 CMES, vol.41, no.1, pp.1-26, 2009

The Fourth-Order Group Preserving Methods for the Integrations of Ordinary Differential Equations

Hung-Chang Lee[1] **and Chein-Shan Liu**[2]

Abstract: The group-preserving schemes developed by Liu (2001) for integrating ordinary differential equations system were adopted the Cayley transform and Padé approximants to formulate the Lie group from its Lie algebra. However, the accuracy of those schemes is not better than second-order. In order to increase the accuracy by employing the group-preserving schemes on ordinary differential equations, according to an efficient technique developed by Runge and Kutta to raise the order of accuracy from the Euler method, we combine the Runge-Kutta method on the group-preserving schemes to obtain the higher-order numerical methods of group-preserving type. They provide single-step explicit time integrators for differential equations. Several numerical examples are examined, showing that the higher-order group-preserving schemes have good computational efficiency and high accuracy.

Keywords: Nonlinear dynamical system, Ordinary differential equations, Cone, Minkowski space, Group preserving scheme (GPS), Fourth-order accuracy, Runge-Kutta method

1 Introduction

There are a lot of numerical methods to integrate ordinary differential equations (ODEs). In general, those methods are effective. However, when the considered ODEs possess certain structure like as the first integrals, the Lie symmetry in the tangent bundle, or a symplectic form on the cotangent bundle, the numerical methods used for general purpose cannot retain these properties very well, unless they are designed to do so. In the past few years there has been a substantial development in the geometric integrators of ODEs evolving on the Lie groups and more generally on the homogeneous spaces as shown by Iserles, Munthes-Kaas, Norsett

[1] Department of Mechanical Engineering, Kun Shan University, Tainan, Taiwan

[2] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan. E-mail: liuchei.nshan@msa.hinet.net

and Zanna (2000), and Hairer, Lubich and Wanner (2002).

For more than a century, the Lie groups have played a decisive role in our understanding of the geometry of differential equations. It is believed that the concept of Lie groups, within the wider terminology and machinery of differential geometry, is very helpful in devising superior numerical methods to discretize the ODEs to retain the invariant property. By sharing the geometric structure and invariance with the original ODEs, the new methods are thought to be more accurate, more stable and more effective than the conventional numerical methods.

In an attempt of devising the geometric integrators that retain the invariance of the underlying ODEs, the second author has presented numerical methods to integrate the augmented ordinary differential equations that evolve on a matrix Lie group as shown by Liu (2001). The schemes apply to the problem of finding numerical approximations to the solution of

$$\dot{\mathbf{Y}} = \mathbf{A}(t,\mathbf{Y})\mathbf{Y}, \;\; \mathbf{Y}(0) = \mathbf{Y}_0, \tag{1}$$

whereby the exact solution $\mathbf{Y}$ evolves in a matrix Lie group and $\mathbf{A}$ is a matrix function on the associated Lie algebra.

Many practical engineering problems of interest can be modeled by systems of differential equations whose solutions satisfy some invariants. In the past several decades, a particular attention has been paid to developing numerical methods which approximate the solution of such a system while preserving invariants to a machinery precision; see, e.g., Baumgarte (1972), Führer and Leimkuhler (1991), Ascher and Petzold (1991), März (1991, 2002), Ascher, Chin and Reich (1994), Campbell and Moore (1995), Ascher (1997), Chan, Chartier and Murua (2002), Arevalo, Campbell and Selva (2004), and references therein.

We begin with the following k-dimensional ODEs system:

$$\dot{\mathbf{x}} = \mathbf{f}(t,\mathbf{x}), \;\; \mathbf{x}(0) = \mathbf{x}_0, \;\; t \in \mathbb{R}, \;\; \mathbf{x} \in \mathbb{R}^k. \tag{2}$$

It is an important feature of the above initial value problem, that the solution value at any given point on the trajectory determines the solution at all later points on the trajectory. In effect, the solutions $\mathbf{x}(t;\mathbf{x}_0)$ of the differential equations define a one-parameter mappings $\{\phi_t\}_{t\geq 0}$, which take initial data to later points along trajectories:

$$\phi_t(\mathbf{x}_0) = \mathbf{x}(t;\mathbf{x}_0). \tag{3}$$

We term the map $\phi_t : \mathbb{R}^k \mapsto \mathbb{R}^k$ the flow map of the given system.

In general, the flow map has the following property. If we solve the differential equations from a given initial point $\mathbf{x}_0$ up to a time t_1, then solve from the resulting point forward t_2 units of time, the effect is the same as solving the differential equations with the initial value $\mathbf{x}_0$ up to a time t_1+t_2. In terms of the mapping, that is, $\phi_{t_1}\circ\phi_{t_2}=\phi_{t_1+t_2}$. Such a mapping is referred to as a one-parameter semigroup.

In general, the differential equations have such a semigroup property, but they not necessarily have the Lie-group property. Group-preserving scheme (GPS) can preserve the internal symmetry group of the considered system. Although we do not know previously the symmetry group of the nonlinear differential equations systems, Liu (2001) has embedded them into the augmented dynamical systems, which concern with not only the evolution of state variables but also the evolution of state vector's magnitude. That is, for the k-dimensional ordinary differential equations system (2), we can embed it to the following $k+1$-dimensional augemented dynamical system:

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}\\ \|\mathbf{x}\|\end{bmatrix}=\begin{bmatrix}\mathbf{0}_{k\times k} & \frac{\mathbf{f}(t,\mathbf{x})}{\|\mathbf{x}\|}\\ \frac{\mathbf{f}^{\mathrm{T}}(t,\mathbf{x})}{\|\mathbf{x}\|} & 0\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \|\mathbf{x}\|\end{bmatrix}. \tag{4}$$

It is obvious that the first equation in Eq. (4) is the same as the original equation (2), but the addition of the second equation gives us a Minkowskian structure of the augmented state variables of $\mathbf{X}:=(\mathbf{x}^{\mathrm{T}},\|\mathbf{x}\|)^{\mathrm{T}}$, satisfying the cone condition:

$$\mathbf{X}^{\mathrm{T}}\mathbf{g}\mathbf{X}=0, \tag{5}$$

where

$$\mathbf{g}=\begin{bmatrix}\mathbf{I}_k & \mathbf{0}_{k\times 1}\\ \mathbf{0}_{1\times k} & -1\end{bmatrix} \tag{6}$$

is a Minkowski metric, $\mathbf{I}_k$ is the identity matrix of order k, and the superscript T stands for the transpose. In terms of $(\mathbf{x},\|\mathbf{x}\|)$, Eq. (5) becomes

$$\mathbf{X}^{\mathrm{T}}\mathbf{g}\mathbf{X}=\mathbf{x}\cdot\mathbf{x}-\|\mathbf{x}\|^2=\|\mathbf{x}\|^2-\|\mathbf{x}\|^2=0, \tag{7}$$

where the dot between two k-dimensional vectors denotes their Euclidean inner product. The cone condition (5) is thus a natural constraint that we have to impose on the augmented dynamical system (4).

As a consequence, we have a $k+1$-dimensional augmented ODEs system:

$$\dot{\mathbf{X}}=\mathbf{A}(t,\mathbf{X})\mathbf{X} \tag{8}$$

with a constraint (5), where

$$\mathbf{A}:=\begin{bmatrix}\mathbf{0}_{k\times k} & \frac{\mathbf{f}(t,\mathbf{x})}{\|\mathbf{x}\|}\\ \frac{\mathbf{f}^{\mathrm{T}}(t,\mathbf{x})}{\|\mathbf{x}\|} & 0\end{bmatrix}, \tag{9}$$

satisfying

$$\mathbf{A}^{\mathrm{T}}\mathbf{g}+\mathbf{g}\mathbf{A}=\mathbf{0}, \tag{10}$$

is a Lie algebra $so(k,1)$ of the proper orthochronous Lorentz group $SO_o(k,1)$. This fact prompts us to devise the so-called group-preserving scheme, whose discretized mapping $\mathbf{G}$ must exactly preserve the following Lie-group properties [Liu (2001)]:

$$\mathbf{G}^{\mathrm{T}}\mathbf{g}\mathbf{G}=\mathbf{g}, \tag{11}$$

$$\det\mathbf{G}=1, \tag{12}$$

$$G_0^0>0, \tag{13}$$

where G_0^0 is the 00-th component of $\mathbf{G}$. Such $\mathbf{G}$ is a proper orthochronous Lorentz group denoted by $SO_o(k,1)$. The term orthochronous used in the special relativity theory is referred to as the preservation of time orientation. Here, it should be understood as the preservation of the sign of $\|\mathbf{x}\|$.

Therefore, the original k-dimensional dynamical system (2) in $\mathbb{E}^k$ can be embedded naturally and mathematically equivalently into an augmented $k+1$-dimensional dynamical system (8) in $\mathbb{M}^{k+1}$. Although the dimension of the new system is raising one more, it has been shown that under the Lipschitz condition of

$$\|\mathbf{f}(t,\mathbf{x})-\mathbf{f}(t,\mathbf{y})\|\leq\mathscr{L}\|\mathbf{x}-\mathbf{y}\|,\ \ \forall\ (t,\mathbf{x}),(t,\mathbf{y})\in\mathbb{D}, \tag{14}$$

where $\mathbb{D}$ is a domain of $\mathbb{R}\times\mathbb{R}^k$, and $\mathscr{L}$ is known as a Lipschitz constant, the new system has the advantage of allowing us to develop the group-preserving numerical scheme as follows [Liu (2001)]:

$$\mathbf{X}_{n+1}=\mathbf{G}(n)\mathbf{X}_n, \tag{15}$$

where $\mathbf{X}_n$ denotes the numerical value of $\mathbf{X}$ at a discrete time t_n, and $\mathbf{G}(n)\in SO_o(k,1)$ is the group value at the time t_n.

The main purpose of Lie-group solver is for providing a better algorithm that retains the orbit generated from numerical solution on the manifold which associated with the Lie-group. The retention of Lie-group structure under discretization is a vital task in the recovery of qualitatively correct behavior in the minimization of numerical error. Indeed, the GPS method is very effective to deal with ODEs with special structures as shown by Lee, Chen and Hung (2002) and Liu (2005) for stiff ODEs, and by Liu (2006a) for ODEs with constraints. Chen, Liu and Chang (2007) have modified the GPS to a time-adaptive numerical scheme for ODEs. Many profound extensions of the GPS to other fiels are mainly contributed by Liu and his coworkers; see, for example, Liu and Ku (2005), Chang, Liu and Chang (2005), Liu (2006b, 2006c, 2006d, 2006e, 2006f, 2006g, 2007), and Liu, Chang and Chang (2006, 2009).

The theory of Lie-group and Lie-algebra has been developed for a long time. However, the Lie-group methods to be employed on the numerical methods are only developed very recently. The methods have been developed mainly by Munthe-Kaas and Iserles. The method is used to construct the numerical solution of differential equations evolving on manifold. The numerical solutions have formed by evolving on the same manifold as the analytical solutions are. At present, there are many famous methods such as the Crouch-Grossman methods, the RKMK methods, the Magnus methods, the Fer methods, etc. The past studies clearly indicated that the Lie-group methods not only produce an improved qualitative behavior but also allow for a more accurate long time integration than that with the general purpose methods, like as forward Euler method and RK4 method. Zhang and Deng (2006) have extended the GPS by combining it with the above mentioned RKMK methods.

The idea of extending the Euler method by allowing for a multiplicity of evaluations of the vector field functions within each step was originally proposed by Runge in 1895. Further contributions were made by Kutta in 1901, who completely characterized the set of Runge-Kutta method of order four, today known as the RK4 method. This paper will develop highly effective schemes by an extension of the group-preserving scheme developed by Liu (2001) by using the above technique due to Runge and Kutta. The new method provides an explicit single-step algorithm, and renders a more compendious numerical implementation than other geometric integrator schemes to solve ODEs. It would be found in this study that the new method greatly reduces the computation time that is important for conducting a long-term calculation.

Of course, Eq. (8) like as Eq. (1) is a good starting point to construct the higher-order GPS. Basically, there are two approaches: one is using the Lie algebra property of $\mathbf{A}$, and another is using the Lie group property of $\mathbf{G}$. The reader can refer the many works of the so-called geometric integrators, for example, Iserles (1984), Iserles and Norsett (1999), Munthes-Kaas (1998, 1999), Iserles, Munthes-Kaas, Norsett and Zanna (2000), and Hairer, Lubich and Wanner (2002). A very recent discussion of the Lie-group integration method for ODEs starting from Eq. (8) was made by Zhang and Deng (2004, 2005, 2006). In this paper we give a different approach by following an efficient method developed by Runge and Kutta one hundred years ago to raise the first-order accuracy of the Euler method to the fourth-order accuracy of the RK4 method.

2 The GPS for differential equations system

2.1 The Cayley transform

The group generated from $\mathbf{A} \in so(k,1)$ is known as a proper orthochronous Lorentz group, one of which is obtained from the Cayley transform

$$\mathrm{Cay}(\tau\mathbf{A}) = (\mathbf{I}_k - \tau\mathbf{A})^{-1}(\mathbf{I}_k + \tau\mathbf{A}), \tag{16}$$

a mapping from $\mathbf{A}$ to an element of $SO_o(k,1)$ for $\tau \in \mathbb{R}$. Substituting Eq. (9) for $\mathbf{A}(n)$, which denotes the value of $\mathbf{A}$ at the discrete time t_n, into the above equation yields

$$\mathbf{G}(n) = \mathrm{Cay}[\tau\mathbf{A}(n)] = \begin{bmatrix} \mathbf{I}_k + \frac{2\tau^2}{\|\mathbf{x}_n\|^2 - \tau^2\|\mathbf{f}_n\|^2}\mathbf{f}_n\mathbf{f}_n^{\mathrm{T}} & \frac{2\tau\|\mathbf{x}_n\|}{\|\mathbf{x}_n\|^2 - \tau^2\|\mathbf{f}_n\|^2}\mathbf{f}_n \\ \frac{2\tau\|\mathbf{x}_n\|}{\|\mathbf{x}_n\|^2 - \tau^2\|\mathbf{f}_n\|^2}\mathbf{f}_n^{\mathrm{T}} & \frac{\|\mathbf{x}_n\|^2 + \tau^2\|\mathbf{f}_n\|^2}{\|\mathbf{x}_n\|^2 - \tau^2\|\mathbf{f}_n\|^2} \end{bmatrix}, \tag{17}$$

where τ is one half of the time increment, i.e., $\tau := h/2$.
Inserting the above $\mathbf{G}(n)$ into Eq. (15) and taking its first row, we obtain

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta^c\mathbf{f}_n, \tag{18}$$

where

$$\eta^c = \frac{4\|\mathbf{x}_n\|^2 + 2h\mathbf{f}_n \cdot \mathbf{x}_n}{4\|\mathbf{x}_n\|^2 - h^2\|\mathbf{f}_n\|^2}h \tag{19}$$

is an adaptive factor. In the above $\mathbf{x}_n$ denotes the numerical value of $\mathbf{x}$ at the discrete time t_n, and $\mathbf{f}_n$ denotes $\mathbf{f}(t_n,\mathbf{x}_n)$ for a simple notation.
In order to meet the property (13), we require the stepsize of scheme (18) being constrained by $h < 2\|\mathbf{x}_n\|/\|\mathbf{f}_n\|$. Under this condition we have

$$h < \frac{2\|\mathbf{x}_n\|}{\|\mathbf{f}_n\|} \Longleftrightarrow G_0^0 > 0 \Longrightarrow \eta^c > 0. \tag{20}$$

Some properties of preserving the fixed point behavior of the above numerical scheme (18) have been investigated by Liu (2001), and when applying it to the stiff ODEs has revealed that it is easy to implement numerically and has high computational efficiency and accuracy as discussed by Liu (2005).

2.2 Exponential mapping

An exponential mapping of $\mathbf{A}(n)$ admits a closed-form representation:

$$\mathbf{G}(n) = \exp[h\mathbf{A}(n)] = \begin{bmatrix} \mathbf{I}_k + \frac{(a_n-1)}{\|\mathbf{f}_n\|^2}\mathbf{f}_n\mathbf{f}_n^{\mathrm{T}} & \frac{b_n\mathbf{f}_n}{\|\mathbf{f}_n\|} \\ \frac{b_n\mathbf{f}_n^{\mathrm{T}}}{\|\mathbf{f}_n\|} & a_n \end{bmatrix}, \tag{21}$$

where

$$a_n := \cosh\left(\frac{h\|\mathbf{f}_n\|}{\|\mathbf{x}_n\|}\right), \;\; b_n := \sinh\left(\frac{h\|\mathbf{f}_n\|}{\|\mathbf{x}_n\|}\right). \tag{22}$$

Substituting the above $\mathbf{G}(n)$ into Eq. (15) and taking its first row, we obtain

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta^e\mathbf{f}_n, \tag{23}$$

where

$$\eta^e := \frac{(a_n-1)\mathbf{f}_n\cdot\mathbf{x}_n + b_n\|\mathbf{x}_n\|\|\mathbf{f}_n\|}{\|\mathbf{f}_n\|^2}. \tag{24}$$

From $a_n > 1$, $\forall h > 0$ and $\|\mathbf{f}_n\|\|\mathbf{x}_n\| \geq \mathbf{f}_n\cdot\mathbf{x}_n \geq -\|\mathbf{f}_n\|\|\mathbf{x}_n\|$, we can prove that

$$\left[\exp\left(\frac{h\|\mathbf{f}_n\|}{\|\mathbf{x}_n\|}\right)-1\right]\frac{\|\mathbf{x}_n\|}{\|\mathbf{f}_n\|} \geq \eta^e \geq \frac{\|\mathbf{x}_n\|}{\|\mathbf{f}_n\|}\left[1-\exp\left(-\frac{h\|\mathbf{f}_n\|}{\|\mathbf{x}_n\|}\right)\right] > 0, \;\; \forall h > 0. \tag{25}$$

This scheme is group properties preserved for all $h > 0$, and does not endure the same shortcoming as the one for scheme (18). However, by applying Eq. (23) to ODEs, a suitable stepsize h must be chosen, in order to avoid overflow.

3 The higher-order GPS for differential equations system

3.1 Higher-order GPS

Here we first prove that the construction of higher-order GPS can be carried out in the state space of $\mathbf{x}$. For this purpose let us prove that

$$\mathbf{X}_{n+1} = \mathbf{G}(n)\mathbf{X}_n \Longleftrightarrow \mathbf{x}_{n+1} = \mathbf{x}_n + \eta\mathbf{f}_n, \tag{26}$$

where η is used for η^c or η^e.

Proof: We first consider $\eta = \eta^c$. For Eq. (26), the left-hand side implying the right-hand side is already proven in Section 2.1. We prove the right-hand side implying the left-hand side. To begin with the right-hand side we have

$$\|\mathbf{x}_{n+1}\|^2 = \|\mathbf{x}_n\|^2 + 2\eta^c\mathbf{x}_n\cdot\mathbf{f}_n + (\eta^c)^2\|\mathbf{f}_n\|^2. \tag{27}$$

Inserting Eq. (19) for η^c into the above equation and through some algebraic manipulations we can get

$$\|\mathbf{x}_{n+1}\|^2 = \frac{\|\mathbf{x}_n\|^2[4\|\mathbf{x}_n\|^2 + h^2\|\mathbf{f}_n\|^2 + 4h\mathbf{f}_n \cdot \mathbf{x}_n]^2}{[4\|\mathbf{x}_n\|^2 - h^2\|\mathbf{f}_n\|^2]^2}. \tag{28}$$

Under the condition (20) with the time stepsize smaller than $2\|\mathbf{x}_n\|/\|\mathbf{f}_n\|$, and then taking the square roots of both the sides of the above equation, we obtain

$$\|\mathbf{x}_{n+1}\| = \frac{4\|\mathbf{x}_n\|^2 + h^2\|\mathbf{f}_n\|^2 + 4h\mathbf{f}_n \cdot \mathbf{x}_n}{4\|\mathbf{x}_n\|^2 - h^2\|\mathbf{f}_n\|^2}\|\mathbf{x}_n\|. \tag{29}$$

In terms of $\mathbf{X}_n = (\mathbf{x}_n^{\mathrm{T}}, \|\mathbf{x}_n\|)^{\mathrm{T}}$, from Eqs. (18), (19) and (29) we get the mapping in Eq. (15) with its $\mathbf{G}(n)$ given by Eq. (17). Therefore, we have proven Eq. (26).

Next, we consider $\eta = \eta^e$. For Eq. (26), the left-hand side implying the right-hand side is already proven in Section 2.2. We prove the right-hand side implying the left-hand side. To begin with the right-hand side we have

$$\|\mathbf{x}_{n+1}\|^2 = \|\mathbf{x}_n\|^2 + 2\eta^e \mathbf{x}_n \cdot \mathbf{f}_n + (\eta^e)^2\|\mathbf{f}_n\|^2. \tag{30}$$

Inserting Eq. (24) for η^e into the above equation and through some algebraic operations we can get

$$\|\mathbf{x}_{n+1}\|^2 = \frac{1}{\|\mathbf{f}_n\|^2}\left\{(1+b_n^2)\|\mathbf{x}_n\|^2\|\mathbf{f}_n\|^2 + 2a_nb_n\mathbf{f}_n \cdot \mathbf{x}_n\|\mathbf{x}_n\|\|\mathbf{f}_n\| + (a_n^2-1)(\mathbf{f}_n \cdot \mathbf{x}_n)^2\right\}. \tag{31}$$

By considering the equality $a_n^2 - 1 = b_n^2$ and taking the square roots of both the sides of the above equation, we obtain

$$\|\mathbf{x}_{n+1}\| = \frac{a_n\|\mathbf{x}_n\|\|\mathbf{f}_n\| + b_n\mathbf{f}_n \cdot \mathbf{x}_n}{\|\mathbf{f}_n\|}. \tag{32}$$

In terms of $\mathbf{X}_n = (\mathbf{x}_n^{\mathrm{T}}, \|\mathbf{x}_n\|)^{\mathrm{T}}$, from Eqs. (23), (24) and (32) we get the mapping in Eq. (15) with its $\mathbf{G}(n)$ given by Eq. (21). Therefore, we have proven Eq. (26). □

Eq. (26) is very significant for helping us to construct the higher-order Lie group methods in the state space of $\mathbf{x}$ rather that in the augmented state space of $\mathbf{X}$. It means that the equation

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta\mathbf{f}_n \tag{33}$$

is already exhibiting two structures: the cone and the Lie group mapping $\mathbf{G}$; however, η should be defined by one of Eqs. (19) and (24).

Even every numerical scheme may have an artificial cone structure but it usually does not have the Lie group mapping $\mathbf{G}$. Let us state this concept by the forward Euler scheme:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}_n. \tag{34}$$

Taking the square of the above equation one has

$$\|\mathbf{x}_{n+1}\|^2 = \|\mathbf{x}_n\|^2 + 2h\mathbf{f}_n \cdot \mathbf{x}_n + h^2\|\mathbf{f}_n\|^2. \tag{35}$$

In terms of the augmented state $\mathbf{X}_n = (\mathbf{x}_n^{\mathrm{T}}, \|\mathbf{x}_n\|)^{\mathrm{T}}$, one can easily check that Eq. (5) is satisfied, but from the above two equations we have

$$\mathbf{X}_{n+1} = \begin{bmatrix} \mathbf{I}_k & \mathbf{0}_{k\times 1} \\ \mathbf{0}_{1\times k} & \frac{\sqrt{\|\mathbf{x}_n\|^2+2h\mathbf{f}_n\cdot\mathbf{x}_n+h^2\|\mathbf{f}_n\|^2}}{\|\mathbf{x}_n\|} \end{bmatrix} \mathbf{X}_n + \begin{bmatrix} h\mathbf{f}_n \\ 0 \end{bmatrix}. \tag{36}$$

Nothing can be said about its Lie group transformation between $\mathbf{X}_n$ and $\mathbf{X}_{n+1}$.

However, the situation is very different if one replaces the h in Eq. (34) by η^c in Eq. (19) or by η^e in Eq. (24). The imposed cone for the Euler scheme is not a natural one. However, for the Cayley transform in Section 2.1 or the exponential mapping in Section 2.2, the cone condition is as natural as its Lie group mapping between two points $\mathbf{X}_n$ and $\mathbf{X}_{n+1}$ on the cone.

Next, we demonstrate the construction of the third-order Runge-Kutta method in the space of $\mathbf{X}$. Starting from Eq. (8) we can derive the third-order Runge-Kutta method (RK3) by

$$\mathbf{K}_1 = \mathbf{A}(t_n, \mathbf{X}_n)\mathbf{X}_n, \tag{37}$$

$$\mathbf{K}_2 = \mathbf{A}(t_n + \tau, \mathbf{X}_n + \tau\mathbf{K}_1)(\mathbf{X}_n + \tau\mathbf{K}_1), \tag{38}$$

$$\mathbf{K}_3 = \mathbf{A}(t_n + 1.5\tau, \mathbf{X}_n + 1.5\tau\mathbf{K}_2)(\mathbf{X}_n + 1.5\tau\mathbf{K}_2), \tag{39}$$

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \frac{h}{9}[2\mathbf{K}_1 + 3\mathbf{K}_2 + 4\mathbf{K}_3]. \tag{40}$$

The above RK3 does not guarantee that the mapping between $\mathbf{X}_n$ and $\mathbf{X}_{n+1}$ is a Lorentz transformation, and of course does not preserve the cone structure of $\mathbf{X}$. This case shows that we cannot directly apply the usual Runge-Kutta method to the differential equations which have special structures, and attempt to retain those structures without extra effort, because it is not designed to preserve these properties.

According to the method by Munthes-Kaas (1998, 1999), we can construct the third-order GPS, namely MKRK3:

$$\mathbf{K}_1 = \mathbf{A}(t_n, \mathbf{X}_n), \tag{41}$$

$$\mathbf{K}_2 = \mathbf{A}(t_n + \tau, \exp(\tau \mathbf{K}_1)\mathbf{X}_n), \tag{42}$$

$$\mathbf{K}_3 = \mathbf{A}(t_n + 1.5\tau, \exp(1.5\tau \mathbf{K}_2)\mathbf{X}_n), \tag{43}$$

$$\mathbf{K}_a = \frac{h}{9}(2\mathbf{K}_1 + 3\mathbf{K}_2 + 4\mathbf{K}_3), \tag{44}$$

$$\mathbf{X}_{n+1} = \exp\left(\mathbf{K}_a + \frac{h}{9}[\mathbf{K}_a, \mathbf{K}_1]\right)\mathbf{X}_n, \tag{45}$$

where $[\mathbf{K}_a, \mathbf{K}_1]$ denotes the commutator of $\mathbf{K}_a$ and $\mathbf{K}_1$. In the above construction we have derived three Lie algebra elements $\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3 \in so(k,1)$ and then map the augmented state $\mathbf{X}_n$ into the cone through three Lie group mappings $\exp(\tau \mathbf{K}_1)$, $\exp(1.5\tau \mathbf{K}_2)$, $\exp\left(\mathbf{K}_a + \frac{h}{9}[\mathbf{K}_a, \mathbf{K}_1]\right) \in SO_o(k,1)$. It can be seen that this construction is carried out in the space of $\mathbf{X}$, and its process is rather complex. In Sections 3.3-3.5 we will construct the higher-order GPS in the space of $\mathbf{x}$ directly with the help of Eq. (26). At there one can see that the new methods are simple and effective.

3.2 The second-order GPS

Below, we give a Lie algebra setting to construct the second-order GPS. We can apply the fourth-order Runge-Kutta method (RK4) to Eq. (2) by the following formula:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{F}_n, \tag{46}$$

where

$$\mathbf{F}_n = \frac{1}{6}[\mathbf{f}_1 + 2\mathbf{f}_2 + 2\mathbf{f}_3 + \mathbf{f}_4] \tag{47}$$

is a weighting average of the following vector fields:

$$\mathbf{f}_1 = \mathbf{f}(t_n, \mathbf{x}_n), \tag{48}$$

$$\mathbf{f}_2 = \mathbf{f}(t_n + \tau, \mathbf{x}_n + \tau \mathbf{f}_1), \tag{49}$$

$$\mathbf{f}_3 = \mathbf{f}(t_n + \tau, \mathbf{x}_n + \tau \mathbf{f}_2), \tag{50}$$

$$\mathbf{f}_4 = \mathbf{f}(t_n + h, \mathbf{x}_n + h\mathbf{f}_3). \tag{51}$$

Then, we can combine these two advantages of the accuracy by the fourth-order Runge-Kutta scheme and the preservation of $\|\mathbf{X}_n\|$ by the group preserving scheme into a single program with a calculation from a given $\mathbf{x}_n$ to a tentative new value of $\mathbf{x}(n+1)$ by the fourth-order Runge-Kutta scheme. Inserting the average $\mathbf{x}_{\bar{n}} = [\mathbf{x}_n + \mathbf{x}(n+1)]/2$ into Eq. (17) or Eq. (21) to calculate $\mathbf{f}_{\bar{n}}$, we can calculate $\mathbf{G}$ at the $\bar{n} = n+1/2$-th time step. Then, we can calculate the next time step $\mathbf{x}_{n+1}$ by the following group preserving schemes:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta^c_{\bar{n}} \mathbf{f}_{\bar{n}} = \mathbf{x}_n + \frac{4h\|\mathbf{x}_{\bar{n}}\|\|\mathbf{x}_n\| + 2h^2 \mathbf{f}_{\bar{n}} \cdot \mathbf{x}_n}{4\|\mathbf{x}_{\bar{n}}\|^2 - h^2\|\mathbf{f}_{\bar{n}}\|^2} \mathbf{f}_{\bar{n}}, \tag{52}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta^e_{\bar{n}} \mathbf{f}_{\bar{n}} = \mathbf{x}_n + \frac{b_{\bar{n}}\|\mathbf{x}_n\|\|\mathbf{f}_{\bar{n}}\| + (a_{\bar{n}} - 1)\mathbf{f}_{\bar{n}} \cdot \mathbf{x}_n}{\|\mathbf{f}_{\bar{n}}\|^2} \mathbf{f}_{\bar{n}}, \tag{53}$$

where $\mathbf{f}_{\bar{n}}$ is calculated from Eq. (48) at the point $(t_{\bar{n}}, \mathbf{x}_{\bar{n}})$, and

$$a_{\bar{n}} := \cosh\left(\frac{h\|\mathbf{f}_{\bar{n}}\|}{\|\mathbf{x}_{\bar{n}}\|}\right), \quad b_{\bar{n}} := \sinh\left(\frac{h\|\mathbf{f}_{\bar{n}}\|}{\|\mathbf{x}_{\bar{n}}\|}\right). \tag{54}$$

Through this modification we can find that the new methods are more accurate.

We have taken advantage of the accuracy of RK4 into a new numerical method, which is realized by a suitable combination of GPS and RK4, namely GPS2. The flow diagram is summarized in Fig. 1. Mathematically speaking, the execution of GPS2 is carried out by a fourth order accurate approximation of the Lie algebra $\mathbf{A}$; however, the group $\mathbf{G}$ obtained from this way is only accurate in the two orders. According to this point we say that the GPS2 numerical method has a second-order accuracy, or better.

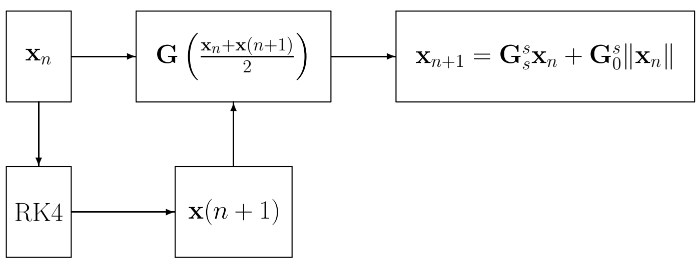

Figure 1: Instead of using GPS or RK4 alone to calculate $\mathbf{x}_{n+1}$ from $\mathbf{x}_n$, we calculate $\mathbf{x}_{n+1}$ by a combination of GPS and RK4 to name the GPS2.

3.3 The direct construction of GPS3 in the space of $\mathbf{x}$

First, we directly apply the RK3 to Eq. (2) by the following formula:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{F}_n, \tag{55}$$

where

$$\mathbf{F}_n = \frac{1}{9}[2\mathbf{f}_1 + 3\mathbf{f}_2 + 4\mathbf{f}_3] \tag{56}$$

is a weighting average of the following vector fields:

$$\mathbf{f}_1 = \mathbf{f}(t_n, \mathbf{x}_n), \tag{57}$$

$$\mathbf{f}_2 = \mathbf{f}(t_n + \tau, \mathbf{x}_n + \tau\mathbf{f}_1), \tag{58}$$

$$\mathbf{f}_3 = \mathbf{f}(t_n + 1.5\tau, \mathbf{x}_n + 1.5\tau\mathbf{f}_2). \tag{59}$$

From Eqs. (55)-(59) we can see that the RK3 obtaining its formulation is through the calculations of the three vector fields of $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$ at three points $\mathbf{x}_n^1 = \mathbf{x}_n$, $\mathbf{x}_n^2 = \mathbf{x}_n + \tau\mathbf{f}_1$, and $\mathbf{x}_n^3 = \mathbf{x}_n + 1.5\tau\mathbf{f}_2$, and then with a suitable weighting average of these three contributions made by the three vector fields of $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3$.

This prompts us to use the GPS on these points which guarantees that the three argumented points $(\mathbf{x}_n^i, \|\mathbf{x}_n^i\|)$, $i = 1, 2, 3$, are located on the cone and that at each point we can generate a Lie group mapping satisfying Eqs. (11)-(13).

We only consider the mapping generated from the Cayley transformation as that similar to Eq. (52), which will be called the GPS3. As demonstrated by Eq. (26) this can be furnished by the mapping (18) but replaced $\mathbf{f}_n$ by the following vector fields:

$$\mathbf{f}_1 = \mathbf{f}(t_n, \mathbf{x}_n^1), \tag{60}$$

$$\mathbf{f}_2 = \mathbf{f}(t_n + \tau, \mathbf{x}_n^2), \tag{61}$$

$$\mathbf{f}_3 = \mathbf{f}(t_n + 1.5\tau, \mathbf{x}_n^3), \tag{62}$$

where

$$\mathbf{x}_n^1 = \mathbf{x}_n, \tag{63}$$

$$\mathbf{x}_n^2 = \mathbf{x}_n + \eta_1^c \mathbf{f}_1, \tag{64}$$

$$\mathbf{x}_n^3 = \mathbf{x}_n + \eta_2^c \mathbf{f}_2, \tag{65}$$

and the two adaptive factors are given by

$$\eta_1^c = \frac{2\tau^2 \mathbf{f}_1 \cdot \mathbf{x}_n + 4\tau\|\mathbf{x}_n^1\|\|\mathbf{x}_n\|}{4\|\mathbf{x}_n^1\|^2 - \tau^2\|\mathbf{f}_1\|^2}, \tag{66}$$

$$\eta_2^c = \frac{3\tau^2 \mathbf{f}_2 \cdot \mathbf{x}_n + 6\tau\|\mathbf{x}_n^2\|\|\mathbf{x}_n\|}{4\|\mathbf{x}_n^2\|^2 - \tau^2\|\mathbf{f}_2\|^2}. \tag{67}$$

Then, let

$$\mathbf{F}_n = \frac{1}{9}[2\mathbf{f}_1 + 3\mathbf{f}_2 + 4\mathbf{f}_3], \tag{68}$$

and inserting the above $\mathbf{F}_n$ into Eq. (55) we get a third-order numerical integrating method, which is named the GPS3.

The last step to obtain the group preserving scheme on the cone is obtained by projecting the point $(\mathbf{x}_{n+1}, \|\mathbf{x}_{n+1}\|)$ into the cone:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta_3^c \mathbf{F}_n, \tag{69}$$

where

$$\eta_3^c = \frac{4h\|\mathbf{x}_n\|^2 + 2h^2 \mathbf{F}_n \cdot \mathbf{x}_n}{4\|\mathbf{x}_n\|^2 - h^2\|\mathbf{F}_n\|^2}. \tag{70}$$

This integrating method is called the Cone-GPS3. The above constructions of the GPS3 and the Cone-GPS3 are simpler than that given in Section 3.1.

3.4 The Cayley fourth-order GPS

By respecting Eqs. (46)-(51) we can see that the RK4 increasing its accuracy is through the fourfold calculations of the four vector fields $\mathbf{f}_1, \ldots, \mathbf{f}_4$ at four points $\mathbf{x}_n^1 = \mathbf{x}_n$, $\mathbf{x}_n^2 = \mathbf{x}_n + \tau \mathbf{f}_1$, $\mathbf{x}_n^3 = \mathbf{x}_n + \tau \mathbf{f}_2$ and $\mathbf{x}_n^4 = \mathbf{x}_n + h\mathbf{f}_3$, and then with a suitable weighting average of these four contributions made by the four vector fields of $\mathbf{f}_1, \ldots, \mathbf{f}_4$.

This prompts us to use the GPS on these points which guarantees that the four argumented points $(\mathbf{x}_n^i, \|\mathbf{x}_n^i\|)$, $i = 1, \ldots, 4$, are located on the cone and that at each point we can generate a Lie group mapping satisfying Eqs. (11)-(13).

We first consider the mapping generated from the Cayley transformation, which will be called the Cayley fourth-order GPS (GPS4) with the mapping from $\mathbf{x}_n$ to $\mathbf{x}_{n+1}$ also calculated by Eq. (46) but with the following vector fields:

$$\mathbf{f}_1 = \mathbf{f}(t_n, \mathbf{x}_n), \tag{71}$$

$$\mathbf{f}_2 = \mathbf{f}(t_n + \tau, \mathbf{x}_n + \eta_1^c \mathbf{f}_1), \tag{72}$$

$$\mathbf{f}_3 = \mathbf{f}(t_n + \tau, \mathbf{x}_n + \eta_2^c \mathbf{f}_2), \tag{73}$$

$$\mathbf{f}_4 = \mathbf{f}(t_n + h, \mathbf{x}_n + \eta_3^c \mathbf{f}_3), \tag{74}$$

where the three adaptive factors are given by

$$\eta_1^c = \frac{2\tau^2 \mathbf{f}_1 \cdot \mathbf{x}_n + 4\tau \|\mathbf{x}_n^1\| \|\mathbf{x}_n\|}{4\|\mathbf{x}_n^1\|^2 - \tau^2 \|\mathbf{f}_1\|^2}, \tag{75}$$

$$\eta_2^c = \frac{2\tau^2 \mathbf{f}_2 \cdot \mathbf{x}_n + 4\tau \|\mathbf{x}_n^2\| \|\mathbf{x}_n\|}{4\|\mathbf{x}_n^2\|^2 - \tau^2 \|\mathbf{f}_2\|^2}, \tag{76}$$

$$\eta_3^c = \frac{4\tau^2 \mathbf{f}_3 \cdot \mathbf{x}_n + 8\tau \|\mathbf{x}_n^3\| \|\mathbf{x}_n\|}{4\|\mathbf{x}_n^3\|^2 - \tau^2 \|\mathbf{f}_3\|^2}. \tag{77}$$

In the GPS4 we have calculated the four vector fields $\mathbf{f}_1, \ldots, \mathbf{f}_4$ at the following four points:

$$\mathbf{x}_n^1 = \mathbf{x}_n, \tag{78}$$

$$\mathbf{x}_n^2 = \mathbf{x}_n + \eta_1^c \mathbf{f}_1, \tag{79}$$

$$\mathbf{x}_n^3 = \mathbf{x}_n + \eta_2^c \mathbf{f}_2, \tag{80}$$

$$\mathbf{x}_n^4 = \mathbf{x}_n + \eta_3^c \mathbf{f}_3. \tag{81}$$

Upon comparing with the RK4, it can be seen that the new method is different on the calculation of the four vector fields $\mathbf{f}_1, \ldots, \mathbf{f}_4$.

Now let

$$\mathbf{F}_n = \frac{1}{6}[\mathbf{f}_1 + 2\mathbf{f}_2 + 2\mathbf{f}_3 + \mathbf{f}_4]. \tag{82}$$

Then the mapping in Eq. (46) is simply of the Euler type, but with a more accurate vector field. Inserting the above $\mathbf{F}_n$ into Eq. (46) we get a fourth-order numerical integrating method, which is named the GPS4.

The last step to obtain the group preserving scheme on the cone is obtained by projecting the point $(\mathbf{x}_{n+1}, \|\mathbf{x}_{n+1}\|)$ into the cone:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta_4^c \mathbf{F}_n, \tag{83}$$

where

$$\eta_4^c = \frac{4h\|\mathbf{x}_n\|^2 + 2h^2 \mathbf{F}_n \cdot \mathbf{x}_n}{4\|\mathbf{x}_n\|^2 - h^2 \|\mathbf{F}_n\|^2}. \tag{84}$$

This integrating method is called the Cone-GPS4.

3.5 The exponential fourth-order GPS

Next, we consider the mapping generated from the exponential transformation, which is also called the Cone-GPS4:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \eta_4^e \mathbf{F}_n = \mathbf{x}_n + \frac{(a_n - 1)\mathbf{F}_n \cdot \mathbf{x}_n + b_n \|\mathbf{x}_n\| \|\mathbf{F}_n\|}{\|\mathbf{F}_n\|^2} \mathbf{F}_n, \tag{85}$$

where

$$a_n := \cosh\left(\frac{h\|\mathbf{F}_n\|}{\|\mathbf{x}_n\|}\right), \quad b_n := \sinh\left(\frac{h\|\mathbf{F}_n\|}{\|\mathbf{x}_n\|}\right), \tag{86}$$

and $\mathbf{F}_n$ is still given by Eq. (82) but with

$$\mathbf{f}_1 = \mathbf{f}(t_n, \mathbf{x}_n), \tag{87}$$

$$\mathbf{f}_2 = \mathbf{f}(t_n + \tau, \mathbf{x}_n + \eta_1^e \mathbf{f}_1), \tag{88}$$

$$\mathbf{f}_3 = \mathbf{f}(t_n + \tau, \mathbf{x}_n + \eta_2^e \mathbf{f}_2), \tag{89}$$

$$\mathbf{f}_4 = \mathbf{f}(t_n + h, \mathbf{x}_n + \eta_3^e \mathbf{f}_3). \tag{90}$$

The three adaptive factors are given by

$$\eta_1^e = \frac{(a_1 - 1)\mathbf{f}_1 \cdot \mathbf{x}_n + b_1 \|\mathbf{f}_1\| \|\mathbf{x}_n\|}{\|\mathbf{f}_1\|^2}, \tag{91}$$

$$\eta_2^e = \frac{(a_2 - 1)\mathbf{f}_2 \cdot \mathbf{x}_n + b_2 \|\mathbf{f}_2\| \|\mathbf{x}_n\|}{\|\mathbf{f}_2\|^2}, \tag{92}$$

$$\eta_3^e = \frac{2(a_3 - 1)\mathbf{f}_3 \cdot \mathbf{x}_n + 2b_3 \|\mathbf{f}_3\| \|\mathbf{x}_n\|}{\|\mathbf{f}_3\|^2} \tag{93}$$

with

$$a_1 := \cosh\left(\frac{\tau\|\mathbf{f}_1\|}{\|\mathbf{x}_n^1\|}\right), \quad b_1 := \sinh\left(\frac{\tau\|\mathbf{f}_1\|}{\|\mathbf{x}_n^1\|}\right), \tag{94}$$

$$a_2 := \cosh\left(\frac{\tau\|\mathbf{f}_2\|}{\|\mathbf{x}_n^2\|}\right), \quad b_2 := \sinh\left(\frac{\tau\|\mathbf{f}_2\|}{\|\mathbf{x}_n^2\|}\right), \tag{95}$$

$$a_3 := \cosh\left(\frac{\tau\|\mathbf{f}_3\|}{\|\mathbf{x}_n^3\|}\right), \quad b_3 := \sinh\left(\frac{\tau\|\mathbf{f}_3\|}{\|\mathbf{x}_n^3\|}\right). \tag{96}$$

Similarly, we can also get the GPS4 by inserting the above $\mathbf{F}_n$ into Eq. (46).

4 Numerical examples

In order to assess the performance of the newly developed schemes let us investigate the following examples.

4.1 Example 1

Let us consider the following periodic system:

$$\dot{x}_1 = x_2, \;\; x_1(0) = 0.0, \tag{97}$$

$$\dot{x}_2 = -2.25x_1 - (x_1 - 1.5\sin t)^3 + 2\sin t, \;\; x_2(0) = 1.59929. \tag{98}$$

The exact solutions are

$$x_1(t) = 1.59941\sin t - 0.00004\sin 3t, \tag{99}$$

$$x_2(t) = 1.59941\cos t - 0.00012\cos 3t. \tag{100}$$

In Fig. 2 we compare the numerical errors obtained by applying the GPS, GPS2, RK4 and GPS4 on the above system with a stepsize $h = 10^{-3}$ sec. Under this small stepsize, all the GPS2, GPS4 and RK4 have the same accuracy in the orders of $10^{-9} - 10^{-5}$; however, the GPS gives slightly inaccurate results with the errors in the orders of $10^{-6} - 10^{-2}$. Then we use a stepsize $h = 0.1$ sec in Fig. 3. Under this stepsize the Cayley type GPS in Section 2.1 is not applicable due to numerical instability; the RK4 has the accuracy in the orders of $10^{-8} - 10^{-5}$, the GPS3 has the accuracy in the orders of $10^{-8} - 10^{-5}$, the GPS4 has the accuracy in the orders of $10^{-8} - 10^{-4}$, and the GPS2 is worse with the accuracy in the orders of $10^{-5} - 10^{-3}$. For this case it is interesting that the GPS3 is accurate than the GPS4.

For this example, Zhang and Deng (2004) have calculated it by the Magnus type Lie integrator. Under the same stepsize $h = 0.1$ sec, our numerical errors due to GPS3 and GPS4 are much smaller than that calculated by Zhang and Deng (2004) as shown in Figures 1 and 2 therein. More importantly, our approach to constructing the higher-order GPS is more effective than that using the Magnus expansion as adopted by Zhang and Deng (2004), and other Lie integrators by using the Crouch-Grossman method, the RKMK method and also the Fer method.

4.2 Example 2

There are many physical systems that are subjected to multiple constraints; for example, for the Euler equations of rigid body dynamics:

$$\frac{d}{dt}\begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \end{bmatrix} = \begin{bmatrix} 0 & \frac{\Pi_3}{I_3} & \frac{-\Pi_2}{I_2} \\ \frac{-\Pi_3}{I_3} & 0 & \frac{\Pi_1}{I_1} \\ \frac{\Pi_2}{I_2} & \frac{-\Pi_1}{I_1} & 0 \end{bmatrix} \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \end{bmatrix}, \tag{101}$$

$I_1, I_2, I_3 > 0$ are the three principal moments of inertia of the rigid body, and Π_1, Π_2, Π_3 are the three components of the rigid body angular momentum.

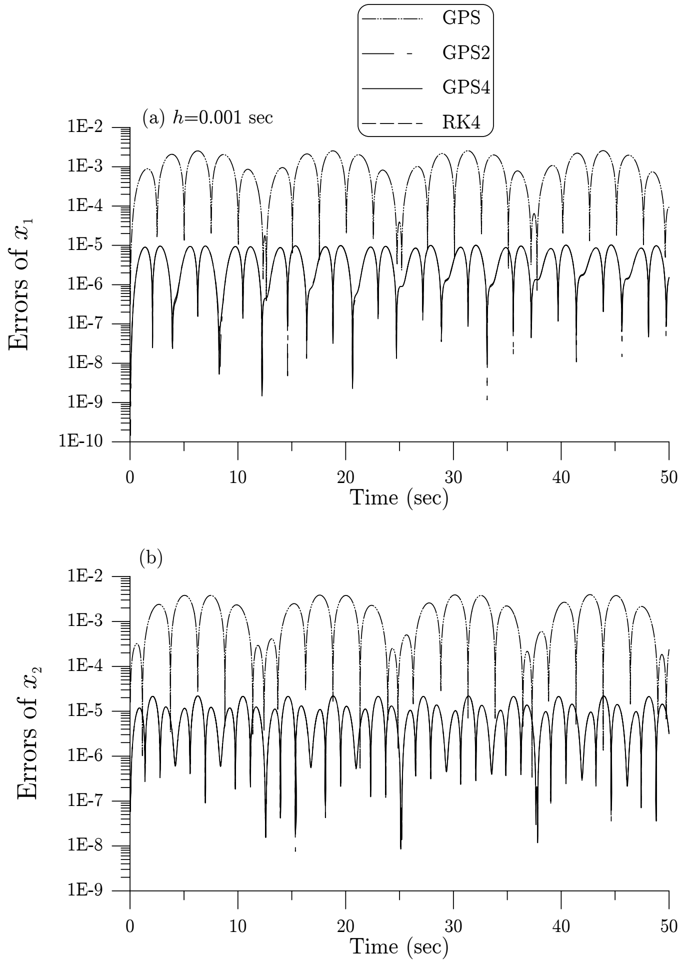

Figure 2: For Example 1 we compare the numerical errors by the numerical methods GPS, GPS2, GPS4 and RK4 with the stepsize $h = 10^{-3}$ sec.

We know that the system of Euler equations possesses two invariants; the first is the momentum, a Casimir function:

$$C := \frac{1}{2}\|\mathbf{\Pi}\|^2, \tag{102}$$

and the second is the energy, a Hamiltonian:

$$H := \frac{1}{2}\mathbf{\Pi} \cdot \mathbf{J}^{-1}\mathbf{\Pi}, \tag{103}$$

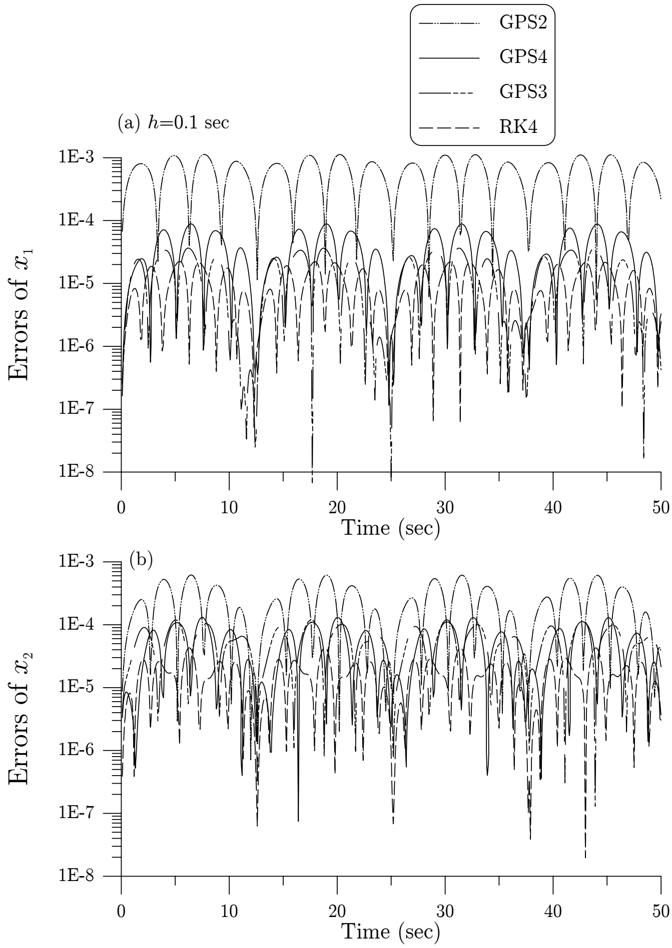

Figure 3: For Example 1 we compare the numerical errors by the numerical methods GPS2, GPS3, GPS4 and RK4 with the stepsize $h = 0.1$ sec.

where $\mathbf{J}$ is the inertia tensor of the body:

$$\mathbf{J} := \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix}. \tag{104}$$

For the special case of $I_1 = I_2 > I_3$, the closed-form solution of the Euler equations is available:

$$\Pi_1(t) = \Pi_1(0)\cos\frac{(I_3-I_1)\Pi_3(0)}{I_1I_3}t - \Pi_2(0)\sin\frac{(I_3-I_1)\Pi_3(0)}{I_1I_3}t, \tag{105}$$

$$\Pi_2(t) = \Pi_2(0)\cos\frac{(I_3-I_1)\Pi_3(0)}{I_1I_3}t + \Pi_1(0)\sin\frac{(I_3-I_1)\Pi_3(0)}{I_1I_3}t, \tag{106}$$

$$\Pi_3(t) = \Pi_3(0). \tag{107}$$

In Fig. 4, the results calculated by using the new schemes with $h = 10^{-3}$ sec were compared with the closed-form solutions. The accuracy of GPS3, GPS4 and RK4 is much better than that of the GPS2. The momentum and energy errors were shown in Fig. 5. It can be seen that the GPS4 provides more accurate results for the energy and momentum than does the RK4. At the same time, the errors of $\Pi_1(t)$ and $\Pi_2(t)$ are also greatly reduced by the GPS3 and GPS4 than the GPS2.

4.3 Example 3

Consider the following two-dimensional stiff ODEs:

$$\dot{x}_1 = 9x_1 + 24x_2 + 5\cos t - \frac{1}{3}\sin t, \;\; x_1(0) = \frac{4}{3}, \tag{108}$$

$$\dot{x}_2 = -24x_1 - 51x_2 - 9\cos t + \frac{1}{3}\sin t, \;\; x_2(0) = \frac{2}{3}, \tag{109}$$

whose solutions are given by

$$x_1(t) = 2e^{-3t} - e^{-39t} + \frac{1}{3}\cos t, \tag{110}$$

$$x_2(t) = -e^{-3t} + 2e^{-39t} - \frac{1}{3}\cos t. \tag{111}$$

We apply RK4 and the cone-GPS4 to Eqs. (108) and (109). The errors are displayed in Fig. 6, where $h = 10^{-4}$ sec was used.

4.4 Example 4

Next we consider a linear stiff system:

$$\dot{x}_1 = -500000.5x_1 + 499999.5x_2, \;\; x_1(0) = 0, \tag{112}$$

$$\dot{x}_2 = 499999.5x_1 - 500000.5x_2, \;\; x_2(0) = 2 \tag{113}$$

in the range of $0 \le t < 2$ sec. The exact solutions are

$$x_1(t) = -\exp(-10^6 t) + \exp(-t), \tag{114}$$

$$x_2(t) = \exp(-10^6 t) + \exp(-t). \tag{115}$$

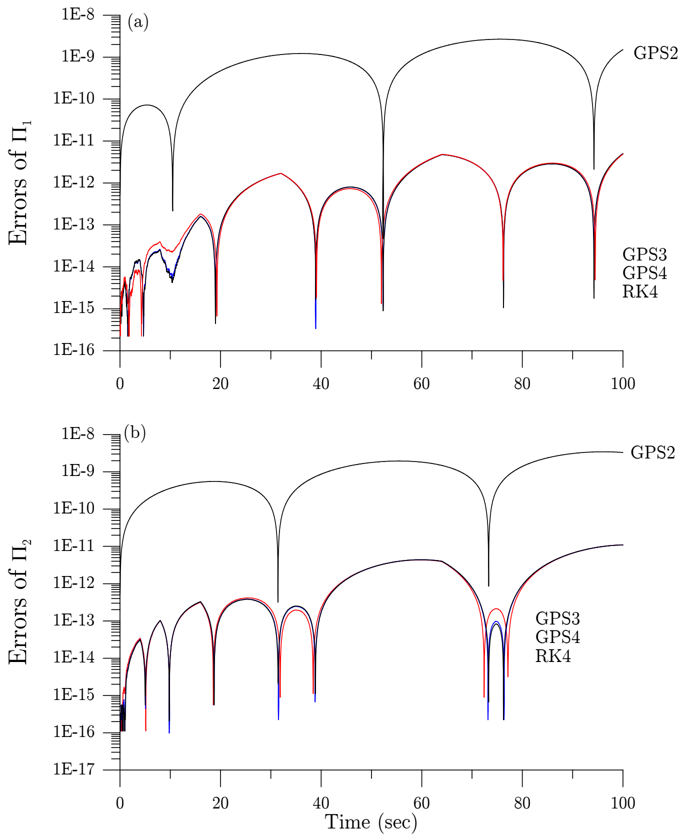

Figure 4: For Example 2 we compare the numerical errors by the numerical methods GPS2, GPS3, GPS4 and RK4 with the stepsize $h = 10^{-3}$ sec.

In the calculations we fix the stepsize to be $h = 10^{-6}$ sec, and apply the RK4 and the cone-GPS4 to Eqs. (112) and (113). The errors are displayed in Fig. 7.

5 Conclusions

The group-preserving schemes developed by Liu (2001) for integrating ODEs were extended to the higher-order numerical schemes by according to the technique used by Runge and Kutta to raise the order of accuracy from the Euler method. The new numerical methods not only preserved the cone structure for any nonlinear dynamical system but also retained the inherent Lie-group behavior. All that based on an important result that the construction of higher-order GPS can be carried out directly in the state space as shown by Eq. (26). The new methods provide us

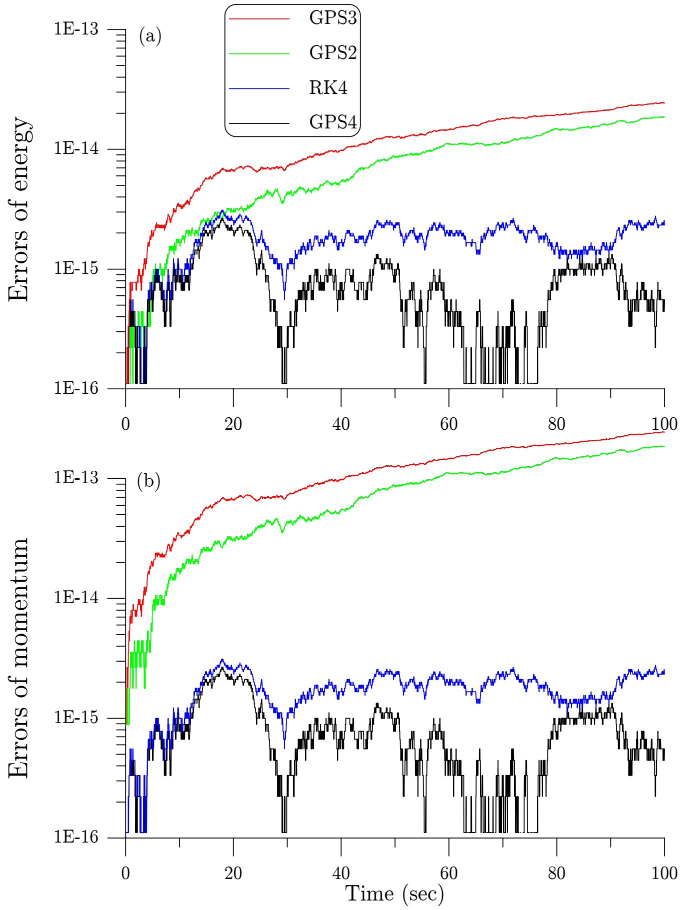

Figure 5: For Example 2 we compare the errors of momentum and energy by the numerical methods GPS2, GPS3, GPS4 and RK4 with the stepsize $h = 10^{-3}$ sec.

single-step explicit time integrators for ODEs, which are proposed by combining the conventional or nonstandard finite difference method and monotonous type of explicit Runge-Kutta method into the group-preserving type obtained by Lee and Liu (2006, 2007,2008) to the higher-order group-preserving numerical methods. They are cheap to run and easy to implement, and the numerical examples show that they have good computational efficiency and high accuracy. More importantly, our approach to constructing the higher-order GPS is more easily to follow than that by using the Magnus expansion, the Crouch-Grossman method, the RKMK method, as well as the Fer method.

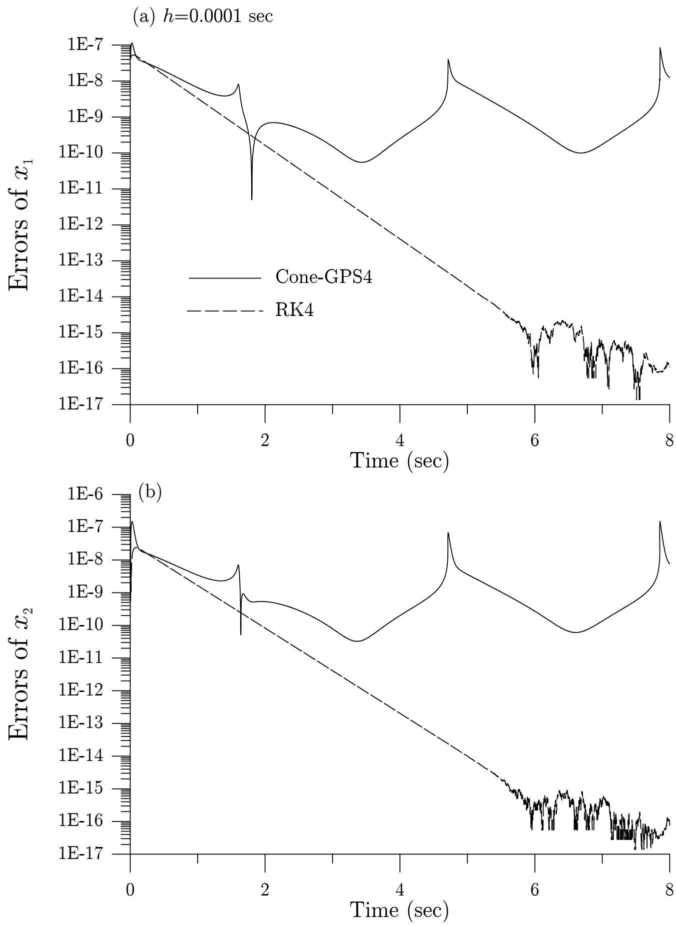

Figure 6: For Example 3 we compare the numerical errors by the numerical methods cone-GPS4 and RK4 with the stepsize $h = 10^{-4}$ sec.

Acknowledgement: The financial supports provided by the National Science Council of Taiwan under the Grant NSC 93-2212-E-168-016 for HC, and NSC-96-2221-E-019-027-MY3 and NSC-97-2221-E-019-009-MY3 for CS are gratefully acknowledged.

References

Arevalo, C.; Campbell, S. L.; Selva, M. (2004): Unitary partitioning in general constraint preserving DAE integrators. *Math. Comp. Model.*, vol. 40, pp. 1273-1284.

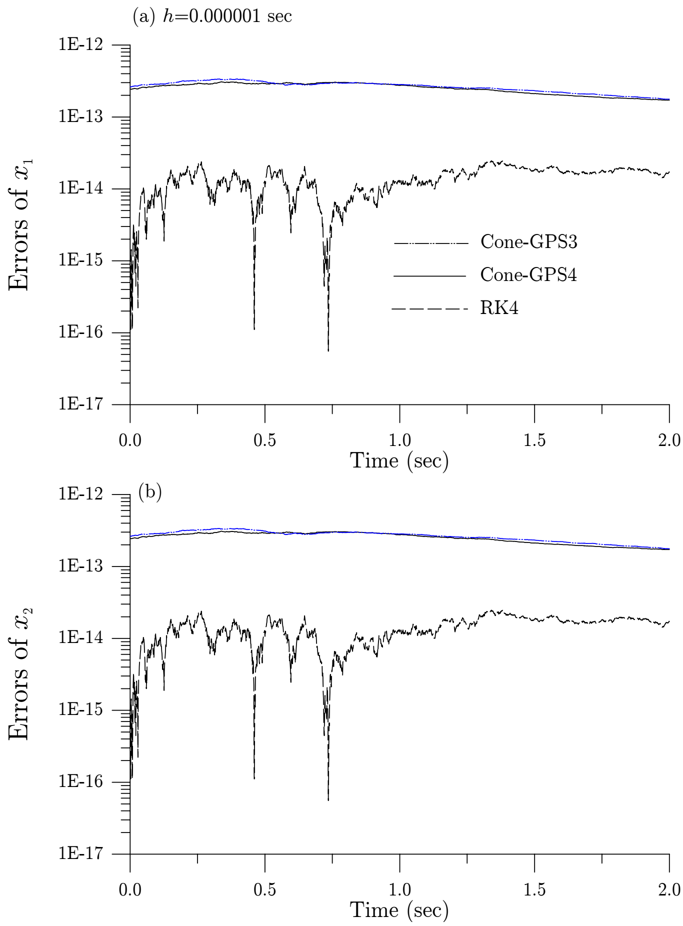

Figure 7: For Example 4 we compare the numerical errors by the numerical methods cone-GPS3, cone-GPS4 and RK4 with the stepsize $h = 10^{-6}$ sec.

Ascher, U. M. (1997): Stabilization of invariants of discretized differential systems. *Num. Algor.*, vol. 14, pp. 1-24.

Ascher, U. M.; Chin, H.; Reich, S. (1994): Stabilization of DAEs and invariant manifolds. *Num. Math.*, vol. 67, pp. 131-149.

Ascher, U. M.; Petzold, L. (1991): Projected implicit Runge-Kutta methods for differential-algebraic equations. *SIAM J. Num. Anal.*, vol. 28, pp. 1097-1120.

Baumgarte, J. (1972): Stabilization of constraints and integrals of motion in dynamical systems. *Comp. Math. Appl. Mech. Eng.*, vol. 1, pp. 1-16.

Campbell, S. L.; Moore, E. (1995): Constraint preserving integrators for general nonlinear higher index DAEs. *Num. Math.*, vol. 69, pp. 383-399.

Chan, R. P. K.; Chartier, P.; Murua, A. (2002): Post-projected Runge-Kutta methods for index-2 differential-algebraic equations. *Appl. Num. Math.*, vol. 42, pp. 77-94.

Chang, C. W.; Liu, C.-S.; Chang, J. R. (2005): A group preserving scheme for inverse heat conduction problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 10, pp. 13-38.

Chen, Y. W.; Liu, C.-S.; Chang, J. R. (2007): A chaos detectable and time step-size adaptive numerical scheme for non-linear dynamical systems. *J. Sound Vib.*, vol. 299, pp. 977-989.

Führer, C.; Leimkuhler, B. J. (1991): Numerical solution of differential-algebraic equations for constrained mechanical motion. *Num. Math.*, vol. 59, pp. 55-69.

Hairer, E.; Lubich, C.; Wanner, G. (2002): Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations. Springer-Verlag, Berlin.

Iserles, A. (1984): Solving linear ordinary differential equations by exponentials of iterated commutators. *Numer. Math.*, vol. 45, pp. 183-199.

Iserles, A.; Norsett, S. P. (1999): On the solution of linear differential equations in Lie group. *Phil. Trans. Roy. Soc. Lond. A*, vol. 357, pp. 983-1019.

Iserles, A.; Munthe-Kaas, H. Z.; Norsett, S. P.; Zanna, A. (2000): Lie-group methods. *Acta Numerica*, pp. 215-365.

Lee, H.-C.; Chen, C.-K.; Hung, C.-I. (2002): A modified group-preserving scheme for solving the initial value problems of stiff ordinary differential equations. *Appl. Math. Comp.*, vol. 133, pp. 445-459.

Lee, H.-C.; Liu, C.-S. (2006): A new group-preserving scheme for solving the initial value problems of stiff ordinary differential equations. Proceedings of the National Conference on Mechanical Engineering, The Chinese Society of Mechanical Engineers, Tainan, Taiwan. pp. 339-344.

Lee, H.-C.; Liu, C.-S. (2007): A new group-preserving scheme for nonlinear dynamic systems. Proceedings of the National Conference on Mechanical Engineering, The Chinese Society of Mechanical Engineers, Chungli, Taiwan, pp. 1618-1623.

Lee, H.-C.; Liu, C.-S. (2008): A new group-preserving scheme and its application to vibrating systems. Proceedings of the National Conference on Mechanical Engineering, The Chinese Society of Mechanical Engineers, Changhua, Taiwan, paper No. B10-20.

Liu, C.-S. (2001): Cone of non-linear dynamical system and group preserving schemes. *Int. J. Non-Linear Mech.*, vol. 36, pp. 1047-1068.

Liu, C.-S. (2005): Nonstandard group-preserving schemes for very stiff ordinary differential equations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 9, pp. 255-272.

Liu, C.-S. (2006a): Preserving constraints of differential equations by numerical methods based on integrating factors. *CMES: Computer Modeling in Engineering & Sciences*, vol. 12, pp. 83-107.

Liu, C.-S. (2006b): An efficient backward group preserving scheme for the backward in time Burgers equation. *CMES: Computer Modeling in Engineering & Sciences*, vol. 12, pp. 55-65.

Liu, C.-S. (2006c): A group preserving scheme for Burgers equation with very large Reynolds number. *CMES: Computer Modeling in Engineering & Sciences*, vol. 12, pp. 197-211.

Liu, C.-S. (2006d): The Lie-group shooting method for nonlinear two-point boundary value problems exhibiting multiple solutions. *CMES: Computer Modeling in Engineering & Sciences*, vol. 13, pp. 149-163.

Liu, C.-S. (2006e): Efficient shooting methods for the second order ordinary differential equations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 15, pp. 69-86.

Liu, C.-S. (2006f): The Lie-group shooting method for singularly perturbed two-point boundary value problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 15, pp. 179-196.

Liu, C.-S. (2006g): The computations of large rotation through an index two nilpotent matrix. *CMES: Computer Modeling in Engineering & Sciences*, vol. 16, pp. 157-175.

Liu, C.-S. (2007): New integrating methods for time-varying linear systems and Lie-group computations. *CMES: Computer Modeling in Engineering & Sciences*, vol.20, pp.157-176.

Liu, C.-S.; Chang, C. W.; Chang, J. R. (2006): Past cone dynamics and backward group preserving schemes for backward heat conduction problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 12, pp. 67-81.

Liu, C.-S.; Chang, C. W.; Chang, J. R. (2009): The backward group preserving scheme for 1D backward in time advection-dispersion equation. *Numer. Meth. Partial Diff. Eq.*, on-line available.

Liu, C.-S.; Ku, Y. L. (2005): A combination of group preserving scheme and Runge-Kutta method for the integration of Landau-Lifshitz equation. *CMES: Com-*

puter Modeling in Engineering & Sciences, vol. 9, pp. 151-178.

März, R. (1991): Numerical methods for differential algebraic equations. *Acta Num.*, pp. 141-198.

März, R. (2002): Differential algebraic systems anew. *Appl. Num. Math.*, vol. 42, pp. 315-335.

Munthe-Kaas, H. Z. (1998): Runge-Kutta methods on Lie group. *BIT*, vol. 38, pp. 92-111.

Munthe-Kaas, H. Z. (1999): High order Runge-Kutta methods on manifolds. *Appl. Numer. Math.*, vol. 29, pp. 115-127.

Zhang, S.; Deng, Z. (2004): A simple and efficient fourth-order approximation solution for nonlinear dynamic system. *Mech. Res. Commu.*, vol. 31, pp. 221-228.

Zhang, S.; Deng, Z. (2005): The Geometrical Integration of Nonlinear Dynamical Systems and Applications. Northwestern Polytechnical University, Xi'an, China (in Chinese).

Zhang, S.; Deng, Z. (2006): Group preserving schemes for nonlinear dynamic system based on RKMK methods . *Appl. Math. Comp.*, vol. 175, pp. 497-507.