Chapter 2
Numerical Integrators

At its most basic level, molecular dynamics is about mapping out complicated point sets using trajectories of a system of ordinary differential equations (or, in Chaps. 6–8, a stochastic-differential equation system). The sets are typically defined as the collection of probable states for a certain system. In the case of Hamiltonian dynamics, they are directly associated to a region of the energy landscape. The trajectories are the means by which we efficiently explore the energy surface. In this chapter we address the design of numerical methods to calculate trajectories.

When we use the equations of motion for an atomic system and compute a trajectory, we are producing what we hope to be a representative path of the system. For the moment, we discount any external interactions (such as due to heating at a boundary or other driving forces), so we can think of the system as a closed, Hamiltonian dynamics model. This introduces the *microscopic* perspective which concerns the detailed description of the instantaneous atomic positions and velocities as time is varied. These microscopic paths are the cornerstone of statistical mechanical theory, which is the tool that we will eventually develop for understanding molecular systems at a more abstract level, thus it is essential in developing a computational methodology that we have an understanding of how to generate trajectories reliably and efficiently.

The challenge before us is to compute solutions of

$$\dot{\boldsymbol{q}} = \boldsymbol{M}^{-1}\boldsymbol{p}, \qquad \dot{\boldsymbol{p}} = \boldsymbol{F}(\boldsymbol{q}) = -\nabla U(\boldsymbol{q}),$$

or, more compactly, with $\boldsymbol{z}$ representing the collection of all positions and momenta,

$$\dot{\boldsymbol{z}} = \boldsymbol{f}(\boldsymbol{z}), \qquad \boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{J}\nabla H, \tag{2.1}$$

where $\boldsymbol{J} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix}$, and $H = \boldsymbol{p}^T\boldsymbol{M}^{-1}\boldsymbol{p}/2 + U(\boldsymbol{q})$ is the Hamiltonian.

© Springer International Publishing Switzerland 2015

B. Leimkuhler, C. Matthews, *Molecular Dynamics*, Interdisciplinary Applied Mathematics 39, DOI 10.1007/978-3-319-16375-8_2

In order to correctly model the different possible states of the system, it will be necessary to cover a large part of the accessible phase space, so either trajectories must be very long or we must use many initial conditions. There are many ways to solve initial value problems such as (2.1) combined with an initial condition $z(0) = \boldsymbol{\zeta}$. The methods introduced here all rely on the idea of a discretization with a finite stepsize h, and an iterative procedure that computes, starting from $z_0 = \boldsymbol{\zeta}$, a sequence $z_1, z_2, \ldots$, where $z_n \approx z(nh)$. The simplest scheme is certainly Euler's method which advances the solution from timestep to timestep by the formula[1]

$$z_{n+1} = z_n + h\boldsymbol{f}(z_n).$$

The method is based on the observation that $z(t+h) \approx z(t)+h\dot{z}(t)$, i.e. the beginning of a Taylor series expansion in powers of h, and the further observation that the solution satisfies the differential equation, hence $\dot{z}(t)$ may be replaced by $\boldsymbol{f}(z(t))$.

In order to be able to easily compare the properties of different methods in a unified way, we focus in this chapter primarily on a particular class of schemes, generalized *one-step methods*. Suppose that the system under study has a well defined flow map $\mathscr{F}_t$ defined on the phase space (which is assumed to exclude any singular points of the potential energy function). The solution of the initial value problem, $\dot{z} = \boldsymbol{f}(z)$, $z(0) = \boldsymbol{\zeta}$, may be written $z(t, \boldsymbol{\zeta})$ (with $z(0, \boldsymbol{\zeta}) = \boldsymbol{\zeta}$), and the flow-map $\mathscr{F}_t$ satisfies $\mathscr{F}_t(\boldsymbol{\zeta}) = z(t, \boldsymbol{\zeta})$. A *one-step method*, starting from a given point, approximates a point on the solution trajectory at a given time h units later. Such a method defines a map $\mathscr{G}_h$ of the phase space as illustrated in Fig. 2.1.

We assume here a basic understanding of ordinary differential equations; some good references for review of this topic are [16, 51, 177, 362]. For basic concepts in the numerical analysis of ordinary differential equations the definitive reference is [167].

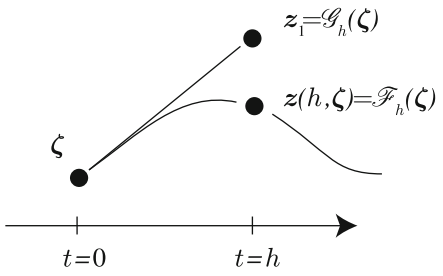

Fig. 2.1 A step with the flow map approximation $\mathscr{G}_h$ is illustrated in comparison to the corresponding step along the solution curve defined by the flow map $\mathscr{F}_h$

[1] Subscripts were used previously to indicate the components of vectors and here they are used to indicate the indexing of timesteps. Although in theory this could lead to some confusion, it normally does not in practice, since we index components in descriptions of details of models and we discuss timesteps in context of defining numerical methods for general classes of systems. Moreover, we use boldface for vectors, so a subscript on a boldface vector indicates a timestep index. When we wish to refer to both the timestep and the component index, we may write $z_{n,i}$ to denote the ith component at timestep n.

Let us emphasize that the issues arising in the design and analysis of numerical methods for molecular dynamics are slightly different than those confronted in other application areas. For one thing the systems involved are highly structured having conservation properties such as first integrals and Hamiltonian structure. We address the issues related to the inherent structure of the molecular N-body problem in both this and the next chapter; wherein we shall learn that *symplectic* discretizations are typically the most appropriate methods.

Another special aspect of the molecular system is that normally it is sensible to use a fixed stepsize, that is each timestep corresponds to a fixed interval of real time. This is in contrast to other applications where it is found to be important to vary the step during simulation. The reason that a uniform stepsize is used is because, in a simulation of many particles, the complexity of the system ensures that if a strong local force is encountered in some corner of the system, a force of similar magnitude will be found somewhere else at the next instant. Even if, occasionally, an instantaneous event is observed that could be controlled by reducing the stepsize, the necessary adaptive machinery can impair the geometric properties and reduce the efficiency of the numerical procedure.[2] There is no trivial way of selecting the stepsize a priori. In typical practice, one performs several trial runs, examining the fluctuations in energy or other easily computable quantities and makes the choice of stepsize in order to keep these within a tolerable range. The molecular dynamics timesteps are typically quite small, measured in femtoseconds, in order to capture the rapid fluctuations of the atoms (in Chap. 4, we discuss ways of increasing the timestep).

2.1 Error Growth in Numerical Integration of Differential Equations

In this section we discuss the issues of convergence and accuracy in numerical integration methods for solving ordinary differential equations.

Let us begin by considering Euler's method in a bit more detail to understand its convergence order. The *convergence* of a numerical method refers to the ability of the method to provide an arbitrary level of accuracy by using small enough timesteps. The *order of accuracy* is the exponent in the power law by which the error in the method is related to the stepsize. For example, when we say that a method is third order accurate, we mean that the global error (on a fixed finite time interval) can be bounded by Kh^3, where h is a sufficiently small timestep and K is a number which depends on the length of the time interval and the features of the problem, but which is independent of h.

The flow map approximation in the case of Euler's method is

[2] A variable stepsize is only used where one expects extreme changes in particle velocities over the course of a simulation (see e.g. [75, 390] for examples arising in radiation damage cascades).

$$\mathscr{G}_h(z) = z + hf(z),$$

and we have

$$z_{n+1} = \mathscr{G}_h(z_n), \qquad n = 0, 1, 2, \ldots \tag{2.2}$$

with $z_0 = \zeta$. The points $\{z_n\}$ are intended to be approximations to the values of the solution. The obvious question is this: how good an approximation is z_n of $z(nh)$?

Let the approximate solution vectors at successive timesteps be $z_0, z_1, \ldots, z_\nu$ where $\nu h = \tau$. We assume that τ, the length of the time interval, is fixed, and ν is an integer parameter representing the total number of timesteps. In order to improve the quality of the approximation, the parameter ν may be increased, as the stepsize is proportionately decreased. The error at step n is defined by $e_n = \|z_n - z(t_n)\|$, where $t_n = nh$; it clearly depends on h. With these definitions one can prove the following result, which is typical of the sorts of results that are available for numerical methods for initial value problems:

Theorem 2.1 *Let $\mathcal{D}$ be a bounded, open region in $\mathbb{R}^m$ such that $f : \mathcal{D} \to \mathbb{R}^m$ is continuously differentiable. Let ζ be an interior point of $\mathcal{D}$ and suppose the initial value problem (2.1) has a unique solution that remains in $\mathcal{D}$ for $t \in [0, \tau]$. Then there exists a constant $C(\tau) > 0$ such that for sufficiently large $\nu \in \mathbb{N}$ the numerical solution z_n remains in $\mathcal{D}$ for $n = 0, 1, \ldots, \nu$, where $h\nu = \tau$, and, moreover, the maximum global error in Euler's method satisfies*

$$\bar{e} := \max_{0 \le n \le \nu} e_n \le C(\tau)h.$$

In short, the error in the approximation obtained on $[0, \tau]$ is reduced in direct proportion to the number of steps taken to cover this interval. Another way to say this is that $\bar{e} \propto h$, or, using the order notation, $\bar{e} = \mathcal{O}(h)$. Because the global error is of order h^r, where $r = 1$, we say that Euler's method is a *first order* method, or that it *converges with order* $r = 1$.

2.1.1 Example: The Trimer

Let us test Euler's method on the trimer model (formulated as a system in $\mathbb{R}^4$). Fixing initial values ($q_0 = (0.5, 1.0); p_0 = (0.1, 0.1)$) we solve the trimer on $[0, 4]$ using the Euler method first with 40 time-steps of length 0.1, next using 80 time-steps of length 0.05 and then with 160 steps of size 0.025, etc. Each calculation results in a different "discrete trajectory." The solutions are graphed in the xy projection in the left panel of Fig. 2.2 with line segments connecting the points; these piecewise linear paths take on the appearance of smooth curves as the stepsize is reduced.

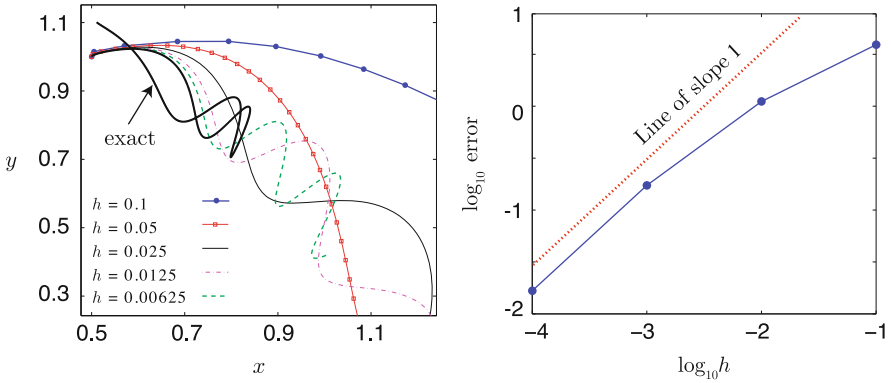

Fig. 2.2 Euler's method applied to the trimer. *Left*, the solution trajectories for different step-sizes; *right*: the maximum global errors in position projections of solutions computed using four different time-steps appear to show an asymptotic linear relationship to step-size, when plotted in log-log scale

A natural question is how can one measure the error in the discrete approximations, since for the trimer, no exact analytical solution is available? Although it is not possible to compute this error exactly, if it is assumed that the process is convergent (as certainly appears to be the case from our experiment) we may use the accurate solution computed with very small steps as a reference and compute the differences between iterates along the trajectories and corresponding points on the reference trajectory. The maximum of the differences between positions of corresponding points is then used as a measure of the global error in the solution for a given step-size h. For the reference solution in the case of the trimer, we have used a simulation with a superior numerical method (the Verlet method) and very small steps of size 10^{-7}. This is the "exact" solution that has been plotted in the left panel of Fig. 2.2. In the right hand panel of Fig. 2.2, the global error $\bar{e}(h)$ is calculated in this way and plotted against the stepsize, using a logarithmic scale. Studying this figure, we see that in log-log scale, the observed relationship between $\bar{e}$ and h is linear, with slope 1 (for h sufficiently small):

$$\ln \bar{e} \approx \ln h + \alpha,$$

where α is a constant. Hence, by exponentiating both sides we obtain

$$\bar{e} \approx Ch,$$

where $C = e^{\alpha}$, confirming the first order relationship between the global error and the step-size.

In the simulations of the example above, it is apparent that the errors are larger at the end of the interval than at earlier times. We know that molecular dynamics trajectories need to be very long compared to the time-step used in

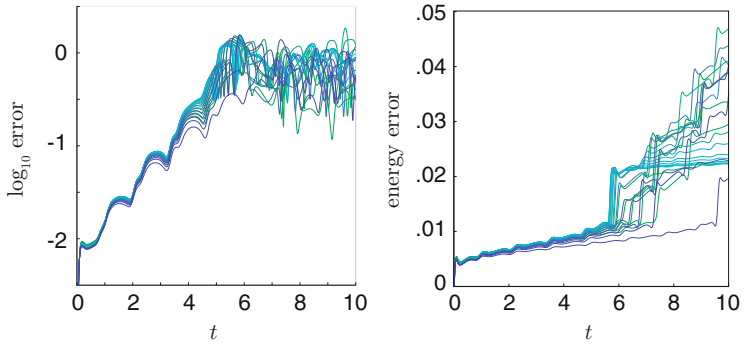

Fig. 2.3 These graphs show the way in which the error grows in relation to time in simulations of the trimer using Euler's method. *Left*: the growth of the logarithm of the trajectory error; *right*: the energy error growth as a function of time

order for them to be useful, so how the error grows in long simulations is quite important. Convergence theorems for numerical methods like that mentioned above are normally formulated for computations on a finite interval, and often do not tell us much about how the error depends on time (i.e. on the length of the interval, τ).

To examine the issue of error growth we perform a simulation using a particular timestep ($h = 0.001$) on the time interval $[0, 10]$ and compare against the reference solution on this interval. The logarithm of the error at each time-step, e_n, is then computed and the result plotted against n. We repeated this for 20 separate initial conditions having the same initial energy. In the left panel of Fig. 2.3, the error graphs are shown, for each of these numerical trajectories, demonstrating that the error in each case grows very rapidly in the early going. Eventually the error growth tapers off, but only when the error is around the same size as the diameter of the system, thus there is little meaningful information remaining regarding a particular trajectory after $t \approx 5$. The right panel shows the errors in energies seen in the same set of simulations, showing that the energy errors initially grow only linearly while the global error is rising exponentially.

To summarize, in this example, with the time interval fixed, the error in Euler's method increases in proportion to the stepsize. On the other hand, the global error grows with the length of the time interval, and at a rapid rate, until it is clear that the numerical trajectory is entirely unrelated to the exact trajectory. Moreover, we observe that the energy errors grow much more slowly than the trajectory errors.

2.1.2 Higher Order Methods

One approach to higher accuracy is to decrease the step-size while continuing to use Euler's method. Since we know that the error on a given fixed interval is proportional to h, using a smaller stepsize should decrease the error in proportion, albeit at the

cost of requiring more timesteps to cover the given time interval. A more efficient means to get better accuracy is to use a higher order method. Higher order methods are ones that satisfy a global error estimate (for finite time intervals) of the form

$$\bar{e} \approx C(\tau)h^r,$$

where $r > 1$.

Example 2.1 The Taylor series expansion of the solution may be written $z(t+h) = z(t) + h\dot{z}(t) + (h^2/2)\ddot{z}(t) + \ldots$. Whereas a first order truncation of this series leads to Euler's method, retaining terms through second order leads to

$$z_{n+1} = z_n + h\dot{z}_n + \frac{h^2}{2}\ddot{z}_n,$$

which is referred to as the 2nd order Taylor series method. In this formula $\dot{z}_n = \boldsymbol{f}(z_n)$, and the second derivative is obtained by differentiating the differential equation itself:

$$\ddot{z}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\dot{z}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f}(z(t)) = \boldsymbol{f}'(z(t))\dot{z}(t) = \boldsymbol{f}'(z(t))\boldsymbol{f}(z(t)),$$

so one may write the 2nd order Taylor series method as

$$z_{n+1} = z_n + h\boldsymbol{f}(z_n) + \frac{h^2}{2}\boldsymbol{f}'(z_n)\boldsymbol{f}(z_n).$$

This method generates the flow map approximation

$$\mathscr{G}_h(z) = z + h\boldsymbol{f}(z) + \frac{h^2}{2}\boldsymbol{f}'(z)\boldsymbol{f}(z).$$

Note that by the notation $\boldsymbol{f}'(z)$ where $z \in \mathbb{R}^m$ and $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^m$, is meant the $m \times m$ Jacobian matrix whose ij-component is $(\boldsymbol{f}'(z))_{ij} = \partial f_i/\partial z_j$. An alternative notation for $\boldsymbol{f}'$ is $\partial \boldsymbol{f}/\partial z$.

Methods like the Taylor series method offer the prospect of better accuracy in the local approximation, and smaller global error in a given simulation, but they do not necessarily resolve the more important issue relevant for very long time integrations (which we will need in molecular dynamics): the unlimited growth of perturbations from the energy surface. In molecular dynamics, we have already seen that the Euler method has growing energy error which suggests that it will be a poor scheme where the goal is to approximate the behavior of a constant energy trajectory. This same qualitative behavior is seen in other numerical methods, such as the Taylor series method. However, there are alternatives that give both higher order of accuracy and, typically, improved energy accuracy. We discuss one of the most popular schemes of this type in the next section.

2.2 The Verlet Method

The Verlet method (also known as leapfrog or Störmer-Verlet) is a second order method that is popular for molecular simulation. It is specialized to problems that can be expressed in the form $\dot{\boldsymbol{q}} = \boldsymbol{v}, \boldsymbol{M}\dot{\boldsymbol{v}} = \boldsymbol{F}(\boldsymbol{q})$, with even dimensional phase space $\mathbb{R}^{2N_c}$, which includes constant energy molecular dynamics. Some generalizations exist for other classes of Hamiltonian systems.

The Verlet method is a numerical method that respects certain conservation principles associated to the continuous time ordinary differential equations, i.e. it is a geometric integrator. Maintaining these conservation properties is essential in molecular simulation as they play a key role in maintaining the physical environment. As a prelude to a more general discussion of this topic, we demonstrate here that it is possible to derive the Verlet method from the variational principle. This is not the case for every convergent numerical method. The Verlet method is thus a special type of numerical method that provides a discrete model for classical mechanics.

2.2.1 Hamilton's Principle and Equations of Motion

Hamilton's principle of least action provides a mechanism for deriving equations of motion from a Lagrangian. Recall from Chap. 1 that the Lagrangian for the N-body system is defined by

$$L(\boldsymbol{q}, \boldsymbol{v}) \stackrel{\text{def}}{=} \frac{\boldsymbol{v}^T \boldsymbol{M} \boldsymbol{v}}{2} - U(\boldsymbol{q}),$$

where $\boldsymbol{M}$ is the mass matrix and the potential energy function U is, for simplicity, taken to be smooth (C^2). We consider the collection of all twice continuously differentiable curves in the configuration space which start from a certain point $\boldsymbol{Q}$ and end at another given point $\boldsymbol{Q}'$. We may think of any such curve as being represented by a parameterization $\boldsymbol{q}(t)$, $t \in [\alpha, \beta]$ with $\boldsymbol{q}(\alpha) = \boldsymbol{Q}$ and $\boldsymbol{q}(\beta) = \boldsymbol{Q}'$, where the components of $\boldsymbol{q}(t)$ are C^∞ functions. Denote by $G = G(\boldsymbol{Q}, \boldsymbol{Q}', \alpha, \beta)$ the class of smooth parameterized curves such that $\boldsymbol{q}(\alpha) = \boldsymbol{Q}, \boldsymbol{q}(\beta) = \boldsymbol{Q}'$. Then the *classical action* (or, simply, *action*) of the Lagrangian L is defined for any $\Gamma \in G$ by

$$\mathcal{A}_L(\Gamma) \stackrel{\text{def}}{=} \int_\alpha^\beta L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) \mathrm{d}t.$$

The variational calculus approach to classical mechanics is based on minimizing the action $\mathcal{A}_L$ over the class G of parameterized curves. This is normally referred to as the "principle of least action". It is difficult to provide a physical motivation for this concept, but it is normally taken as a foundation stone for classical mechanics.

Given Γ in G with parameterization $\boldsymbol{q}(t)$, $t \in [\alpha, \beta]$, we consider the curve Γ^ε with parameterization $\boldsymbol{q}^\varepsilon$ defined by

$$\boldsymbol{q}^\varepsilon(t) = \boldsymbol{q}(t) + \varepsilon\boldsymbol{\eta}(t), \qquad t \in [\alpha, \beta], \tag{2.3}$$

where $\boldsymbol{\eta}(t)$ satisfies $\boldsymbol{\eta}(\alpha) = \boldsymbol{\eta}(\beta) = 0$. Thus $\boldsymbol{\eta}$ is a C^∞ parameterized curve linking $\mathbf{0}$ to $\mathbf{0}$. In defining variations in this way, we are using, implicitly, the fact that we can add together functions in C^∞ and multiply them by scalars (e.g. ε) and remain in C^∞. We also make use of the intuitive concept that (2.3) defines $\boldsymbol{q}^\varepsilon$ in such a way that it is "close" to $\boldsymbol{q}$ when ε is small. This can be made precise by a little more elaboration, but we forego this here. Effectively, we are using our understanding of C^∞ as a *normed function space* to restrict attention to variations of the base curve Γ in a particular "direction."

Using a Taylor series expansion of the Lagrangian,[3] we have

$$\begin{aligned}\mathcal{A}_L(\Gamma^\varepsilon) - \mathcal{A}_L(\Gamma) &= \int_\alpha^\beta [L(\boldsymbol{q}(t) + \varepsilon\boldsymbol{\eta}(t), \dot{\boldsymbol{q}}(t) + \varepsilon\dot{\boldsymbol{\eta}}(t)) - L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))]\,\mathrm{d}t,\\ &= \int_\alpha^\beta \left[\varepsilon\left(\frac{\partial L}{\partial \boldsymbol{q}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\boldsymbol{\eta}(t) + \frac{\partial L}{\partial \dot{\boldsymbol{q}}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\dot{\boldsymbol{\eta}}(t)\right) + \mathcal{O}(\varepsilon^2)\right]\mathrm{d}t.\end{aligned}$$

Hamilton's principle states that the natural motion of the system described by the Lagrangian L is a stationary point of the classical action which implies that the $\mathcal{O}(\varepsilon)$ term above should vanish for any smooth variation $\boldsymbol{\eta}(t)$ with $\boldsymbol{\eta}(\alpha) = \boldsymbol{\eta}(\beta) = 0$, i.e.,

[3] In the multidimensional setting, Taylor's theorem states that given a C^{k+1} function $f : \mathbb{R}^m \to \mathbb{R}$ and a point z_0, we have

$$\begin{aligned}f(z) - f(z_0) = \nabla f(z_0)\cdot(z - z_0) + f^{(2)}\langle z - z_0, z - z_0\rangle + f^{(3)}\langle z - z_0, z - z_0, z - z_0\rangle\\ + \ldots f^{(k)}\langle z - z_0, z - z_0, \ldots, z - z_0\rangle + \mathcal{O}(\|z - z_0\|^{k+1})\end{aligned}$$

where $\nabla f = \partial f/\partial z$ is the gradient, i.e. a vector with m components, $f^{(2)}$ is the $m \times m$ Hessian matrix of f (the matrix whose ij component is $\partial^2 f/\partial z_i \partial z_j$), and $f^{(2)}\langle \boldsymbol{u}, \boldsymbol{v}\rangle$, $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m$, represents the quadratic form $\boldsymbol{u}^T f^{(2)}\boldsymbol{v}$. In a similar way we interpret $f^{(3)}$ as a tensor which we can think of as a $m \times m \times m$ triply-indexed array, the ijk element being $\partial^3 f/\partial z_i \partial z_j \partial z_k$ and

$$f^{(3)}\langle \boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}\rangle = \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m (\partial^3 f/\partial z_i \partial z_j \partial z_k) u_i v_j w_k.$$

$$I = \int_{\alpha}^{\beta} \left[\frac{\partial L}{\partial \boldsymbol{q}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\boldsymbol{\eta}(t) + \frac{\partial L}{\partial \dot{\boldsymbol{q}}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))\dot{\boldsymbol{\eta}}(t) \right] \mathrm{d}t = 0.$$

We use integration by parts to remove the differentiation of $\boldsymbol{\eta}$, thus (in light of the boundary conditions on $\boldsymbol{\eta}(t)$),

$$I = \int_{\alpha}^{\beta} \left[\frac{\partial L}{\partial \boldsymbol{q}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) - \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{\boldsymbol{q}}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) \right] \boldsymbol{\eta}(t) \mathrm{d}t = 0.$$

Since the variation $\boldsymbol{\eta}$ is meant to be arbitrary, it requires

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial L}{\partial \dot{\boldsymbol{q}}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)) = \frac{\partial L}{\partial \boldsymbol{q}}(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t)),$$

which is precisely the Lagrangian formulation of the equations of motion.

The method of derivation described here may be formulated in direct analogy to the traditional method of minimizing a function of several variables based on finding critical points. Define the *variational derivative* of a functional $\mathcal{F}$, $\delta\mathcal{F}/\delta\boldsymbol{q}$ so that

$$\mathcal{F}(\boldsymbol{q} + \varepsilon\boldsymbol{\eta}) = \varepsilon \frac{\delta \mathcal{F}}{\delta \boldsymbol{q}} \boldsymbol{\eta} + O(\varepsilon^2),$$

for all suitable (say, C^∞) functions $\boldsymbol{\eta}$. Viewing the action $\mathcal{A}_L(\Gamma)$ as the functional $\mathcal{F}$, the stationarity condition (the Euler-Lagrange equations) may be written

$$\frac{\delta \mathcal{F}}{\delta \boldsymbol{q}} = 0.$$

Thus the calculus of variations becomes a generalization of the traditional method of minimizing smooth functions by finding critical points.[4]

[4]This discussion is a great simplification. Any curve which satisfies this equation will represent a "stationary point" (actually, "stationary curve" would be more accurate) of the classical action. Such curves could include smooth local action minimizers, local action maximizers, or "saddle points" of the actional functional in a generalized sense. Deciding whether a given stationary curve is an actual minimizer of the action would require analysis of the second variation (the coefficient of ε^2 in the expansion above), which introduces additional complexity. For a more comprehensive treatment, see e.g. [210].

2.2.2 Derivation of the Verlet Method

Now let us consider the discrete version of "minimizing the action". We work on the time interval $[0, \tau]$. Consider the $\nu + 1$ points $\boldsymbol{q}_0$ to $\boldsymbol{q}_\nu$ in configuration space

$$\hat{\Gamma} = (\boldsymbol{q}_0, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_\nu).$$

We must first define the action for such a discrete path. One perspective is that we first formulate a discrete version of the Lagrangian in terms of only the point set $\boldsymbol{q}_0, \boldsymbol{q}_1, \ldots \boldsymbol{q}_\nu$, but this somehow requires that we define velocities. Noting that

$$\dot{\boldsymbol{q}}(t) \approx \frac{\boldsymbol{q}(t+h) - \boldsymbol{q}(t)}{h},$$

we are led to consider the approximation at time level n

$$\boldsymbol{v}_n \stackrel{\text{def}}{=} \frac{\boldsymbol{q}_{n+1} - \boldsymbol{q}_n}{h},$$

and thus, by Riemann summation

$$\mathcal{A}_L \approx \hat{\mathcal{A}} \stackrel{\text{def}}{=} \sum_{n=0}^{\nu-1} L\left(\boldsymbol{q}_n, \frac{\boldsymbol{q}_{n+1} - \boldsymbol{q}_n}{h}\right) h.$$

At first glance, it looks like this is a crude approximation, since we have employed a one-sided difference, however, let us proceed anyway to see the implications of this choice. In the case of a mechanical system with Lagrangian $L(\boldsymbol{q}, \boldsymbol{v}) = \dot{\boldsymbol{v}}^T \boldsymbol{M} \boldsymbol{v}/2 + U(\boldsymbol{q})$, we then have

$$\hat{\mathcal{A}} = \sum_{n=0}^{\nu-1} \left[\frac{(\boldsymbol{q}_{n+1} - \boldsymbol{q}_n)^T \boldsymbol{M} (\boldsymbol{q}_{n+1} - \boldsymbol{q}_n)}{2h^2} - U(\boldsymbol{q}_n) \right] h.$$

We should think of $\hat{\mathcal{A}}$ as a function of all the positions $\boldsymbol{q}_0, \boldsymbol{q}_1, \ldots, \boldsymbol{q}_\nu$ defining the discrete path. Critical points of this function satisfy

$$\nabla \hat{\mathcal{A}} = 0,$$

where the gradient must be taken with respect to all configurational points on the path (and all coordinates). This condition leads to the equations

$$\frac{\partial \hat{\mathcal{A}}}{\partial \boldsymbol{q}_n} = 0, \qquad n = 1, \ldots, \nu,$$

since we think of the starting point $\boldsymbol{q}_0$ as fixed. To calculate the derivative of $\hat{\mathcal{A}}$ with respect to $\boldsymbol{q}_n$, $n = 1, 2, \ldots, \nu - 1$, note that only a few terms of the discrete action involve this configurational point, thus

$$\begin{aligned}\frac{\partial \hat{\mathcal{A}}}{\partial \boldsymbol{q}_n} &= \frac{\partial}{\partial \boldsymbol{q}_n}\left[\frac{1}{2h}\left((\boldsymbol{q}_n - \boldsymbol{q}_{n-1})^T \boldsymbol{M}(\boldsymbol{q}_n - \boldsymbol{q}_{n-1}) + (\boldsymbol{q}_{n+1} - \boldsymbol{q}_n)^T\right.\right.\\ &\quad \left.\left. \times\ \boldsymbol{M}(\boldsymbol{q}_{n+1} - \boldsymbol{q}_n)\right) - U(\boldsymbol{q}_n)h\right]\\ &= \frac{1}{h}\left[\boldsymbol{M}(\boldsymbol{q}_n - \boldsymbol{q}_{n-1}) - M(\boldsymbol{q}_{n+1} - \boldsymbol{q}_n)\right] - \nabla U(\boldsymbol{q}_n)h,\end{aligned}$$

which yields the equations

$$\boldsymbol{M}(\boldsymbol{q}_{n+1} - 2\boldsymbol{q}_n + \boldsymbol{q}_{n-1}) = -h^2 \nabla U(\boldsymbol{q}_n), \qquad n = 1, 2, \ldots, \nu - 1. \tag{2.4}$$

We can think of

$$\frac{1}{h^2}(\boldsymbol{q}_{n+1} - 2\boldsymbol{q}_n + \boldsymbol{q}_{n-1})$$

as a centered finite difference approximation of the acceleration (the second derivative of position), thus the Eq. (2.4) is a direct discretization of Newton's equations.

Somewhat surprisingly, the one-sided approximation of velocities in the discrete Lagrangian has led to a symmetric discretization of the equations of motion. There is a slight issue of what to do at the endpoints. In the variational formulation we think of the endpoints of the curve as fixed points which means that we are effectively solving a boundary value problem and Eq. (2.4) tells us how to compute all the interior points along the path. In the case of an initial value problem (the usual issue in molecular dynamics) we do not know in advance the right endpoint value, but we assume that we can, in some way, calculate $\boldsymbol{q}_1$ (by a *starting procedure*) and so Eq. (2.4) defines $\boldsymbol{q}_2, \boldsymbol{q}_3, \ldots, \boldsymbol{q}_\nu$.

The method (2.4) is commonly referred to as *Störmer's rule*. It was used by the mathematician Störmer for calculations in the first decade of the 1900s. In molecular dynamics this method is referred to as the Verlet method since it was used by Verlet in his important 1967 paper [387].

The scheme is usually given in an alternative "velocity Verlet" form that takes a step from a given vector $\boldsymbol{q}_n$, $\boldsymbol{v}_n$ to $\boldsymbol{q}_{n+1}$, $\boldsymbol{v}_{n+1}$ by the sequence of operations

$$\boldsymbol{v}_{n+1/2} = \boldsymbol{v}_n + (h/2)\boldsymbol{M}^{-1}\boldsymbol{F}_n, \tag{2.5}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h\boldsymbol{v}_{n+1/2}, \tag{2.6}$$

$$\boldsymbol{v}_{n+1} = \boldsymbol{v}_{n+1/2} + (h/2)\boldsymbol{M}^{-1}\boldsymbol{F}_{n+1}, \tag{2.7}$$

where $\boldsymbol{F}_n = \boldsymbol{F}(\boldsymbol{q}_n) = -\nabla U(\boldsymbol{q}_n)$. The force computed at the end of a given step may be reused at the start of the following step, thus effectively a single force evaluation is needed at each timestep (the method has a similar "cost" to the Euler method, if we measure cost in terms of force evaluations). The derivation of the Störmer form from the velocity Verlet form is straightforward: write down two consecutive steps of (2.5)–(2.7) then eliminate velocities.

The method is widely used but is often stated in other forms, so let us consider these here.

The most straightforward rewriting of the Verlet method is to put the equations and the discretization in Hamiltonian form, i.e. introducing momenta $\boldsymbol{p} = \boldsymbol{M}\boldsymbol{v}$, and thus $\boldsymbol{p}_n = \boldsymbol{M}\boldsymbol{v}_n$, which results in the flow map approximation (taking us from any point in phase space $(\boldsymbol{q}, \boldsymbol{p})$ to a new point $(\boldsymbol{Q}, \boldsymbol{P})$):

$$\boldsymbol{Q} = \boldsymbol{q} + h\boldsymbol{M}^{-1}\boldsymbol{p} + \frac{h^2}{2}\boldsymbol{M}^{-1}\boldsymbol{F}(\boldsymbol{q}), \qquad \boldsymbol{P} = \boldsymbol{p} + \frac{h}{2}[\boldsymbol{F}(\boldsymbol{q}) + \boldsymbol{F}(\boldsymbol{Q})]. \tag{2.8}$$

Alternatively,

$$\boldsymbol{p}_{n+1/2} = \boldsymbol{p}_n + (h/2)\boldsymbol{F}_n, \quad \boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h\boldsymbol{M}^{-1}\boldsymbol{p}_{n+1/2}, \quad \boldsymbol{p}_{n+1} = \boldsymbol{p}_{n+1/2} + (h/2)\boldsymbol{F}_{n+1}.$$

Returning to (2.5)–(2.7), write out the formulas for two successive steps ($(q_{n-1}, v_{n-1}) \mapsto (q_n, v_n)$ and $(q_n, v_n) \mapsto (q_{n+1}, v_{n+1})$) and note that, from

$$\boldsymbol{v}_n = \boldsymbol{v}_{n-1/2} + (h/2)\boldsymbol{M}^{-1}\boldsymbol{F}_n,$$

one has

$$\begin{aligned} \boldsymbol{v}_{n+1/2} &= \boldsymbol{v}_{n-1/2} + h\boldsymbol{M}^{-1}\boldsymbol{F}_n, \\ \boldsymbol{q}_{n+1} &= \boldsymbol{q}_n + h\boldsymbol{v}_{n+1/2}. \end{aligned}$$

How do we use this if, as is typical, initial conditions of the form $\boldsymbol{q}(0) = \boldsymbol{q}_0, \boldsymbol{v}(0) = \boldsymbol{v}_0$ are specified? It is necessary to define an initialization procedure for $\boldsymbol{v}_{-1/2}$:

$$\boldsymbol{v}_{-1/2} = \boldsymbol{v}_0 - (h/2)\boldsymbol{M}^{-1}\boldsymbol{F}(\boldsymbol{q}_0).$$

And it is also necessary to use, at any subsequent step,

$$\boldsymbol{v}_n = \boldsymbol{v}_{n-1/2} + (h/2)\boldsymbol{M}^{-1}\boldsymbol{F}_n,$$

if $\boldsymbol{q}_n, \boldsymbol{v}_n$ are both needed, e.g. for the evaluation of the energy or other velocity-dependent observable.

The difference between formulations is typically subtle, and largely influence the details of computer software design. For many applications the difference can

be safely ignored, but it can have implications when used as part of a more complex algorithm (see [29] and the discussion therein). The use of different formulations can result in differences in the accumulation of rounding error [167].[5]

2.2.3 Convergence and the Order of Accuracy

A typical integrator computes successive steps from the formulas

$$z_{n+1} = \mathscr{G}_h(z_n), \qquad z_0 = \boldsymbol{\zeta}.$$

Assume that $\mathscr{G}_h$ is a smooth map for all $h > 0$. The exact solution satisfies

$$z(t_{n+1}) = \mathscr{F}_h(z(t_n)).$$

To each $h > 0$ we may associate a finite set of phase space points $z_0, z_1, z_2, \ldots, z_\nu$; these represent the numerical solution at $t_0 = 0, t_1 = h, t_2 = 2h, \ldots, t_\nu = \nu h = \tau$.

Taking the difference of the numerical and exact solutions, we have

$$z_{n+1} - z(t_{n+1}) = \mathscr{G}_h(z_n) - \mathscr{F}_h(z(t_n)). \tag{2.9}$$

The first assumption is that $\mathscr{G}_h$ is an $\mathcal{O}(h^{p+1})$ approximation of $\mathscr{F}_h$ in the sense that there is a constant $K \geq 0$ and a constant $\Delta > 0$ such that, for $t \in [0, \tau]$, we have

$$\|\mathscr{F}_h(z(t)) - \mathscr{G}_h(z(t))\| \leq \bar{K} h^{p+1}, \qquad h < \Delta. \tag{2.10}$$

This assumption is usually verified by expanding the numerical and exact solutions in powers of h, using Taylor series expansions.

To tackle the question of the growth of local error, we still must make an important assumption on $\mathscr{G}_h$, namely that it satisfies a *Lipschitz condition* of the form

$$\|\mathscr{G}_h(\boldsymbol{u}) - \mathscr{G}_h(\boldsymbol{w})\| \leq (1 + hL)\|\boldsymbol{u} - \boldsymbol{w}\|, \qquad \boldsymbol{u}, \boldsymbol{w} \in D, h \leq \Delta. \tag{2.11}$$

The set D should be a domain containing the exact solution for $[0, \tau]$, and it is assumed that, for all $h \leq \Delta$ the numerical solution is also contained in D for $n =$

[5] Rounding error is the error introduced when numbers are forced into the finite word length representation in a typical digital computer. Adding together two "computer numbers," then rounding, results in another computer number. Rounding errors may accumulate in long computations, but in molecular dynamics they are normally dominated by the much larger "truncation errors" introduced in the process of discretization, that is, due to replacing the differential equation by a difference equation such as the Euler or Verlet method. For an example of the role of rounding error in the context of constrained molecular dynamics, see [237].

$0, 1, \ldots, \nu$. (This is a simplifying assumption the removal of which would require somewhat more intricate assumptions regarding the differential equations.)

With these assumptions in hand, begin from (2.9) and write

$$z_{n+1} - z(t_{n+1}) = \mathscr{G}_h(z_n) - \mathscr{G}_h(z(t_n)) + \mathscr{G}_h(z(t_n)) - \mathscr{F}_h(z(t_n)),$$

then take norms and use the triangle inequality and (2.10), (2.11) to get the following recurrent inequality for the error $\varepsilon_n = \|z_n - z(t_n)\|$:

$$\varepsilon_{n+1} \leq (1 + Lh)\varepsilon_n + \bar{K}h^{p+1}.$$

From this, the bound

$$\varepsilon_n \leq \frac{\bar{K}}{L} e^{Lnh} h^p, \qquad n = 0, 1, \ldots \nu, \tag{2.12}$$

follows by a straightforward calculation, for $h \leq \Delta$.

The assumption (2.10) that the local error is of order $p + 1$, $p > 0$, is termed the *consistency* of the numerical method. We say the method is *consistent of order p*.

The assumption (2.11) that the method does not increase the separation between two nearby trajectories by more than a factor of the form $1 + hL$ in each step is referred to as the *stability* of the method.

This result shows that a method which is consistent of order p and stable is convergent of order p.

Example 2.2 (The Verlet Method is 2nd Order) In this example, assume a single degree of freedom system, i.e. $q, p \in \mathbb{R}$, and take $M = 1$. The Verlet method can be written in the form of a map, as in (2.8), or, in slightly more detail, as

$$Q = q + hp + \frac{h^2}{2} F(q), \tag{2.13}$$

$$P = p + \frac{h}{2}\left[F(q) + F(q + hp + \frac{h^2}{2}F(q))\right]. \tag{2.14}$$

The first equation is already a polynomial, i.e. it is in the form of a series expansion in powers of h where the coefficients are functions of the starting point (q, p). The second equation may be written as a series expansion in powers of h as well:

$$P = p + \frac{h}{2}F(q) + \frac{h}{2}\left[F(q) + hF'(q)(p + \frac{h}{2}F(q)) + \frac{h^2}{2}F''(q)(p + \frac{h}{2}F(q))^2 + \ldots\right].$$

Note that the neglected terms will involve 4th (and higher) powers of h.

Combining terms of like powers of h, we have

$$P = p + hF + \frac{h^2}{2}pF' + \frac{h^3}{4}\left[F'F + p^2F''\right] + \mathcal{O}(h^4).$$

We carry these terms to compare against the expansion of the exact solution. Since $\dot{q} = p$, we have $\ddot{q} = \dot{p} = F(q)$, and the third derivative is $q^{(3)} = F'(q)p$. On the other hand $\dot{p} = F(q)$ implies that $\ddot{p} = F'(q)\dot{q} = pF'$, and thus

$$p^{(3)} = p^2F'' + F'F.$$

The Taylor expansion of the solution is (taking $q(t) = q, p(t) = p$):

$$q(t+h) = q + hp + \frac{h^2}{2}F + \frac{h^3}{6}F'p + \mathcal{O}(h^4),$$
$$p(t+h) = p + hF + \frac{h^2}{2}pF' + \frac{h^3}{6}\left[p^2F'' + F'F\right] + \mathcal{O}(h^4).$$

We now examine the series expansions for the exact and Verlet solutions and find that these differ in the third (and higher) order terms.

$$Q - q(t+h) = \frac{h^3}{6}F'p + \mathcal{O}(h^4),$$

and

$$P - p(t+h) = \frac{h^3}{12}\left[p^2F'' + F'F\right] + \mathcal{O}(h^4).$$

These relations can be summarized as telling us that

$$\|\mathscr{G}_h(\boldsymbol{z}) - \mathscr{F}_h(\boldsymbol{z})\| = \kappa(\boldsymbol{z})h^3 + \mathcal{O}(h^4),$$

where $\kappa(\boldsymbol{z}) = \kappa(q,p)$ is a function of the position and momentum.

We may then define

$$\bar{K} = \max_{t\in[0,\tau]} \kappa(\boldsymbol{z}(t))$$

bounding the local error by (with neglect of the fourth order terms) $\bar{K}h^3$. Thus the Verlet method is consistent of order two.

To complete the convergence proof for Verlet's method, we would still need to verify the second assumption. This requires the assumption that the force field $\boldsymbol{F}$ satisfy a Lipschitz condition:

$$\|\boldsymbol{F}(\boldsymbol{u}) - \boldsymbol{F}(\boldsymbol{w})\| \leq \hat{L}\|\boldsymbol{u} - \boldsymbol{w}\| \tag{2.15}$$

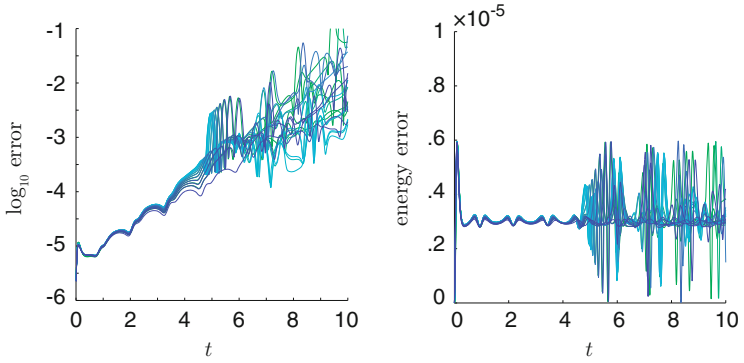

Fig. 2.4 The performance of the Verlet method is demonstrated by repeating the tests of error growth discussed for Euler's method (compare Fig. 2.3). *Left panel*: In the case of the Verlet method, the error is much smaller in an absolute sense than that for the Euler method with the same step-size, but the growth still appears to be exponential. *Right*: the energy error growth as a function of time, very remarkably, compared to the equivalent figure for Euler's method the energy error appears not to grow at all beyond some reasonably small limiting value, suggesting that the Verlet numerical approximation will remain near the desired energy surface

for all $\boldsymbol{u}, \boldsymbol{w}$. Generally speaking this could be taken to hold in a neighborhood of the solution where all approximate solutions for $h < \bar{h}$ are assumed to lie. With a bit of effort, it is then possible to demonstrate the stability condition for the numerical method (see Exercise 4).

In the left-hand panel of Fig. 2.4 we report the Verlet error growth as a function of time for the Lennard-Jones trimer. Verlet is substantially more accurate in this simulation, although the error still appears to grow exponentially.

Due to the chaotic nature of molecular dynamics, which implies a sensitivity to perturbations of the initial condition or the differential equations themselves, it is to be expected that the global error due to using a numerical method will always grow rapidly (exponentially) in time. As we shall see in later chapters, this does not necessarily mean that a long trajectory is entirely without value. In molecular dynamics it turns out that the real importance of the trajectory is that it provides a mechanism for calculating averages that maintain physical parameters. The simplest example of such a parameter is the energy.

What is more relevant for using Verlet to simulate molecular dynamics is the remarkable stability of energy shown in the right-hand panel of Fig. 2.4. Notice that the energy error does not appear to exhibit a steady accumulation in time (unlike for Euler's method, where it exhibited a linear-in-time growth). The explanation for this unexpected behavior lies in the structural properties of the method, a topic we explore in this and the next chapter.

As another illustration of the performance of the Verlet method, we mention that the dynamical trajectories given in the previous chapter (in particular those given Examples 1.8 and 1.9) were computed using this method (with stepsize $h = 0.00001$). In the case of the calculation of the exponential separation of

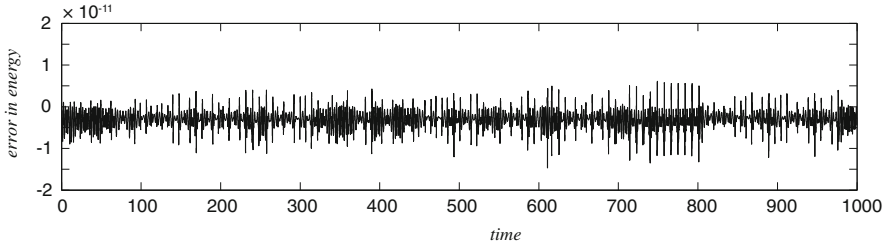

Fig. 2.5 The error in energy in a numerically computed trajectory (Verlet method) with stepsize $h = 0.00001$, for the anisotropic oscillator, showing that it remains bounded by 1.5×10^{-11} in very long runs

trajectories for the anisotropic oscillator shown in Fig. 1.25, the energy errors (see Fig. 2.5) remain bounded and small, suggesting that we may have some confidence in the numerical results presented. This example also provides additional evidence for the lack of "secular growth"[6] of the energy error in Verlet simulations even in a relatively large number (10^8) of timesteps.

2.2.4 *First Integrals and Their Preservation Under Discretization*

Recall that the condition for a given function $I : \mathbb{R}^m \to \mathbb{R}$ to be a first integral is that

$$\nabla I(z) \cdot f(z) = \sum_{j=1}^{m} \frac{\partial I}{\partial z_j}(z) f_j(z) \equiv 0.$$

It is important in this definition that this is an equivalence that holds *everywhere* (or at least in some open set in $\mathbb{R}^m$), so the statement is not just that $I(z(t)) = I(z(0))$ for some particular trajectory, but, moreover, I is conserved for all nearby initial conditions. The flow map preserves the first integral, thus $I(\mathscr{F}_t(z)) \equiv I(z)$. For example, in a Hamiltonian system, the flow map conserves the energy:

$$H \circ \mathscr{F}_t = H.$$

[6]The term "secular growth" in this context is a reference to the long-term growth of perturbations in celestial mechanics. For example, the precession of the Earth's polar axis occurs on a long period relative to its orbital motion and much longer period than its rotation, and so may be classed as a secular motion. In the context of molecular simulations, we use this to refer to accumulation of drift that takes the system steadily away from the energy surface.