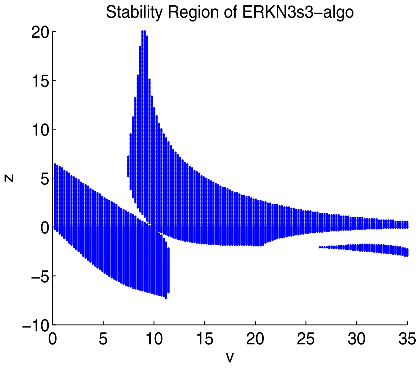


Fig. 7. Stability regions for the method ERKN3s3.

It follows that

$$\begin{pmatrix} q_{n+1} \\ hq'_{n+1} \end{pmatrix} = S(V,z) \begin{pmatrix} q_n \\ hq'_n \end{pmatrix}$$

with the stability matrix $S(V,z)$ given by

$$S(V,z) = \begin{pmatrix} \phi_0(V) - z\bar{b}^T(V)N^{-1}\phi_0(c^2V) & \phi_1(V) - z\bar{b}^T(V)N^{-1}(c\cdot\phi_1(c^2V)) \\ -V\phi_1(V) - zb^T(V)N^{-1}\phi_0(c^2V) & \phi_0(V) - zb^T(V)N^{-1}(c\cdot\phi_1(c^2V)) \end{pmatrix}, \tag{21}$$

with $N = I + z\bar{A}(V)$. The spectral radius $\rho\big(S\big)$ characterize the stability of the method.

Definition 5.2. The region in the (V,z)-plane $\bar{R}_s = \{(V,z)|\ V > 0,\ \rho(S) < 1\}$ is called the stability region of a multi-frequency and multidimensional ERKN method.

For the one-dimensional test equation (17), the stability region of the methods ERKN3s3 used in this paper without approximating the functions $\phi_0(V)$ and $\phi_1(V)$ is depicted in Fig. 7.

Similarly, in numerical implementation, we obtain

$$\hat{S}(V,z) = \begin{pmatrix} \hat{\phi}_0(V) - z\bar{b}^T(V)N^{-1}\hat{\phi}_0(c^2V) & \phi_1(V) - z\bar{b}^T(V)N^{-1}(c\cdot\hat{\phi}_1(c^2V)) \\ -V\hat{\phi}_1(V) - zb^T(V)N^{-1}\hat{\phi}_0(c^2V) & \hat{\phi}_0(V) - zb^T(V)N^{-1}(c\cdot\hat{\phi}_1(c^2V)) \end{pmatrix} \tag{22}$$

with $\hat{\phi}_0(V)$ and $\hat{\phi}_1(V)$ the numerical approximation of $\phi_0(V)$ and $\phi_1(V)$, respectively in (21). The corresponding stability region is $\hat{\bar{R}}_s = \{(V,z)|V > 0,\ \rho(\hat{S}) < 1\}$ in the (V,z)-plane.

The stability regions of the integrators ERKN3s3-8, ERKN3s3-12 and ERKN3s3-algo based on the stability matrix (22) are depicted in Fig. 8. It is also shown that the stability region of the method ERKN3s3-algo is the most close to the exact stability region which is given in Fig. 7.

It can be observed from Figs. 1–8 that the numerical method with large time stepsize might produce very large global errors that cannot be displayed in the figures. The reason is that the method with this stepsize is out of the corresponding stability region. That is, the limited truncated series expansions of $\phi_0(V)$ and $\phi_1(V)$ in place of the functions $\phi_0(V)$ and $\phi_1(V)$ can not only make notable impact on the efficiency of the numerical methods but also destroy the stability properties of the original ARKN and ERKN schemes to some extent.

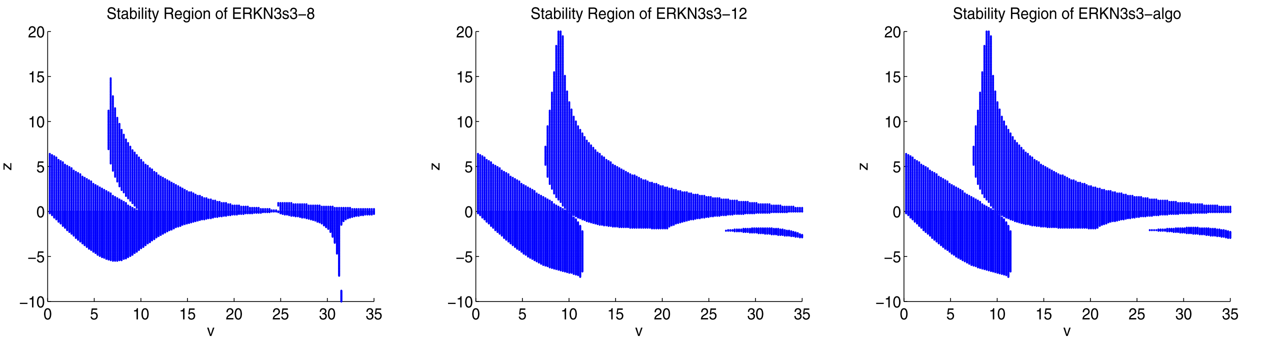


Fig. 8. Stability regions for the methods ERKN3s3-8 (left), ERKN3s3-12 (middle), and ERKN3s3-algo (right).

6. Conclusions

The design and analysis of numerical methods for multi-frequency oscillatory systems is an important problem that has received a great deal of attention in the last few years. In this paper, we paid attention to the efficient implementation issues of the multi-frequency and multidimensional ARKN and ERKN integrators which are closely related to the calculations of the matrix-valued functions $\phi_0(V)$ and $\phi_1(V)$. For the multi-frequency oscillatory second-order oscillatory problem (1) with multiple time scales considered in this paper, M is not necessarily symmetric nor diagonal, therefore, the computation of the matrix-valued functions is required to be independent of matrix decompositions. We analyzed and derived our algorithm by employing variable-degree approximations of series expansions, where the degrees were chosen to minimize the computational cost, and by employing the relative error bounds expressed in terms of the norm of the matrix. Based on the properties of $\phi_0(V)$ and $\phi_1(V)$ and their relations, the two matrix-valued functions can be evaluated simultaneously at lower cost than evaluated separately. Illuminating experiments show that, the accuracy of the computation of functions $\phi_0(V)$ and $\phi_1(V)$ makes notable impact on achieving the high efficiency of ARKN and ERKN methods. Moreover, we presented an important stability analysis when the numerical approximations to $\phi_0(V)$ and $\phi_1(V)$ are considered which shows the importance of the accuracy of calculating $\phi_0(V)$ and $\phi_1(V)$ from the viewpoint of the numerical stability.

References

[1] A.H. Al-Mohy, N.J. Higham, S.D. Relton, New algorithms for computing the matrix sine and cosine separately or simultaneously, SIAM J. Sci. Comput. 37 (2015) A456–A487.
[2] J.M. Franco, New methods for oscillatory systems based on ARKN methods, Appl. Numer. Math. 56 (2006) 1040–1053.
[3] B. García-Archilla, J.M. Sanz-Serna, R.D. Skeel, Long-time-step methods for oscillatory differential equations, SIAM J. Sci. Comput. 20 (1999) 930–963.
[4] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd ed., Johns Hopkins Univ. Press, Baltimore, MD, 1996.
[5] A.B. González, P. Martín, J.M. Farto, A new family of Runge-Kutta type methods for the numerical integration of perturbed oscillators, Numer. Math. 82 (1999) 635–646.
[6] E. Hairer, C. Lubich, Long-time energy conservation of numerical methods for oscillatory differential equations, SIAM J. Numer. Anal. 38 (2000) 414–441.
[7] E. Hairer, C. Lubich, G. Wanner, Geometric Numerical Integration: Structure-Preserving Algorithms, 2nd edition, Springer-Verlag, Berlin, Heidelberg, 2006.
[8] G.I. Hargreaves, N.J. Higham, Efficient algorithms for the matrix cosine and sine, Numer. Algorithms 40 (2005) 383–400.
[9] M. Hochbruck, C. Lubich, A Gautschi-type method for oscillatory second-order differential equations, Numer. Math. 83 (1999) 403–426.
[10] S. Jimánez, L. Vázquez, Analysis of four numerical schemes for a nonlinear Klein-Gordon equation, Appl. Math. Comput. 35 (1990) 61–93.
[11] S.M. Serbin, S.A. Blalock, An algorithm for computing the matrix cosine, SIAM J. Sci. Stat. Comput. 1 (2) (1980) 198–204.
[12] P.J. Van der Houwen, B.P. Smmeijer, Explicit Runge-Kutta(-Nyström) methods with reduced phase errors for computing oscillating solution, SIAM J. Numer. Anal. 24 (1987) 595–617.
[13] B. Wang, X. Wu, J. Xia, Error bounds for explicit ERKN integrators for systems of multi-frequency oscillatory second-order differential equations, Appl. Numer. Math. 74 (2013) 17–34, https://doi.org/10.1016/j.apnum.2013.08.002.
[14] X. Wu, B. Wang, Multidimensional adapted Runge-Kutta-Nyström methods for oscillatory systems, Comput. Phys. Commun. 181 (2010) 1955–1962.
[15] X. Wu, X. You, J. Xia, Order conditions for ARKN methods solving oscillatory systems, Comput. Phys. Commun. 180 (2009) 2250–2257.
[16] X. Wu, X. You, W. Shi, B. Wang, ERKN integrators for systems of oscillatory second-order differential equations, Comput. Phys. Commun. 181 (2010) 1873–1887.
[17] X. Wu, X. You, B. Wang, Structure-Preserving Algorithms for Oscillatory Differential Equations, Springer-Verlag, Berlin, Heidelberg, 2013.
[18] H. Yang, X. Wu, X. You, Y. Fang, Extended RKN-type methods for numerical integration of perturbed oscillators, Comput. Phys. Commun. 180 (2009) 1777–1794.