Nonlinear Dyn
DOI 10.1007/s11071-016-3290-3

ORIGINAL PAPER

A nilpotent algebra approach to Lagrangian mechanics and constrained motion

Aaron D. Schutte

Received: 11 December 2015 / Accepted: 12 December 2016

Abstract Lagrangian mechanics is extended to the so-called nilpotent Taylor algebra $\mathbb{T}$. It is shown that this extension yields a practical computational technique for the evaluation and analysis of the equations of motion of general constrained dynamical systems. The underlying $\mathbb{T}$-algebra utilized herein permits the analysis of constrained dynamical systems without the need for analytical or symbolic differentiations. Instead, the algebra produces the necessary exact derivatives inherently through binary operations, thus permitting the numerical analysis of constrained dynamical systems using only the defining scalar functions (the Lagrangian $\mathcal{L}$ and the imposed constraints). The extension of the Lagrangian framework to the $\mathbb{T}$-algebra is demonstrated analytically for a problem of constrained motion in a central field and numerically for the calculation of Lyapunov exponents of N-pendulum systems.

Keywords Constrained motion · Automatic differentiation · Nilpotent algebra · Lagrangian mechanics · Lyapunov exponents

1 Introduction

The traditional approach to Lagrangian mechanics involves the evaluation of real-valued scalar functions. Within the Lagrangian framework, one is compelled to construct the requisite equations of motion of a mechanical system by first considering the kinematics of the system and then by evaluating the total system kinetic energy T and potential energy U in order to arrive at the Lagrangian $\mathcal{L}$ of the system. Invariably, the process requires one to find the partial derivatives of $\mathcal{L}$ with respect to (1) the generalized coordinates that describe the system state and (2) time. Furthermore, expressions for U may themselves require many differentiation evaluations with respect to the system configuration. This process can become nontractable if a large number of coordinates are involved. In the presence of constraints, the situation is further complicated since derivative information of the constraints may be needed for both the development of the constrained equations of motion and their numerical integration. Implementation of analysis techniques involving system linearization and sensitivity calculation of Lagrangian systems also relies on differentiation and can become quite challenging for complex systems. By extensions of the Lagrangian to the nilpotent Taylor algebra developed herein, it is the goal of this paper to show that the derivation of equations of motion for general constrained dynamical systems and their analyses can be transformed from an analytical to a purely numerical process given knowledge of only the Lagrangian and the constraints on the motion.

Schemes for numerical simulation of dynamical systems typically rely on formulations wherein the system topology is determined before actual implementa-

A. D. Schutte (✉)
The Aerospace Corporation, 2310 E. El Segundo Blvd,
El Segundo, CA 90245-4691, USA
e-mail: aaron.d.schutte@aero.org

Published online: 26 December 2016

tion. This is partly due to the fact that derivatives play a central role in relating the kinematics of a system to its dynamics. Practical numerical implementation of the equations of motion of a constrained dynamical system essentially relies on three different forms of differentiation, which include (1) analytical differentiation by hand, (2) symbolic differentiation, or (3) automatic differentiation. The first option is feasible if all the necessary derivatives can be obtained by hand prior to implementation. However, the process must be repeated with changing system topology and is error prone for complex systems. Symbolic differentiation utilizes a computer algebra system to accomplish the same feat in a more automated fashion, and as a result is less error prone. Both of these forms of differentiation lead to algorithms that rely on concrete analytical expressions. In contrast, automatic differentiation obtains derivatives of functions numerically. Common automatic differentiation methods apply a judicious use of the chain rule to evaluate the derivative of a given function by source code transformation or operator overloading [1,2]. This assumes that the function of interest is suitably differentiable and can be implemented in a particular programming language. These methods fall into the classification of forward or reverse mode techniques, where forward and reverse are used to indicate the direction traversed by the chain rule. Software implementation of these techniques is widely available [3]. Another popular form of numerical differentiation includes the complex-step derivative [4–6], which has been used to numerically compute first- and second-order derivatives. However, this approach is based on the finite difference approximation and is subject to truncation errors, making it a generally unfeasible option for the numerical development and analysis of equations of motion for constrained systems.

Yet another approach to automatic differentiation involves the use of the so-called dual number algebra. The dual numbers, which can be traced back to the work of Clifford [7], have been shown to produce exact first derivatives of real-valued functions by simply extending the function to the dual number algebra. In mechanics, dual numbers have been primarily used in kinematics analysis; e.g., see [8]. A dual number extension to second-order derivatives has recently been developed [9], which was successfully applied to a Navier–Stokes solver. In what follows, a superset of the dual numbers called the Taylor numbers is constructed. It is shown that the Taylor algebra $\mathbb{T}$ can be used to produce derivatives to arbitrary order of continuous real-valued multivariate functions. The Jacobian and Hessian numbers are constructed from truncated $\mathbb{T}$-algebras and used in the description of Lagrangian mechanics. More importantly, it is demonstrated how the algebras are used to numerically evaluate the elements needed to formulate general constrained equations of motion for dynamical systems and to perform linearization and sensitivity analyses. Finally, the exact nature of the derivatives produced by the algebras is demonstrated analytically for a problem of constrained motion in a central field and numerically for the calculation of Lyapunov exponents for Lagrangian systems.

2 Nilpotent $\mathbb{J}$, $\mathbb{H}$, and $\mathbb{T}$ algebras

Consider a commutative ring R. By definition, the binary operations of addition (+) and multiplication ($\cdot$) on R satisfy the usual commutative, associative, and distributive axioms. The element $0 \in R$ is the additive identity and $1 \in R$ is the multiplicative identity. While all elements in R have an additive inverse, we do not assume multiplicative inverses. Thus, R is not a field. Elements in R that have a multiplicative inverse are called units. It is also essential to note that an element $\epsilon \in R$ is nilpotent to order k if $\epsilon^k = 0$ for minimal $k \in \mathbb{N}^+$.

Definition 1 Let $R[x_1, \ldots, x_n]$ be a polynomial ring over the real numbers in the n algebraically independent indeterminates $x_1, \ldots, x_n$. $R[x_1, \ldots, x_n]$ is called the set of polynomials over the ring R, and it is commutative since R is commutative.

Definition 2 An ideal $\mathcal{I}$ is a nonempty subset of the commutative ring R such that the following closure properties are satisfied

1. $\forall a, b \in \mathcal{I}$: $a \pm b \in \mathcal{I}$.
2. $\forall a \in \mathcal{I}$ and $\forall b \in R$: $ab \in \mathcal{I}$.

An ideal allows the construction of a quotient ring $R/\mathcal{I}$ (also written as R mod $\mathcal{I}$). This is essentially a new ring with the elements of the ideal $\mathcal{I}$ removed from the ring R. The quotient of a polynomial ring in multiple indeterminates is given by $R[x_1, \ldots, x_n]/\mathcal{I}$. This is an important type of ring since many rings are conveniently expressed as a quotient of polynomial rings. Here, it is assumed the ideal $\mathcal{I} = (\alpha_1, \ldots, \alpha_m)$ is a

set of generators such that $\alpha_i \in R[x_1, \ldots, x_n]$. This allows the construction of the Taylor algebra

$$\mathbb{T}(\epsilon_1, \ldots, \epsilon_n) \simeq R[x_1, \ldots, x_n] / \{x_i x_j \cdots x_m\}_{i,j,m \in \mathbb{N}^+}, \quad (1)$$

where the ideal $\mathcal{I} = \{x_i x_j \cdots x_m\}_{i,j,m \in \mathbb{N}^+}$ is both maximal and nilpotent to order m for $1 \leq i, j, \ldots, m \leq n$. An ideal is nilpotent when $\mathcal{I}^k = \{0\}$ for any $k \in \mathbb{N}^+$. Note that the $\mathbb{T}$-algebra actually depends on the truncation order m. Obscurely, the most germane and interesting property of this algebra lies in its ability to produce exact derivatives to arbitrary order of continuous real-valued multivariate functions. This can be accomplished by simply extending the function of interest to the $\mathbb{T}$–algebra of appropriate truncation order.

The dual numbers are a well-known subset of a truncated $\mathbb{T}$–algebra. Abstractly, the dual numbers are derived from the Jacobian algebra, $\mathbb{J}$, in a single indeterminate ϵ so that

$$\mathbb{J}(\epsilon) \simeq R[x]/\{x^2\}; \quad (2)$$

i.e., $\mathbb{J}(\epsilon)$ is isomorphic to the quotient ring $R[x]/\{x^2\}$. They are commonly written in the form

$$z = a_0 + a_1\epsilon, \quad (3)$$

where $a_0, a_1 \in \mathbb{R}$. Thus, the dual number is a type of Jacobian number. Namely, a Jacobian number with a single indeterminate ϵ. The Jacobian number z in Eq. (3) is then a particular 2-vector (a_0, a_1) over the real numbers, where multiplication must follow the rule $\epsilon^k = 0$ for $k \in \{2, 3, \ldots\}$. Addition and multiplication of the Jacobian numbers are defined by

$$(a_0, a_1) + (b_0, b_1) = (a_0 + b_0, a_1 + b_1), \quad (4)$$

and

$$(a_0, a_1) \cdot (b_0, b_1) = (a_0 b_0, a_0 b_1 + b_0 a_1). \quad (5)$$

Division of two Jacobian numbers is determined by

$$\frac{b_0 + b_1\epsilon}{a_0 + a_1\epsilon} \cdot \frac{a_0 - a_1\epsilon}{a_0 - a_1\epsilon} = \left(\frac{b_0}{a_0}, \frac{b_1 a_0 - b_0 a_1}{a_0^2}\right). \quad (6)$$

Using Eq. (6), it can be verified that the Jacobian number $a_0 + a_1\epsilon$ is a unit for all nonzero a_0. The utility of the Jacobian number becomes obvious when we consider the extension of the map $f : \mathbb{R} \to \mathbb{R}$ to the map $\tilde{f} : \mathbb{J} \to \mathbb{J}$. For sufficiently smooth functions f, this leads to the following evaluation

$$\tilde{f}(z) = \tilde{f}(a_0 + a_1\epsilon) = f(a_0) + f'(a_0)a_1\epsilon. \quad (7)$$

Clearly, Eq. (7) represents a first degree Taylor polynomial. Higher degree terms vanish since higher powers of ϵ are zero. Thus, $\tilde{f}(z)$ returns the evaluated function $f(a_0)$ and information about the first derivative $f'(a_0)$. The exact derivative $f'(a_0)$ is obtained by evaluating $\tilde{f}(a_0 + \epsilon)$. This is one of the most useful properties of the Jacobian algebra.

Extending the Jacobian algebra to higher derivatives yields another subset of a truncated $\mathbb{T}$–algebra, which has a third-order nilpotent element ϵ such that $\epsilon^3 = 0$. In a single indeterminate ϵ, the Hessian algebra, $\mathbb{H}$, is given in the single indeterminate form as

$$\mathbb{H}(\epsilon) \simeq R[x]/\{x^3\}. \quad (8)$$

The Hessian number is given by

$$z = a_0 + a_1\epsilon + a_2\epsilon^2, \quad (9)$$

where $a_0, a_1, a_2 \in \mathbb{R}$. Addition and multiplication of the Hessian number follow the definitions

$$(a_0, a_1, a_2) + (b_0, b_1, b_2) = (a_0 + b_0, a_1 + b_1, a_2 + b_2) \quad (10)$$

and

$$(a_0, a_1, a_2) \cdot (b_0, b_1, b_2) = (a_0 b_0, a_0 b_1 + b_0 a_1, a_0 b_2 + b_0 a_2 + a_1 b_1). \quad (11)$$

An extension of the sufficiently smooth map $f : \mathbb{R} \to \mathbb{R}$ to the map $\tilde{f} : \mathbb{H} \to \mathbb{H}$ then yields

$$\tilde{f}(z) = \tilde{f}(a_0 + a_1\epsilon + a_2\epsilon^2) = f(a_0) + f'(a_0)a_1\epsilon + \left(f'(a_0)a_2 + \frac{f''(a_0)}{2!}a_1^2\right)\epsilon^2. \quad (12)$$

In order to extract the first and second derivatives of f, it is apparent from Eq. (12) that one should evaluate the Hessian function $\tilde{f}(a_0 + \epsilon)$ and multiply the resulting ϵ^2 element by a factor of 2.

Generalizing ϵ to higher powers, the construction of the more general Taylor algebra in single indeterminate form is given as

$$\mathbb{T}(\epsilon) \simeq R[x]/\{x^n\}. \quad (13)$$

The Taylor number is simply

$$z = a_0 + a_1\epsilon + a_2\epsilon^2 + \cdots + a_n\epsilon^n = a_0 + a^{\mathsf{T}}E, \quad (14)$$

where $a_0 \in \mathbb{R}$, $a \in \mathbb{R}^n$, $E = [\epsilon, \epsilon^2, \ldots, \epsilon^n]^{\mathsf{T}}$, and $\epsilon^{n+1} = 0$. The basic operations of addition and multiplication of Taylor numbers are defined by the tuples

$$(a_0, a^{\mathsf{T}}E) + (b_0, b^{\mathsf{T}}E) = \left[a_0 + b_0, (a^{\mathsf{T}} + b^{\mathsf{T}})E\right] \quad (15)$$

and

$$(a_0, a^{\mathsf{T}}E)\cdot(b_0, b^{\mathsf{T}}E) = \left[a_0 b_0, \left(a_0 b^{\mathsf{T}} + b_0 a^{\mathsf{T}}\right)E, a^{\mathsf{T}}EE^{\mathsf{T}}b\right]. \quad (16)$$

In Eq. (16), it can be shown that

$$EE^{\mathsf{T}} = \begin{bmatrix} \epsilon^2 & \epsilon^3 & \epsilon^4 & \cdots & \epsilon^n & 0 \\ \epsilon^3 & \iddots & & \epsilon^n & \iddots & \vdots \\ \epsilon^4 & & \iddots & \iddots & & \vdots \\ \vdots & \epsilon^n & \iddots & & & \vdots \\ \epsilon^n & \iddots & & & & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & 0 \end{bmatrix}. \quad (17)$$

Once again, for sufficiently smooth real-valued functions $f : \mathbb{R} \to \mathbb{R}$, the map $\tilde{f} : \mathbb{T} \to \mathbb{T}$ is generalized to an nth-order Taylor series expansion in ϵ as

$$\tilde{f}(z) = \tilde{f}(a_0 + a^{\mathsf{T}}E) = f(a_0) + \sum_{k=1}^{n} \frac{f^{(k)}(a_0)}{k!}\left(a^{\mathsf{T}}E\right)^k, \quad (18)$$

where $\left(a^{\mathsf{T}}E\right)^k$ is the kth-order piece in ϵ. Consequently, the k-th derivative of f is recovered by evaluating $\tilde{f}(a_0+\epsilon)$ and then by multiplying the ϵ^k element by a factor of $k!$.

In multiple indeterminates, the Hessian algebra is given by

$$\mathbb{H}(\epsilon_1, \ldots, \epsilon_n) \simeq R[x_1, \ldots, x_n]/(x_i x_j x_k), \quad 1 \le i, j, k \le n. \quad (19)$$

This algebra has great practical applicability in problems involving multivariate functions $f : \mathbb{R}^n \to \mathbb{R}^m$. For example, consider $z \in \mathbb{H}^n$ of the form

$$z = a + B\epsilon + \begin{bmatrix} \epsilon^{\mathsf{T}}C_1\epsilon \\ \vdots \\ \epsilon^{\mathsf{T}}C_n\epsilon \end{bmatrix}, \quad (20)$$

where $\epsilon = [\epsilon_1, \ldots, \epsilon_n]^{\mathsf{T}}$, $a \in \mathbb{R}^n$, $B \in \mathbb{R}^{n\times n}$, and $C_i \in \mathbb{R}^{n\times n}$ for $i = 1, \ldots, n$. Addition and multiplication of two elements of the vector $z \in \mathbb{H}^n$ are defined as

$$z_1 + z_2 = a_1 + a_2 + (B_1 + B_2)\epsilon + \epsilon^{\mathsf{T}}(C_1 + C_2)\epsilon \quad (21)$$

and

$$z_1 \cdot z_2 = a_1 a_2 + (a_1 B_2 + a_2 B_1)\epsilon + \epsilon^{\mathsf{T}}\left(a_1 C_2 + a_2 C_1 + \frac{1}{2}B_1^{\mathsf{T}}B_2 + \frac{1}{2}B_2^{\mathsf{T}}B_1\right)\epsilon, \quad (22)$$

where B_i is the i-th 1 by n row matrix of B. By extending the sufficiently smooth vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$ to the map $\tilde{f} : \mathbb{H}^n \to \mathbb{H}^m$ we therefore obtain

$$\tilde{f}(z) = f(a) + \mathrm{J}f(a)\epsilon + \begin{bmatrix} \epsilon^{\mathsf{T}}\frac{1}{2!}\mathrm{H}f_1(a)\epsilon \\ \vdots \\ \epsilon^{\mathsf{T}}\frac{1}{2!}\mathrm{H}f_m(a)\epsilon \end{bmatrix}, \quad (23)$$

where z is evaluated at $B = \mathrm{I}_n$ and $C_i = 0$ for $i = 1, \ldots, n$. Thus, the function $\tilde{f}(z)$ in Eq. (23) yields (in a single function evaluation) the real-valued vector function $f(a)$, the real-valued Jacobian $\mathrm{J}f(a)$, and the real-valued Hessians $\mathrm{H}f_1(a), \ldots, \mathrm{H}f_m(a)$.

3 Lagrangian mechanics over the Hessian numbers

Consider a mechanical system with kinetic energy $T(q, \dot{q}, t)$ and potential energy $U(q)$. The Lagrangian of the system is given by the scalar function $\mathcal{L}(q, \dot{q}, t) = T(q, \dot{q}, t) - U(q)$, where $q, \dot{q} \in \mathbb{R}^n$ are the generalized position and velocity vectors and $t \in [0, \infty)$ is the time. Assuming the components of the generalized coordinate vector q are independent from one another, the Lagrange equations are most often applied using the component form

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \Gamma_i, \quad i = 1, \ldots, n, \quad (24)$$

where the n-vector $\Gamma = (\Gamma_1, \ldots, \Gamma_n)^{\mathsf{T}}$ is an arbitrary generalized force vector and n is the number of generalized coordinates. Departing from the standard component form, the Newtonian form of Eq. (24) is recovered by considering the differential of $\mathcal{L}_{\dot{q}} = \partial\mathcal{L}/\partial\dot{q}$ as

$$\mathrm{d}\mathcal{L}_{\dot{q}} = \frac{\partial \mathcal{L}_{\dot{q}}}{\partial q}\mathrm{d}q + \frac{\partial \mathcal{L}_{\dot{q}}}{\partial \dot{q}}\mathrm{d}\dot{q} + \frac{\partial \mathcal{L}_{\dot{q}}}{\partial t}\mathrm{d}t. \quad (25)$$

By taking the differential $\mathrm{d}\mathcal{L}_{\dot{q}}$ with respect to time, the matrix-vector representation of Lagrange's equation of motion is

$$M(q,t)\ddot{q} := \frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\ddot{q} = \Gamma - \frac{\partial^2 \mathcal{L}}{\partial q \partial \dot{q}}\dot{q} - \frac{\partial^2 \mathcal{L}}{\partial t \partial \dot{q}} + \frac{\partial \mathcal{L}}{\partial q} := Q(q, \dot{q}, t), \quad (26)$$

where $M \in \mathbb{R}^{n\times n}$ and $Q \in \mathbb{R}^n$. Equation (26) describes the unconstrained motion of the system, wherein the terminology 'unconstrained' is used to imply that the n

generalized coordinates q are to be treated as independent of one another. The motion of the unconstrained system described by Eq. (26) can be uniquely determined at time t since $M > 0$, and assuming the initial position $q(0) = q_0$ and velocity $\dot{q}(0) = \dot{q}_0$ are provided at initial time t_0.

When the system is subjected to constraints, an additional force of constraint, $Q_c(q, \dot{q}, t)$, arises such that

$$M\ddot{q} = Q + Q_c. \tag{27}$$

Additionally, the mass matrix may only be positive semi-definite ($M \geq 0$) and non-invertible if Lagrange's equation is applied with dependent generalized coordinates. The constraints may have the form

$$\phi_i(q, t) = 0 \quad i = 1, \ldots, s_1, \tag{28}$$

and

$$\psi_i(q, \dot{q}, t) = 0 \quad i = 1, \ldots, s_2, \tag{29}$$

which constitutes a general set of holonomic and nonholonomic bilateral constraints, respectively. The initial conditions q_0 and $\dot{q}_0$ are now assumed to satisfy the constraints in Eqs. (28) and (29) at initial time t_0. The force of constraint Q_c must be devised to ensure that these constraints are satisfied at each instant in time.

Assuming the constraints in Eqs. (28) and (29) are continuously differentiable with respect to time, they can take the form of the constraint matrix equation

$$\begin{aligned} A\ddot{q} &:= \begin{bmatrix} \frac{\partial \phi_1}{\partial q} \\ \vdots \\ \frac{\partial \phi_{s_1}}{\partial q} \\ \frac{\partial \psi_1}{\partial \dot{q}} \\ \vdots \\ \frac{\partial \psi_{s_2}}{\partial \dot{q}} \end{bmatrix} \ddot{q} \\ &= \begin{bmatrix} -\dot{q}^{\mathsf{T}} \frac{\partial^2 \phi_1}{\partial q^2} \dot{q} - 2 \frac{\partial^2 \phi_1}{\partial t \partial q} \dot{q} - \frac{\partial^2 \phi_1}{\partial t^2} \\ \vdots \\ -\dot{q}^{\mathsf{T}} \frac{\partial^2 \phi_{s_1}}{\partial q^2} \dot{q} - 2 \frac{\partial^2 \phi_{s_1}}{\partial t \partial q} \dot{q} - \frac{\partial^2 \phi_{s_1}}{\partial t^2} \\ -\frac{\partial \psi_1}{\partial q} \dot{q} - \frac{\partial \psi_1}{\partial t} \\ \vdots \\ -\frac{\partial \psi_{s_2}}{\partial q} \dot{q} - \frac{\partial \psi_{s_2}}{\partial t} \end{bmatrix} := b, \end{aligned} \tag{30}$$

where A is a matrix with $m = s_1 + s_2$ rows and n columns and b is an m-vector. We note that the set of constraints in Eq. (30) should be consistent ($AA^+b = b$), while the rows of A may be linearly dependent. If the matrix $[M \ A^{\mathsf{T}}]$ has full rank, then the auxiliary mass matrix

$$\mathcal{M} = M + A^{\mathsf{T}} W A > 0 \tag{31}$$

is positive definite with positive definite diagonal weighting matrix

$$W = \mathrm{diag}(w_1, \ldots, w_m) > 0.$$

The weights $w_1, \ldots, w_m$ simply scale the m-constraints relative to the mass matrix M. Under the condition given by Eq. (31), the constrained equations of motion can then become [10–12]

$$\mathcal{M}\ddot{q} = Q + A^{\mathsf{T}}(A\mathcal{M}^{-1}A^{\mathsf{T}})^{+}(b - A\mathcal{M}^{-1}Q), \tag{32}$$

where the superscript $+$ symbol denotes the Moore-Penrose matrix inverse.

Obtaining closed-form expressions of the constrained equations of motion depends on the determination of four elements. Namely, the mass matrix M, the generalized force vector Q, the constraint matrix A, and the constraint vector b. In comparison with other formulations [13], Eq. (32) does not require the use of Lagrange multipliers, which will eventually simplify the Lagrangian evaluations in what follows. It is evident from the expressions above that M, Q, A, and b are constructed using only the partial derivatives of $\mathcal{L}$, ϕ, and ψ with respect to q, $\dot{q}$, and t. Normally, one would derive these expressions and evaluate the derivatives in closed-form before arriving at the required equations of motion. Here, we show that these objects can also be obtained by extending the real-valued Lagrangian map $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ to the Hessian-valued $\mathbb{H}(\epsilon_1, \ldots, \epsilon_n)$ Lagrangian map $\tilde{\mathcal{L}} : \mathbb{H}^n \times \mathbb{H}^n \times \mathbb{H} \to \mathbb{H}$, or equivalently $\tilde{\mathcal{L}} : \mathbb{H}^{2n+1} \to \mathbb{H}$.

Define a generalized state vector $z \in \mathbb{H}^{2n+1}$ so that

$$z = \begin{bmatrix} z_q \\ z_{\dot{q}} \\ z_t \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \\ t \end{bmatrix} + B\epsilon + \begin{bmatrix} \epsilon^{\mathsf{T}} C_1 \epsilon \\ \vdots \\ \epsilon^{\mathsf{T}} C_{2n+1} \epsilon \end{bmatrix}, \tag{33}$$

where

$$\epsilon = [\epsilon_q^{\mathsf{T}}, \epsilon_{\dot{q}}^{\mathsf{T}}, \epsilon_t]^{\mathsf{T}} = [\epsilon_{q_1}, \ldots, \epsilon_{q_n}, \epsilon_{\dot{q}_1}, \ldots, \epsilon_{\dot{q}_n}, \epsilon_t]^{\mathsf{T}},$$

$$B \in \mathbb{R}^{2n+1 \times 2n+1},$$

and

$$C_i \in \mathbb{R}^{2n+1 \times 2n+1}, \quad i = 1, \ldots, 2n+1.$$

Given the Lagrangian of the system $\mathcal{L}(q, \dot{q}, t)$, the extended Hessian-valued Lagrangian is then given by $\tilde{\mathcal{L}}(z)$. At each instant in time, we can evaluate $\tilde{\mathcal{L}}(z)$ at $z = z^*$, where

$$z^* = \begin{bmatrix} q^* + \epsilon_q \\ \dot{q}^* + \epsilon_{\dot{q}} \\ t^* + \epsilon_t \end{bmatrix} \tag{34}$$

is obtained using the values $B = \mathrm{I}_n$ and $C_i = 0$ for $i = 1, \ldots, 2n + 1$. This yields

$$\tilde{\mathcal{L}}(z^*) = \mathcal{L} + \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial q} & \frac{\partial \mathcal{L}}{\partial \dot{q}} & \frac{\partial \mathcal{L}}{\partial t} \end{bmatrix} \epsilon + \epsilon^{\mathsf{T}} \frac{1}{2!} \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial q^2} & \frac{\partial^2 \mathcal{L}}{\partial \dot{q} \partial q} & \frac{\partial^2 \mathcal{L}}{\partial t \partial q} \\ \frac{\partial^2 \mathcal{L}}{\partial q \partial \dot{q}} & \frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2} & \frac{\partial^2 \mathcal{L}}{\partial t \partial \dot{q}} \\ \frac{\partial^2 \mathcal{L}}{\partial q \partial t} & \frac{\partial^2 \mathcal{L}}{\partial \dot{q} \partial t} & \frac{\partial^2 \mathcal{L}}{\partial t^2} \end{bmatrix} \epsilon. \tag{35}$$

The Hessian-valued Lagrangian map $\tilde{\mathcal{L}}(z^*)$ consists of (1) the Lagrangian function evaluation, (2) the gradient of the Lagrangian, and (3) the Hessian matrix of the Lagrangian with respect to q, $\dot{q}$, and t; each evaluated at $q = q^*$, $\dot{q} = \dot{q}^*$, and $t = t^*$. The elements needed to construct the unconstrained equations of motion of the system are obtained directly from Eq. (35) as is apparent in Eq. (26); i.e., Eq. (35) produces the mass matrix M and the generalized force vector Q evaluated at $q = q^*$, $\dot{q} = \dot{q}^*$, and $t = t^*$.

When constraints are imposed, the holonomic constraints, $\phi(q, t)$, and the nonholonomic constraints, $\dot{\psi}(q, \dot{q}, t)$, require second- and first-order derivative information, respectively, as shown in Eq. (30). For $z \in \mathbb{H}^{n+1}$, the i-th Hessian-valued constraint $\tilde{\phi}_i(z)$ evaluated at

$$z^* = \begin{bmatrix} q^* + \epsilon_q \\ t^* + \epsilon_t \end{bmatrix} \tag{36}$$

yields

$$\tilde{\phi}_i(z^*) = \phi_i + \begin{bmatrix} \frac{\partial \phi_i}{\partial q} & \frac{\partial \phi_i}{\partial t} \end{bmatrix} \epsilon + \epsilon^{\mathsf{T}} \frac{1}{2!} \begin{bmatrix} \frac{\partial^2 \phi_i}{\partial q^2} & \frac{\partial^2 \phi_i}{\partial t \partial q} \\ \frac{\partial^2 \phi_i}{\partial q \partial t} & \frac{\partial^2 \phi_i}{\partial t^2} \end{bmatrix} \epsilon. \tag{37}$$

For $z \in \mathbb{J}^{2n+1}$, the i-th Jacobian-valued constraint $\tilde{\psi}_i(z)$ evaluated at

$$z^* = \begin{bmatrix} q^* + \epsilon_q \\ \dot{q}^* + \epsilon_{\dot{q}} \\ t^* + \epsilon_t \end{bmatrix} \tag{38}$$

yields

$$\tilde{\psi}_i(z^*) = \psi_i + \begin{bmatrix} \frac{\partial \psi_i}{\partial q} & \frac{\partial \psi_i}{\partial \dot{q}} & \frac{\partial \psi_i}{\partial t} \end{bmatrix} \epsilon. \tag{39}$$

Equations (37) and (39) then contain all the elements needed to construct the constraint matrix A and the constraint vector b evaluated at $q = q^*$, $\dot{q} = \dot{q}^*$, and $t = t^*$. Observe that the constrained Lagrange equations of motion given by Eq. (32) can then be obtained by pure numerical evaluation of the extended functions $\tilde{\mathcal{L}}(z^*)$, $\tilde{\phi}(z^*)$, and $\tilde{\psi}(z^*)$!

4 Numerical linearization of Lagrangian systems

Given the exact nature of the derivatives produced by the $\mathbb{T}$-algebras, analyses requiring system linearization or sensitivity calculation are easily carried out numerically without truncation error using extended Lagrangian $\tilde{\mathcal{L}}(z^*)$ evaluations. Consider a Lagrangian system with configuration expressible as a bijective function of the minimum number of generalized coordinates

$$\upsilon = \upsilon(q), \tag{40}$$

where $\upsilon \in \mathbb{R}^n$ are the reference coordinates and $q \in \mathbb{R}^n$ are the generalized coordinates. The acceleration of the system (Eq. (32)) is given as

$$\ddot{q} = M^{-1} Q = \left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2} \right)^{-1} \left(\Gamma - \frac{\partial^2 \mathcal{L}}{\partial q \partial \dot{q}} \dot{q} + \frac{\partial \mathcal{L}}{\partial q} \right) \tag{41}$$

since no constraints are imposed on the motion. Differentiating Eq. (40) once with respect to time yields

$$\dot{\upsilon} = \frac{\partial \upsilon}{\partial q} \dot{q} := \Phi(q) \dot{q}, \tag{42}$$

where $\Phi \in \mathbb{R}^{n \times n}$ is nonsingular. A second differentiation with respect to time yields

$$\ddot{\upsilon} = \dot{\Phi} \dot{q} + \Phi \ddot{q}. \tag{43}$$

Substitution of Eq. (41) into Eq. (43) yields the acceleration of the reference coordinates as a function of the generalized coordinates q and $\dot{q}$ as

$$\ddot{\upsilon} = \dot{\Phi} \dot{q} + \Phi M^{-1} Q. \tag{44}$$

A first-order realization of Eq. (44) is given by

$$\dot{x} := \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \ddot{\upsilon} \end{bmatrix}, \tag{45}$$

where $x_1 = \upsilon$, $x_2 = \dot{\upsilon}$, and $x = [x_1^\mathsf{T}, x_2^\mathsf{T}]^\mathsf{T}$. Note that the acceleration $\ddot{\upsilon}$ in Eq. (45) is obtained directly from Eq. (44).

Ignoring any input or output relations for simplicity, a linear model of Eq. (45) is obtained by considering the perturbation $\delta \in \mathbb{R}^{2n}$ about the operating point $x_0 \in \mathbb{R}^{2n}$ such that $x = x_0 + \delta$. The linearization with minimum-sized Jacobian, $\mathcal{A}$, is then given by

$$\dot{\delta} = \left[\begin{array}{cc} 0 & \mathrm{I}_n \\ \frac{\partial \dot{x}_2}{\partial q}\frac{\partial q}{\partial x_1} + \frac{\partial \dot{x}_2}{\partial \dot{q}}\frac{\partial \dot{q}}{\partial x_1} & \frac{\partial \dot{x}_2}{\partial \dot{q}}\frac{\partial \dot{q}}{\partial x_2} \end{array} \right]\Bigg|_{\substack{x=x_0 \\ q=q_0 \\ \dot{q}=\dot{q}_0}}$$

$$\delta := \mathcal{A}\delta, \tag{46}$$

where $\frac{\partial q}{\partial x_2} = \frac{\partial q}{\partial \dot{\upsilon}} = 0$ and $(q_0, \dot{q}_0)$ is the operating point in terms of the generalized coordinates. Clearly, the linearization in Eq. (46) requires (1) the inverse function of Eq. (40)

$$q = q(\upsilon) \tag{47}$$

and (2) the differentiation of the matrices M^{-1} and Q with respect to q and $\dot{q}$. If the quantities M and Q are known analytically, one could theoretically compute $\mathcal{A}$ analytically by using the formula

$$\mathrm{d}M^{-1} = -M^{-1}\mathrm{d}M M^{-1}. \tag{48}$$

For complex Lagrangians $\mathcal{L}$, this process is formidable and most likely not possible. However, numerical calculation of these objects using the $\mathbb{T}$-algebra is trivial. For example, using Eq. (48), the derivative of the inverse mass matrix M^{-1} with respect to q_i in terms of the Lagrangian $\mathcal{L}$ is given by

$$\begin{aligned} \frac{\partial M^{-1}}{\partial q_i} &= -M^{-1}\frac{\partial M}{\partial q_i}M^{-1} \\ &= -\left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1} \frac{\partial^3 \mathcal{L}}{\partial q_i \partial \dot{q}^2}\left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1}. \end{aligned} \tag{49}$$

Since the second relation in Eq. (49) requires third-order derivatives, it is apparent that Eq. (49) can be calculated numerically by extending the Lagrangian $\mathcal{L}$ to a third-order truncated $\mathbb{T}$-algebra (see Sect. 2) evaluation. In this way, the linearization in Eq. (46) can also be calculated numerically and directly from the Lagrangian $\mathcal{L}$. Once the $\mathbb{T}$-algebra has been implemented computationally, it inheres the burden of derivative computation, and the linearization is easily obtained once the Lagrangian $\mathcal{L}$ and the desired minimum coordinate representation $\upsilon(q)$ and its inverse are defined.

5 Implementation and application

Actual computational implementation of a particular truncated $\mathbb{T}$-algebra can be accomplished in software using object-oriented programming techniques such as operator overloading. A class structure can be created that encapsulates the appropriate $\mathbb{T}$-algebra number. For example, the class data definition of an $\mathbb{H}(\epsilon)$-algebra would include three distinct floating-point declarations that represent the scalar, ϵ, and ϵ^2 parts. The methods of the class are then defined to override the binary operations and special functions. In particular, the binary operators of addition (+) and multiplication (*) of two $\mathbb{H}(\epsilon)$ numbers would be coded to follow the relations in Eqs. (10) and (11), while the special functions such as $\sin(z)$ would be coded to follow the evaluation in Eq. (12) as

$$\begin{aligned} \sin(z) = \sin(a + b\epsilon + c\epsilon^2) &= \sin(a) + b\cos(a)\epsilon \\ &+ \left(c\cos(a) - \frac{b^2}{2}\sin(a)\right)\epsilon^2. \end{aligned} \tag{50}$$

In hardware, a field programmable gate array (FPGA) coprocessor could theoretically be designed to implement a particular $\mathbb{T}$-algebra natively. A typical microprocessor (CPU) could then offload numerical derivative computations to the FPGA in parallel.

Once an appropriately truncated $\mathbb{T}$-algebra has been implemented computationally, it is quite trivial to functionally evaluate the extended Lagrangian and motion constraints. The function evaluation using the $\mathbb{T}$-algebra implementation provides all the derivatives needed for the formulation of constrained equations of motion and their linearization as demonstrated in Sects. 3 and 4. This also simplifies sensitivity analyses with respect to many system parameters. On the contrary, using analytical derivatives for sensitivity analyses with respect to arbitrary system parameters is quite difficult, or impossible, for complex systems.

In the following, two simple yet fairly comprehensive examples are provided to demonstrate how the extended Lagrangian and constraint functions can yield the derivatives needed to construct the constrained equations of motion of a Lagrangian system in terms of the matrices M, Q, A, and b, and for the computation of Lyapunov exponents for a given Lagrangian system. Increasingly complex examples can obviously be handled simply by deriving the appropriate Lagrangian and the required constraints.

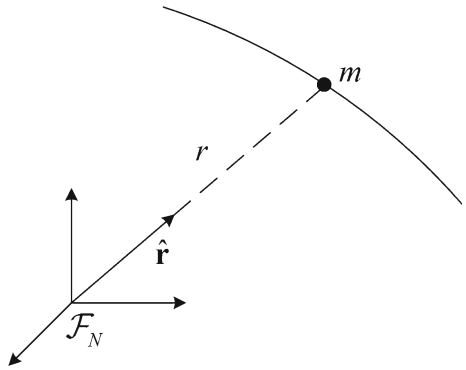

Fig. 1 A particle moving in a central field

5.1 Illustrative example: constrained motion in a central field

Consider a single particle with mass m moving relative to an inertial frame of reference $\mathcal{F}_N$ with orthonormal basis. The position of the particle in the inertial frame is given by the 3-vector $\mathbf{r}$. As illustrated in Fig. 1, the position vector, $\mathbf{r}$, can be decomposed into magnitude and normalized vector form by

$$\mathbf{r} = r\hat{\mathbf{r}}, \tag{51}$$

where $\hat{\mathbf{r}}$ is a unit 3-vector and r is the magnitude of the position vector $\mathbf{r}$. The position of the particle is therefore defined by four coordinates from which three of the coordinates are constrained by the relation

$$\phi(q) = \hat{\mathbf{r}}^{\mathsf{T}}\hat{\mathbf{r}} - 1 = 0, \tag{52}$$

where $q = [r, \hat{\mathbf{r}}^{\mathsf{T}}]^{\mathsf{T}}$ is the generalized coordinate 4-vector describing the configuration of the system. The velocity of the particle is given by

$$\dot{\mathbf{r}} = \frac{\partial \mathbf{r}}{\partial q}\dot{q} := \begin{bmatrix} \hat{\mathbf{r}} & r\mathrm{I}_3 \end{bmatrix}\begin{bmatrix} \dot{r} \\ \dot{\hat{\mathbf{r}}} \end{bmatrix}. \tag{53}$$

The kinetic energy is subsequently derived as

$$T(q,\dot{q}) = \frac{1}{2}m\dot{\mathbf{r}}^{\mathsf{T}}\dot{\mathbf{r}} = \frac{1}{2}m\left(\dot{r}^2 + r^2\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}}\right). \tag{54}$$

Furthermore, we assume the particle is moving in a uniform gravitational field so that we have the potential

$$U(q) = -\frac{\mu m}{r}, \tag{55}$$

where $\mu > 0$. This yields the Lagrangian

$$\mathcal{L}(q,\dot{q}) = \frac{1}{2}m\left(\dot{r}^2 + r^2\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}}\right) + \frac{\mu m}{r}. \tag{56}$$

To evaluate the constrained equations of motion, we need to obtain the elements M, Q, A, and b as described in Sect. 3. This is accomplished by extending the Lagrangian $\mathcal{L}(q,\dot{q}) \to \tilde{\mathcal{L}}(z)$ and the constraint $\phi(q) \to \tilde{\phi}(z)$. The Hessian state vector $z \in \mathbb{H}^8$ is given as

$$z = \begin{bmatrix} z_r \\ z_{\hat{\mathbf{r}}} \\ z_{\dot{r}} \\ z_{\dot{\hat{\mathbf{r}}}} \end{bmatrix} = \begin{bmatrix} r \\ \hat{\mathbf{r}} \\ \dot{r} \\ \dot{\hat{\mathbf{r}}} \end{bmatrix} + B\epsilon + \begin{bmatrix} \epsilon^{\mathsf{T}}C_1\epsilon \\ \vdots \\ \epsilon^{\mathsf{T}}C_8\epsilon \end{bmatrix}, \tag{57}$$

where $\epsilon = [\epsilon_r, \epsilon_{\hat{\mathbf{r}}}^{\mathsf{T}}, \epsilon_{\dot{r}}, \epsilon_{\dot{\hat{\mathbf{r}}}}^{\mathsf{T}}]^{\mathsf{T}} = [\epsilon_1, \ldots, \epsilon_8]^{\mathsf{T}}$, $B \in \mathbb{R}^{8\times 8}$, and $C_i \in \mathbb{R}^{8\times 8}$ for $i = 1, \ldots, 8$. Thus, we can evaluate

$$\tilde{\mathcal{L}}(z) = \frac{1}{2}m\left(z_{\dot{r}}^2 + z_r^2 z_{\dot{\hat{\mathbf{r}}}}^{\mathsf{T}} z_{\dot{\hat{\mathbf{r}}}}\right) + \frac{\mu m}{z_r} \tag{58}$$

at $z = z^*$, where

$$z^* = \begin{bmatrix} r + \epsilon_r \\ \hat{\mathbf{r}} + \epsilon_{\hat{\mathbf{r}}} \\ \dot{r} + \epsilon_{\dot{r}} \\ \dot{\hat{\mathbf{r}}} + \epsilon_{\dot{\hat{\mathbf{r}}}} \end{bmatrix}. \tag{59}$$

By appropriately carrying out the Hessian-based binary operations, we obtain

$$z_{\dot{r}}^2 = \dot{r}^2 + 2\dot{r}\epsilon_{\dot{r}} + \epsilon_{\dot{r}}^2, \tag{60}$$

$$z_r^2 z_{\dot{\hat{\mathbf{r}}}}^{\mathsf{T}} z_{\dot{\hat{\mathbf{r}}}} = r^2\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}} + 2r\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}}\epsilon_r + 2r^2\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\epsilon_{\dot{\hat{\mathbf{r}}}} + \dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}}\epsilon_r^2 + 4r\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\epsilon_r\epsilon_{\dot{\hat{\mathbf{r}}}} + r^2\epsilon_{\dot{\hat{\mathbf{r}}}}^{\mathsf{T}}\epsilon_{\dot{\hat{\mathbf{r}}}}, \tag{61}$$

and

$$\frac{1}{z_r} = \frac{1}{r} - \frac{1}{r^2}\epsilon_r + \frac{1}{r^3}\epsilon_r^2. \tag{62}$$

Combining like terms in ϵ, Eq. (58) then yields

$$\tilde{\mathcal{L}}(z^*) = \frac{1}{2}m\left(\dot{r}^2 + r^2\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}}\right) + \frac{\mu m}{r} + \left[mr\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}} - \frac{\mu m}{r^2}\ 0\ m\dot{r}\ mr^2\dot{\hat{\mathbf{r}}}\right]\epsilon + \epsilon^{\mathsf{T}}\frac{1}{2}\begin{bmatrix} m\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}} + 2\frac{\mu m}{r^3} & 0\ 0 & 2mr\dot{\hat{\mathbf{r}}}^{\mathsf{T}} \\ 0 & 0\ 0 & 0 \\ 0 & 0\ m & 0 \\ 2mr\dot{\hat{\mathbf{r}}} & 0\ 0 & mr^2\mathrm{I}_3 \end{bmatrix}\epsilon. \tag{63}$$

Similarly, the evaluation of

$$\tilde{\phi}(z) = z_{\hat{\mathbf{r}}}^{\mathsf{T}} z_{\hat{\mathbf{r}}} - 1 \tag{64}$$

at $z = z^*$ for

$$z^* = \begin{bmatrix} r + \epsilon_r \\ \hat{\mathbf{r}} + \epsilon_{\hat{\mathbf{r}}} \end{bmatrix} \tag{65}$$

yields

$$\tilde{\phi}(z^*) = \hat{\mathbf{r}}^{\mathsf{T}}\hat{\mathbf{r}} - 1 + \begin{bmatrix} 0 & 2\hat{\mathbf{r}}^{\mathsf{T}} \end{bmatrix}\epsilon + \epsilon^{\mathsf{T}}\frac{1}{2}\begin{bmatrix} 0 & 0 \\ 0 & 2\mathrm{I}_3 \end{bmatrix}\epsilon. \tag{66}$$

In regard to the constrained equations of motion of the system, Eqs. (63) and (66) contain all the needed information. The mass matrix is

$$M = \begin{bmatrix} m & 0 \\ 0 & mr^2 \mathrm{I}_3 \end{bmatrix}, \tag{67}$$

the generalized force vector is

$$Q = \begin{bmatrix} mr\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}} - \dfrac{\mu m}{r^2} \\ -2mr\dot{r}\dot{\hat{\mathbf{r}}} \end{bmatrix}, \tag{68}$$

the constraint matrix is

$$A = \begin{bmatrix} 0 & 2\hat{\mathbf{r}}^{\mathsf{T}} \end{bmatrix}, \tag{69}$$

and the constraint vector (a 1-vector here) is

$$b = -2\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}}. \tag{70}$$

Given a constraint weight $w > 0$, we can compile these elements into Eq. (32) to obtain the constrained acceleration of the system as

$$\begin{bmatrix} \ddot{r} \\ \ddot{\hat{\mathbf{r}}} \end{bmatrix} = \begin{bmatrix} r\left(\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}}\right) - \dfrac{\mu}{r^2} \\ -\dfrac{2\dot{r}}{r}\dot{\hat{\mathbf{r}}} - \left(\dot{\hat{\mathbf{r}}}^{\mathsf{T}}\dot{\hat{\mathbf{r}}}\right)\hat{\mathbf{r}} \end{bmatrix}. \tag{71}$$

Equation (71) is provided analytically for completeness with the understanding that we could also produce this equation at any given instant in time using only the numerical calculations $\tilde{\mathcal{L}}(z^*)$ and $\tilde{\phi}(z^*)$.

5.2 Numerical example: Lyapunov exponents of N-pendulum systems

Lyapunov exponents are quantities used to characterize the asymptotic behavior of a dynamical system. They are commonly used to determine whether the system is sensitive to initial conditions and to help establish chaos. Given an autonomous dynamical system

$$\dot{y} = f(y), \quad y \in \mathbb{R}^n, \quad f : \mathbb{R}^n \to \mathbb{R}^n, \tag{72}$$

the variational equation

$$\dot{Y} = \frac{\partial f}{\partial y} Y, \quad Y \in \mathbb{R}^{n \times n}, \tag{73}$$

can be used to compute the Lyapunov exponents of the system in Eq. (72). The spectrum of Lyapunov exponents of Eq. (72) are defined as

$$\lambda_i = \lim_{t \to \infty} \frac{1}{t} \ln(|m_i|), \quad i = 1, \ldots, n, \tag{74}$$

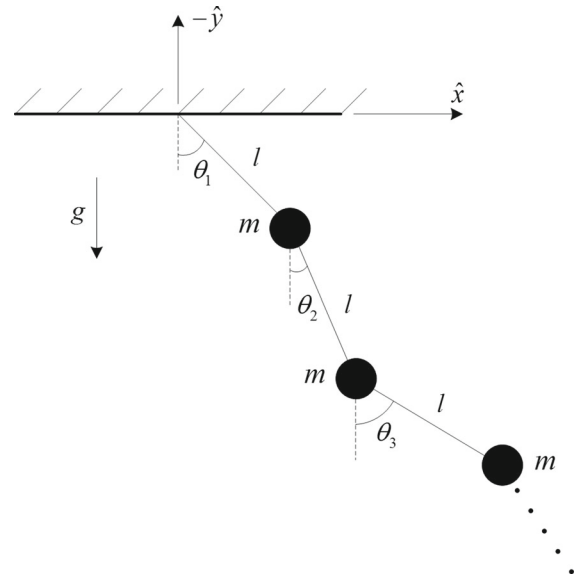

Fig. 2 A planar N-pendulum system

where m_i is the i-th eigenvalue of Y. The largest Lyapunov exponent (usually of most interest) is termed the maximal Lyapunov exponent (MLE). Computation of the n Lyapunov exponents of Eq. (72) can be accomplished by solving the initial value problem [14]

$$\dot{y} = f(y), \quad y(0) = y_0 \tag{75}$$

$$\dot{\mathcal{Q}} = \mathcal{Q}S, \quad \mathcal{Q}(0) = \mathrm{I}_n, \tag{76}$$

$$\dot{\rho}_i = \left(\mathcal{Q}^{\mathsf{T}} \frac{\partial f}{\partial y} \mathcal{Q}\right)_{i,i}, \quad \rho_i(0) = 0, \quad i = 1, \ldots, n, \tag{77}$$

where the n by n matrix $\mathcal{Q}$ is the orthogonal matrix obtained from the QR–decomposition

$$Y = \mathcal{Q}R \tag{78}$$

and the n by n matrix S is defined as

$$S = \begin{cases} \left(\mathcal{Q}^{\mathsf{T}} \dfrac{\partial f}{\partial y} \mathcal{Q}\right)_{i,j}, & i > j \\ 0, & i = j \\ -\left(\mathcal{Q}^{\mathsf{T}} \dfrac{\partial f}{\partial y} \mathcal{Q}\right)_{j,i}, & i < j \end{cases}. \tag{79}$$

The time evolution of the Lyapunov exponent spectrum is then given by

$$\lambda_i(t) = \frac{\rho_i(t)}{t}, \quad i = 1, \ldots, n, \tag{80}$$

and the value of each Lyapunov exponent is given by

$$\lambda_i = \lim_{t \to \infty} \lambda_i(t), \quad i = 1, \ldots, n. \tag{81}$$

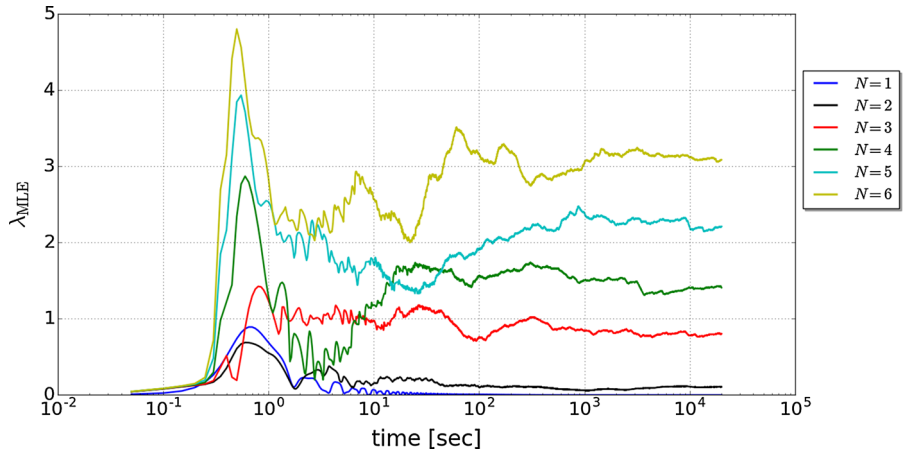

Fig. 3 Time evolution of the N-pendulum MLE

Now consider the MLE calculation of the planar N-pendulum system illustrated in Fig. 2, where the Lyapunov exponent spectrum is $2N$ dimensional. The calculation of the MLE clearly depends on the realizations of the n-vector $f(y)$ and the n by n matrix $\partial f/\partial y$, where the state vector is given by

$$y = (q, \dot{q}) = (\theta_1, \ldots, \theta_N, \dot{\theta}_1, \ldots, \dot{\theta}_N), \quad y \in \mathbb{R}^{2N}. \tag{82}$$

For the N-pendulum system in Fig. 2, f is obtained from Eq. (26) as

$$f = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M^{-1}Q \end{bmatrix} = \begin{bmatrix} \dot{q} \\ \left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1} \left(\frac{\partial \mathcal{L}}{\partial q} - \frac{\partial^2 \mathcal{L}}{\partial q \partial \dot{q}}\dot{q}\right) \end{bmatrix} \tag{83}$$

since no constraints are present and $\mathcal{L}$ has no explicit dependence on time; i.e.,

$$\mathcal{L} = \frac{1}{2}m \sum_{i=1}^{N} \dot{r}_i \cdot \dot{r}_i - mg \sum_{i=1}^{N} h_i, \tag{84}$$

where for $i > 1$

$$\dot{r}_i = \dot{r}_{i-1} + l\dot{\theta}_i[\cos\theta_i, -\sin\theta_i]^{\mathsf{T}} \tag{85}$$

and

$$h_i = h_{i-1} - l\cos\theta_i. \tag{86}$$

The matrix $\partial f/\partial y$ is then given by

$$\frac{\partial f}{\partial y} = \begin{bmatrix} 0 & \mathrm{I}_N \\ \frac{\partial \ddot{q}}{\partial q} & \frac{\partial \ddot{q}}{\partial \dot{q}} \end{bmatrix}, \tag{87}$$

where the i-th column of $\partial \ddot{q}/\partial q$ is

$$\frac{\partial \ddot{q}}{\partial q_i} = -\left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1} \frac{\partial^3 \mathcal{L}}{\partial q_i \partial \dot{q}^2} \left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1} \left(\frac{\partial \mathcal{L}}{\partial q} - \frac{\partial^2 \mathcal{L}}{\partial q \partial \dot{q}}\dot{q}\right) + \left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1} \left(\frac{\partial^2 \mathcal{L}}{\partial q_i \partial q} - \frac{\partial^3 \mathcal{L}}{\partial q_i \partial q \partial \dot{q}}\dot{q}\right) \tag{88}$$

and the i-th column of $\partial \ddot{q}/\partial \dot{q}$ is

$$\frac{\partial \ddot{q}}{\partial \dot{q}_i} = -\left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1} \frac{\partial^3 \mathcal{L}}{\partial \dot{q}_i \partial \dot{q}^2} \left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1} \left(\frac{\partial \mathcal{L}}{\partial q} - \frac{\partial^2 \mathcal{L}}{\partial q \partial \dot{q}}\dot{q}\right) + \left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}^2}\right)^{-1} \left(\frac{\partial^2 \mathcal{L}}{\partial \dot{q}_i \partial q} - \frac{\partial^3 \mathcal{L}}{\partial \dot{q}_i \partial q \partial \dot{q}}\dot{q} - \frac{\partial^2 \mathcal{L}}{\partial q \partial \dot{q}}\frac{\partial \dot{q}}{\partial \dot{q}_i}\right) \tag{89}$$

All of the derivatives in Eqs. (83), (88), and (89) are obtained numerically using a third-order nilpotent algebra as described in Sect. 2. This eliminates the need to re-derive analytical expressions as N (the number of pendulums) increases. The numerical evaluation of $\mathcal{L}(z^*)$ is all that is needed.

A Fortran 2008 module that implements the third-order multivariate nilpotent algebra was constructed to enable numerical calculation of Eqs. (83), (88), and (89). A simulation that implements Eqs. (75) – (77) using numerical derivatives for the N pendulum system was then constructed. As N increases, the initial conditions were chosen as

$$q(0) = [\theta_1, \theta_2, \ldots, \theta_N]^{\mathsf{T}} = [75^\circ, 0, \ldots, 0]^{\mathsf{T}} \tag{90}$$

and

$$\dot{q}(0) = [\dot{\theta}_1, \dot{\theta}_2, \ldots, \dot{\theta}_N]^{\mathsf{T}} = [0, 0, \ldots, 0]^{\mathsf{T}}. \tag{91}$$

The mass and length of each pendulum is chosen as $m = 1$ and $l = 1$, and the gravitational constant

Table 1 Approximate maximal Lyapunov exponent with increasing N

N	λ_{MLE}
1	3×10^{-5}
2	0.1
3	0.8
4	1.4
5	2.2
6	3.1

$g = 9.81$. Using a fixed-step fifth-order Dormand-Prince integration scheme with step size of 0.05 seconds, the time history of the resulting MLE for each system (up to $N = 6$) is shown in Fig. 3. Approximate values of λ_{MLE} for each system are reported in Table 1. As expected, λ_{MLE} approaches zero when $N = 1$, which is consistent with the fact that the system is conservative and two dimensional. The other higher-order cases show increasingly positive λ_{MLE} values. Finally, Figs. 4 and 5 illustrate the evolution of the errors in the energy conservation of the system and the orthogonality of the matrix $\mathcal{Q}$. Since the system is conservative, the total energy of the system

$$E = \frac{1}{2} m \sum_{i=1}^{N} \dot{r}_i \cdot \dot{r}_i + mg \sum_{i=1}^{N} h_i \tag{92}$$

should not deviate from the initial value E_0 throughout the system trajectory. The growth in the errors shown in Figs. 4 and 5 indicates that a variable time step and/or higher-order integrator may be needed for improved accuracy with increasing N. It is important to note that the entire analysis was completed without analytically differentiating any expressions. Only the Lagrangian in Eq. (84) was needed.

6 Conclusions

In this paper, a nilpotent algebra approach is developed for comprehensive numerical analysis of constrained

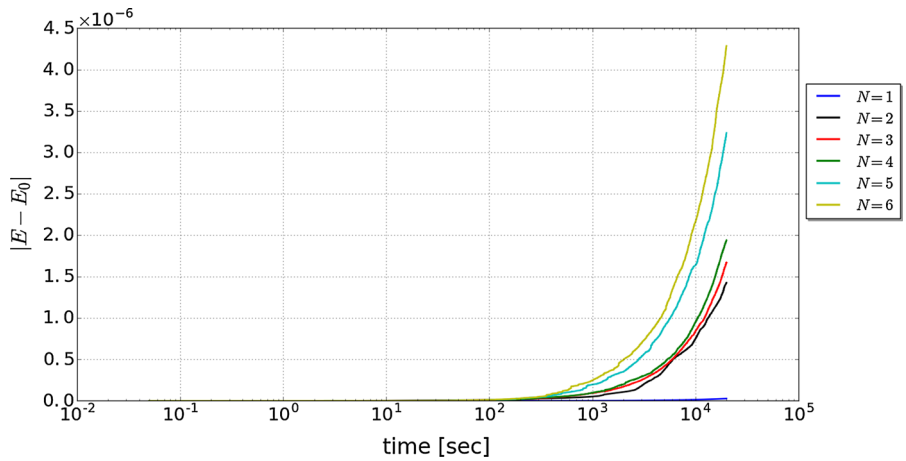

Fig. 4 Time evolution of the error in total energy conservation $|E - E_0|$

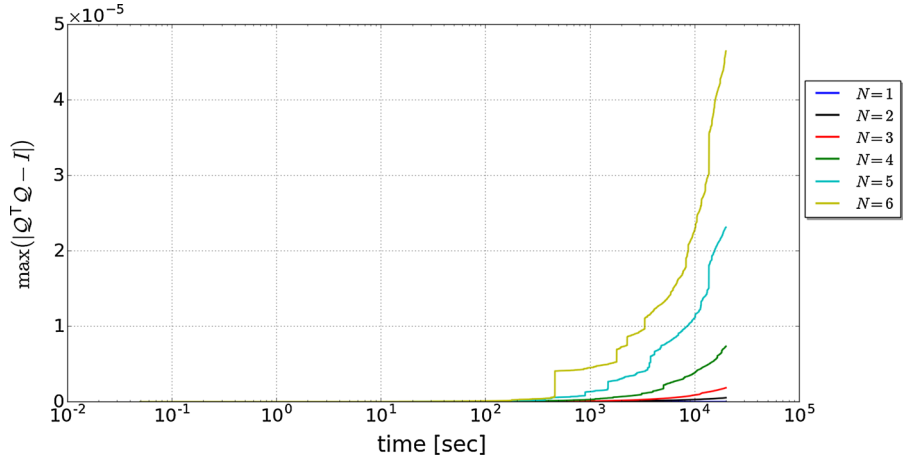

Fig. 5 Time evolution of the maximum component of $|\mathcal{Q}^{\mathsf{T}}\mathcal{Q} - I|$

dynamical systems in the Lagrangian framework. This was accomplished by extending the Lagrangian $\mathcal{L}(q, \dot{q}, t)$ and the constraints $\phi(q, t)$ and $\psi(q, \dot{q}, t)$ to truncated $\mathbb{T}$-algebras (multivariate generalizations of the so-called dual number algebra). Conceptually, it was shown that the process for numerical derivation of constrained equations of motion can be quite simple since only four elements are needed, namely the mass matrix M, the generalized force vector Q, the constraint matrix A, and the constraint vector b. It was shown that these elements can be constructed *numerically* by evaluating the extended $\tilde{\mathcal{L}}(z)$, $\tilde{\phi}(z)$, and $\tilde{\psi}(z)$ functions with a generalized state vector $z \in \mathbb{H}^{2n+1}$. In addition, it was shown that exact numerical linearization and sensitivity calculations can be carried out directly by extending the Lagrangian to a third-order truncated $\mathbb{T}$-algebra, which was demonstrated numerically for the computation of Lyapunov exponents in N-pendulum systems. The approach eliminates the need to obtain analytical derivatives prior to computational implementation. This is a significant capability when considering algorithm development for analysis of Lagrangian systems in general, especially for systems with complex Lagrangians and many interacting subsystems. In practice, these algebras can be implemented in software using standard operator overloading techniques, or in hardware using Hardware Description Language (HDL). As a result, numerical algorithms for simulations based on Lagrangian dynamical systems can be automated more intelligently as it is apparent that one only needs to define the requisite Lagrangian $\mathcal{L}$ and motion constraints ϕ and ψ for a given system.

References

1. Griewank, A., Walther, A.: Evaluating Derivatives, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (2008)
2. Rall, L.B.: Automatic Differentiation: Techniques and Applications, vol. 120. Springer, Heidelberg (1981)
3. Griewank, A., Juedes, D., Utke, J.: Algorithm 755: ADOL-C: a package for the automatic differentiation of algorithms written in C/C++. ACM Trans. Math. Softw. **22**, 131167 (1996)
4. Lyness, J.N., Moler, C.B.: Numerical differentiation of analytic functions. J. Numer. Anal. **4**, 202210 (1967)
5. Kroo, J.R.R.A., Alonso, J.J.: An Automated Method for Sensitivity Analysis Using Complex Variables. American Institute of Aeronautics and Astronautics, Reston (2000). AIAA-2000-0689
6. Lai, K.L., Crassidis, J.L.: Extensions of the first and second complex-step derivative approximations. J. Comput. Appl. Math. **219**, 276293 (2008)
7. Clifford, W.K.: Preliminary sketch of biquaternions. Proc. Lond. Math. Soc. **4**, 381–395 (1873)
8. Veldkamp, G.R.: On the use of dual numbers, vectors, and matrices in instantaneous, spatial kinematics. Mech. Mach. Theory **11**, 141–156 (1976)
9. Fike, J.A., Alonso, J.J.: Automatic differentiation through the use of hyper-dual numbers for second derivatives. Lecture notes in Computational Science and Engineering: Recent Advances in Algorithmic Differentiation, vol. 87, pp. 163–173 (2012)
10. Udwadia, F.E., Schutte, A.D.: Equations of motion for general constrained systems in lagrangian mechanics. Acta Mech. **213**, 111–129 (2010)
11. Udwadia, F.E., Schutte, A.D.: A unified approach to rigid body rotational dynamics and control. Proc. Roy. Soc. A **468**, 395–414 (2012)
12. Udwadia, F.E., Wanichanon, T.: On general nonlinear constrained mechanical systems. Numer. Algebra Control Optim. **3**(3), 425–443 (2013)
13. Bayo, E., Ledesma, R.: Augmented lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. Nonlinear Dyn. **9**, 113–130 (1996)
14. Udwadia, F.E., von Bremen, H.: Computation of Lyapunov characteristic exponents for continuous dynamical systems. Zeitschrift fur Angewandte Mathematik und Physik **53**, 123–146 (2002)