DYNAMICAL SYSTEMS AND GEOMETRIC CONSTRUCTION OF ALGORITHMS ①②

动力系统与算法的几何构造

Abstract

We give a brief survey on the research, undertaken by the senior author and his group, on the construction and analysis of numerical algorithms for ordinary differential equations, including specifically Hamiltonian systems and other systems with algebraico-geometric structures from the view-point of dynamical systems and ∞-dimensional Lie algebras of vector fields on manifolds and the associated ∞-dimensional Lie transformation groups.

§1 Introduction

Consider the autonomous system on $\mathbf{R}^m$

$$\dot{z} = a(z), \quad z = (z_1, \cdots, z_m)^T, \quad a(z) = (a_1(z), \cdots, a_m(z))^T \tag{1.1}$$

defined by its "right hand function", i.e., a smooth vector field $z \to a(z)$ on $\mathbf{R}^m$. The phase flow operator of dynamical system (1.1) defined by the vector field a is determined by the one parameter family $e_a^t = e^t$ of near-identity diffeomorphisms (or synonymously, transformations, operators) on $\mathbf{R}^m$ satisfying

$$\begin{cases} \dfrac{d}{dt} e_a^t = a \circ e_a^t \\ e_a^0 = \text{ identity } := 1_m. \end{cases} \tag{1.2}$$

It completely determines the dynamical evolution of the given continuous-time dynamical system. The phase flow operator e_a^t satisfies the one parameter group property

$$\begin{cases} e_a^{t+s} = e_a^t \circ e_a^s, \quad \forall t, s \in \mathbf{R}, \\ e_a^0 = 1_m \end{cases} \tag{1.3}$$

①Joint with Wang D L

②In Shi Z C, Yang C J, ed. Computational Mathematics of China. Comtemporary Mathematics. 1994 163: 1–32 and depends analytically on t, expressible as a convergent, at least locally in space and time, power series

$$e_a^t = \sum_{k=0}^{\infty} t^k e_k, \tag{1.4}$$

where the coefficients are smooth vector functions from $\mathbf{R}^m$ to $\mathbf{R}^m$ and can be determined recursively

$$e_0 = e^0 = 1_m,\ e_1 = a,\ e_k = \frac{1}{k}(e_{k-1})_* e_1, \quad k = 1, 2, \cdots, \tag{1.5}$$

here Jacobian matrix of $u : \mathbf{R}^m \to \mathbf{R}^m$ is denoted by u_*.

Now to any difference scheme solving the system (1.1) there corresponds a step-transition operator $g^s = g_a^s$ for step size s which completely (for single-step methods) or essentially (for multi-step methods) determines the dynamical evolution of the discrete-time dynamical system. Operator g^s is a one parameter family in step size s of near-identity diffeomorphisms

$$g_a^s = \sum_{k=0}^{\infty} s^k g_k, \quad g_0 = 1_m. \tag{1.6}$$

The algorithm g_a^s is said to be consistent with system (1.1) if $g_1 = a$; it is said to be of (accuracy) order $p(\geq 1)$ if

$$g_k = e_k, \quad k = 0, 1, \cdots, p; \quad g_{p+1} \neq e_{p+1}. \tag{1.7}$$

If g_a^s is consistent with system (1.1) and is a one-parameter group in s, then $g_a^s = e_a^s$, i.e., the difference scheme is exact. So, apart from some exceptional cases, g_a^s does not satisfy the group property $g_a^{s+t} = g_a^s \circ g_a^t$. Therefore, in the most cases, g_a^s is an approximation of e_a^s. The problem is to construct the algorithmic approximation g^s to the phase flow e_a^s in a *proper* way.

As is well known, all (smooth) vector fields on $\mathbf{R}^m$ form an ∞-dim Lie algebra, denoted by $\mathbf{V}(\mathbf{R}^m) = \mathbf{V}_m$, under the Lie bracket $[a, b] = a_* b - b_* a$, where a and b are two vector fields on $\mathbf{R}^m$. The Lie algebra $\mathbf{V}_m$ is associated with the ∞-dimensional Lie group $\mathbf{D}(\mathbf{R}^m) = \mathbf{D}_m$ of near-identity diffeomorphisms on $\mathbf{R}^m$. Every Lie subalgebra $\mathbf{L}$ of $\mathbf{V}_m$ is associated with a Lie subgroup $\mathbf{G}$ of $\mathbf{D}_m$. For any vector field a in $\mathbf{L}$, the phase flow e_a^t is a one-parameter subgroup in $\mathbf{G}$.

The three classical cases of dynamical systems with Lie-algebraic structures are: 1. Hamiltonian systems on $\mathbf{R}^{2n}$ — with the symplectic structure. $\mathbf{L}$ is the Lie algebra $\mathbf{SpV}_{2n}$ of Hamiltonian vector fields, $\mathbf{G}$ is the Lie group $\mathbf{SpD}_{2n}$ of symplectic transformations. 2. Contact systems on $\mathbf{R}^{2n+1}$ — with the contact structure. $\mathbf{L}$ is the Lie algebra $\mathbf{KV}_{2n+1}$ of contact vector fields, $\mathbf{G}$ is the Lie group $\mathbf{KD}_{2n+1}$ of contact transformations. 3. Source-free systems on $\mathbf{R}^m$ — with the volume structure. $\mathbf{L}$ is the Lie algebra $\mathbf{SV}_m$ of source-free vector fields, $\mathbf{G}$ is the Lie group $\mathbf{SD}_m$ of volume-preserving transformations. It is natural and even mandatory to require that the numerical algorithms for solving dynamical systems with structures to be structure-preserving, i.e., g^s belongs to the associated group $\mathbf{G}$ for small $|s|$. This is the main theme of this survey.

In section 2, we discuss the revertibility property of a one-parameter family of operators. Since the phase flows of all autonomous systems are revertible, their algorithms should be revertible too. In section 3, we give a kind of explicit structure-preserving algorithms for systems with Lie algebraic structures. In section 4, based on explicit Euler schemes, a class of explicit symplectic schemes for separable Hamiltonian systems are given. Symplectic schemes are implicit in general. But for specific Hamiltonians explicit Euler schemes may be symplectic and even exact. Consequently, if a Hamiltonian can be decomposed as a sum of specific Hamiltonians, then explicit symplectic schemes can be constructed by composing explicit Euler schemes for these specific Hamiltonians and higher order explicit symplectic schemes can be obtained as well. This kind of Hamiltonians is not too rare but can cover many important cases. In section 5 we give simple families of algorithms which are symplectic for all Hamiltonian systems. The general and systematic method to construct unconditional symplectic schemes is the generating function method which is discussed in section 6. Section 7 is to construct Hamiltonian algorithms for Hamiltonian systems with a perturbation parameter. In sections 8 and 9 we consider contact algorithms for contact systems and volume preserving algorithms for source-free systems respectively. In section 10 we consider the general theory of numerical approximations for ordinary differential equations based on the apparatus of formal power series and their exponential and logarithmic transforms.

§2 Revertibility of Operators

Let g^t be a one-parameter family of near-1 diffeomorphisms expansible as a convergent power series in t,

$$g^t = \sum_0^\infty t^k g_k, \quad g_0 = 1_m. \tag{2.1}$$

g^t is called ***revertible*** if $g^t \circ g^{-t} = 1_m$, $\forall t$. We define the reversion of g^t as

$$\check{g}^t = \left(g^{-t}\right)^{-1}. \tag{2.2}$$

g^t is revertible if and only if $g^t = \check{g}^t$. We have

(1) $\check{g}^t \circ g^t$ and $g^t \circ \check{g}^t$ are always revertible for all g^t;

(2) $\check{\check{g}}^t = g^t$, $(f^t \circ g^t)^\vee = \check{g}^t \circ \check{f}^t$;

(3) f^t and g^t are revertible $\Longrightarrow$ $f^{\alpha t} \circ g^{\beta t} \circ f^{\alpha t}$ and $f^{\alpha t} \circ g^{\beta t} \circ g^{\beta t} \circ f^{\alpha t}$ are revertible $\forall \alpha, \beta \in \mathbf{R}$;

(4) f_i^t are revertible, $i = 1, \cdots, k \Longrightarrow f_1^{\alpha_1 t} \circ \cdots \circ f_k^{\alpha_k t} \circ f_k^{\alpha_k t} \circ \cdots \circ f_1^{\alpha_1 t}$ are revertible for $\forall \alpha_i \in \mathbf{R}$, $i = 1, \cdots, k$;

(5) When f^t and g^t are revertible then $f^t \circ g^t$ is revertible if and only if $f^t \circ g^t = g^t \circ f^t$.

Since revertibility is a weaker form of the group property (1.3) all one-parameter groups of diffeomorphisms are always revertible.

For example, consider the system (1.1). The explicit and implicit Euler schemes are

$$z \to \hat{z} = E_a^s z: \quad \hat{z} = z + sa(z), \quad E_a^s = 1_m + sa; \tag{2.3}$$

$$z \to \hat{z} = I_a^s z: \quad \hat{z} = z + sa(\hat{z}), \quad I_a^s = (1_m - sa)^{-1}, \tag{2.4}$$

where E_a^s and I_a^s are step transition operators of the explicit and implicit Euler schemes respectively. So $I_a^s = (E_a^{-s})^{-1} = \check{E}_a^s$, i.e., the reversion of the explicit Euler operator is the implicit Euler operator.

Two-leg and one-leg weighted Euler schemes are

$$
\begin{aligned}
z \to \hat{z} = T^s_{a,c} z: \ \hat{z} &= z + s(ca(\hat{z}) + (1-c)a(z)), \\
T^s_{a,c} &= (1_m - sca)^{-1} \circ (1_m + s(1-c)a); \qquad (2.5) \\
z \to \hat{z} = E^s_{a,c} z: \ \hat{z} &= z + sa(c\hat{z} + (1-c)z), \\
E^s_{a,c} &= (1_m + s(1-c)a) \circ (1_m - sca)^{-1}. \qquad (2.6)
\end{aligned}
$$

So

$$\check{T}^s_{a,c} = T^s_{a,1-c}, \qquad \check{E}^s_{a,c} = E^s_{a,1-c}.$$

Consequently, $T^s_{a,c}$ and $E^s_{a,c}$ are revertible for any vector field a if and only if $c = \frac{1}{2}$, i.e., they are trapezoidal and centered Euler schemes respectively.

We write $g^s \approx e^s_a$ if $g^s = e^s_a + O(s^2)$, $g^s \approx e^s_a$, ord p if $g^s = e^s_a + O(s^{p+1})$, $g^s \approx e^s_a$, ord ∞ if $g^s = e^s_a$. A consistent revertible step transition operator is always of even order $2l (l \geq 1)$. For $g^s \approx e^s_a$, ord 1, $g^{s/2} \circ \check{g}^{s/2}$ and $\check{g}^{s/2} \circ g^{s/2} \approx e^s_a$ are revertible and of order 2. If $g^s \approx e^s_a$ is revertible and of order 2, then the revertible composite of g^s:

$$g^{\alpha s} \circ g^{\beta s} \circ g^{\alpha s} \approx e^s_a \qquad (2.7)$$

is of order 4 when

$$2\alpha + \beta = 1, \quad 2\alpha^3 + \beta^3 = 0,$$

i.e.,

$$\alpha = \frac{1}{2 - 2^{1/3}} > 0, \qquad \beta = 1 - 2\alpha < 0. \qquad (2.7')$$

Generally, if $g^s \approx e^s_a$ is revertible and of order $2l$, then the revertible composite (2.7) of g^s is of order $2(l+1)$, when

$$2\alpha + \beta = 1, \quad 2\alpha^{2l+1} + \beta^{2l+1} = 0, \qquad (2.8)$$

i.e.,

$$\alpha = \frac{1}{2 - 2^{1/(2l+1)}} > 0, \qquad \beta = 1 - 2\alpha < 0. \qquad (2.8')$$

For more details, see [19, 28, 31, 44].

We give some elementary algorithms, i.e., the explicit Euler method E, the implicit Euler method I, the centered Euler method C. E, I, C, especially the oldest and simplest E, will be the basic components for the composition of structure preserving algorithms. Then transition maps for step-size s for $x \to$ next step $\hat{x}$ will be denoted by E^s, I^s, C^s.

$$
\begin{aligned}
E: \quad & \hat{x} = x + sa(x), \quad E^s_a = 1_m + sa \approx e^s_a, \\
& \text{ord 1, non-revertible.} \qquad (2.9) \\
I: \quad & \hat{x} = x + sa(\hat{x}), \quad I^s_a = (1_m - sa)^{-1} = \check{E}^s_a \approx e^s_a, \\
& \text{ord 1, non-revertible.} \qquad (2.10) \\
C: \quad & \hat{x} = x + sa\Big(\frac{1}{2}\hat{x} + \frac{1}{2}x\Big), \quad \text{or in equivalent 2-stage form} \\
& \bar{x} = x + \frac{s}{2}a(\bar{x}), \quad \hat{x} = 2\bar{x} - x. \qquad (2.11) \\
& C^s_a = E^{s/2}_a \circ I^{s/2}_a = E^{s/2}_a \circ \check{E}^{s/2}_a = 2I^{s/2}_a - 1_m \approx e^s_a, \\
& \text{ord 2, revertible.}
\end{aligned}
$$

A vector field a is called *nilpotent of degree 2* if

$$a^2 = a_* a = a^* a = 0 \Longleftrightarrow e_a^s = 1_m + sa = E_a^s, \tag{2.12}$$

so with such "simple" vector fields, the method E gives exact solutions.

§3 Explicit Structure-preserving Algorithms

Consider a dynamical system

$$\frac{dz}{dt} = a(z), \quad a \in \mathbf{L} \tag{3.1}$$

where $\mathbf{L} \subset \mathbf{V}_m$, a Lie algebra of vector fields under the Lie bracket. Then the phase flow e_a^t is a one parameter subgroup of $\mathbf{G} \subset \mathbf{D}_m$, the Lie group associated with the Lie algebra $\mathbf{L}$.

We want to construct algorithms g_a^s approximating to e_a^s in the same group $\mathbf{G}$, i.e., $g_a^s \in \mathbf{G}$, for $|s|$ small. Such algorithm g_a^s is called *structure-preserving*, or $\mathbf{L}$-preserving. We have the following general propositions.

Let

$$a = \sum_{i=1}^{k} a^i, \qquad a^i \in \mathbf{L} \tag{3.2}$$

and let e_i^s be the phase flow of a^i, $e_i^s \in \mathbf{G}$, and $f_i^s \approx e_i^s$, ord ≥ 1, $f_i^s \in \mathbf{G}$, $\forall i = 1, \cdots, k$. Then

$$f^s := f_1^s \circ \cdots \circ f_k^s \approx e_a^s, \quad \text{ord } 1, \mathbf{L}\text{-preserving}, \tag{3.3}$$

$$g^s := f^{s/2} \circ \check{f}^{s/2} = f_1^{s/2} \circ \cdots \circ f_k^{s/2} \circ \check{f}_k^{s/2} \circ \cdots \circ \check{f}_1^{s/2} \approx e_a^s,$$
$$\text{ord } 2, \text{ revertible } \mathbf{L}\text{-preserving}, \tag{3.4}$$

$$h^s := g^{\alpha s} \circ g^{\beta s} \circ g^{\alpha s} \approx e_a^s, \quad \text{with the } \alpha, \beta \text{ in (2.7')}$$
$$\text{ord } 4, \text{ revertible } \mathbf{L}\text{-preserving; etc.} \tag{3.5}$$

A case with $a = a^1 + a^2 + a^3$ for construction of 3D volume preserving algorithms will be shown in §9. In particular, as an ideal case, take $f_i^s = e_{a^i}^s = e_i^s$, $i = 1, \cdots k$, then

$$f^s := e_1^s \circ \cdots \circ e_k^s \approx e_a^s, \quad \text{ord } 1, \mathbf{L}\text{-preserving}, \tag{3.6}$$

$$g^s := e_1^{s/2} \circ \cdots \circ e_{k-1}^{s/2} \circ e_k^s \circ e_{k-1}^{s/2} \circ \cdots \circ e_1^s \approx e_a^s,$$
$$\text{ord } 2, \text{ revertible } \mathbf{L}\text{-preserving}. \tag{3.7}$$

This construction gives actually a *numerical algorithm* only when the component phase flows e_i^s are available and algorithmically implementable, this is quite often the case.

We call a vector field $a \in \mathbf{L}$ to be $\mathbf{L}$-separable if it admits a decomposition

$$a = \sum_{i=1}^{k} a^i, \quad a^i \in \mathbf{L} \text{ and nilpotent, i.e., } a_*^i a^i = 0. \tag{3.8}$$

In this case, by (2.12), $e_{a^i}^s = E_{a^i}^s =: E_i^s$, so f^s, g^s become

$$f^s := E_1^s \circ \cdots \circ E_k^s \approx e_a^s, \quad \text{ord } 1, \mathbf{L}\text{-preserving}, \tag{3.9}$$

$$g^s := E_1^{s/2} \circ \cdots \circ E_{k-1}^{s/2} \circ E_k^s \circ E_{k-1}^{s/2} \circ \cdots \circ E_1^{s/2} \approx e_a^s,$$
$$\text{ord } 2, \text{ revertible } \mathbf{L}\text{-preserving}, \tag{3.10}$$

and all the related revertible $\mathbf{L}$-preserving algorithms are explicit. A simple $\mathbf{L}$-separable case with decomposition $a = a^1 + a^2$ is important for the construction of explicit symplectic method for Hamiltonian systems. See immediately the next section.

§4 Explicit Euler-type Hamiltonian Algorithms

We consider now Hamiltonian systems on $\mathbf{R}^{2n}$. The Hamiltonian vector field is of the form

$$a = J\nabla H, \quad J = J_{2n} = \begin{pmatrix} 0 & -I_n \\ I_n & 0 \end{pmatrix} \tag{4.1}$$

defined by an energy function $H(z)$ on $\mathbf{R}^{2n}$, which defines the Hamiltonian system

$$\frac{dz}{dt} = a(z) = J\nabla H(z), \quad z \in \mathbf{R}^{2n}, \tag{4.2}$$

where $\nabla H(z) = H_z = (H_{z_1}(z), \cdots, H_{z_{2n}}(z))^T$ is the gradient of H, or

$$\begin{aligned} \frac{dp}{dt} &= -H_q, \\ \frac{dq}{dt} &= H_p, \end{aligned} \qquad \begin{pmatrix} p \\ q \end{pmatrix} = z \in \mathbf{R}^{2n}, \quad p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix}, \quad q = \begin{pmatrix} q_1 \\ \vdots \\ q_n \end{pmatrix}.$$

Intrinsic to all Hamiltonian systems is the symplectic structure defined by the differential 2-form

$$\omega = \sum_{i=1}^{n} dp_i \wedge dq_i.$$

A *symplectic* (or *canonical*) transformation is a diffeomorphism $g : z \to \hat{z}$ on $\mathbf{R}^{2n}$ preserving the symplectic structure

$$\sum d\hat{p}_i \wedge d\hat{q}_i = \sum dp_i \wedge dq_i, \qquad \text{i.e.} \qquad \big(g_*(z)\big)^T J_{2n}\big(g_*(z)\big) \equiv J_{2n}.$$

All Hamiltonian vector fields form a Lie algebra $\mathbf{L} = \mathbf{SpV}_{2n}$ under Lie bracket. The associated Lie group $\mathbf{G} = \mathbf{SpD}_{2n}$ consists of all near-identity symplectic transformations on $\mathbf{R}^{2n}$. The phase flows $e^t_{J\nabla H}$, abbreviated as e^t_H, of Hamiltonian systems are symplectic transformations. So it is natural and mandatory to require the algorithms with transition operator $g^s_{J\nabla H}$, abbreviated as g^s_H, to be symplectic too, such algorithms are called *symplectic* (or *Hamiltonian* or *canonical*). Non-symplectic algorithms inevitably bring in distortions alien to Hamiltonian dynamics, such as artificial dissipation and other parasitic effects. See [2–5, 14–15, 20].

The phase flow e^t_H of (4.2) is a one parameter group of symplectic maps

$$e^t_H = e_0 + te_1 + t^2 e_2 + \cdots, \tag{4.3}$$

where

$$\begin{aligned} &e_0 = 1_{2n},\ e_1 = J\nabla H,\ e_2 = \frac{1}{2}(J\nabla H)_* J\nabla H, \\ &e_k = \frac{1}{k}(e_{k-1})_* J\nabla H,\ k \geq 3. \end{aligned} \tag{4.4}$$

We need to construct *symplectic* algorithms.

By (2.12), a Hamiltonian vector field $J\nabla H$ is nilpotent of degree 2 if

$$(J\nabla H)_* J\nabla H = 0 \quad \text{or} \quad H_{zz} J H_z = 0, \tag{4.5}$$

for simplicity we say H is nilpotent of degree 2. For the Hamiltonian H nilpotent of degree 2, the explicit Euler scheme E^s_H is just the phase flow of the system (4.2) at $t = s$

$$e^s_H(z) = z + se_1(z) = z + sJ\nabla H(z) \equiv E^s_H(z).$$

So it must be symplectic.

If

$$H(z) = \sum_{i=1}^{k} H_i(z), \tag{4.6}$$

and every $H_i(z)$ is nilpotent of degree 2, then $H(z)$ is said to be ***symplectically separable***. The following composite

$$z \to \hat{z} = f^s z, \quad f^s := E^s_{H_1} \circ E^s_{H_2} \circ \cdots \circ E^s_{H_k}, \tag{4.7}$$

is explicit, symplectic and of order 1. The composite

$$\begin{aligned} &z \to \hat{z} = g^s z, \\ &g^s := f^{s/2} \circ \check{f}^{s/2} = E^{s/2}_{H_1} \circ \cdots \circ E^{s/2}_{H_{k-1}} \circ E^s_{H_k} \circ E^{s/2}_{H_{k-1}} \circ \cdots \circ E^{s/2}_{H_1} \end{aligned} \tag{4.8}$$

is explicit, symplectic and revertible of order 2. High order revertible symplectic schemes can be constructed from these schemes by the procedure (2.7) and (2.8).

Examples. If

$$H(z) = H(p,q) = \phi(p) \quad \text{or} \quad \psi(q),$$

where ϕ and ψ are functions of n variables, then $H(z)$ is nilpotent of degree 2

$$\begin{pmatrix} H_p(p,q) \\ H_q(p,q) \end{pmatrix} = \begin{pmatrix} \phi_p(p) \\ 0 \end{pmatrix} \quad \text{or} \quad \begin{pmatrix} 0 \\ \psi_q(q) \end{pmatrix}.$$

The corresponding explicit Euler schemes

$$\begin{aligned} \hat{p} &= p, \\ \hat{q} &= q + s\phi_p(p), \end{aligned}$$

or

$$\begin{aligned} \hat{p} &= p - s\psi_q(q), \\ \hat{q} &= q, \end{aligned}$$

are symplectic. If

$$H(p,q) = \phi(p) + \psi(q), \tag{4.9}$$

which is symplectically separable, then the composite schemes

$$E^s_\psi \circ E^s_\phi : \quad \hat{p} = p - s\psi_q(\hat{q}), \quad \hat{q} = q + s\phi_p(p), \tag{4.10}$$

$$E^s_\phi \circ E^s_\psi : \quad \hat{p} = p - s\psi_q(q), \quad \hat{q} = q + s\phi_p(\hat{p}), \tag{4.11}$$

are explicit, symplectic and of order 1. The revertible scheme

$$\begin{aligned} &g^s := (E^{s/2}_\psi \circ E^{s/2}_\phi)^\vee \circ (E^{s/2}_\psi \circ E^{s/2}_\phi) = E^{s/2}_\phi \circ E^s_\psi \circ E^{s/2}_\phi : \\ &q^1 = q + \frac{s}{2}\phi_p(p), \quad \hat{p} = p - s\psi_q(q_1), \quad \hat{q} = q^1 + \frac{s}{2}\phi_p(\hat{p}). \end{aligned} \tag{4.12}$$

is symplectic and of order 2. Its revertible composite

$$h^s := g^{\alpha s} \circ g^{\beta s} \circ g^{\alpha s}$$

gives a symplectic scheme of order 4 with the parameters (2.7′), i.e.,

$$\begin{aligned} p^1 &= p - c_1 s\psi_q(q), & q^1 &= q + d_1 s\phi_p(p^1),\\ p^2 &= p^1 - c_2 s\psi_q(q^1), & q^2 &= q^1 + d_2 s\phi_p(p^2),\\ p^3 &= p^2 - c_3 s\psi_q(q^2), & q^3 &= q^2 + d_3 s\phi_p(p^3),\\ \hat{p} &= p^3 - c_4 s\psi_q(q^3), & \hat{q} &= q^3 + d_4 s\phi_p(\hat{p}), \end{aligned} \tag{4.13}$$

with the parameters $\alpha = (2 - 2^{1/3})^{-1}$, $\beta = 1 - 2\alpha$ and either

$$c_1 = 0, \quad c_2 = c_4 = \alpha, \quad c_3 = \beta, \quad d_1 = d_4 = \alpha/2, \quad d_2 = d_3 = (\alpha + \beta)/2,$$

or

$$c_1 = c_4 = \alpha/2, \quad c_2 = c_3 = (\alpha + \beta)/2, \quad d_1 = d_3 = \alpha, \quad d_2 = \beta, \quad d_4 = 0.$$

Symplectically separable Hamiltonians have wide coverage in applications. It is easy to see that, the Hamiltonian of the form

$$H(p, q) = \phi(Ap + Bq), \quad AB^T = BA^T, \tag{4.14}$$

where $\phi(x)$ is a function of n variables, A and B are $n \times n$ matrices, is also nilpotent of degree 2. Then the explicit Euler scheme $E^s_H = E^s(\phi) = e^s_H$ is

$$E^s(\phi): \quad \begin{aligned} \hat{p} &= p - sB^T \phi_x(Ap + Bq)\\ \hat{q} &= q + sA^T \phi_x(Ap + Bq) \end{aligned} \tag{4.15}$$

so we get a class of symplectically separable Hamiltonians [19]

$$H(p, q) = \sum_{i=1}^{k} H_i(p, q), \quad H_i(p, q) = \phi_i(A_i p + B_i q), \quad A_i B_i^T = B_i A_i^T, \tag{4.16}$$

where $\phi_i(x)$ are functions of n variables, A_i, B_i are $n \times n$ matrices. Moreover, all polynomial Hamiltonians belong to this class; in fact, it has been proved in [19] that every polynomial $H(p, q)$ in $2n$ variables p, q can be decomposed in the form of (4.16), where $\phi_i(x)$ are polynomials of n variables x, $A_i = \mathrm{diag}(a^i_1, \cdots, a^i_n)$ and $B_i = \mathrm{diag}(b^i_1, \cdots, b^i_n)$.

For symplectically separable Hamiltonians of the above class, all explicit revertible symplectic schemes formed by compositions contain solely the Euler schemes $E^s(\phi_i)$ in the form (4.15) as basic components. A related conjecture is: this class already covers all the symplectically separable cases, i.e., H is nilpotent of degree 2 if and only if it is expressible in the form (4.14).

More generally, if we have a decomposition $H = H_1 + \cdots + H_k$, for which each H_i is integrable and its phase flow $e^s_{H_i}$ is algorithmically implementable, then by (3.4), (3.5) the symmetrical composites

$$g^s_H = e^{s/2}_{H_1} \circ \cdots \circ e^{s/2}_{H_{k-1}} \circ e^s_{H_k} \circ e^{s/2}_{H_{k-1}} \circ \cdots \circ e^{s/2}_{H_1} \tag{4.17}$$

$$h^s_H = g^{\alpha s}_H \circ g^{\beta s}_H \circ g^{\alpha s}_H \tag{4.18}$$

give explicit symplectic, revertible algorithms of order 2, 4, etc. This general approach is widely applicable to *many body problems* in different physical contexts for which the *2-body problem* is *solvable*. Since in such problems the Hamiltonian usually admits a natural decomposition $H = \sum_{i<j} H_{ij}$, each H_{ij} accounts for a 2-body problem.

An interesting result from this approach is a construction [47] of explicit symplectic method for computing the Hamiltonian system of N vortices $z_i = (x_i, y_i)$ with intensities k_i

$$k_i \frac{dx_i}{dt} = \frac{\partial H}{\partial y_i}, \qquad k_i \frac{dy_i}{dt} = -\frac{\partial H}{\partial x_i}, \quad i = 1, \cdots, N \tag{4.19}$$

with symplectic structure $\omega_k = \sum \frac{1}{k_i} dx_i \wedge dy_i$ and

$$H = \sum_{i<j} H_{ij}, \qquad H_{ij} = -\frac{1}{2\pi} k_i k_j \ln r_{ij},$$
$$r_{ij} = ((x_i - x_j)^2 + (y_i - y_j)^2)^{1/2}. \tag{4.20}$$

Each H_{ij} accounts for a solvable 2-vortex motion in which both vortices z_i, z_j rotate about their center of vorticity $z = (k_i z_i + k_j z_j)(k_i + k_j)^{-1}$ with angular velocity $a = (k_i + k_j)/2\pi r_{ij}^2$ if $k_i + k_j \neq 0$ or translate with linear velocity $(b(y_i - y_j), -b(x_i - x_j))$, $b = k_i/2\pi r_{ij}^2$ if $k_i + k_j = 0$. These simple phase flows $e^s_{H_{ij}}$ serve as the basic algorithmic components for successive compositions, resulting in efficient explicit revertible methods, symplectic for this specific system. They are promising in application to incompressible ideal flows for tracking the vortex particles, which ought to be done in the structure-preserving way.

§5 Unconditional Hamiltonian Algorithms

Now we want to construct *unconditional* Hamiltonian algorithms, i.e., they are symplectic for *all* Hamiltonian systems.

It can be shown that

(a) The two-leg weighted Euler schemes (2.5) are generically non-symplectic for any real number c or real $2n \times 2n$ matrix c.

(b) The one-leg weighted Euler schemes (2.6), i.e.,

$$\hat{z} = g^s_H z: \quad \hat{z} = z + sJH_z(c\hat{z} + (1-c)z), \tag{5.1}$$

with real number c is unconditionally symplectic if and only if $c = \frac{1}{2}$, which corresponds to the *centered Euler scheme*

$$\hat{z} = z + sJH_z\left(\frac{\hat{z} + z}{2}\right). \tag{5.2}$$

These simple propositions illustrate a general situation: apart from some very rare exceptions, the vast majority of conventional schemes are non-symplectic. However, if we allow c in (5.1) to be a real matrix of order $2n$, we get a far-reaching generalization: (5.1) is symplectic iff

$$c = \frac{1}{2}(I_{2n} + J_{2n}B), \quad B^T = B, \quad c^T J + Jc = J. \tag{5.3}$$

The simplest and important cases are

$$\begin{aligned}
C: &\quad c = \frac{1}{2} I_{2n}, & &\hat{z} = z + sJH_z\left(\frac{\hat{z} + z}{2}\right), \\
P: &\quad c = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}, & &\begin{aligned}\hat{p} &= p - sH_q(\hat{p}, q), \\ \hat{q} &= q + sH_p(\hat{p}, q),\end{aligned} \\
Q: &\quad c = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix}, & &\begin{aligned}\hat{p} &= p - sH_q(p, \hat{q}), \\ \hat{q} &= q + sH_p(p, \hat{q}),\end{aligned}
\end{aligned} \tag{5.4}$$

For $H(p,q)=\phi(p)+\psi(q)$, the above schemes P and Q reduce to explicit schemes (4.11) and (4.10).

Note that we may split the Hamiltonian vector field as

$$J\nabla H = cJ\nabla H + (I-c)J\nabla H \tag{5.5}$$

and it is more convenient to implement (5.1) as the composite

$$\begin{aligned} &z \to \bar{z} = z + scJ\nabla H(\bar{z}) = I^s_{cJ\nabla H}(z),\\ &\bar{z} \to \hat{z} = \bar{z} + s(I-c)J\nabla H(\bar{z}) = E^s_{(I-c)J\nabla H}(\bar{z}),\\ &g^s_{H,c} = E^s_{(I-c)J\nabla H} \circ I^s_{cJ\nabla H}. \end{aligned} \tag{5.6}$$

The first step is the implicit Euler method for the vector field $cJ\nabla H$, the second step is the explicit Euler method for the vector field $(I-c)J\nabla H$, both maps are not symplectic, but their composite is symplectic.

Scheme (5.1) is revertible of order 2 for $c=\frac{1}{2}I$, this is (5.2). (5.1) is of order 1 for $c \neq \frac{1}{2}I$, from §2 it follows that $\check{g}^s_{H,c} = g^s_{H,I-c}$. Then the composites

$$g^{s/2}_{H,c} \circ g^{s/2}_{H,I-c} \quad \text{and} \quad g^{s/2}_{H,I-c} \circ g^{s/2}_{H,c}$$

are symplectic, revertible and of order 2 for step size s.

So we get a great variety of simple symplectic schemes of order 1 or 2, classified according to type matrices $B \in sm(2n) :=$ space of symmetric matrices of order $2n$, which is a linear space of dimension $2n^2+n$.

The conservation properties of the above symplectic schemes are well-understood [4, 15, 18, 43]. In fact we have proved:

Let $\phi_A(z) = \frac{1}{2}z^TAz$, $A=A^T$ be a quadratic form and invariant under the system with Hamiltonian H. Then ϕ_A is invariant under symplectic difference scheme $g^s_{H,c}$ iff A commutes symplectically with the type matrix B in (5.3), i.e., $AJB=BJA$.

We list some of the most important weight matrices c, the type matrices B, together with the corresponding form of symmetric matrices A of the conserved quadratic invariants ϕ_A:

$$c = I - c = \frac{1}{2}I, \quad B=0, \quad A \text{ arbitrary}.$$

$$c = \begin{pmatrix} I_n & 0 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & -I_n \\ -I_n & 0 \end{pmatrix}, \qquad c = \begin{pmatrix} 0 & 0 \\ 0 & I_n \end{pmatrix}, \quad B = \begin{pmatrix} 0 & I_n \\ I_n & 0 \end{pmatrix}, \qquad A = \begin{pmatrix} 0 & b \\ b^T & 0 \end{pmatrix}, \ b \text{ arbitrary; angular momentum type}.$$

$$c = \frac{1}{2}\begin{pmatrix} I_n & \pm I_n \\ \mp I_n & I_n \end{pmatrix}, \quad B = \mp I_{2n}, \quad A = \begin{pmatrix} a & b \\ -b & a \end{pmatrix}, \ a^T = a,\ b^T = -b; \ \text{Hermitian type}.$$

$$c = \frac{1}{2}\begin{pmatrix} I & \pm I \\ \pm I & I \end{pmatrix}, \quad B = \pm\begin{pmatrix} I_n & 0 \\ 0 & -I_n \end{pmatrix}, \quad A = \begin{pmatrix} a & b \\ -b & -a \end{pmatrix}, \ a^T = a,\ b^T = -b.$$

Numerical evidences and theoretical analysis so far accumulated show unanimously and indisputably with the qualitative and the quantitative superiority of performance of symplectic methods over conventional non-symplectic ones for Hamiltonian systems. The contrasts are overwhelming and striking especially in crucial aspects of global behavior, orbital stability and long-time tracking capabilities. For the first time one now knows ***consciously*** a ***proper*** or descent way to compute ***Newton's equations*** of ***motion***, i.e., first put them in Hamiltonian form, then compute by symplectic algorithms [4–5, 10, 15].

§6 Generating Function Methods

The construction of unconditional symplectic algorithms in §5 presents a special case of a general methodology based on generating functions explained in this section.

A matrix α of order $4n$ is called a ***Darboux matrix*** if

$$\alpha^T J_{4n}\alpha = \tilde{J}_{4n}, \quad J_{4n} = \begin{pmatrix} 0 & -I_{2n} \\ I_{2n} & 0 \end{pmatrix}, \qquad \tilde{J}_{4n} = \begin{pmatrix} J_{2n} & 0 \\ 0 & -J_{2n} \end{pmatrix},$$
$$\alpha = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad \alpha^{-1} = \begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}.$$

Every Darboux matrix induces a (linear) ***fractional transform*** between symplectic and symmetric matrices

$$\begin{aligned} &\sigma_\alpha : Sp(2n) \to sm(2n), \\ &\quad \sigma_\alpha(S) = (aS+b)(cS+d)^{-1} = A, \quad \text{for} \quad |cS+d| \neq 0 \end{aligned}$$

with the inverse transform $\sigma_\alpha^{-1} = \sigma_{\alpha^{-1}}$

$$\begin{aligned} &\sigma_\alpha^{-1} : sm(2n) \to Sp(2n), \\ &\quad \sigma_\alpha^{-1}(A) = (a_1 A + b_1)(c_1 A + d_1)^{-1} = S, \quad \text{for } |c_1 A + d_1| \neq 0, \end{aligned}$$

where $Sp(2n) = \left\{ S \in GL(2n, \mathbf{R}) |\ S^T J_{2n} S = J_{2n} \right\}$ is the group of symplectic matrices.

The above machinery can be extended to generally non-linear operators on $\mathbf{R}^{2n}$. Denote $\mathbf{SpD}_{2n}$ the totality of symplectic operators, and $symm(2n)$ the totality of symmetric operators (not necessary one-one). Every $f \in symm(2n)$ corresponds, at least locally, a real function ϕ (unique up to a constant) such that f is the gradient of ϕ: $f(w) = \nabla\phi(w)$, where $\nabla\phi(w) = (\phi_{w_1}(w), \cdots, \phi_{w_{2n}}(w))^T = \phi_w(w)$. Then we have

$$\begin{aligned} &\sigma_\alpha :\ \mathbf{SpD}_{2n} \to symm(2n), \\ &\sigma_\alpha(g) = (a \circ g + b) \circ (c \circ g + d)^{-1} = \nabla\phi, \quad \text{for} \quad |cg_z + d| \neq 0, \end{aligned}$$

or alternatively

$$ag(z) + bz = (\nabla\phi)(cg(z) + dz),$$

where ϕ is called the ***generating function*** of Darboux type α for the symplectic operator g. Then

$$\begin{aligned} &\sigma_\alpha^{-1} : \quad symm(2n) \to \mathbf{SpD}_{2n}, \\ &\sigma_\alpha^{-1}(\nabla\phi) = (a_1 \circ \nabla\phi + b_1) \circ (c_1 \circ \nabla\phi + d_1)^{-1} = g, \\ &\qquad\qquad \text{for} \quad |c_1\phi_{ww} + d_1| \neq 0 \end{aligned} \tag{6.1}$$

or alternatively

$$a_1\nabla\phi(w) + b_1(w) = g(c_1\nabla\phi(w) + d_1 w), \tag{6.2}$$

where g is called the symplectic operator of Darboux type α for the generating function ϕ.

For the study of symplectic difference scheme we may narrow down the class of Darboux matrices to the subclass of ***normal Darboux matrices***, i.e., those satisfying $a+b=0$, $c+d=I_{2n}$. The normal Darboux matrices α can be characterized as

$$\alpha=\begin{pmatrix} a & b \\ c & d \end{pmatrix}=\begin{pmatrix} J & -J \\ c & I-c \end{pmatrix}, \quad c=\frac{1}{2}(I+JB), \quad B^T=B, \tag{6.3}$$

$$\alpha^{-1}=\begin{pmatrix} a_1 & b_1 \\ c_1 & d_1 \end{pmatrix}=\begin{pmatrix} (c-I)J & I \\ cJ & I \end{pmatrix}. \tag{6.4}$$

The fractional transform induced by a normal Darboux matrix establishes a 1-1 correspondence between ***symplectic operators near identity*** and ***symmetric operators near nullity***. Then the determinantal conditions could be taken for granted. Those B's listed in section 5 correspond to the most important normal Darboux matrices.

For every Hamiltonian H with its phase flow e_H^t and for every normal Darboux matrix α, we get the ***generating function*** $\phi(w,t)=\phi_H^t(w)=\phi_{H,\alpha}^t(w)$ of ***normal Darboux type*** α for the ***phase flow*** of H by

$$\nabla\phi_{H,\alpha}^t=(Je_H^t-J)\circ(ce_H^t+I-c)^{-1}, \quad \text{for small} \quad |t|. \tag{6.5}$$

$\phi_{H,\alpha}^t$ satisfies the ***Hamilton-Jacobi*** equation

$$\frac{\partial}{\partial t}\phi(w,t)=-H(w+a_1\nabla\phi(w,t))=-H(w+c_1\nabla\phi(w,t)) \tag{6.6}$$

and can be expressed by Taylor series in t

$$\phi(w,t)=\sum_{k=1}^{\infty}\phi^{(k)}(w)t^k, \quad |t| \quad \text{small}. \tag{6.7}$$

The coefficients can be determined recursively

$$\begin{aligned} \phi^{(1)}(w) &= -\,H(w), \quad \text{and for} \quad k\ge 0, \quad a_1=(c-I)J: \\ \phi^{(k+1)}(w) &= \frac{-1}{k+1}\sum_{m=1}^{k}\frac{1}{m!}\sum_{\substack{j_1+\cdots+j_m=k \\ j_l\ge 1}} D^m H(w)\times \\ &\quad \times(a_1\nabla\phi^{(j_1)}(w),\cdots,a_1\nabla\phi^{(j_m)}(w)), \end{aligned} \tag{6.8}$$

where we use the notation of the m-linear form

$$\begin{aligned} &D^m H(w)(a_1\nabla\phi^{(j_1)}(w),\cdots,a_1\nabla\phi^{(j_m)}(w)) \\ &:= \sum_{i_1,\cdots,i_m=1}^{2n} H_{z_{i_1}\cdots z_{i_m}}(w)(a_1\nabla\phi^{(j_1)}(w))_{i_1}\cdots(a_1\nabla\phi^{(j_m)}(w))_{i_m}. \end{aligned}$$

Let ψ^s be a truncation of $\phi_{H,\alpha}^s$ up to a certain power s^m, say. Using inverse transform σ_α^{-1} we get the symplectic operator

$$g^s=\sigma_\alpha^{-1}(\nabla\psi^s), \quad |s| \quad \text{small}, \tag{6.9}$$

which depends on s, H, α (or equivalently B) and the mode of truncation. It is a symplectic approximation to the phase flow e_H^s and can serve as the transition operator of a symplectic difference scheme (for the Hamiltonian system (4.2))

$$z\to\hat z=g^s z: \quad \hat z=z-J\nabla\psi^s(c\hat z+(I-c)z), \quad c=\frac{1}{2}(I+JB). \tag{6.10}$$

Thus, using the machinery of phase flow generating functions we have constructed, for every H and every normal Darboux matrix, a hierarchy of symplectic schemes by truncation. The simple symplectic schemes (5.1, 5.3) correspond to the lowest truncation. The conservation properties of all these higher order schemes are the same as stated in section 5.

§7 Hamiltonian Algorithms for Hamiltonian Systems with a Perturbation Parameter

The machinery above can also be applied to construct symplectic algorithms for perturbed Hamiltonian systems defined by the perturbed Hamiltonian

$$H(z;\epsilon)=\sum_{k=0}^{\infty}\epsilon^k H_k(z)=H_0(z)+\sum_{k=1}^{\infty}\epsilon^k H_k(z), \tag{7.1}$$

where ϵ is the small perturbation parameter. $H_0(z)$ is usually an integrable Hamiltonian. The corresponding perturbed Hamiltonian system is

$$\frac{dz}{dt}=JH_z(z,\epsilon), \qquad z\in \mathbf{R}^{2n}. \tag{7.2}$$

Its phase flow, denoted by $e^t_\epsilon = e^t_{H,\epsilon}$, depends on the parameter ϵ. The Hamilton-Jacobi equation and generating function are also parameterized by ϵ. That means, the parameterized generating function $\phi^t_\epsilon(w)=\phi^t(w,\epsilon)=\phi(w,t,\epsilon)$ satisfies the parameterized Hamilton-Jacobi equation

$$\frac{\partial}{\partial t}\phi(w,t,\epsilon)=-H(w+a_1\nabla\phi(w,t,\epsilon),\epsilon). \tag{7.3}$$

$\phi(w,t,\epsilon)$ can be expanded as a power series in ϵ in stead of t

$$\phi(w,t;\epsilon)=\sum_{k=0}^{\infty}\epsilon^k\phi^{(k)}(w,t). \tag{7.4}$$

The coefficients $\phi^{(k)}(w,t)$ satisfy the following equations:

$$\phi^{(0)}_t(w,t)=-\,H_0(w+a_1\nabla\phi^{(0)}(w,t)), \tag{7.5}$$

$$\begin{aligned} k\ge 1:\ \phi^{(k)}_t(w,t)=&-\,H_k(w^*)-\sum_{i=1}^{k}\sum_{m=1}^{i}\frac{1}{m!}\sum_{\substack{i_1+\cdots+i_m=i\\ i_j\ge 1}}D^mH_{k-i}(w^*)\\ &\times(a_1\nabla\phi^{(i_1)}(w,t),\cdots,a_1\nabla\phi^{(i_m)}(w,t))\end{aligned} \tag{7.6}$$

with the initial values $\phi^{(i)}(w,0)=0$, where $w^*=w+a_1\nabla\phi^{(0)}(w,t)$.

(7.5) is just the Hamilton-Jacobi equation for the unperturbed Hamiltonian system with Hamiltonian $H_0(z)$. The right hand side of (7.6) can be written as

$$-DH_0(w^*)\cdot a_1\nabla\phi^{(k)}(w,t)+R^{(k)}(w,t),$$

where the remainder $R^{(k)}(w,t)$ depends only on $H_i(w^*)$, $i=0,1,\cdots,k$ and $\phi^{(i)}(w,t)$, $i=0,1,\cdots,k-1$. Once $\phi^{(0)}(w,t),\cdots,\phi^{(k-1)}(w,t)$ are known, $R^{(k)}(w,t)$ is also known. Therefore, if $\phi^{(0)}(w,t)$ can be solved from (7.5), then for $k\ge 1$,

$$\phi^{(k)}_t(w,t)=-DH_0(w^*)\cdot a_1\nabla\phi^{(k)}(w,t)+R^{(k)}(w,t)$$

are the linear partial differential equations for $\phi^{(k)}(w,t)$. They have the same coefficients, only those for $R^{(k)}(w,t)$ are different.

In some cases, we can solve (7.5) easily, refer to [38]. In general, it is difficult to solve (7.5) by analytical method. Nevertheless we can always give an approximative solution, for example, using the methods discussed above.

Let now $\psi_\epsilon^s(w) = \psi(w, s, \epsilon)$ be a truncation of $\phi(w, s, \epsilon)$ up to a certain power ϵ^m. Using the inverse transform σ_α^{-1} we get the symplectic operator

$$g_\epsilon^s = \sigma_\alpha^{-1}(\nabla \psi_\epsilon^s), \quad |s| \quad \text{small}. \tag{7.7}$$

It is a symplectic approximation to the phase flow e_ϵ^s of order m in ϵ and can serve as the transition operator of a symplectic difference scheme for the perturbed Hamiltonian system (7.2)

$$z \to \hat{z} = g_\epsilon^s z : \quad \hat{z} = z - J\nabla\psi(c\hat{z} + (I - c)z, s, \epsilon), \quad c = \frac{1}{2}(I + JB). \tag{7.8}$$

For general perturbed Hamiltonians $H(z, \epsilon)$, these schemes are only consistent in the time stepsize s. But for the perturbed Hamiltonians with the form

$$H(z, \epsilon) = H_0(z) + \epsilon H_1(z), \tag{7.9}$$

the order of the schemes (7.8) in the time stepsize s is the same as in ϵ. More precisely, for the m-th order scheme g_ϵ^s for the perturbed Hamiltonian (7.9),

$$g_\epsilon^s = e_\epsilon^s + O\big((s\epsilon)^{m+1}\big). \tag{7.10}$$

Therefore, for small ϵ the time stepsize s can be taken quite large. For more details, refer to [38].

§8 Contact Algorithms for Contact Systems

Contact systems occur only on space $\mathbf{R}^{2n+1}$ of odd dimensions. We use 3-symbol notation to denote the coordinates and vectors on $\mathbf{R}^{2n+1}$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}, \qquad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad z = (z),$$

$$\begin{pmatrix} a(x,y,z) \\ b(x,y,z) \\ c(x,y,z) \end{pmatrix}, \qquad a = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}, \quad c = (c).$$

A *contact* system is defined by a function $K(x, y, z)$, called contact Hamiltonian, on $\mathbf{R}^{2n+1}$ as follows

$$\begin{aligned} \frac{dx}{dt} &= -K_y + K_z x = a, \\ \frac{dy}{dt} &= K_x = b, \\ \frac{dz}{dt} &= K_e = c, \quad K_e(x,y,z) := K(x,y,z) - \langle x, K_x(x,y,z)\rangle. \end{aligned} \tag{8.1}$$

The right hand side is the general form of a *contact* vector field.

Intrinsic to all contact systems is the *contact* structure on $\mathbf{R}^{2n+1}$ defined by the differential 1-form

$$\alpha = \sum_{i=1}^{n} x_i dy_i + dz = (0 \ x^T \ 1) \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} \tag{8.2}$$

up to an everywhere non-vanishing multiplier function. A *contact* transformation f is a diffeomorphism on $\mathbf{R}^{2n+1}$

$$f: \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} \to \begin{pmatrix} \hat{x}(x,y,z) \\ \hat{y}(x,y,z) \\ \hat{z}(x,y,z) \end{pmatrix}$$

preserving the contact structure, that means

$$\sum_{i=1}^{n} \hat{x}_i d\hat{y}_i + d\hat{z} = \mu_f \left(\sum_{i=1}^{n} x_i dy_i + dz \right) \tag{8.3}$$

for some function $\mu_f \neq 0$ everywhere on $\mathbf{R}^{2n+1}$, called the multiplier of f. The explicit expression of (8.3) is

$$(0 \ \hat{x}^T \ 1) \begin{pmatrix} \hat{x}_x & \hat{x}_y & \hat{x}_z \\ \hat{y}_x & \hat{y}_y & \hat{y}_z \\ \hat{z}_x & \hat{z}_y & \hat{z}_z \end{pmatrix} = \mu_f (0 \ x^T \ 1).$$

All contact vector fields on $\mathbf{R}^{2n+1}$ form a Lie algebra $\mathbf{L} = \mathbf{KV}_{2n+1}$ under Lie bracket. The associated Lie group $\mathbf{G} = \mathbf{KD}_{2n+1}$ consists of all near-identity contact transformations on $\mathbf{R}^{2n+1}$. For $f \in \mathbf{G}$, $\mu_f > 0$ everywhere. The phase flows e^t_K of contact systems defined by contact Hamiltonians K on $\mathbf{R}^{2n+1}$ are contact transformations. So the proper algorithms for contact systems should be *contact* too. For elements of contact geometry, see [1].

All the existing numerical methods including the symplectic ones are generically non-contact even for linear contact systems. A linear contact system is necessarily of the form,

$$\begin{aligned} \frac{dx}{dt} &= (\lambda I - L^T)x, \qquad L \in gl(n), \quad \lambda \in \mathbf{R}, \\ \frac{dy}{dt} &= Ly, \\ \frac{dz}{dt} &= \lambda z, \end{aligned} \tag{8.4}$$

the 3 equations are independent. The matrix of a linear contact transformation is necessarily of the block-diagonal form

$$\begin{pmatrix} \mu M^{-T} & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & \mu \end{pmatrix}, \qquad M \in \mathbf{GL}(n), \quad \mu > 0. \tag{8.5}$$

Using the trapezoidal method (identical with the centered Euler method in linear cases), one gets

$$\hat{x} = \left(\left(1-\frac{s\lambda}{2}\right)I+\frac{s}{2}L^T\right)^{-1}\left(\left(1+\frac{s\lambda}{2}\right)I-\frac{s}{2}L^T\right)x =: Nx,$$
$$\hat{y} = \left(I-\frac{s}{2}L\right)^{-1}\left(I+\frac{s}{2}L\right) =: My,$$
$$\hat{z} = \left(1-\frac{s\lambda}{2}\right)^{-1}\left(1+\frac{s\lambda}{2}\right)z =: \mu z.$$

The map is contact $\Longleftrightarrow N = \mu M^{-T}, \forall s \Longleftrightarrow \lambda = 0$ or $L^2 = \lambda L$; but this is highly exceptional.

The construction of explicit Euler-type structure preserving algorithms in §3 is applicable to contact systems when the vector field is contact-separable, i.e., decomposable into

$$a = \sum_{i=1}^{n} a^i, \qquad a^i \in \mathbf{KL}_{2n+1}, \quad a^i_* a^i = 0, \quad i = 1, \cdots, n. \tag{8.6}$$

Our main concern will be the construction of unconditional contact algorithms for general contact systems. The method is based on the well-known correspondence between contact geometry on $\mathbf{R}^{2n+1}$ and homogeneous (or conic) symplectic geometry on $\mathbf{R}^{2n+2}$.

We use 4-symbol notation for the coordinates on $\mathbf{R}^{2n+2}$

$$\begin{pmatrix} p_0 \\ p_1 \\ q_0 \\ q_1 \end{pmatrix} \in \mathbf{R}^{2n+2}, \quad p_0 = (p_0), \; q_0 = (q_0), \quad p_1 = \begin{pmatrix} p_{11} \\ \vdots \\ p_{1n} \end{pmatrix}, \quad q_1 = \begin{pmatrix} q_{11} \\ \vdots \\ q_{1n} \end{pmatrix}.$$

Consider ***homogeneous*** near-identity transformation $g : (p_0, p_1, q_0, q_1) \to (\hat{p}_0, \hat{p}_1, \hat{q}_0, \hat{q}_1)$ and ***homogeneous*** function $H(p_0, p_1, q_0, q_1)$ on $\mathbf{R}^{2n+2}$, they are defined for $p_0 \neq 0$ and satisfy the conditions

$$\begin{aligned} \forall \lambda \neq 0 : \qquad & \hat{p}_i(\lambda p_0, \lambda p_1, q_0, q_1) = \lambda \hat{p}_i(p_0, p_1, q_0, q_1), \\ & \hat{q}_i(\lambda p_0, \lambda p_1, q_0, q_1) = \hat{q}_i(p_0, p_1, q_0, q_1), \qquad i = 1, 2, \\ & H(\lambda p_0, \lambda p_1, q_0, q_1) = \lambda H(p_0, p_1, q_0, q_1). \end{aligned}$$

They depend essentially only on $2n+1$ variables

$$\begin{aligned} p_0 \neq 0 : \qquad & \hat{p}_i(p_0, p_1, q_0, q_1) = p_0 \hat{p}_i(1, \frac{p_1}{p_0}, q_0, q_1), \\ & \hat{q}_i(p_0, p_1, q_0, q_1) = \hat{q}_i(1, \frac{p_1}{p_0}, q_0, q_1), \\ & H(p_0, p_1, q_0, q_1) = p_0 H(1, \frac{p_1}{p_0}, q_0, q_1). \end{aligned}$$

The phase flow e^t_H for Hamiltonian system defined by homogeneous function H on $\mathbf{R}^{2n+2}$ is homogeneous and symplectic.

Near-identity homogeneous symplectic transformation g and homogeneous function H on $\mathbf{R}^{2n+2}$ can be put in correspondence with near-identity contact transformation f and function K on $\mathbf{R}^{2n+1}$ as follows:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \to \begin{pmatrix} p_0 \\ p_1 \\ q_0 \\ q_1 \end{pmatrix} = \begin{pmatrix} p_0 \\ p_0 x \\ z \\ y \end{pmatrix} \xrightarrow{g} \begin{pmatrix} \hat{p}_0 \\ \hat{p}_1 \\ \hat{q}_0 \\ \hat{q}_1 \end{pmatrix} = \begin{pmatrix} \hat{p}_0 \\ \hat{p}_0 \hat{x} \\ \hat{z} \\ \hat{y} \end{pmatrix}, \tag{8.7}$$

where $p_0 > 0$ is an arbitrary chosen parameter. Then $\hat{p}_0 > 0$ and the map

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \xrightarrow{f} \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} = \begin{pmatrix} \hat{p}_1/\hat{p}_0 \\ \hat{q}_1 \\ \hat{q}_0 \end{pmatrix}, \quad \mu_f := \hat{p}_0/p_0 > 0 \tag{8.8}$$

is a near-identity contact transformation independent of p_0 and μ_f is the multiplier of f.

$$\begin{aligned}
&(x,y,z) \longrightarrow K(x,y,z) := H(1,x,z,y), \\
&H(p_0,p_1,q_0,q_1) = p_0 K(\frac{p_1}{p_0}, q_1, q_0). \\
&H_{p_0}(p_0,p_1,q_0,q_1) = K(x,y,z) - \langle x, K_x(x,y,z)\rangle = K_e(x,y,z) \\
&H_{p_1}(p_0,p_1,q_0,q_1) = K_x(x,y,z), \\
&H_{q_0}(p_0,p_1,q_0,q_1) = p_0 K_z(x,y,z), \\
&H_{q_1}(p_0,p_1,q_0,q_1) = p_0 K_y(x,y,z), \\
&x = p_1/p_0, \quad y = q_1, \quad z = q_0.
\end{aligned}$$

The above process and its inverse are called *contactization* and *symplectization* respectively. Under this correspondence we have

1. $K(x,y,z) \longleftrightarrow H(p_0,p_1,q_0,q_1)$.
2. Contact system (8.1) $\longleftrightarrow$ homogeneous Hamiltonian system

$$\begin{aligned}
\frac{dp_0}{dt} &= -H_{q_0}, & \frac{dq_0}{dt} &= H_{p_0}, \\
\frac{dp_1}{dt} &= -H_{q_1}, & \frac{dq_1}{dt} &= H_{p_1}.
\end{aligned} \tag{8.9}$$

3. Contact phase flow e^t_K on $\mathbf{R}^{2n+1}$ $\longleftrightarrow$ homogeneous symplectic phase flow e^t_H on $\mathbf{R}^{2n+2}$

Now symplectic phase flow e^t_H can be approximated by various unconditional symplectic algorithms, e.g., $g^s_{H,c}$ generated by (5.4) and (6.10) on $\mathbf{R}^{2n+2}$. For weight matrices c on $\mathbf{R}^{2n+2}$ compatible with homogeneity conditions of functions and mappings, $g^s_{H,c}$ are homogeneous symplectic, the generating relations (5.4) and (6.10) are contactized to give the contactization $f^s_{K,c}$ on $\mathbf{R}^{2n+2}$ of $g^s_{H,c}$. They provide unconditional contact algorithms for contact systems. We give some basic contact algorithms in the following. For more details and the theory of contact generating functions see [12–13].

$\widetilde{C}$. $c = \frac{1}{2} I_{2n+2}$, contact version of method C, two stage form (write $\bar{K}_x = K_x(\bar{x},\bar{y},\bar{z})$, etc.)

$$\begin{aligned}
&\bar{x} = x + \frac{s}{2}(-\overline{K}_y + x\overline{K}_z), \quad \bar{y} = y + \frac{s}{2}\overline{K}_x, \quad \bar{z} = z + \frac{s}{2}\overline{K}_e, \\
&\hat{x} = (\bar{x} - \frac{s}{2}\overline{K}_y)/(1 - \frac{s}{2}\overline{K}_z), \quad \hat{y} = 2\bar{y} - y, \quad \hat{z} = 2\bar{z} - z.
\end{aligned} \tag{8.10}$$

$\widetilde{P}$. $c = \begin{pmatrix} I_{n+1} & 0 \\ 0 & 0_{n+1} \end{pmatrix}$, contact analog of symplectic method P:

$$\text{1-stage form}: \quad \begin{aligned} &\hat{x} = x + s(-K_y(\hat{x}, y, z) + xK_z(\hat{x}, y, z)), \\ &\hat{y} = y + sK_x(\hat{x}, y, z), \\ &\hat{z} = z + sK_e(\hat{x}, y, z). \end{aligned} \tag{8.11}$$

$$\text{2-stage form}: \quad \begin{aligned} &\bar{x} = x + s(-\overline{K}_y + x\overline{K}_z), \quad \bar{y} = y, \quad \bar{z} = z, \\ &\hat{x} = \bar{x}, \quad \hat{y} = \bar{y} + s\overline{K}_x, \quad \hat{z} = \bar{z} + s\overline{K}_e. \end{aligned}$$

$\widetilde{Q}$. $c = \begin{pmatrix} 0_{n+1} & 0 \\ 0 & I_{n+1} \end{pmatrix}$, contact analog of symplectic method Q:

$$\text{1-stage form}: \quad \begin{aligned} &\hat{x} = x + s(-K_y(x, \hat{y}, \hat{z}) + \hat{x}K_z(x, \hat{y}, \hat{z})), \\ &\hat{y} = y + sK_x(x, \hat{y}, \hat{z}), \\ &\hat{z} = z + sK_e(x, \hat{y}, \hat{z}). \end{aligned} \tag{8.12}$$

$$\text{2-stage form}: \quad \begin{aligned} &\bar{x} = x, \quad \bar{y} = y + s\overline{K}_x, \quad \bar{z} = z + s\overline{K}_e, \\ &\hat{x} = \bar{x} + s(-\overline{K}_y + \hat{x}\overline{K}_z), \quad \hat{y} = \bar{y}, \quad \hat{z} = \bar{z}. \end{aligned}$$

One might suggest, by analogy with (5.4), for example, the following scheme for (8.1):

$$\hat{x} = x + sa(\hat{x}, y, z), \quad \hat{y} = y + sb(\hat{x}, y, z), \quad \hat{z} = z + sc(\hat{x}, y, z).$$

It differs from (8.11) only in one term for $\hat{x}$, i.e., $\hat{x}K(\hat{x}, y, z)$ instead of $xK(\hat{x}, y, z)$. This minute but delicate difference makes (8.11) contact and other non-contact! For further developments of contact algorithms, see [10, 12–13]. For numerical comparisons of 3D contact vs non-contact computations, see [10].

§9 Volume-preserving Algorithms for Source-free Systems

The source-free systems on $\mathbf{R}^m$ are governed by the m-D volume structure, the underlying Lie algebra $\mathbf{L} = \mathbf{SV}_m$ consists of all *source-free* vector fields a, $\operatorname{div} a = \operatorname{tr} a_* = 0$, the underlying Lie group $\mathbf{G} = \mathbf{SD}_m$ consists of all near-identity *volume-preserving* transformations g, $\det g_* = 1$.

The phase flow e_a^t of source-free field a is always volume preserving. Hence the proper algorithms for source-free systems should be also *volume-preserving*, since otherwise the dynamics would be polluted by artificial sources and sinks.

For $m = 2$, source-free fields = Hamiltonian fields, area-preserving maps = symplectic maps; so the problem for area-preserving algorithms has been solved in principle.

For $m \geq 3$, the problem is new, since all the conventional methods plus even the symplectic methods are generically not volume-preserving, even for linear source-free systems. As an illustration, solve on $\mathbf{R}^3$

$$\frac{dx}{dt} = Ax, \qquad \operatorname{tr} A = 0$$

by the method C, we get

$$x \to \hat{x} = G^s x, \quad G^s = \Big(I - \frac{s}{2}A\Big)^{-1}\Big(I + \frac{s}{2}A\Big).$$

$\det G^s = 1 \Longleftrightarrow \det A = 0$, which is exceptional.

The construction of explicit Euler type structure-preserving algorithms by composition is applicable here when the vector field is source-free separable, i.e., decomposable as

$$a = \sum_{i=1}^{n} a^i, \quad \operatorname{div} a^i = 0, \quad a^i_* a^i = 0, \quad i = 1, \cdots, n. \tag{9.1}$$

A special case is

$$\begin{aligned} &a = (a_1, \cdots, a_m), \quad \frac{\partial a_k}{\partial x_k} = 0, \quad a^k = (0, \cdots, 0, a_k, 0, \cdots, 0), \quad k = 1, \cdots, m; \\ &a = \sum_{k=1}^{m} a^k. \end{aligned} \tag{9.2}$$

Example 1. Euler equation for free rigid body and Jacobian elliptic functions of modulus k,

$$\begin{aligned} \frac{dx_1}{dt} &= c_1 x_2 x_3 = a_1, \\ \frac{dx_2}{dt} &= c_2 x_1 x_3 = a_2, \\ \frac{dx_3}{dt} &= c_3 x_1 x_2 = a_3, \end{aligned} \qquad \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} c_1 x_2 x_3 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ c_2 x_1 x_3 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ c_3 x_1 x_2 \end{pmatrix}$$

where (x_1, x_2, x_3) are angular momenta along principle axes of inertia with diagonal elements I_1, I_2, I_3, $c_1 = (I_2 - I_3)/I_2 I_3$, cyclic. Take $c_1 = 1$, $c_2 = -1$, $c_3 = -k^2$, it gives elliptic functions for $x_1(0) = 0$, $x_2(0) = x_3(0) = 1$, $x_1(t) = \operatorname{sn}(t, k)$, $x_2(t) = \operatorname{cn}(t, k)$, $x_3(t) = \operatorname{dn}(t, k)$.

Example 2. ABC flows [1]

$$\begin{aligned} \frac{dx_1}{dt} &= A \sin x_3 + C \cos x_2 = a_1, \\ \frac{dx_2}{dt} &= B \sin x_1 + A \cos x_3 = a_2, \\ \frac{dx_3}{dt} &= C \sin x_2 + B \cos x_1 = a_3, \end{aligned}$$

$$\begin{aligned} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} &= \begin{pmatrix} A \sin x_3 + C \cos x_2 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ B \sin x_1 + A \cos x_3 \\ 0 \end{pmatrix} \\ &\quad + \begin{pmatrix} 0 \\ 0 \\ C \sin x_2 + B \cos x_1 \end{pmatrix} \\ &= \begin{pmatrix} A \sin x_3 \\ A \cos x_3 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ B \sin x_1 \\ B \cos x_1 \end{pmatrix} + \begin{pmatrix} C \sin x_2 \\ 0 \\ C \cos x_2 \end{pmatrix}. \end{aligned}$$

All decompositions on the right satisfy (9.1), then (3.10) gives corresponding volume-preserving revertible algorithms.

In $\mathbf{R}^m$ all source-free fields have "vector potential" representations: under such representation, a field always decomposes into a sum of essentially 2-D source-free fields, for which essentially area-preserving algorithms can always be constructed, then one can apply the decomposition-composition method in §3 to obtain m-D volume-preserving algorithms [16, 39].

In $\mathbf{R}^2$, every source-free field (a_1, a_2) corresponds to a stream function or 2-D Hamiltonian ψ, unique up to a constant:

$$a_1 = -\frac{\partial\psi}{\partial x_2}, \qquad a_2 = \frac{\partial\psi}{\partial x_1}, \tag{9.3}$$

then area-preserving algorithms = 2-D symplectic algorithms.

In $\mathbf{R}^3$, every source-free field (a_1, a_2, a_3) corresponds to a vector potential (b_1, b_2, b_3), unique up to a gradient:

$$a = \operatorname{curl} b, \quad a_1 = \frac{\partial b_3}{\partial x_2} - \frac{\partial b_2}{\partial x_3}, \quad a_2 = \frac{\partial b_1}{\partial x_3} - \frac{\partial b_3}{\partial x_1}, \quad a_3 = \frac{\partial b_2}{\partial x_1} - \frac{\partial b_1}{\partial x_2}, \tag{9.4}$$

then one get source-free decomposition

$$\begin{aligned} a &= (a_1, a_2, a_3) \\ &= \Big(0, \frac{\partial b_1}{\partial x_3}, -\frac{\partial b_1}{\partial x_2}\Big) + \Big(-\frac{\partial b_2}{\partial x_3}, 0, \frac{\partial b_2}{\partial x_1}\Big) + \Big(\frac{\partial b_3}{\partial x_2}, -\frac{\partial b_3}{\partial x_1}, 0\Big) \\ &= a^1 + a^2 + a^3. \end{aligned} \tag{9.5}$$

Each field a^i is 2-D source-free and 0 in the 3rd dimension, apply area-preserving method P, say, with 3rd dimension fixed, then one get three volume-preserving maps

$$\begin{aligned} P_1^s &\approx e^s(a^1) = e^s\Big(0, \frac{\partial b_1}{\partial x_3}, -\frac{\partial b_1}{\partial x_2}\Big), \quad \text{ord 1, volume-preserving}, \\ P_2^s &\approx e^s(a^2) = e^s\Big(-\frac{\partial b_2}{\partial x_3}, 0, \frac{\partial b_2}{\partial x_1}\Big), \quad \text{ord 1, volume-preserving}, \\ P_3^s &\approx e^s(a^3) = e^s\Big(\frac{\partial b_3}{\partial x_2}, -\frac{\partial b_3}{\partial x_1}, 0\Big), \quad \text{ord 1, volume-preserving}, \\ P_1^s &= E^s\Big(0, \frac{\partial b_1}{\partial x_3}, 0\Big) I^s\Big(0, 0, -\frac{\partial b_1}{\partial x_2}\Big), \\ \check{P}_1^s &= E^s\Big(0, 0, -\frac{\partial b_1}{\partial x_2}\Big) I^s\Big(0, \frac{\partial b_1}{\partial x_3}, 0\Big), \\ P_2^s &= E^s\Big(0, 0, \frac{\partial b_2}{\partial x_1}\Big) I^s\Big(-\frac{\partial b_2}{\partial x_3}, 0, 0\Big), \\ \check{P}_2^s &= E^s\Big(-\frac{\partial b_2}{\partial x_3}, 0, 0\Big) I^s\Big(0, 0, \frac{\partial b_2}{\partial x_1}\Big), \\ P_3^s &= E^s\Big(\frac{\partial b_3}{\partial x_2}, 0, 0\Big) I^s\Big(0, -\frac{\partial b_3}{\partial x_1}, 0\Big), \\ \check{P}_3^s &= E^s\Big(0, -\frac{\partial b_3}{\partial x_1}, 0\Big) I^s\Big(\frac{\partial b_3}{\partial x_2}, 0, 0\Big), \end{aligned} \tag{9.6}$$

then one get volume-preserving algorithms for field a

$$\begin{aligned} f^s &= P_1^s P_2^s P_3^s \approx e^s(a), \quad \text{ord 1, non-revertible}, \\ g^s &= P_1^{s/2} P_2^{s/2} P_3^{s/2} \circ \check{P}_3^{s/2} \check{P}_2^{s/2} \check{P}_1^{s/2} \approx e^s(a), \ \text{ord 2, revertible}, \\ h^s &= g^{\alpha s} g^{\beta s} g^{\alpha s} \approx e^s(a), \quad \text{ord 4, revertible}. \end{aligned} \tag{9.7}$$

The basic elements of all these composites consist only in 6 explicit Euler and 6 implicit Euler in one variable appeared in (9.6).

On $\mathbf{R}^m$, to every source-free field (a_i), there exists, as a generalization of cases $m = 2$, 3, skew-symmetric tensor field of order 2, $b = (b_{ik})$, $b_{ik} = -b_{ki}$ so that

$$a_i = \sum_{k=1}^{m} \frac{\partial b_{ik}}{\partial x_k}, \qquad i = 1, \cdots, m. \tag{9.8}$$

By (9.8) we can decompose

$$a = \sum_{i<k} a_{(ik)}, \ a_{(ik)} = \left(0, \cdots, 0, \frac{\partial b_{ik}}{\partial x_k}, 0, \cdots, 0, -\frac{\partial b_{ik}}{\partial x_i}, 0, \cdots, 0\right)^T, \ i < k. \tag{9.9}$$

Every vector field $a_{(ik)}$ is 2-D source-free on the i-k plane and 0 in other dimensions so we can approximate the phase flow of $a_{(ik)}$ by area-preserving method which implies m-D volume-preserving, similar to the cases discussed above for $m = 3$. For more details see [16].

The tensor potential (b_{ik}) for a given (a_i) is by far not unique. For uniqueness one may impose normalizing conditions in many different ways. One way is to impose

$$N_0: \quad b_{ik} = 0, \ |i - k| \geq 2, \tag{9.10}$$

(this condition is ineffective for $m = 2$). The non-zero components are

$$b_{12} = -b_{21}, \quad b_{23} = -b_{32}, \quad \cdots \quad b_{m-1,m} = -b_{m,m-1}.$$
$$N_1: \quad b_{12}|_{x_1=x_2=0} = 0.$$
$$N_k, \ 2 \leq k \leq m-1: \quad b_{k,k+1}|_{x_k=0} = 0,$$

(this condition is ineffective for $m = 2$). Then all $b_{k,k+1}$ are uniquely determined by quadrature

$$b_{12} = \int_0^{x_2} a_1 \, dx_2 - \int_0^{x_1} a_2\Big|_{x_2=0} dx_1,$$
$$b_{k,k+1} = -\int_0^{x_k} a_{k+1} \, dx_k, \quad 2 \leq k \leq m-1; \tag{9.11}$$
$$a = \sum_{k=1}^{m-1} a^k, \quad a^k = (0, \cdots, 0, \frac{\partial b_{k,k+1}}{\partial x_{k+1}}, -\frac{\partial b_{k,k+1}}{\partial x_k}, 0, \cdots, 0)^T. \tag{9.12}$$

When the vector field a is polynomial, then by (9.10) and (9.11) the tensor components b_{ik} are polynomials. Then the theorem on symplectic separability of all polynomial Hamiltonians (4.16) implies the source-free separability in the sense of (9.1) for all polynomial source-free systems. For numerical comparisons of 3D volume-preserving vs non-volume-preserving computations, see [10].

§10 Formal Dynamical Systems and Numerical Algorithms

In order to facilitate the study of the problem $f^s \approx e^s_a$, the apparatus of formal power series is useful. We first introduce some notation. For $a = (a_1, \cdots, a_m)^T : \mathbf{R}^m \to \mathbf{R}^m$,

$$a^* := \sum a_i \frac{\partial}{\partial x_i} = \text{ linear differential operator of first order associated to } a.$$

The association $a \to a^*$ is linear, a^* operates on scalar functions $\phi : \mathbf{R}^m \to \mathbf{R}$ and on vector functions $b : \mathbf{R}^m \to \mathbf{R}^m$ as

$$a^*\phi = \sum a_i \frac{\partial \phi}{\partial x_i}$$
$$a^*b = a^*(b_1, \cdots, b_m)^T = (a^*b_1, \cdots, a^*b_m)^T = b_*a, \quad a^*1_m = a.$$

Multiple applications of linear differential operators are naturally defined such as a^*b^*, $(a^*b^*)c^*$, $a^*(b^*c^*)$, etc. The operations are multilinear, associative but non-commutative; thus powers can be defined

$$a^{*k} = a^*a^* \cdots a^* (k \text{ times}), \qquad a^k := a^{*k}1_m$$

The identity operator I operates on scalar and vector functions ϕ and b as $I\phi = \phi$, $Ib = b$. Lie bracket of vector functions a, b on $\mathbf{R}^m$ can be written as

$$[a, b] := a_*b - b_*a = b^*a - a^*b = (b^*a^* - a^*b^*)1_m.$$

Among the formal power series $\sum_0^\infty s^k a_k$, $a_k : \mathbf{R}^m \to \mathbf{R}^m$, we pick out two special classes. The first class consists of those with $a_0 = 0$, called near-0 formal vector fields; the second class consists of those with $a_0 = 1_m$, called near-1 formal maps (diffeomorphisms).

All near-0 formal vector fields $a^s = \sum\limits_1^\infty s^k a_k$ form a (∞-dim.) real Lie algebra $\mathbf{FV}_m$ under the Lie bracket

$$[a^s, b^s] = \Big[\sum_1^\infty s^k a_k, \sum_1^\infty s^k b_k\Big] := \sum_{k=2}^\infty s^k \sum_{i+j=k} [a_i, b_j].$$

The associated near-0 formal differential operators and their products are

$$(a^s)_* := \Big(\sum_1^\infty s^k a_k\Big)_* := \sum_1^\infty s^k a_{k*}$$
$$a^{s*} = \Big(\sum_1^\infty s^k a_k\Big)^* := \sum_1^\infty s^k a_k^*,$$
$$a^{s*}b^{s*} := \sum_2^\infty s^k \sum_{i+j=k} a_i^* b_j^*, \quad (a^{s*})^2 := a^{s*}a^{s*}, \text{ etc.}$$

For any vector function $a = (a_1, \cdots, a_m)^T : \mathbf{R}^m \to \mathbf{R}^m$ and any near-1 formal map $g^s = 1 + \sum s^k g_k$ we define the composition

$$(a \circ g^s)(x) = a(g^s(x)) = a(x) + \sum_1^\infty s^k (a \circ g)_k(x),$$
$$(a \circ g)_k = \sum_{n=1}^k \sum_{j_1 + \cdots + j_n = k} \frac{1}{n!}(D^n a)(g_{j_1}, \cdots, g_{j_n}),$$

where

$$D^n a = (D^n a_1, \cdots, D^n a_m)^T,$$
$$D^n a_i(v_1, \cdots, v_n) = \sum_{j_1, \cdots, j_n = 1}^m \frac{\partial^n a_i}{\partial x_{j_1} \cdots \partial x_{j_n}} v_{1j_1} \cdots v_{nj_n}$$

is the usual n-th differential multi-linear form for n tangent vectors $v_j = (v_{j_1}, \cdots, v_{j_m})^T$, $j = 1, \cdots, n$ at point $x \in \mathbf{R}^m$ which is invariant under permutation of vectors. Using the identities

$$\begin{aligned}
&(D^1 a)(b) = b^* a, \\
&(D^2 a)(b, c) = (c^* b^* - (c^* b)^*) a, \\
&(D^3 a)(b, b, b) = (b^{*3} + 2b^{3*} - 3b^* b^{2*}) a,
\end{aligned}$$

we get in particular

$$\begin{aligned}
&(a \circ g)_1 = g_1^* a, \\
&(a \circ g)_2 = g_2^* a + \frac{1}{2}(g_1^{*2} - g_1^{2*}) a, \\
&(a \circ g)_3 = g_3^* a + ((g_2^* g_1^* - (g_2^* g_1)^*) a + \frac{1}{3!}(g_1^{*3} + 2g_1^{3*} - 3g_1^* g_1^{2*}) a.
\end{aligned}$$

For any two near-1 formal maps $f^s = 1_m + \sum s^k a_k$, $g^s = 1_m + \sum s^k g_k$, the composition $f^s \circ g^s$ is defined in a term by term way:

$$\begin{aligned}
(f^s \circ g^s)(x) &= f^s(g^s(x)) = 1_m(g^s(x)) + \sum_{k=1}^{\infty} s^k f_k(g^s(x)) \\
&=: 1_m(x) + \sum_{k=1}^{\infty} s^k (f \circ g)_k(x), \\
(f \circ g)_1 &= f_1 + g_1, \\
(f \circ g)_k &= f_k + g_k + \delta(f_1, \cdots, f_{k-1}; g_1, \cdots, g_{k-1}), \quad k \geq 2, \\
\delta(f_1, \cdots, f_{k-1}; g_1, \cdots, g_{k-1}) &= \sum_{n=1}^{k-1} \sum_{j=1}^{n} \sum_{i_1 + \cdots + i_j = n} \frac{1}{j!}(D^j f_{k-n})(g_{i_1}, \cdots, g_{i_j}),
\end{aligned}$$

we get in particular,

$$\begin{aligned}
(f \circ g)_2 =& f_2 + g_2 + g_1^* f_1, \\
(f \circ g)_3 =& f_3 + g_3 + g_1^* f_2 + g_2^* f_1 + \frac{1}{2}(g_1^{*2} - g_1^{2*}) f_1, \\
(f \circ g)_4 =& f_4 + g_4 + g_1^* f_3 + g_2^* f_2 + g_3^* f_1 + \frac{1}{2}(g_1^{*2} - g_1^{2*}) f_2 \\
&+ (g_2^* g_1^* - (g_2^* g_1)^*) f_1 + \frac{1}{3!}(g_1^{*3} + 2g_1^{3*} - 3g_1^* g_1^{2*}) f_1.
\end{aligned}$$

Under this composition rule, all near-1 formal maps $f^s = 1 + \sum_1^{\infty} s^k f_k$ form a (∞-dim) formal Lie group $\mathbf{FD}_m$. In group $\mathbf{FD}_m$ inverse elements, square roots, rational powers, etc., always exist, their coefficients can always be determined recursively by the defining composition relations. For example, the inverse $(f^s)^{-1} := 1 + \sum s^k h_k = h^s$ is defined by $(f^s \circ h^s) = 1_N$, hence

$$f_1 + h_1 = 0, \quad f_k + h_k + \delta(f_1, \cdots, f_{k-1}; h_1, \cdots, h_{k-1}) = 0, \quad k \geq 2.$$

In particular,

$$\begin{aligned}
h_1 &= -f_1, \\
h_2 &= -f_2 + f_1^2, \\
h_3 &= -f_3 + f_1^* f_2 + (f_2^* - f_1^{2*}) f_1 - \frac{1}{2} f_1^3 + \frac{1}{2} f_1^{2*} f_1.
\end{aligned}$$

There is an obvious one-one correspondence between the Lie algebra $\mathbf{FV}_m$ and the Lie group $\mathbf{FD}_m$, established simply by $+1_m$ and by -1_m. However, the more significant one-one correspondence between them is given by exp and its inverse log.

$$\begin{aligned}
&\exp : \mathbf{FV}_m \to \mathbf{FD}_m \\
&a^s = \sum_1^\infty s^k a_k \to \exp a^s := 1_m + \sum_{k=1}^\infty \frac{1}{k!} (a^{s*})^k 1_m \\
&\qquad\qquad\qquad\qquad =: 1_m + \sum_1^\infty s^k f_k = f^s. \qquad (10.1)
\end{aligned}$$

Note that

$$(a^{s*})^k = \left(\sum s^{i_1} a_{i_1}^* \right) \cdots \left(\sum s^{i_k} a_{i_k}^* \right) = \sum_{i_1, \cdots, i_k = 1}^\infty s^{i_1 + \cdots + i_k} a_{i_1}^* \cdots a_{i_k}^*,$$

so we get easily

$$\begin{aligned}
f_k &= \sum_{n=1}^k \frac{1}{n!} \sum_{k_1 + \cdots + k_n = k} a_{k_1}^* \cdots a_{k_n}^* 1_m, \qquad k \geq 1, \quad f_1 = a_1, \qquad (10.2) \\
f_k &= a_k + \sum_{n=2}^k \frac{1}{n!} \sum_{k_1 + \cdots + k_n = k} a_{k_1}^* \cdots a_{k_n}^* 1_m, \qquad k \geq 2, \quad f_2 = a_2 + \frac{1}{2} a_1^2.
\end{aligned}$$

Note that (10.2) provides a 2-way recursion formula from $a_1, \cdots, a_k$ to $f_1, \cdots, f_k$ and vice versa. Therefore exp maps $\mathbf{FV}_m$ one-one onto $\mathbf{FD}_m$ and its inverse, i.e., log is defined by the same (10.2):

$$\log = (\exp)^{-1} : \mathbf{FD}_m \to \mathbf{FV}_m, \quad \log \exp a^s = a^s, \quad \exp \log f^s = f^s.$$

In particular

$$\begin{aligned}
a_1 &= f_1, \quad a_2 = f_2 - \frac{1}{2} a_1^2, \quad a_3 = f_3 - \frac{1}{2} (a_1^* a_2 + a_2^* a_1) - \frac{1}{3!} a_1^3, \\
a_4 &= f_4 - \frac{1}{2} (a_1^* a_3 + a_2^2 + a_3^* a_1) \\
&\quad - \frac{1}{3!} (a_1^* a_1^* a_2 + a_1^* a_2^* a_1 + a_2^* a_1^* a_1) - \frac{1}{4!} a_1^4, \qquad (10.3) \\
a_k &= f_k - \sum_{n=2}^{k-1} \frac{1}{n!} \sum_{k_1 + \cdots + k_n = k} a_{k_1}^* \cdots a_{k_n}^* 1_m - \frac{1}{k!} a_1^k, \quad k \geq 3.
\end{aligned}$$

An equivalent construction of $\log f^s = a^s$ is

$$\log f^s = \sum_{k=1}^\infty \frac{(-1)^{k-1}}{k} h_k^s, \qquad (10.4)$$

where

$$h_1^s = f^s - 1_m, \quad h_k^s = h_{k-1}^s \circ f^s - h_{k-1}^s.$$

It is easy to compute

$$\begin{aligned}
h_1^s &= \sum_{k=1}^{\infty} s^k f_k = \sum_{k_1=1}^{\infty} s^{k_1} (1_m \circ f)_{k_1} \\
h_2^s &= \sum_{k_1=k_2=1}^{\infty} s^{k_1+k_2} ((1_m \circ f)_{k_1} \circ f)_{k_2}, \\
h_3^s &= \sum_{k_1,k_2,k_3=1}^{\infty} s^{k_1+k_2+k_3} (((1_m \circ f)_{k_1} \circ f)_{k_2} \circ f)_{k_3}, \text{ etc.}
\end{aligned}$$

Substituting in (10.4) and equating with $\sum_1^\infty s^k a_k$, we get

$$a_k = \sum_{n=1}^{k} \frac{(-1)^{n-1}}{n} \sum_{k_1+\cdots+k_n=k} (\cdots((1_m \circ f)_{k_1} \circ f)_{k_2} \cdots \circ f)_{k_n}. \tag{10.5}$$

It is easy to verify $\log \exp a^s = a^s$ for this log, so this is precisely the inverse of exp, thus agrees with the previous one.

We use the above construction (10.5) to establish the *formal* Campbell-Hausdorff formula:
For arbitrary near-1 formal maps f^s, g^s

$$\log(f^s \circ g^s) = \log f^s + \log g^s + \sum_{k=1}^{\infty} d_k(\log f^s, \log g^s), \tag{10.6}$$

where, for $\log f^s = a^s$, $\log g^s = b^s$,

$$d_k(a^s, b^s) = \frac{1}{k} \sum_{n=1}^{k} \frac{(-1)^{n-1}}{n} \sum_{\substack{p_1+q_1+\cdots+p_n+q_n=k \\ p_i+q_i\geq 1,\, p_i\geq 0,\, q_i\geq 0}} \frac{[(a^s)^{p_1}(b^s)^{q_1} \cdots (a^s)^{p_n}(b^s)^{q_n}]}{p_1!q_1!\cdots p_n!q_n!} \tag{10.7}$$

where

$$(x)^p = xx\cdots x \ (p\text{-times}), \quad [x_1x_2x_3\cdots x_n] = [[\cdots[[x_1,x_2],x_3],\cdots],x_n].$$

In particular

$$d_1 = \frac{1}{2}[a^s, b^s], \quad d_2 = \frac{1}{12}([a^s b^s b^s] + [b^s a^s a^s]), \quad d_3 = -\frac{1}{24}[a^s b^s b^s a^s].$$

Let $\log(f^s \circ g^s) = c^s = \sum_1^\infty s^k c_k$. Then

$$\begin{aligned}
c_1 =& a_1 + b_1, \qquad c_2 = a_2 + b_2 + \frac{1}{2}[a_1 b_1], \\
c_3 =& a_3 + b_3 + \frac{1}{2}([a_1 b_2] + [a_2 b_1]) + \frac{1}{12}([a_1 b_1 b_1] + [b_1 a_1 a_1]),
\end{aligned}$$

$$\begin{aligned} c_4 =& a_4 + b_4 + \frac{1}{12}([a_1b_3] + [a_2b_2] + [a_3b_1]) \\ &+ \frac{1}{12}([a_1b_1b_2] + [a_1b_2b_1] + [a_2b_1b_1] + [b_1a_1a_2] \\ &+ [b_1a_2a_1] + [b_2a_1a_1]) - \frac{1}{24}[a_1b_1b_1a_1], \\ &\text{etc.} \end{aligned}$$

Note that the classical CH formula is restricted to the composition of two one-parameter groups, where $\log f^s = sa_1$, $\log g^s = sb_1$.

The log transform reduces matters at the Lie group level to those at the easier level of Lie algebra. All properties of near-1 formal maps have their logarithmic interpretations. We list some of them, let $\log f^s = a^s = \sum_1^\infty s^k a_k$:

1. f^s is a phase flow, i.e., $f^{s+t} = f^s \circ f^t \Leftrightarrow \log f^s = sa_1$.
2. f^s is revertible, i.e., $f^s \circ f^{-s} = 1_m \Leftrightarrow \log f^s$ is odd in s.
3. f^s raised to real μth power $(f^s)^\mu \Leftrightarrow \log(f^s)^\mu = \mu \log f^s$. In particular $\log(f^s)^{-1} = -\log f^s$, $\log\sqrt{f^s} = \frac{1}{2}\log f^s$.
4. f^s scaled to $f^{\alpha s} \Leftrightarrow \log(f^{\alpha s}) = (\log f)^{\alpha s}$. In particular $\log(f^{-s}) = (\log f)^{-s}$.
5. $f^s - g^s = O(s^{p+1}) \Leftrightarrow \log f^s - \log g^s = O(s^{p+1})$.
6. $f^s \circ g^s = g^s \circ f^s \Leftrightarrow [\log f^s, \log g^s] = 0 \Leftrightarrow \log(f^s \circ g^s) = \log f^s + \log g^s$.
7. $(f^s \circ g^s) = h^s \Leftrightarrow \log h^s = \log(f^s \circ g^s) = \log f^s + \log g^s + \sum_1^\infty d_k(\log f^s, \log g^s)$.
8. f^s symplectic $\Leftrightarrow$ all a_k are Hamiltonian fields.
9. f^s contact $\Leftrightarrow$ all a_k are contact fields.
10. f^s volume-preserving $\Leftrightarrow$ all a_k are source-free fields.

The log transform has important bearing on dynamical systems with Lie algebra structure, the structure-preserving property of maps f^s at the Lie group ($\mathbf{G} \subset \mathbf{D}_m$) level can be characterized through their logarithms at the associated Lie algebra ($\mathbf{L} \subset \mathbf{V}_m$) level.

We return to the main problem of approximation to the phase flow for dynamical system $dx/dt = a(x)$.

$$f_a^s = f^s = 1_m + \sum s^k f_k \approx e_a^s = 1_m + \sum s^k e_k, \quad e_k = a^k/k!. \tag{10.8}$$

If $f_k = e_k$, $1 \leq k \leq p$, we say f_a^s is accurate to order $\geq p$, if moreover, $f_{p+1} \neq e_{p+1}$, we say it is accurate to order p.

Let $\log f^s = a^s = \sum s^k a_k$. Note that the first $p+1$ equations in (10.2) completely determine $a_1, a_2, \cdots, a_{p+1}$ and $f_1, f_2, \cdots, f_{p+1}$ each other. It is then easy to establish

$$\begin{aligned} &f_k = e_k, \quad 1 \leq k \leq p; \quad f_{p+1} \neq e_{p+1} \Longleftrightarrow \\ &a = a_1 = e_1; \quad a_k = 0, \quad 1 < k \leq p; \quad a_{p+1} = f_{p+1} - e_{p+1} \neq 0. \end{aligned} \tag{10.9}$$

So the orders of approximation for $f_a^s \approx e_a^s$ and for $\log f_a^s - sa$ are the same.

Moreover, note that we have a *formal field*

$$s^{-1}\log f^s = s^{-1}a^s = a + \sum_1^\infty s^k a_{k+1} = a + O(s^p) \tag{10.10}$$

which is equal to the original field a up to a near-0 perturbation and defines a *formal dynamical system*

$$\frac{dx}{dt} = (s^{-1} \log f^s)(x) = a(x) + \sum_1^\infty s^k a_{k+1}(x) \tag{10.11}$$

having a *formal phase flow* (in two parameters t and s with group property in t) $e^t_{s^{-1}a^s} = \exp ts^{-1}a^s$ whose *diagonal formal flow* $e^t_{s^{-1}a^s}|_{t=s}$ is exactly f^s. This means that any compatible algorithm f^s_a of order p gives perturbed solution of a right equation with field a; however, it gives the right solution of a perturbed equation with field $s^{-1}\log f^s_a = a + O(s^p)$. There could be many methods with the same formal order of accuracy but with quite different qualitative behavior. The problem is to choose among them those leading to allowable perturbations in the equation. For systems with geometric structure, the propositions 8, 9 and 10 provide guidelines for a proper choice. The structure-preservation requirement for the algorithms precludes all unallowable perturbations alien to the pertinent type of dynamics. Take, for example, Hamiltonian systems. A transition map f^s_a for Hamiltonian field a is symplectic if and only if all fields a_k are Hamiltonian, i.e., the induced perturbations in the equation are Hamiltonian [4, 15]. So symplectic algorithms are clean, inherently free from all kinds of perturbations alien to Hamiltonian dynamics (such as artificial dissipation inherent in the vast majority of conventional methods), this accounts for their superior performance. The situations are analogous for contact and volume-preserving algorithms. Propositions 8, 9 lead to the existence of formal energy for symplectic algorithms and that of formal contact energy for contact algorithms, see [8–9, 12–13, 37, and also 45]. For the incompatibility of symplectic structure preservation with energy conservation for algorithms, see [24, 26].

Appended Remarks. Our results on structure-preserving algorithms for dynamical systems and the constructive theory of generating functions for symplectic maps as surveyed above has been extended and further developed in various aspects as follows: For Poisson manifolds and Poisson maps, see [21–23, 25, 36]. For infinite-dimensional Hamiltonian systems, see [24, 27, 29, 35]. For quantum mechanical systems and unitary algorithms, see [30, 46]. For Hamiltonization of ordinary differential equations, see [39, 41, 42]. For comparative numerical studies for Hamiltonian vs non-Hamiltonian algorithms, see [4, 5, 14, 15, 34]. For the problem of symplecticity of multi-step methods, see [7, 17]. For KAM theorem for symplectic algorithms for Hamiltonian systems, see [32].

Acknowledgements The authors would like to thank their collaborators Qin Meng-zhao, Li Wang-yao, Ge Zhong, Shang Zai-jiu, Wu Yu-hua, Li Chun-wang, Zhang Mei-qing, Tang Yi-fa, Zhu Wen-jie, Jiang Li-xin and Shu Hai-bing for fruitful cooperation.

References

1 Arnold V I. Mathematical Methods of Classical Mechanics. New York: Springer. 1978

2 Feng K. On difference schemes and symplectic geometry. In Feng K, ed. Proc 1984 Beijing Symp Diff Geometry and Diff Equations. Beijing: Science Press, 1985. 42–58

3 Feng K. Difference schemes for Hamiltonian formalism and symplectic geometry. J Comp Math. 1986 4(3): 279–289

4 Feng K. The Hamiltonian way for computing Hamiltonian dynamics. In Spigler R, ed. Applied and Industrial Mathematics. Netherlands: Kluwer, 1991. 17–35

5 Feng K. How to compute properly Newton's equation of motion. In Ying L A, Guo B Y, ed. Proc of 2nd Conf on Numerical Methods for Partial Differential Equations. Singapore: World Scientific, 1992. 15–22
6 Feng K. Formal power series and numerical methods for differential equations. In Chan T, Shi Z C, ed. Proc of Inter Conf on Scientific Computation. Singapore: World Scientific, 1992. 28–35
7 Feng K. The step transition operators for multi-step methods of ODE's: [Preprint]. Beijing: CAS. Computing Center, 1991
8 Feng K. The calculus of generating functions and the formal energy for Hamiltonian algorithms: [Preprint]. Beijing: CAS. Computing Center, 1991
9 Feng K. The calculus of formal power series for diffeomorphisms and vector fields: [Preprint]. Beijing: CAS. Computing Center, 1991
10 Feng K. Symplectic, contact and volume-preserving algorithms. In Shi Z C, Ushijima T, ed. Proc 1st China-Japan Conf on Numer Math. Singapore: World Scientific, 1993. 1–28
11 Feng K. Formal dynamical systems and numerical algorithms. In Feng K, Shi Z C, ed. Proc Inter Conf on Computation of Differential Equations and Dynamical Systems. Singapore: World Scientific, 1993. 1–10
12 Feng K. Contact algorithms for contact dynamical systems: [Preprint]. Beijing: CAS. Computing Center, 1992
13 Feng K. Theory of contact generating functions: [Preprint]. Beijing: CAS. Computing Center, 1992
14 Feng K, Qin M Z. The symplectic methods for computation of Hamiltonian equations. In Zhu Y L, Guo Ben-yu, ed. Proc Conf on Numerical Methods for PDE's. Berlin: Springer, 1987. 1–37. Lect Notes in Math 1297
15 Feng K, Qin M Z. Hamiltonian algorithms and a comparative numerical study. Comput Phys Comm. 1991 65: 173–187
16 Feng K, Shang Z J. Volume preserving algorithms for source-free dynamical systems. Numerische Mathematik. 1995
17 Feng K, Tang Y F. Non-symplecticity of linear multi-step methods: [Preprint]. Beijing: CAS. Computing Center, 1991
18 Feng K, Wang D L. A note on conservation laws of symplectic difference schemes for Hamiltonian systems. J Comp Math. 1991 9(3): 229–237
19 Feng K, Wang D L. Variations on a theme by Euler. Preprint, Acad. Sin. Comp. Ctr., 1991
20 Feng K, Wu H M, Qin M Z, Wang D L. Construction of canonical difference schemes for Hamiltonian formalism via generating functions. J Comp Math. 1989 7(1): 71–96
21 Ge Z. Generating functions for the Poisson map: [Preprint]. Beijing: CAS. Computing Center, 1987
22 Ge Z. Geometry of symplectic difference schemes and generating functions: [Preprint]. Beijing: CAS. Computing Center, 1988
23 Ge Z. Generating functions, Hamilton-Jacobi equations and symplectic groupsoids on Poisson manifolds: [Preprint]. Berkeley: MSRI, 1988
24 Ge Z, Feng K. On the approximation of Hamiltonian systems. J Comp Math. 1988 6(1): 88–97
25 Ge Z, Marsden J. Lie-Poisson Hamilton-Jacobi theory and Lie-Poisson integrators. Phys Lett A. 1988 133(3): 137–139
26 Li C W. The incompatibility of symplectic structure preservation with energy conservation for numerical computation of Hamiltonian systems: [Preprint]. Beijing: CAS. Computing Center, 1987
27 Li C W, Qin M Z. Symplectic difference schemes for infinite dimensional Hamiltonian systems. J Comp Math. 1988 6(2): 164–174
28 Qin M Z, Wang D L, Zhang M Q. Explicit symplectic difference schemes for separable Hamiltonian systems. J Comp Math. 1991 9(3): 211–221

29 Qin M Z, Zhang M Q. Multi-stage symplectic schemes of two kind of Hamiltonian systems of wave equations. Computer Math Applic. 1990 19(10): 51–62

30 Qin M Z, Zhang M Q. Explicit Runge-Kutta like unitary schemes to solve certain quantum operator equations of motion. J Stat Phys. 1990 60: 837–843

31 Qin M Z, Zhu W J. Construction of higher order symplectic schemes by composition. Computing. 1992 47: 309–321

32 Shang Z J. On KAM theorem for symplectic algorithms for Hamiltonian systems: [dissertation]. Beijing: CAS. Computing Center, 1991

33 Shu H B. A new approach to generating functions for contact systems: [Preprint]. Beijing: CAS. Computing Center, 1992

34 Tang Y F. Hamiltonian systems and algorithms for geodesic flows on compact Riemannian manifolds: [Master Thesis]. Beijing: CAS. Computing Center, 1990

35 Wang D L. Semi-discrete Fourier spectral approximations of infinite dimensional Hamiltonian systems and conservation laws. Computer Math Applic. 1991 21(4): 63–75

36 Wang D L. Poisson difference schemes for Hamiltonian systems on Poisson manifolds. J Comp Math. 1991 9(2): 115–124

37 Wang D L. Some aspects of Hamiltonian systems and symplectic algorithms. Physica D. 1994 73: 1–16

38 Wang D L. Symplectic difference schemes for perturbed Hamiltonian systems: [Preprint]. Berlin: ZIB, 1990

39 Wang D L. Decomposition of vector fields and composition of algorithms. In Feng K, Shi Z C, ed. Proc Inter Conf on Computation of Differential Equations and Dynamical Systems. Singapore: World Scientific, 1993, 179–184

40 Wang D L, Wu Y H. Generating function methods for the construction of one-step schemes for ODE's: [Preprint]. Beijing: CAS. Computing Center, 1988

41 Wu Y H. The generating function of the solution of ODE and its discrete methods. Computer Math Applic. 1988 15(12): 1041–1050

42 Wu Y H. Symplectic transformations and symplectic difference schemes. Chinese J Numer Math Applic. 1990 12(1): 23–31

43 Wu Y H. Discrete variational principle to the Euler-Lagrange equation. Computer Math Applic 1990 20(8): 61–75

44 Yoshida H. Construction of higher order symplectic integrators. Phys Letters A. 1990 150: 262–268

45 Yoshida H. Conserved quantities of symplectic integrators for Hamiltonian systems: [Preprint]. Japan, 1990

46 Zhang M Q. Explicit unitary schemes to solve quantum operator equations of motion. J Stat Phys. 1991 65(3/4): 793–799

47 Zhang M Q, Qin M Z. Explicit symplectic schemes to solve vortex systems: [Preprint]. Beijing: CAS. Computing Center, 1992