

A Simple Variational Integrator for General Holonomic Mechanical Systems

Ulrich Mutze
Heidelberg Digital Finishing GmbH
73347 Mühlhausen, Germany
ulrich.mutze@heidelberg.com
ulrichmutze@aol.com

Abstract

For an arbitrary holonomic mechanical system an integrator is derived by applying an extended principle of stationary action to the manifold of those system paths which are parabolic with respect to the system of generalized coordinates under consideration. A modified variational derivative of the action integral is shown to agree with the force that the environment of the system exerts on it. This generalizes the characterization of free motion in terms of a vanishing variational derivative. As a numerical example, the method is applied to the damped harmonic oscillator. It is shown that the method follows accurately a change of the amplitude by a factor of 1024 within 20 oscillation periods when taking only 32 steps for one such period. The computed oscillation gets out of phase by an angle of 11.5 degrees over these 20 oscillations.

1 Introduction

Modeling the dynamics of granular media (e.g. [14]) is presently gaining acceptance in industry as a method to generate results of practical relevance. The method of the present paper comes from this field.

Although there is no need to develop fundamentally new methods for numerically solving the equations of motion for granular systems, the experience-based preferences for methods from related application fields such as celestial mechanics and molecular dynamics may not be reliable guides in this field.

In granular media the interaction terms are typically expensive to compute since they depend on many parameters, such as the location and size of the zones where two irregularly shaped bodies overlap. Consecutive evaluations of the interaction term result in a sequence of values which is irregular enough to fool polynomial extrapolation more often than not. So, higher order integration methods can't show the performance which they show in more regular problems.

This defines the topic of the present paper, which is the construction of one-stage, one-step integrators for rather general mechanical systems. The motivation for this study was the desire to understand why a specific such method, which will be called the *direct midpoint method* in the following, performs so well in the granular media application [9]. In the most simple framework this method is given by the following self-explanatory integrator formula

$$\begin{aligned}x(t + \Delta t) &= x(t) + v(t)\Delta t + a \frac{(\Delta t)^2}{2}, \\v(t + \Delta t) &= v(t) + a \Delta t, \\a &:= \frac{F(t + \frac{\Delta t}{2}, x(t) + v(t)\frac{\Delta t}{2})}{m}.\end{aligned}$$

The question of particular interest is how to generalize this scheme to velocity dependent forces which are indispensable for achieving inelastic collisions of particles. In the application mentioned above a particle is idealized as a rigid arrangement of strongly overlapping spheres each of which generates and feels forces according to a soft particle model. Friction is very conveniently introduced as hysteresis of elastic forces: during compression, the spheres exchange a stronger force than during release¹ [1].

The agenda of this paper is not to describe the direct midpoint method and its performance in the complex framework of [9] where particle positions are described by elements of the Euclidean group of motions in space and velocities as elements of the Lie algebra of this group. Instead, the method is studied here within the traditional framework of theoretical mechanics which employs generalized coordinates, generalized forces, and a general Lagrangian.

The actual approach is similar to that in [8]. It applies the action principle directly to the state change caused by a single integrator step and arranges things such that a deterministic equation for this state change results. This differs considerably from the usual method to mimic the derivation of the Euler-Lagrange differential equations as difference equations for discrete trajectories. As is shown in this paper, a significant advantage of this *one step variational method* is its ability to cope with forced systems without formal complications.

This approach leads to a family of simple algorithms, with the direct midpoint method as one member, distinguished by simplicity and, in the example of Section 7, also by performance. With expressive exception of the direct midpoint method, these methods are not presented with the intention to add to the toolbox of the computational physicist. Rather they are considered here since they emerge naturally in the exploitation of predictive applications of the action principle. So, they put the main method into perspective and help to appreciate the unique simplicity of this method.

¹ such forces depend obviously on the velocity, although more on their direction than on their magnitude

This approach also unveils a general relation (the theorem in Section 5) between the Euler-Lagrange differential operator and a differential operator acting on the action integral associated with a single time step. This relation between a construct of time-continuous mechanics and one of time-discrete mechanics, which will be called the *small step action principle*, should be considered the main result of this paper. The main goal of this paper would be reached if readers would start testing the direct midpoint method in the framework of their work. That this could happen also outside computational mechanics is indicated in an informal subsection of Section 6.

2 General framework

Let us consider a holonomic mechanical system Σ and restrict our interest to a part Q of the configuration space on which a single system ϕ of n generalized coordinates can be introduced. It would be misleading to consider the set of coordinate n -tuples simply as a subset of \mathbb{R}^n since the primary arithmetical operations on coordinates (those that make \mathbb{R}^n a linear space over \mathbb{R}) do, as a rule, not correspond to operations that can be defined in terms of physical quantities. Only special linear combinations of coordinate n -tuples are meaningful:

1. Linear combinations the coefficients of which add up to 0 are vectors. The most frequently used type of such linear combinations is the difference of two points which determines a *tangent vector* if the points are sufficiently close together. More complex combinations appear in (27).
2. Linear combinations the coefficients of which add up to 1 (i.e. weighted means) are points. The most frequently used type of such a linear combination is the mean of two points with equal weights. This determines the midpoint if the points are sufficiently close together.

We take this into account by considering coordinate n -tuples as elements of the affine point space X over \mathbb{R}^n . It may happen that a computation concerning Σ returns a point $x \in X$ to which there corresponds no physical system configuration or no system configuration for which the system ϕ of coordinates is adequate. Evaluating some system related function f at x will then not be defined. When this occurs, it may indicate an interesting system behavior that should be analyzed based on the information available in that specific case. In computational physics or physical modeling it is most natural to assume that the mathematical representation of the physical system is done in a way that such ‘pathologies’ can’t happen (at least not in the first place): all pertinent functions are defined for all values of the arguments if only their type² (irrespective of the value) matches the declaration

² see e.g. [13] for type theory, which is a more convenient basis for constructive mathematics and modeling than set theory on which mainstream mathematics still relies.

of the function. If during a computation a function gets evaluated for an argument value for which it is only by artificial regularization that a result can be created, the function should automatically document this event. The documentation of exceptional intermediate results is to be taken into account in drawing conclusions from a computation. Even simpler than making all functions ‘everywhere defined’ is to assure that they are *smooth*, i. e. differentiable to any order. A typical action of this kind is to replace $1/r$ in the Coulomb potential by $1/\sqrt{r^2 + \varepsilon^2}$. Throughout this paper, $C(Y, Z)$ is used as synonymous to $C^\infty(Y, Z)$, the set of smooth functions $f : Y \rightarrow Z$ (and not the continuous functions for which this symbol is more often used).

On the ubiquity of smooth functions

For functions which enter a computational model in the form of discrete data that have to be interpolated, a C^∞ -representation might not be obvious since polynomial interpolation is known to be not adequate for more than a few points, and spline curves are not C^∞ . Although it seems not to have been spelled out in the literature, there is a simple and efficient C^∞ -interpolation method [5] based on the remarkable meromorphic function

$$\rho(z, p) := \frac{1}{1 + e^{(\frac{1}{z} + \frac{1}{z-1})p}},$$

where p is a real parameter for which there are reasons³ to choose $p \approx 0.81$. For real $z \in]0, 1[$ this is a strictly increasing function which connects the points $(0, 0)$ and $(1, 1)$ and has a symmetry center at $(\frac{1}{2}, \frac{1}{2})$ and the function

$$r(x) := (0, \rho(x, p), 1) \quad \text{for } (x \leq 0, 0 < x < 1, x \geq 1) \quad \text{respectively}$$

is $C^\infty(\mathbb{R}, \mathbb{R})$ for all $p > 0$. For a list $(x_i, y_i)_{i \in \{1, \dots, m\}}$ of interpolation points we employ a *partition of unity* (w_i) to build the function f

$$f(x) := \sum_{i=1}^m w_i(x) f_i(x),$$

$$w_i(x) := r_{i-1}(x) - r_i(x), \quad i \in \{1, \dots, m\},$$

$$r_0(x) := 1, \quad r_i(x) := r\left(\frac{x - x_i}{x_{i+1} - x_i}\right), \quad i \in \{1, \dots, m-1\}, \quad r_m(x) := 0,$$

$$f_1 := \text{interpolating line for } (x_1, x_2; y_1, y_2),$$

$$f_i := \text{interpol. parabola for } (x_{i-1}, x_i, x_{i+1}; y_{i-1}, y_i, y_{i+1}), \quad i \in \{2, \dots, m-1\},$$

$$f_m := \text{interpolating line for } (x_{m-1}, x_m; y_{m-1}, y_m),$$

which is easily shown to be a function of the desired kind: $f \in C^\infty(\mathbb{R}, \mathbb{R})$ and $f(x_i) = y_i$ for all $i \in \{1, \dots, m\}$. For each x only two terms w_i contribute so that the method is only twice as expensive as local quadratic interpolation, and unlike splines, it requires no initializing run. It is straightforward to extend the method from \mathbb{R} to \mathbb{R}^n . Also, function r can directly be used to turn piecewise defined functions like $f(x) := g(x)$ for $x > 0$ and $f(x) := h(x)$ else, into smooth functions if g and h are smooth.

Dynamics is concerned with configurations that change with time, i.e. curves (paths, trajectories)

$$q \in C(T, X),$$

³The L^2 -norm of the second derivative of r is minimal for $p = 0.810747\dots$

where $T := [t_i, t_f]$ is the time span under consideration. We assume that the physics of Σ is completely defined by the basic *Lagrange-d'Alembert principle*, e.g. [7] p. 421 (3.1.1), which subsequently will be formulated in a way that it defines all the quantities to be used in the subsequent deductions. It will also be referred to simply as the *action principle*.

For objects Σ , X , and $T = [t_i, t_f]$ as introduced so far, there exist functions

$$\begin{aligned}\mathcal{L} &: T \times C(T, X) \rightarrow \mathbb{R}, \\ \mathfrak{F} &: T \times C(T, X) \rightarrow \mathbb{R}^n, \quad \mathfrak{G} : T \times C(T, X) \rightarrow \mathbb{R}^n,\end{aligned}$$

such that for all $q \in C(T, X)$, and all

$$\delta q \in C(T, \mathbb{R}^n) \quad \text{such that} \quad \delta q(t_i) = \delta q(t_f) = 0 \quad ^4$$

the equation

$$E(\mathcal{L}, q, \delta q) + W(\mathfrak{F}, q, \delta q) + W(\mathfrak{G}, q, \delta q) = 0 \quad (1)$$

holds, where

$$E(\mathcal{L}, q, \delta q) := \frac{d}{d\lambda} \int_{t_i}^{t_f} dt \mathcal{L}(t, q + \lambda \delta q) \Big|_{\lambda=0}, \quad (2)$$

$$W(\mathfrak{F}, q, \delta q) := \int_{t_i}^{t_f} dt \delta q(t) \cdot \mathfrak{F}(t, q), \quad (3)$$

$$W(\mathfrak{G}, q, \delta q) := \int_{t_i}^{t_f} dt \delta q(t) \cdot \mathfrak{G}(t, q). \quad (4)$$

The physical meaning of $\mathfrak{G}(t, q)$ is the *guiding force* that the environment has to exert on system Σ at time t in order to let the path q happen. $\mathfrak{F}(t, q)$ is an internal force at time t that is not assumed to be of the kind that can be included in the *Lagrangian* \mathcal{L} . If $\mathfrak{F}(t, q) = 0$ for all $t \in T$ and all $q \in C(T, X)$ the system Σ is said to be *purely Lagrangian*.

The functions \mathcal{L} , \mathfrak{F} , and \mathfrak{G} are *instantaneous* ('local in time') in the sense that their values at (t, q) can be expressed in terms of $q(t)$ and the time derivatives $\dot{q}(t), \ddot{q}(t), \ddot{\ddot{q}}(t), \dots$.

For mechanical systems that occur in nature or in text books on technical mechanics the Lagrangian needs no derivatives higher than the first one ⁵ and this is what will be assumed in the following. We thus have a function

$$L : T \times X \times \mathbb{R}^n \rightarrow \mathbb{R}, \quad (t, x, v) \mapsto L(t, x, v) \quad (5)$$

⁴ see Figure 1

⁵ man-made mechanisms containing motors, sensors, and signal processors might well need higher derivatives

which allows writing $\mathfrak{L}(t, q) = L(t, q(t), \dot{q}(t))$ and

$$E(\mathfrak{L}, q, \delta q) = \int_{t_i}^{t_f} dt \delta q(t) \cdot [(\nabla_x L)(t, q(t), \dot{q}(t)) - \frac{d}{dt}(\nabla_v L)(t, q(t), \dot{q}(t))] . \quad (6)$$

This allows exploiting the arbitrariness of δq to get from equations (1), (2), (3), (4), and (6)

$$(\nabla_x L)(t, q(t), \dot{q}(t)) - \frac{d}{dt}(\nabla_v L)(t, q(t), \dot{q}(t)) + \mathfrak{F}(t, q) + \mathfrak{G}(t, q) = 0 . \quad (7)$$

Finally we specify the forces \mathfrak{F} such that they depend on the same variables as the Lagrangian: $\mathfrak{F}(t, q) = F(t, q(t), \dot{q}(t))$ and get a function

$$F : T \times X \times \mathbb{R}^n \rightarrow \mathbb{R}^n , \quad (t, x, v) \mapsto F(t, x, v) . \quad (8)$$

In all that follows we assume

$$L \in C(T \times X \times \mathbb{R}^n, \mathbb{R}) , \quad F \in C(T \times X \times \mathbb{R}^n, \mathbb{R}^n) .$$

Then we conclude from (7) that $\mathfrak{G}(t, q) = G(t, q(t), \dot{q}(t), \ddot{q}(t))$ with a function

$$G : T \times X \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n , \quad (t, x, v, a) \mapsto G(t, x, v, a) \quad (9)$$

for which (7) and (8) entail the representation

$$\begin{aligned} G_i(t, x, v, a) &= L_{v_i v_k}(t, x, v) a_k + L_{v_i x_k}(t, x, v) v_k \\ &\quad + L_{v_i t}(t, x, v) - L_{x_i}(t, x, v) - F_i(t, x, v) , \end{aligned} \quad (10)$$

where the indexes to L denote partial derivatives. In terms of differential operators this can be rewritten as:

$$G(t, x, v, a) = ([(\partial_t + v \cdot \nabla_x + a \cdot \nabla_v) \nabla_v - \nabla_x] L - F)(t, x, v) . \quad (11)$$

An autonomous path, i.e. a path that occurs if Σ does not interact with external systems, is characterized by vanishing guiding force so that it satisfies the condition

$$G(t, q(t), \dot{q}(t), \ddot{q}(t)) = 0 \quad \text{for all } t \in T \quad (12)$$

which is the *Euler-Lagrange differential equation* of the system Σ . It is also said to be an *equation of motion* of the system.

The totality of autonomous paths gives rise to the *evolution map* which associates with each subinterval $\tilde{T} = [t_1, t_2]$ of T the function

$$\Psi(\tilde{T}) : X \times \mathbb{R}^n \rightarrow X \times \mathbb{R}^n , \quad (x, v) \mapsto (q(t_2), \dot{q}(t_2)) , \quad (13)$$

where q is the trajectory defined by the equation of motion (12) and the initial conditions $q(t_1) = x$, $\dot{q}(t_1) = v$. For each partition $\tilde{T} = \cup_{i=1}^n \tilde{T}_n$ we have the important property

$$\Psi(\tilde{T}) = \Psi(\tilde{T}_n) \circ \Psi(\tilde{T}_{n-1}) \circ \dots \circ \Psi(\tilde{T}_1) . \quad (14)$$

An *integrator* or *small step evolution map* for the system Σ (or its equation of motion) is a function $\tilde{\Psi}$ which has the same domain and target as Ψ and has the property that replacing the factors $\Psi(\tilde{T}_k)$ on the right hand side of equation (14) by $\tilde{\Psi}(\tilde{T}_k)$ conserves the equation to any reasonably desired degree of accuracy if the intervals \tilde{T}_k are chosen sufficiently small. It is a well understood fact that integrators for very complicated physical systems can be given as rather simple finite formulas whereas a finite formula for $\Psi(\tilde{T})$ may not exist. This is the basis for computational models of complex dynamical systems.

The physical meaning of G

With the advent of programmable stepper motor drives it became easily possible to enforce the space-time path of a mechanical test system and to record the (potentially strong) forces G that the driving mechanism had to exert on the test system to make it follow that path. Even without this technology we can feel the strong forces needed to suddenly elevate a massive body (think of shot-putting) for which the free motion would be the free fall. The usual presentation of the action principle considers only paths that deviate from autonomous ones slightly and therefore need only small guiding forces. These forces don't enter the result of the reasoning. For following the thread of thought in the present article it helps to be aware of G as a measurable response quantity that is associated with a mechanical system just as, say, the impedance with a two-terminal electric circuit. We will encounter this quantity G in many places in this paper, particularly in equation (60) of the main theorem. The most important feature of G is that, unlike forces and the Lagrangian, it depends not only on (t, x, v) , which is a descriptor for the (dynamical) *state* of the system, but also on the acceleration a , as is particularly obvious in the shot-putting example. See also (88) for an example in terms of a formula. Knowing the expression G is more than simply knowing an equation of motion: Multiplying G by any non-zero factor will again provide an equation of motion, but it will lose the capability to predict reaction forces of the system to the out-side world.

Needless to say that the forces F and G are generalized forces referring to the generalized coordinate system ϕ . So they may not be forces by dimension but the quantity $F \cdot \dot{q}$ has always the dimension of a power. As is well known, the concept of generalized coordinates arose in mechanics as an effective method to describe constraint motion and the systems for which this works are named *holonomic*. It is in this sense that the title of this article should be understood.

Finally, we introduce the function arising from solving the equation $G = 0$ for the acceleration a :

$$\begin{aligned} A(t, x, v) &:= (\text{that } a \text{ for which } G(t, x, v, a) = 0) \\ &= [L_{vv}(t, x, v)]^{-1} [F(t, x, v) + L_x(t, x, v) \\ &\quad - L_{vt}(t, x, v) - L_{vx}(t, x, v)v]. \end{aligned} \quad (15)$$

Here, the negative exponent denotes for $n > 1$ (i.e. for more than one degree of freedom) the inversion of a matrix. If there are points (t, x, v) where this inverse does not exist, the behavior of the system certainly deserves special investigation. A useful universal strategy is to understand the inversion as the always existing pseudo-inverse. For systems of point particles in Cartesian coordinates the matrix L_{vv} is diagonal so that matrix inversion becomes inversion of numbers. If F , L_x , L_{vt} , and L_{vx} depend on v trivially, and L_{vx} vanishes, A depends on v trivially which gives the time stepping algorithm in Section 6 a particularly simple form.

The expression A allows writing the equation of motion in explicit form

$$\ddot{x} = A(t, x, \dot{x}) \quad (16)$$

in which the physical context is completely hidden. On the other hand, the schematic construct

$$L(t, x, v) := \frac{1}{2}v \cdot v, \quad F(t, x, v) := A(t, x, v),$$

gives (16) back as an equation of motion.

3 Integrators based on parabolic paths

As is clear from (14), an integrator needs to represent the true time evolution of a system only for small steps of time. As a small step time evolution map it needs no notion of a trajectory between the initial state and the final state which are connected by this mapping. However, the strategy for obtaining formulas for the small step evolution certainly will benefit from a parameterized mathematical model for that hidden trajectory.

It is the methodological decision which underlies the present paper to use a second order, or parabolic, model based on the system of coordinates under consideration. This then implies, as will be worked out subsequently, that position change and velocity change along the short span of time can be parameterized by a single vector which has the physical meaning of an acceleration. It is important to recognize that this vector may (and normally will) differ from the actual acceleration of the system at any instant of time and that the most natural approximate identification is with the true acceleration in the middle of the time interval. Valuable heuristics concerning a closely related situation is given in [3]. For higher order models

more parameters would be needed and, as a consequence, parameters would enter for which physics provides no basic determining relations comparable to the way acceleration is determined by force.

There is a growing popularity of the thought that physical models could reflect information processes that in some sense actually occur in nature and that require only a limited supply of memory and bandwidth (i.e. computational resources), see e.g. [15] and citations therein. From such a point of view, integrators (rather than equations of motion) are the fundamental descriptors of motion since already one variable ranging over an interval between any two different real numbers would need an infinite number of bits for representation. Providing a single ‘dependent variable’ such as $\sin(\omega t)$ would not only need an infinity of bits but also an infinity of computational steps. Who considers integrators as fundamental encodings of physical laws, could expect simplicity to single out ‘realistic’ encodings as it singles out realistic equations of motion. From such a point of view, the restriction to the parabolic path model is mandatory since higher order models only add complexity.

The time interval on which an integrator depends as a variable and that was denoted \tilde{T} in Section 2 will now be denoted $T = [t, t + 2\tau]$. Having decided to make a model for what happens within this time span, we will need a variable ranging over T . Since the name t is no longer available for this purpose, we write this variable as $t + h$, $h \in [0, 2\tau]$ and the derivation dot now denotes derivation with respect to h ⁶. Parabolic paths form a subset

$$C_2(T, X) := \{q \in C(T, X) \mid \ddot{q}(t+h) = 0 \text{ for all } h \in [0, 2\tau]\} \quad (17)$$

of $C(T, X)$ where the index 2 stands for a polynomial degree and not for a degree of differentiability. Obviously $q \in C_2(T, X)$ is uniquely determined by the point $q(t) \in X$ and the two vectors $\dot{q}(t) \in \mathbb{R}^n$, $\ddot{q}(t) \in \mathbb{R}^n$. Given these objects under names x , v , and a we have an explicit representation of q in the form

$$q(t+h) = x + hv + \frac{h^2}{2}a \text{ for all } h \in [0, 2\tau]. \quad (18)$$

Defining a small step evolution map requires to determine $q(t + 2\tau)$ and $\dot{q}(t + 2\tau)$ from $q(t) = x$ and $\dot{q}(t) = v$. All what is not yet fixed is the acceleration a in representation (18): Given a we complete the definition by

$$(x, v) \mapsto (x + 2\tau v + \frac{(2\tau)^2}{2}a, v + 2\tau a). \quad (19)$$

The rule for finding a , when expressed in the form

$$I(t, x, v, a, \tau) = 0 \quad (20)$$

⁶ Latin fortunately has two words for time: *tempus* and - rarely - *hora*.

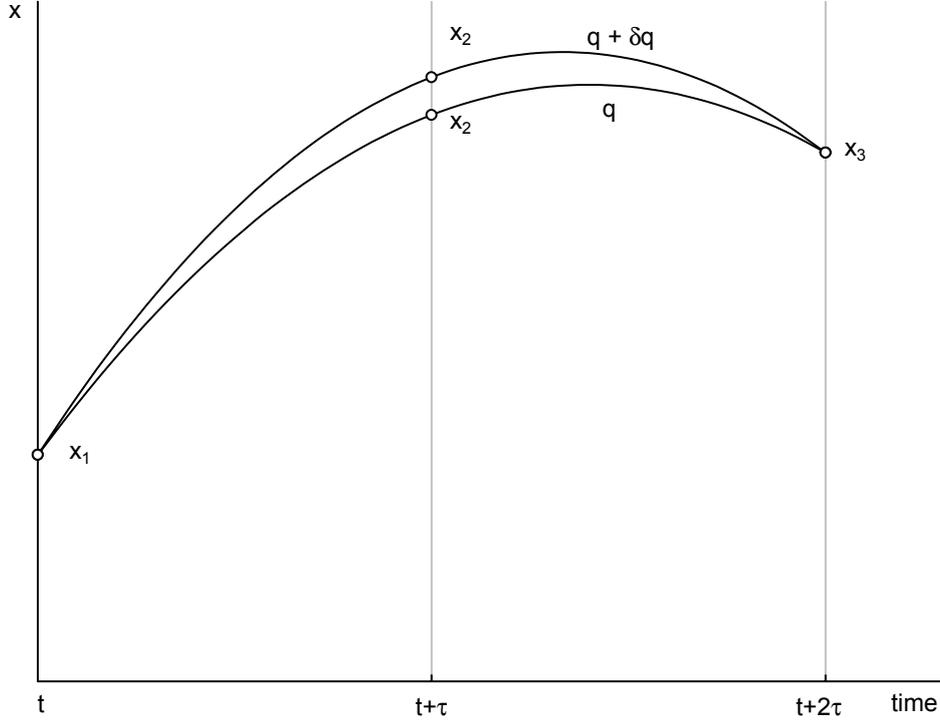


Figure 1: geometry of parabolic paths and variations

with a function $I : \{t\} \times X \times \mathbb{R}^n \times \mathbb{R}^n \times \{\tau\} \rightarrow \mathbb{R}^n$, will be referred to as a *small step equation of motion*⁷. All integrators to be defined in the following will be of this type, differing only in the formula that defines function I . These formulas will be developed systematically in Section 4.

The rest of the present section is devoted to making plausible an important special case. To start with, we express small step equations of motion in the *three-point form* obtained by denoting a parabolic path by three points at equidistant times as shown in Figure 1. To put this into formulas we define the two mappings

$$\begin{aligned}
 \Pi_1 : C_2(T, X) &\rightarrow Y_T, \\
 q &\mapsto (t, x := q(t), v := \dot{q}(t), a := \ddot{q}(t), \tau), \\
 \text{where } Y_T &:= \{t\} \times X \times \mathbb{R}^n \times \mathbb{R}^n \times \{\tau\}, \\
 \Pi_2 : C_2(T, X) &\rightarrow Z_T, \\
 q &\mapsto (t, x_1 := q(t), x_2 := q(t + \tau), x_3 := q(t + 2\tau), \tau), \\
 \text{where } Z_T &:= \{t\} \times X \times X \times X \times \{\tau\}
 \end{aligned} \tag{21}$$

⁷In all other cases the 'prefix' *small step* qualifies a notion that involves a time step so that the word *small* carries most of the intended meaning. In the present case, since an equation of motion does not refer to a time step, both words contribute equally to the intended meaning.

which are obviously bijective. The three-point form \tilde{I} of a small step equation of motion I is given by

$$\tilde{I} \circ \Pi_2 = I \circ \Pi_1 . \quad (22)$$

The action principle is concerned with variations of a path for which the endpoints are held fixed. In the present context such a variation corresponds simply to a variation of the variable x_2 . So it is a natural idea to express the action principle as a small step equation of motion in the following form:

$$\tilde{I}(t, x_1, x_2, x_3, \tau) := \frac{\partial \tilde{S}}{\partial x_2}(t, x_1, x_2, x_3, \tau) ,$$

where

$$\tilde{S}(t, x_1, x_2, x_3, \tau) := \int_0^{2\tau} dh L(t+h, x+vh+a\frac{h^2}{2}, v+ah) , \quad (23)$$

where x, v, a on the right-hand side are given by

$$(t, x, v, a, \tau) := \Pi_1(\Pi_2^{-1}(t, x_1, x_2, x_3, \tau)) .$$

In the next section (23) will be deduced from the action principle. It will also be generalized to not purely Lagrangian systems.

It is interesting to transform also the descriptors of a time step to the three-point notation: Assume that step data (t, x_1, x_2, x_3, τ) are given then the data $(t+2\tau, \underline{x}_1, \underline{x}_2, \underline{x}_3, \underline{\tau})$ for the next time step of arbitrary duration $\underline{\tau}$ are determined by

$$\begin{aligned} \underline{x}_1 &= x_3 \\ \frac{3}{2} \frac{\underline{x}_2 - \underline{x}_1}{\underline{\tau}} - \frac{1}{2} \frac{\underline{x}_3 - \underline{x}_2}{\underline{\tau}} &= \frac{3}{2} \frac{x_3 - x_2}{\tau} - \frac{1}{2} \frac{x_2 - x_1}{\tau} \\ \tilde{I}(t+2\tau, \underline{x}_1, \underline{x}_2, \underline{x}_3, \underline{\tau}) &= 0 . \end{aligned} \quad (24)$$

This follows easily from the fact that

$$q = \Pi_1^{-1}(t, x, v, a, \tau) = \Pi_2^{-1}(t, x_1, x_2, x_3, \tau) \quad (25)$$

implies

$$\begin{aligned} x_1 &= x , \\ x_2 &= x + v\tau + a\frac{\tau^2}{2} , \\ x_3 &= x + v(2\tau) + a\frac{(2\tau)^2}{2} , \end{aligned} \quad (26)$$

and

$$\begin{aligned} x &= x_1 , \\ v &= \frac{-3x_1 + 4x_2 - x_3}{2\tau} = \frac{3}{2} \frac{x_2 - x_1}{\tau} - \frac{1}{2} \frac{x_3 - x_2}{\tau} , \\ a &= \frac{x_1 - 2x_2 + x_3}{\tau^2} = \frac{1}{\tau} \left(\frac{x_3 - x_2}{\tau} - \frac{x_2 - x_1}{\tau} \right) , \end{aligned} \quad (27)$$

and

$$v_3 := \dot{q}(t + 2\tau) = v + 2\tau a = \frac{3}{2} \frac{x_3 - x_2}{\tau} - \frac{1}{2} \frac{x_2 - x_1}{\tau} . \quad (28)$$

Using the original path descriptors the same stepping rule can be expressed much shorter: Assume that step data (t, x, v, a, τ) are given then the data $(t + 2\tau, \underline{x}, \underline{v}, \underline{a}, \underline{\tau})$ for the next time step are determined by

$$\begin{aligned} \underline{v} &= v + 2\tau a , \\ \underline{x} &= x + \tau(v + \underline{v}) , \\ I(t + 2\tau, \underline{x}, \underline{v}, \underline{a}, \underline{\tau}) &= 0 . \end{aligned} \quad (29)$$

The specific small step equation of motion (23) reads then as follows

$$\begin{aligned} I(t, x, v, a, \tau) &:= \left[\frac{2}{\tau^2} (\tau \nabla_v - \nabla_a) S \right] (t, x, v, a, \tau) , \\ \text{where} & \\ S(t, x, v, a, \tau) &:= \int_0^{2\tau} dh L(t + h, x + vh + a \frac{h^2}{2}, v + ah) . \end{aligned} \quad (30)$$

Notice that \underline{a} is not needed to perform the step, it is needed however to enable the next one. Notice also that the formalism allows to change the time step size from one step to the next (i.e. $\underline{\tau}$ needs not to equal τ) so that small step equations of motion generate *asynchronous integrators*. It could be hard to develop an intuitive understanding of the differential operator in (30) if its origination from $\frac{\partial}{\partial x_2}$ would not be known.

4 The action principle applied to parabolic paths

In this section we will see that (23) and (30) are consequences of the action principle as formulated in Section 2 and that the derivation naturally extends to non-conservative forces.

Parabolic path variations δq form a subset of $C(T, \mathbb{R}^n)$

$$\hat{C}_2(T, \mathbb{R}^n) := \{ \delta q \in C_2(T, \mathbb{R}^n) \mid \delta q(t) = \delta q(t + 2\tau) = 0 \} , \quad (31)$$

which together with $C_2(T, X)$ makes up an affine space so that $q + \delta q$ is a path and $q_1 - q_2$ is a path variation for paths q_1 and q_2 satisfying $q_1(t) = q_2(t)$ and $q_1(t + 2\tau) = q_2(t + 2\tau)$. Obviously the space $\hat{C}_2(T, \mathbb{R})$ is one-dimensional and consists of the real multiples of $\sigma \in \hat{C}_2(T, \mathbb{R})$ defined by

$$\sigma(t + h) := \frac{3}{4\tau^3} h(2\tau - h) \quad \text{for all } h \in [0, 2\tau] \quad (32)$$

and each $\delta q \in \hat{C}_2(T, \mathbb{R}^n)$ can be written as

$$\delta q = e\sigma, \quad e \in \mathbb{R}^n, \quad (33)$$

where the vector e is uniquely determined. Here σ is chosen to have the dimension of a frequency, actually $\int_0^{2\tau} dh \sigma(t+h) = 1$, and the components of e may differ in dimension if the components of X do so. Throughout the rest of the paper quantities $\delta q, e, \sigma$ will be understood as satisfying (33). Unlike the general situation in the calculus of variations, the path variations form a space of finite dimension n ; therefore stationarity entails algebraic equations instead of differential equations.

We are now in a position to exploit (1) within the parabolic path framework. Therefore, we compute the quantities (2),(3), and (4). The result will be reached with equations (52) and (53).

In (2) we have to compute a directional derivative (variation). The influence of this variation on the parameters (x, v, a) and (x_1, x_2, x_3) of a path follows from

$$\dot{\sigma}(t) = \frac{3}{2\tau^2}, \quad \ddot{\sigma}(t) = \frac{-3}{2\tau^3}, \quad \sigma(t+\tau) = \frac{3}{4\tau} \quad (34)$$

to be

$$\begin{aligned} q + \lambda \delta q &= q + e\lambda\sigma = \Pi_1^{-1}(t, x, v + e\lambda\frac{3}{2\tau^2}, a + e\lambda\frac{-3}{2\tau^3}, \tau) \\ &= \Pi_2^{-1}(t, x_1, x_2 + e\lambda\frac{3}{4\tau}, x_3, \tau). \end{aligned} \quad (35)$$

For each $\Phi \in C(C_2(T, X), \mathbb{R})$ we thus have

$$\begin{aligned} (\delta\Phi)(q) &:= \left. \frac{d}{d\lambda} \Phi(q + \lambda\delta q) \right|_{\lambda=0} \\ &= e \cdot [D_\tau(\Phi \circ \Pi_1^{-1})](t, x, v, a, \tau), \\ &= e \cdot [\partial_\tau(\Phi \circ \Pi_2^{-1})](t, x_1, x_2, x_3, \tau) \end{aligned} \quad (36)$$

for the following first order, n -component, differential operators ⁸:

$$D_\tau : C(Y_T, \mathbb{R}) \rightarrow C(Y_T, \mathbb{R}), \quad D_\tau := \frac{3}{2\tau^3}(\tau\nabla_v - \nabla_a), \quad (37)$$

$$\partial_\tau : C(Z_T, \mathbb{R}) \rightarrow C(Z_T, \mathbb{R}), \quad \partial_\tau := \frac{3}{4\tau} \frac{\partial}{\partial x_2}. \quad (38)$$

Defining the *action integral* $S_L \in C(Y_T, \mathbb{R})$ by

$$\begin{aligned} \int_0^{2\tau} dh \mathfrak{L}(t+h, q) &= \int_0^{2\tau} dh L(t+h, x + vh + a\frac{h^2}{2}, v + ah) \\ &=: S_L(t, x, v, a, \tau) \end{aligned} \quad (39)$$

⁸each component of D_τ and ∂_τ has source and target as indicated

we get from (36) for the quantity defined in equation (2)

$$E(\mathcal{L}, q, \delta q) = e \cdot (D_\tau S_L)(t, x, v, a, \tau) . \quad (40)$$

Notice that this does not make use of the integration by parts representation (6).

Equations (3) and (4) imply

$$\begin{aligned} \int_0^{2\tau} dh \delta q(t+h) \mathfrak{F}(t+h, q) &= \int_0^{2\tau} dh e \sigma(t+h) F(t+h, x+vh + a \frac{h^2}{2}, v+ah) \\ &=: e \cdot W_F(t, x, v, a, \tau) , \end{aligned} \quad (41)$$

$$\begin{aligned} \int_0^{2\tau} dh \delta q(t+h) \mathfrak{G}(t+h, q) &= \int_0^{2\tau} dh e \sigma(t+h) G(t+h, x+vh + a \frac{h^2}{2}, v+ah, a) \\ &=: e \cdot W_G(t, x, v, a, \tau) \end{aligned} \quad (42)$$

which defines $W_F \in C(Y_T, \mathbb{R}^n)$ and $W_G \in C(Y_T, \mathbb{R}^n)$.

The basic equation (1) expressing the action principle can now be expressed for parabolic paths as

$$D_\tau S_L + W_F + W_G = 0 . \quad (43)$$

Looking for autonomous trajectories we are facing an interesting situation: Since all paths under consideration are parabolic, we should not assume that a strictly autonomous path can be found among them. Each parabolic path will satisfy (43) only with small guiding forces in action. This means that the integrand of the integral W_G will be small but will not vanish. However, setting the whole integral equal to zero results in as many equations as the acceleration a has components so that for given (t, x, v, τ) a value a can be expected to exist such that

$$\boxed{(D_\tau S_L)(t, x, v, a, \tau) + W_F(t, x, v, a, \tau) = 0 .} \quad (44)$$

This is the first of three forms of small step equations of motion that the parabolic path framework suggests. As will be shown soon (see equation (57)) it is closely related to the *discrete Euler-Lagrange equation* of discrete variational mechanics. The path $\Pi_1^{-1}(t, x, v, a, \tau) \in C_2(T, X)$ can be considered the natural solution to the initial value problem within space $C_2(T, X)$ since the quality measure of the approximation (a weighted integral over the guiding force) is taken directly from the physics of the problem and not, for instance, from the geometry of the path which may be badly represented by our generalized coordinates anyway. I'll refer to (44) as the *Euler-Lagrange equation for parabolic paths*.

Let us try to let the forces enter as a contribution to the action integral i.e. transform W_F into $D_\tau S_F$ for a suitable action integral S_F . This is not possible in a consistent

manner for the general case of long paths as lucidly explained in [11], §33, equation 11. A superficial argument in favor of the opposite opinion is given in [4]. For short paths, we have a particular situation that was already exploited in [8]. There the contribution of the forces to the action was defined relative to an autonomous path and fortunately this reference path canceled out in the integrator formula. Here is an improved formulation which uses an explicitly given reference path, namely the linear path

$$q_0 := \Pi_1^{-1}(t, x, v + a\tau, 0, \tau) \quad (45)$$

which satisfies $q(t) = q_0(t)$ and $q(t + 2\tau) = q_0(t + 2\tau)$. This path allows us to form a plausible action integral of forces

$$\begin{aligned} S_F(t, x, v, a, \tau) &:= \int_0^{2\tau} dh [q(t+h) - q_0(t+h)] \cdot \mathfrak{F}(t+h, q) \\ &= \int_0^{2\tau} dh \frac{h}{2} (h - 2\tau) a \cdot F\left(t+h, x + vh + a \frac{h^2}{2}, v + ah\right). \end{aligned} \quad (46)$$

We have

$$\begin{aligned} &e \cdot (D_\tau S_F)(t, x, v, a, \tau) \\ &= \delta \int_0^{2\tau} dh [q(t+h) - q_0(t+h)] \cdot \mathfrak{F}(t+h, q) \\ &= \int_0^{2\tau} dh \delta q(t+h) \cdot \mathfrak{F}(t+h, q) + \int_0^{2\tau} dh [q(t+h) - q_0(t+h)] \cdot (\delta \mathfrak{F})(t+h, q) \\ &= e \cdot W_F(t, x, v, a, \tau) + \int_0^{2\tau} dh \frac{h}{2} (h - 2\tau) \frac{d}{d\lambda} a \cdot \mathfrak{F}(t+h, q + \lambda \delta q) \Big|_{\lambda=0} \\ &= e \cdot W_F(t, x, v, a, \tau) + O(\tau^2). \end{aligned} \quad (47)$$

That the last integral, let us call it I , is indeed $O(\tau^2)$ can be seen as follows:

$$\begin{aligned} I &= \int_0^{2\tau} dh \frac{h}{2} (h - 2\tau) \frac{d}{d\lambda} a \cdot F\left(t+h, x + vh + a \frac{h^2}{2}\right. \\ &\quad \left. + \lambda e \frac{3}{4\tau^3} h(2\tau - h), v + ah + \lambda \frac{3}{2\tau^3} e(\tau - h)\right) \Big|_{\lambda=0} \\ &= \frac{3}{2\tau^3} \int_0^{2\tau} dh \frac{1}{4} [h(h - 2\tau)]^2 f_1\left(t+h, x + vh + a \frac{h^2}{2}, v + ah, a\right) \\ &\quad + \frac{3}{2\tau^3} \int_0^{2\tau} dh \frac{h}{2} (h - 2\tau)(h - \tau) f_2\left(t+h, x + vh + a \frac{h^2}{2}, v + ah, a\right), \end{aligned} \quad (48)$$

where $f_1 := (e \cdot \nabla_x)(a \cdot F)$, $f_2 := (e \cdot \nabla_v)(a \cdot F)$. The f_1 -integral is obviously $O(\tau^2)$ and for the f_2 -integral, integration by parts gives

$$-\frac{3}{2\tau^3} \int_0^{2\tau} dh \frac{1}{8} [h(h-2\tau)]^2 \frac{d}{dh} f_2(t+h, x+vh+a\frac{h^2}{2}, v+ah, a) \quad (49)$$

which is $O(\tau^2)$ too. We thus have

$$D_\tau S_F = W_F + O(\tau^2) . \quad (50)$$

Returning to equation (42) we get by Simpson's rule

$$\begin{aligned} W_G(t, x, v, a, \tau) &= \int_0^{2\tau} dh \frac{3}{4\tau^3} h(2\tau-h) G(t+h, x+vh+a\frac{h^2}{2}, v+ah, a) \\ &= G(t+\tau, x+v\tau+a\frac{\tau^2}{2}, v+a\tau, a) + O(\tau^2) . \end{aligned} \quad (51)$$

Inserting results (50), and (51) into (43) we finally get

$$(D_\tau S)(t, x, v, a, \tau) = -G(t+\tau, x+v\tau+a\frac{\tau^2}{2}, v+a\tau, a) + O(\tau^2) , \quad (52)$$

where

$$S := S_L + S_F \quad (53)$$

which I will refer to as the *small step action principle*. The $O(\tau^2)$ -residual will be discussed in Section 5. There are three ways of reading this principle:

1. Looking at G as a physical quantity (the guiding force) the principle shows how this force is determined by the system path by setting G equal to some explicit expression depending on this path.
2. Looking at G as an expression depending on a path we see the principle determining the path for which G has a certain value.
3. Looking at both sides as explicit expressions made of the quantities L, F, q we see the principle stating a mathematical fact. For the author the most surprising side of this fact was its generality. For a long time some misconception led me to believe that it would hold only for Lagrangians of the special form $L(t, x, v) = T(v) - V(t, x)$ and forced me to look for modifications of growing complexity to cover the case of velocity dependent forces.

For determining autonomous paths the right hand side can be set to zero yielding

$$(D_\tau S)(t, x, v, a, \tau) = 0 . \quad (54)$$

This equation determines a , and hence the path $q = \Pi_1^{-1}(t, x, v, a, \tau)$ if t, x, v, τ are given. This is our second form of a small step equations of motion.

Let us repeat the three-point re-formulation of (54) for reference since in Section 3 it was not stated for non-conservative forces: The *three-point version* \tilde{S} of S is defined by

$$\tilde{S}(t, x_1, x_2, x_3, \tau) := S(t, x, v, a, \tau), \quad (55)$$

where x, v , and a are given by (27). Equations (37), (38) imply

$$\frac{3}{4\tau} \frac{\partial \tilde{S}}{\partial x_2}(t, x_1, x_2, x_3, \tau) = (D_\tau S)(t, x, v, a, \tau) \quad (56)$$

so that (54) is equivalent to

$$\frac{\partial \tilde{S}}{\partial x_2}(t, x_1, x_2, x_3, \tau) = 0 \quad (57)$$

and

$$x_1 = x, \quad \frac{3}{2} \frac{x_2 - x_1}{\tau} - \frac{1}{2} \frac{x_3 - x_2}{\tau} = v$$

expresses the initial condition $q(t) = x, \dot{q}(t) = v$. The two equivalent formulations (54) and (57) of the initial value problem for parabolic paths will be referred to as the *midpoint variation method*.

More on the contribution of forces to the action integral

For use in this subsection we rename the time variable from $t+h$ to t . Let us look into this combined contribution of Lagrangian and forces to a single action integral for paths that are not necessarily parabolic. In the light of the action principle the *raison d'être* of the Lagrangian is to associate a value of the action with each finite piece of a system path. If the Lagrangian would in addition to its usual arguments (t, x, v) depend also on the initial time t_i and the final time t_f of the path, this would not change L 's ability to define a function on path segments. The same would be true if it would depend in addition on the acceleration. Within such a framework we define an *extended Lagrangian*

$$K(t_i, t, t_f, x, v, a) := L(t, x, v) + \frac{(t-t_i)(t-t_f)}{2} a \cdot F(t, x, v) \quad (58)$$

which allows us extend the definition of the action integral (53) to an arbitrary path as follows

$$S(t_i, t_f, q) := \int_{t_i}^{t_f} dt K(t_i, t, t_f, q(t), \dot{q}(t), \ddot{q}(t)). \quad (59)$$

To analyze the behavior of the expression we divide the time interval $[t_i, t_f]$ into subintervals. To each subinterval the definition of S applies, and we have a well defined action integral over each subinterval. Without the contribution of the F -term to K the action integral would be additive with respect to such a partition. The relative contribution of the F -term and the L -term to the value of the action integral S depends strongly on $t_f - t_i$. But the contribution to variations of its value is balanced such that it makes no difference whether we treat a potential as a contribution to S_L or its gradient as a contribution to S_F . With growing values of $t_f - t_i$ the combination of L and F into one expression (58) becomes increasingly meaningless. We thus have the remarkable situation that for small-step evolution (the realm of integrators) a forced mechanical system shows a *unity of behavior* of forces and potentials that gets lost for larger time steps.

Equation (57) is closely related to the *discrete Euler-Lagrange equation* of [7],(1.3.3),p. 372. Setting there

$$S_d(q_1, q_2, q_3) := L_d(q_1, q_2) + L_d(q_2, q_3)$$

this equation can be written as

$$D_2 S_d(q_1, q_2, q_3) = 0 ,$$

which has the same meaning as (57). Equation [7],(1.3.3) is stated for purely Lagrangian systems. The corresponding equation [7],(3.2.2) for forced systems is more complex. Whilst discrete mechanics as developed in [7] has a two-point action function (discrete Lagrangian) as the basic input and considers discrete trajectories, the method of [8] is based on the three-point function. This structure is rich enough to allow a formulation of the dynamical law as a one-stage, one-step integrator. The composition of many steps into a discrete trajectory is then a pragmatic procedure which is not needed for the formulation and solution of the equation of motion. In [8] the three-point function is not introduced as an integral of a continuously defined function along a parabolic path, but by direct discretization just as the discrete Lagrangians are obtained in [7].

In summary, the three- point method enjoys the following remarkable properties:

1. Forces for which there is no potential can be taken into account as a contribution to the three-point function (see (46), (53), (55)).
2. The equation of motion can be written in terms of the three-point function (see (57)).

5 Analysis of the small step action principle in terms of series expansions

In this section we verify and extend (52) by a different method which is very powerful for this purpose but is not suitable for discovering such equations. The derivation of (52) could lead one to expect that $g = \frac{1}{2}$ would be a distinguished value in (60). Thus the last part of the theorem is so far interesting as it falsifies this expectation. We will encounter the factor $\frac{3}{5}$ again in equation (90).

Theorem *Let $t, \tau, L, F, G, S, D_\tau$ be as defined so far. Then*

$$(D_\tau S)(t, x, v, a, \tau) = -G(t + \tau, x + v\tau + a\tau^2 g, v + a\tau, a) + O(\tau^2) \quad (60)$$

for all $(x, v, a) \in X \times \mathbb{R}^n \times \mathbb{R}^n$ and all $g \in \mathbb{R}$. If L and F are those of a harmonic oscillator, the $O(\tau^2)$ -term reduces to $O(\tau^3)$ iff $g = \frac{3}{5}$.

Proof: Verification of this statement is straightforward in principle since both sides can be expanded in powers of τ by Taylor's formula. The number of terms arising that way is very large so that is hard to avoid omissions. So, what is done here is to transform the relevant terms to a form that then can be treated by a computer program capable of executing replacement rules. The program together with its output is reproduced in the appendix.

In order to deal only with scalar expressions, we contract the free indices by an arbitrary $e \in \mathbb{R}^n$ and introduce as abbreviations:

$$\begin{aligned} \text{LHS} &:= e \cdot (D_\tau S)(t, x, v, a, \tau) \\ \text{RHS} &:= -e \cdot G(t + \tau, x + v\tau + a\tau^2 g, v + a\tau, a) . \end{aligned}$$

What we have to show is

$$\text{DIFF} := \text{LHS} - \text{RHS} = O(\tau^2) . \quad (61)$$

We transform LHS and RHS into sums of products of differential operators which finally act on the functions L and F . We will need the argument transformations

$$[H(h, z)f](t, x, v, a) := f(t + h, x + vh + ah^2 z, v + ah, a) \quad (62)$$

for $z = g$ and $z = \frac{1}{2}$. Due to the particular argument structure these can be easily expressed as sums of products of differential operators. This will be carried out for a slightly more general expression, which makes the point clearer:

$$\begin{aligned} [T(\Delta t, \Delta x, \Delta v)f](t, x, v, a) &:= \\ f(t + \Delta t(x, v, a), x + \Delta x(v, a), v + \Delta v(a), a) . \end{aligned} \quad (63)$$

We easily see from Taylor's formula

$$\begin{aligned} T(\Delta t, \Delta x, \Delta v) &= e^{\Delta t \partial_t} e^{\Delta x \cdot \nabla_x} e^{\Delta v \cdot \nabla_v} \\ &=_{\text{up to substitution}} \sum_{j=0}^{\infty} \frac{1}{j!} (\Delta t \partial_t + \Delta x \cdot \nabla_x + \Delta v \cdot \nabla_v)^j, \end{aligned} \quad (64)$$

where the substitution is as follows: any product of two terms from the list $(\partial_t, \Delta x \cdot \nabla_x, \Delta v \cdot \nabla_v)$ in which the order of factors does not match the order in the list has to be replaced by the corresponding product in reversed order. By definition of H we have

$$\text{RHS} = -[H(\tau, g) e \cdot G](t, x, v, a) \quad (65)$$

which established the desired form for this term.

In order to transform LHS we define

$$K(t, x, v, a, h, \tau) := L(t, x, v) + a \cdot F(t, x, v) \frac{h}{2} (h - 2\tau) \quad (66)$$

and get

$$\begin{aligned} S(t, x, v, a, \tau) &= \int_0^{2\tau} [H(h, \frac{1}{2}) K](t, x, v, a, h, \tau) dh \\ &= [H(h, \frac{1}{2}) K](t, x, v, a, h, \tau)_{\text{collect powers of } h, h^k \rightarrow \frac{(2\tau)^{k+1}}{k+1}}. \end{aligned} \quad (67)$$

LHS is obtained from S by multiplication with the differential operator $e \cdot D_\tau$ from (37):

$$\text{LHS} = \frac{3}{2\tau^3} [(\tau e \cdot \nabla_v - e \cdot \nabla_a) S](t, x, v, a, \tau). \quad (68)$$

This establishes the desired form also for this term. In order to simplify the terms RHS and LHS and to render them comparable one has to bring the differential operators into a fixed order. This works in our case since the commutation relations between all differential operators appearing in these expressions can easily be worked out to result in a closed Lie algebra. Then, the products of differential operators can be expanded according to a Birkhoff-Witt basis by repeated application of the commutation relations. This defines a normal form for each sum of products of differential operators.

Although achieving this normal form involves thousands of substitutions it is easily formulated and very efficiently carried out by a computer algebra program such as *FORM* by J.Vermaseren. Fully commented *FORM* code and computed expressions for S , LHS and DIFF are reproduced in the appendix. Particularly, the Lie algebra mentioned above is given explicitly there under the heading COMMUTATION RELATIONS (iii).

Finally we look for circumstances that let the second order terms in DIFF vanish. Let us make first an assumption that is a bit more general than having an harmonic oscillator: $L = T(v) - V(x)$ and $F = 0$. Then the following terms remain in DIFF:

$$\begin{aligned} & \frac{1}{10} v \cdot \nabla_x^2 e \cdot \nabla_x L - \frac{1}{10} a \cdot \nabla_v^3 e \cdot \nabla_v L \\ & \frac{3}{5} a \cdot \nabla_x e \cdot \nabla_x L - g a \cdot \nabla_x e \cdot \nabla_x L . \end{aligned}$$

The first line vanishes if kinetic energy and potential energy are quadratic forms and the last line vanishes just for $g = \frac{3}{5}$. **QED**

The *FORM* program creates the following explicit expressions for S and $D_\tau S$:⁹

$$\begin{aligned} S(t, x, v, a, \tau) &= [2\tau L + 2\tau^2(\partial_t + v \cdot \nabla_x + a \cdot \nabla_v) L \\ &+ \frac{2}{3}\tau^3(-a \cdot F + 2\partial_t^2 L + 4\partial_t v \cdot \nabla_x L + 4\partial_t a \cdot \nabla_v L + 2v \cdot \nabla_x^2 L \\ &+ 4v \cdot \nabla_x a \cdot \nabla_v L + 2a \cdot \nabla_x L + 2a \cdot \nabla_v^2 L) \\ &+ \frac{2}{3}\tau^4(-\partial_t a \cdot F + \partial_t^3 L + 3\partial_t^2 v \cdot \nabla_x L + 3\partial_t^2 a \cdot \nabla_v L \\ &+ 3\partial_t v \cdot \nabla_x^2 L + 6\partial_t v \cdot \nabla_x a \cdot \nabla_v L + 3\partial_t a \cdot \nabla_x L + 3\partial_t a \cdot \nabla_v^2 \\ &- v \cdot \nabla_x a \cdot F + v \cdot \nabla_x^3 L + 3v \cdot \nabla_x^2 a \cdot \nabla_v L + 3v \cdot \nabla_x a \cdot \nabla_x L \\ &+ 3v \cdot \nabla_x a \cdot \nabla_v^2 L + 3a \cdot \nabla_x a \cdot \nabla_v L - a \cdot \nabla_v a \cdot F + a \cdot \nabla_v^3 L)](t, x, v) \\ &+ O(\tau^5) . \end{aligned} \tag{69}$$

$$\begin{aligned} (D_\tau S)(t, x, v, a, \tau) &= [F - \partial_t \nabla_v L - v \cdot \nabla_x \nabla_v L + \nabla_x L - a \cdot \nabla_v \nabla_v L \\ &+ \tau(\partial_t F - \partial_t^2 \nabla_v L - 2\partial_t v \cdot \nabla_x \nabla_v L + \partial_t \nabla_x L - 2\partial_t a \cdot \nabla_v \nabla_v L \\ &+ v \cdot \nabla_x F - v \cdot \nabla_x^2 \nabla_v L + v \cdot \nabla_x \nabla_x L - 2v \cdot \nabla_x a \cdot \nabla_v \nabla_v L - a \cdot \nabla_x \nabla_v L \\ &+ \nabla_x a \cdot \nabla_v L + a \cdot \nabla_v F - a \cdot \nabla_v^2 \nabla_v L)](t, x, v) \\ &+ O(\tau^2) . \end{aligned} \tag{70}$$

A simple consequence of (60) is

$$(D_\tau S)(t, x, v, a, \tau) = -G(t, x, v, a) + O(\tau) . \tag{71}$$

This can also be seen directly by comparing (70) and (11). Thus the general Lagrange equations of mechanics, given by $G = 0$ follow from the small step action principle (54).

⁹ Conversion from *FORM* output into conventional notation has been done manually, although automation would be desirable. The printed part of the program output gives also the next order terms that are not needed for the present $O(\tau^2)$ -form of the theorem. The terms of much higher order can be created by increasing the number p in the first program line.

Aiming at solving equation (54) for a , one could replace $D_\tau S$ by the approximation provided by (70) and find a by an iterative scheme. This is, however, neither elegant nor efficient. Here, the theorem shows its value: The ‘ $O(\tau^2)$ -equivalent’ equation

$$\boxed{G(t + \tau, x + v\tau + a\tau^2 g, v + a\tau, a) = 0} \quad (72)$$

can much more easily be solved for a as will be shown in Section 6. This is our third and last form of a small step equation of motion. An additive $O(\tau^2)$ -term to either side of equation (72) would cause an $O(\tau^3)$ -contribution to the single step state change and an $O(\tau^2)$ -contribution to the state change over a fixed span T of time. So it would not change the order of a second-order integration method as the one to be defined in the next section.

Notice that Euler’s rule just uses equation $G(t, x, v, a) = 0$ instead of (72). *Euler’s rule is the only rational recipe if no information is available on the time span over which a will be used to predict the trajectory. If this time span 2τ is known, we can do much better by using (72), particularly with the simplifying choice $g = 0$.* Section 6 puts this into an algorithm. Section 7 will give an example where (72) works better than (54).

6 Time Stepping Algorithm (Integrator)

Given (t, x, v) as initial state and a value a for the acceleration, the final final state is

$$\begin{aligned} \underline{t} &:= t + 2\tau \\ \underline{v} &:= v + a(2\tau) \\ \underline{x} &:= x + v(2\tau) + a\frac{(2\tau)^2}{2} \end{aligned} \quad (73)$$

or, nicely re-using \underline{v} ,

$$\begin{aligned} \underline{t} &:= t + 2\tau \\ \underline{v} &:= v + a(2\tau) \\ \underline{x} &:= x + (v + \underline{v})\tau . \end{aligned} \quad (74)$$

or, using suitable intermediate quantities ,

$$\begin{aligned} t' &:= t + \tau \\ v' &:= v + a\tau \\ x' &:= x + v\tau \\ \underline{t} &:= t' + \tau \\ \underline{v} &:= v' + a\tau \\ \underline{x} &:= x' + \underline{v}\tau . \end{aligned} \quad (75)$$

So, the remaining task of the time stepping algorithm is to determine the acceleration a . As explained in Section 5, the natural thing to do is to solve (72) for a . By definition (15) of A , we have

$$G(t, x, v, A(t, x, v)) = 0$$

for all values of the arguments. In particular

$$G(t + \tau, x + v\tau + a\tau^2g, v + a\tau, A(t + \tau, x + v\tau + a\tau^2g, v + a\tau)) = 0,$$

so that (72) implies

$$a = A(t + \tau, x + v\tau + a\tau^2g, v + a\tau). \quad (76)$$

For $g = 0$ we have the following particularities:

1. Equation (76) can be written as

$$a = A(t', x', v'). \quad (77)$$

2. If A depends on v trivially, we get a simply as

$$a = A(t + \tau, x + v\tau, v). \quad (78)$$

3. If A depends linearly on v : $A(t, x, v) = A_0(t, x) + A_1(t, x)v$, we get

$$a = (1 - \tau A_1(t + \tau, x + v\tau))^{-1} A(t + \tau, x + v\tau, v), \quad (79)$$

where the inversion means matrix inversion (see commentary to equation (15)) for $n > 1$. Else, the equation is implicit in a harmless manner and can be solved by one or two steps of the following remarkable oscillation damping iteration scheme, which also works well for the general case $0 \leq g \leq 1$ to which we now return:

$$\begin{aligned} a_0 &:= A(t + \tau, x + v\tau, v) \\ a_* &:= A(t + \tau, x + v\tau + a_n\tau^2g, v + a_n\tau) \\ a_{**} &:= A(t + \tau, x + v\tau + a_*\tau^2g, v + a_*\tau) \\ a_{n+1} &:= \frac{1}{2}(a_* + a_{**}). \end{aligned} \quad (80)$$

This iteration solution may be even more economic than (79). In the example of Section 7 the simple iteration scheme (i.e. $a_{n+1} := a_*$) is nearly as accurate as the full version. In more complicated situations, it is hard to find reasons for doing without the extra stability provided by (80). Doing the iteration zero times (i.e. $a := a_0$) also defines an integrator which, however, is only of first order if A depends non-trivially on v . This method is used in [9] for computational efficiency.

It is first order only during collisions and second order else. In the example of the next section this simplified method is more accurate than the Runge-Kutta second order method.

The intermediate quantity (t', x', v') from (75) is not considered a point on a discrete trajectory. It is simply an auxiliary quantity which can be economically computed from the initial conditions and which is designed to yield a good proposal for the overall acceleration during the time span 2τ . In the context of numerical methods for the initial value problem of ordinary differential equations it is said to be an *intermediate stage* of a one-step method. From a geometric point of view it is a very natural one. Note that the computed parabolic trajectory lies in a plane determined by the point x and by the vectors v, a . When parameterizing this curve by $(\text{time} - t)/2\tau$ it is just the *Bézier curve* generated from the *control points* x, x' , and \underline{x} (e.g. [2]).

Let us see how the algorithm fits into the spectrum of ODE-methods. The definitions

$$\begin{aligned} p &:= (t, x, v), & p' &:= (t', x', v'), & \underline{p} &:= (\underline{t}, \underline{x}, \underline{v}), \\ f(p) &:= f(t, x, v) := (1, v, A(t, x, v)) \end{aligned} \quad (81)$$

make equation (16) an autonomous, first-order ODE

$$\dot{p} = f(p). \quad (82)$$

Equations (75),(81) imply

$$\underline{p} = p + 2\tau f(p') \quad (83)$$

which is known under the names *explicit midpoint rule* or *Leap-Frog method* (see [12], p. 224, (3.3.11)) when considered as definition of a two-step method (i.e. assuming p and p' from previous computation, we find the next state \underline{p} by (83)). This is, however, not our present situation. Instead we need a further equation that allows eliminating p' so that we can compute \underline{p} from p alone. This equation is in the case $g = 0$ given by

$$t' = \frac{1}{2}(t + \underline{t}), \quad v' = \frac{1}{2}(v + \underline{v}). \quad (84)$$

This is a special condition that cannot be expressed in terms of p and f , rather it makes use of the special situation that we have an ODE of second order. The most closely related equation that can be formulated in the general theory is

$$p' = p + \tau f(p') \quad (85)$$

which, together with (83), defines the *implicit midpoint rule*. This rule is implicit also for the case that equation (78) is valid. It is just what we get for $g = 1$. The $g = 0$ method is not new to the ODE-literature, see [8] for some identifications, but it has, as it appears, not yet been recognized as a particularly natural and efficient method. Actually, all such identifications deal with the case (78), so that the general case has an even more uncertain literature status. I'll refer to the $g = 0$ method as the *direct midpoint method*.

Informal remark on quantum and field dynamics

It is interesting to observe that (78) can easily be applied to quantum dynamics and to hyperbolic partial differential equations. Let us spell out here the quantum version for a time-dependent Hamiltonian H_t :

$$\begin{aligned} a &:= -\frac{1}{\hbar^2} H_{t+\tau}^2(\psi_t + \tau\dot{\psi}_t) \\ \psi_{t+2\tau} &:= \psi_t + 2\tau\dot{\psi}_t + 2\tau^2 a \\ \dot{\psi}_{t+2\tau} &:= \dot{\psi}_t + 2\tau a, \end{aligned} \tag{86}$$

where the initialization of the velocities is done by $\dot{\psi}_0 := \frac{1}{i\hbar} H_0 \psi_0$. With (86) it is straightforward to set up a one-dimensional computer model of a single quantum particle under the influence of arbitrary potentials. In all the many experiments done in this way, I found no exception to the following remarkable property: the norm and the energy expectation value (since the potentials did not depend on time) of the wave function deviate from strict constancy during phases of very violent motion (where the spatial and temporal variations are clearly too large for the discretization length and time of the model) but come back to the original values with nearly the accuracy of the computer's number representation in all phases of calmer motion. No trend for changing the norm or the energy in the long run was observed. Notice that this is more than unitarity and energy conservation. A method enjoying the property of unitarity would conserve the disturbed value of the norm and would not restore it.

I experienced comparable stability properties whenever I applied the method characterized by equations (74) and (78) to physical systems, ranging from point particles, to rigid bodies, waves, and quantum particles. So, there seems to exist *the* method to start with, when the dynamics of physical systems is to be modeled on a computer. The implementation is nearly as simple as that of the Euler rule and whenever the discretization is chosen with a minimum of insight one will see the system evolve without the explosive auto-acceleration effects that the Euler rule shows after a few integration steps. The simplicity of the algorithm guarantees fast execution. Only if high accuracy is to be achieved with a small number of steps and in systems of sufficiently smooth forces, more specialized methods may offer substantial advantages.

I imagine that Leonhard Euler would have switched soon to that symmetrical mode of updating the acceleration in the center of the time step if he would have been interested in computing a finite piece of a trajectory. For building the trajectory in the mode of a thought experiment, the symmetrical update offers no advantage since the number of acceptable time steps is then unlimited.

7 Application to the Damped Harmonic Oscillator

For conservative systems the direct midpoint method was studied in some detail in [8]. So, let us consider here a non-conservative system, where both L and F are at work. The present choice is the one-dimensional damped harmonic oscillator since it allows all methods under consideration being carried out explicitly and being compared with the exact solution. With a strong influence of friction is not completely trivial as a computational problem. It is not intended to make serious comparison with other numerical methods. As we will see, even with this restriction one can observe a lot of interesting properties that invite further investigation.

Lagrangian and (additional) force of the damped harmonic oscillator can be written as

$$L(t, x, v) = \frac{m}{2}v^2 - \frac{k}{2}x^2, \quad F(t, x, v) = -bv, \quad (87)$$

from which we get (see (11),(15))

$$G(t, x, v, a) = ma + bv + kx, \quad A(t, x, v) = -\frac{1}{m}(kx + bv). \quad (88)$$

This is an example of the complicating situation that (78) is not satisfied. However, it is not the worst case, since obviously A depends linearly on v , so that (79) directly gives

$$a = -\frac{bv + k(x + v\tau)}{m + \tau b}. \quad (89)$$

In this case one easily calculates the action integrals (39), (46), and the force term (41) exactly. One finds that the Euler-Lagrange equation for parabolic path (44) and the equation of the midpoint variation method (54) can be solved exactly and have the same solution, namely

$$a = -\frac{bv + k(x + v\tau)}{m + \tau(b + \frac{3}{5}k\tau)}. \quad (90)$$

Explicit expressions for the exact trajectory result from the solution of the elementary initial value problem for linear differential equation

$$m\ddot{x} + b\dot{x} + kx = 0. \quad (91)$$

For a considerably damped oscillator ($b > 0$) the amplitude quickly decays to zero and it could look difficult to quantify the differences between methods. There is, however, a method of quantifying the error of dynamical approximation methods which is independent of the order of magnitude of the dynamical variables. This method is a re-interpretation of what is known in quantum mechanics as the *interaction picture* and as the method of intermediary orbits in celestial mechanics. Characterizing the dynamics by its time evolution map Φ we compare a state $s(0)$ at time $t = 0$ with the state

$$s_{\text{int}}(t) := \Phi_{\text{exact}}(-t)(\Phi_{\text{approximate}}(t)s(0)) \quad (92)$$

in the course of time t . The exact method for evolving the state back to $t = 0$ is in our case given by a simple explicit formula. For testing numerical methods for which the exact back-evolution method is not given by explicit expressions it can always be simulated by numerical methods that get more computing resources allocated than the method under evaluation. If one likes to stay within the usual context of the interaction picture one has to consider the approximate dynamics as resulting from the exact one by an interaction process. (See e.g. [12], p. 502.) The dynamics caused by this interaction process alone is the one given by the

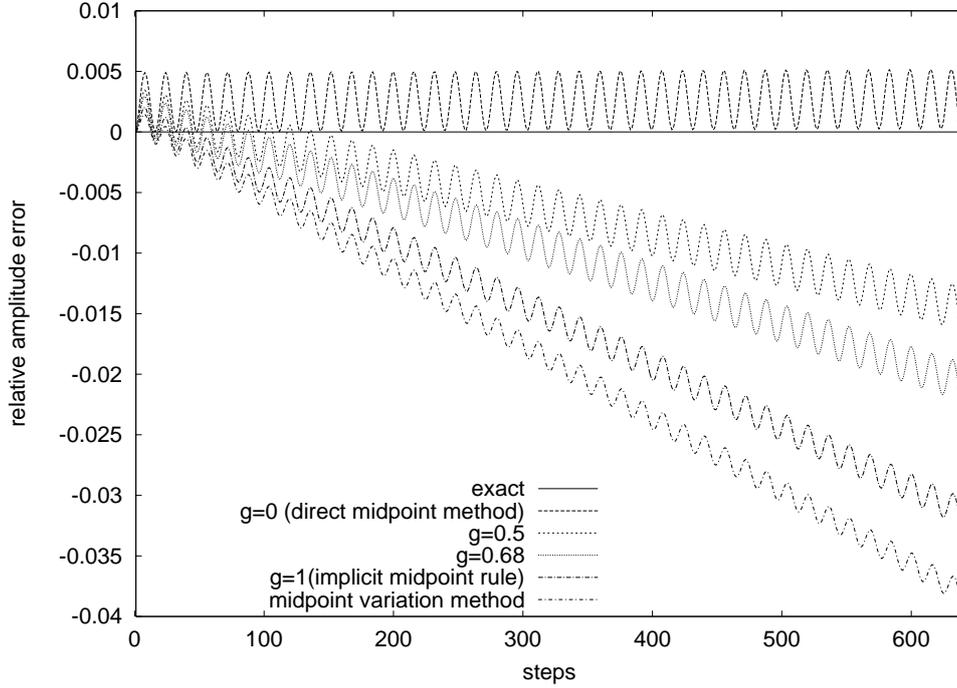


Figure 2: relative amplitude error over 20 oscillations

interaction picture. The full power of the method becomes apparent if one shows not only the trajectory of a single state as is done here with Figures 2 and 3, but if one looks at a cloud of states simultaneously in a computed animation (analogous to [10], Figure 6.1).

Let us look at the present numerical example. We consider the somewhat artificial situation $b < 0$, in which there is no friction but a driving force which increases with speed. Then the amplitude of the exact solution grows exponentially. We consider a rather large value of $|b|$ that doubles the amplitude every two oscillation periods. The number of steps per oscillation period is chosen as 32 (as in the oscillator example in [8]). Let ρ and ω be the real constants for which the general exact solution of (91) is $t \mapsto e^{-\rho t} \cos(\omega t + \varphi)$. Then a convenient descriptor for the state $s(t) = (x(t), v(t))$ is the complex amplitude $s(t)_c := x(t) - i[v(t) + \rho x(t)]/\omega$ of absolute value $s(t)_r := |s(t)_c|$ and phase $s(t)_\varphi := \arg(s(t)_c)$. What we represent graphically are the quantities

$$[s_{\text{int}}(t)_r - s(0)_r]/s(0)_r \quad \text{and} \quad s_{\text{int}}(t)_\varphi - s(0)_\varphi$$

which would be constantly zero for an exact integration method and that are therefore called *relative amplitude error* and *phase error*. The two figures show these quantities for 20 oscillation periods during which the amplitude grows by a factor of 1024. In Figure 2 the direct midpoint method stands out in giving a very low amplitude error without any visible trend. Larger values of $|b|$ show that this is not

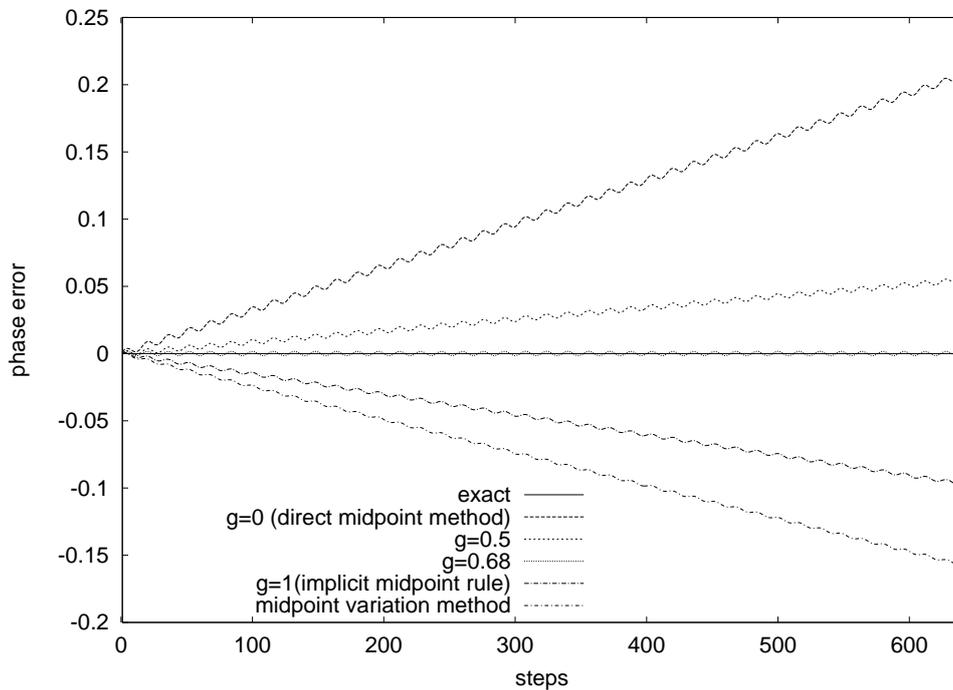


Figure 3: phase error over 20 oscillations

a law. In all cases that I tested, the direct midpoint method was by far the method with the least relative amplitude error. However, in Figure 3, the direct midpoint method shows the largest deviation of the frequency (a linear trend of the phase translates into a shift of the frequency; looking at discretization as a perturbation of the continuous motion we should not be surprised to find renormalization of the parameters of the original equation). That just $g = 0.68$ makes the phase error nearly vanish is not a magic property of just this number. For different values of b , different values for g are needed to achieve the same effect, whereas the property of $g = 0$ to minimize the amplitude error seems to occur uniformly for all values of b .

In the application field of granular media there is a decisive difference in the relevance of the phase error and the amplitude error. If two irregularly shaped particles undergo an inelastic collision one has to cover the phase of overlap typically only by a few time steps of the integrator. Amplitude errors translate into errors in the resulting energy dissipation (even the sign of it may come out wrongly) whereas phase errors translate into deviations from the correct duration of the process which in a soft particle idealization is deliberately accepted.

Given that the midpoint variation method is a direct translation of the fundamental action principle into the parabolic path framework, I would not have been surprised if this method would have turned out to give overall the most realistic trajectories.

Instead, it gives the second worst phase error and the worst amplitude error among the methods under consideration. This renders the expected overall superiority of the method improbable. It should be noted, however, that the Runge Kutta method of second order shows errors approximately 10 times larger so that also the midpoint variation method belongs still to the ‘methods that work better than expected’. Actually it is advantageous that the computationally much simpler direct midpoint method works better, but an understanding for this situation seems still to be missing.

Acknowledgments

I’m grateful to J.Vermaseren for having made his great program FORM publicly available, and to my colleagues Thomas Dera, Helmut Domes, Tom Plutchak, and Eric Stelter for many suggestions on algorithms, computation, and interpretation of results. Further, I acknowledge that my management accepted the publication of work which originated from an industrial project on granular media made of irregularly shaped, magnetized, and charged particles, [9]. I’m grateful to John Zollweg of Cornell University who gave invaluable help and advice during this project which ran thousands of particles over days on dozens of processors in parallel at the Cornell Theory Center. The present work incorporates suggestions of a reviewer of the *Journal of Nonlinear Science*.

References

- [1] The idea to use hysteresis of normal forces for modeling friction was told to me by Dr. Bernhard Eberhardt at the University of Tübingen in 1998.
- [2] G Farin: Shape, p. 465, equations (2),(3) in B. Engquist, W. Schmid (editors): Mathematics Unlimited—2001 and Beyond, Springer, 2001
- [3] Richard P. Feynman, Robert B. Leighton, Matthew Sands: The Feynman Lectures On Physics, Addison Wesley 1963, Volume 1, Paragraph 9-6
- [4] Herbert Goldstein: Klassische Mechanik, Akademische Verlagsgesellschaft, Frankfurt 1963, Chapter II 2–4, equations (2.17), (2.18)
- [5] The basic idea was inherent in Erhard Heinz’ lectures on calculus at the University of Munich 1963. It is worked out in Ulrich Mutze: Smooth interpolation through many points via partition of unity. Kodak Technical Report 278844A 1992
- [6] C. Kane, J.E. Marsden, M. Ortiz and M. West (2000): Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems, Internat. J. Numer. Math. Eng. 49, 1295–1325

- [7] J.E. Marsden and M. West: Discrete mechanics and variational integrators, Acta Numerica (2001), pp. 357–514, Cambridge University Press, 2001
- [8] U. Mutze: Predicting Classical Motion Directly from the Action Principle II. Mathematical Physics Preprint Archive 99–271
http://www.ma.utexas.edu/mp_arc/index-99.html, July 16
- [9] U. Mutze, E. Stelter, T. Dera: Simulation of Electrophotographic Development, Final Program and Proceedings of IS& T's NIP19: International Conference on Digital Printing Technologies September 28 - October 3, 2003 (IS& T: The Society for Imaging Science and Technology, Springfield VA, 2003) p.57.
- [10] J.M. Sanz-Serna and M.P. Calvo: Numerical Hamiltonian Problems, Chapman & Hall, 1994
- [11] Arnold Sommerfeld: Vorlesungen über Theoretische Physik, Mechanik, Verlag Harri Deutsch, 1977
- [12] A.M. Stuart and A.R. Humphries: Dynamical Systems and Numerical Analysis, Cambridge University Press, 1996
- [13] Paul Taylor: Practical Foundations of Mathematics, Cambridge University Press, 1999
- [14] D.E.Wolf: Modelling and Computer Simulation of Granular Media; in K.H. Hoffmann, M. Schreiber (editors) Computational Physics, Springer 1996
- [15] Stephen Wolfram: A new kind of science, Stephen Wolfram, LLC 2002

8 Appendix

```

FORM by J.Vermaseren,version 3.0(Jan 28 2001) Run at: Tue Nov 12 21:33:15 2002
*****
* Ulrich Mutze 2002-11-12
* Symbolic computation using the program FORM (by J.A.M. Vermaseren)
* for proving the theorem in
* 'A Simple Variational Integrator for General
* Holonomic Mechanical Systems'
* Makes no use of Form's vector indexing.
* Thus the number n of generalized coordinates needs not to be specified.
* Notice that all lines starting with '*' are ignored by FORM
*****
#define p "4"
* Highest order to which the series of the exponential function
* will be computed.
* We get all contributions to S of order p+1 (in tau),
* all contributions to LHS of order (p+1)-3,
* and all contributions to RHS of order p.
* Since we need all linear terms in LHS-RHS, we need p>=3.

```

* Even for p=6 the program needs only few seconds to do the work.

Symbols tau,g,h,j,h;

Functions L,aF,eF;

* Lagrangean and Force, aF:=a.F
* In FORM 'Function' declares simply symbols for which the commutative
* law of multiplication is not assumed. This has nothing to do with
* function arguments and evaluation.
* Although not all of these non-commutative quantities are needed from
* the beginning, we list them all here since output formatting depends
* on the order of introduction.

Functions Dt, vDx, aDx, eDx, aDv, eDv, eDa;

* List of differential operators. The order in the list as written here
* is derived from the order t,x,v,a,e of name components.
* When it comes to ordering products, always this order (i) is
* the one to be used.
* The following interpretation of these symbols is understood
* when comparing present FORM-expressions with those of the main text.
* pDq:= scalar product of a vector p with nabla with respect to a
* function argument p or in LATEX-notation:
* $qDp := p_i \cdot \nabla_{q_i}$ (ii)

Off statistics;

*** Argument shift operators Htau and Hh defined by Taylor's formula ****
* See (62),(63),(64)

Local taut=tau*Dt;
Local taux=tau*vDx+g*tau*tau*aDx;
Local tauv=tau*aDv;
Local Htau=1+sum_(j,1,'p',invfac_(j)*(taut+tau*tau+aDx)^j);
* This is Taylor's formula for arguments shift in the t,x,v-slot
* of a function !
* [Htau f](t,x,v,a) := f(t+tau, x+v*tau+a*tau*tau*g, v+a*tau, a)

* now the same with g-->1/2, tau --> h, I don't know the way to do such
* substitutions on a language level. Here, the editor is sufficient

Local ht=h*Dt;
Local hx=h*vDx+(1/2)*h*h*aDx;
Local hv=h*aDv;
Local Hh=1+sum_(j,1,'p',invfac_(j)*(ht+hx+hv)^j);
* [Hh f](t,x,v,a) := f(t+h, x+v*h+a*h*h/2, v+a*h, a)

* For the following commutation relations see (64).

repeat;
id aDv*aDx=aDx*aDv;
id aDv*vDx=vDx*aDv;
id aDv*Dt=Dt*aDv;
id aDx*vDx=vDx*aDx;
id aDx*Dt=Dt*aDx;
id vDx*Dt=Dt*vDx;
endrepeat;

```

.sort

* .sort finishes a module, so that we may enter into a new
* fixed order cycle of
* 1. Declarations: starting with keywords Symbol(s), Function(s), ...
* 2. Specifications: e.g. statistics Off ...
* 3. Definitions: starting with keywords Local, ...
* 4. Executable Statements: starting with keywords id ...
* 5. Output control: such as Print and Bracket

*** Definition of S ****

Symbols k,itag;

Local K=L+(h/2)*(h-2*tau)*aF;
* integrand of the action integral. See (66).

Local S=itag*Hh*K;
* presently S is the integrand of the action integral. Next statement
* transforms S into the evaluated integral.
* itag stands for integration tag

* so far we made definitions, now comes a computation
*(executable statement)

id itag*h^k?=(2*tau)^(k+1)/(k+1);
* doing the integration over h in expression S.
* See (53). The role of 'itag' is simple:
* it delimits the range of the substitution to the h-terms in
* S instead of doing the replacement also in, say, Hh.

.sort

*** Definition of the expressions in Equation (60) ****

Local Ge=((Dt+vDx+aDv)*eDv-eDx)*L-eF;
* expression of equation (10) with indexes contracted by
* vector e. Introduction of e is the trick that allows to avoid
* vector indexing in the computation.

Local RHS=-Htau*Ge;
* this is the right-hand side of the equation of the theorem. See (65).

Local Dtau=(3/(2*tau^3))*(tau*eDv-eDa);
* Differential operator for the midpoint derivative in e-direction.
* See equation (37).

Local LHS=Dtau*S;

Local DIFF=LHS-RHS;
* Theorem 1 says that DIFF is second order in tau. Notice that terms
* of order 0 in tau are printed last, instead of first. The term DIFF
* should contain neither terms of order 0 nor of order 1 in tau.

```

```

* Now we have to bring the non-commuting differential operators into
* a fixed order. This brings all expressions in a normal form so that
* one can see whether two expressions are equal, especially
* whether they are equal to 0.

repeat;
id eDa*aF=eF;
id eDa*L=0;
* destruction operator property. None of the functions on which the
* differential operators act, depends on a. In order to make use of this
* property, one has to move eDa to the utmost right position.

* COMMUTATION RELATIONS (iii)
* The following 6+5+4+3+2+1 commutation relations define
* a universal enveloping algebra. And repeatedly applying these
* relations to a sum of products (SOP) of Dt, vDx, aDx, eDx, aDv, eDv, eDa
* expands this SOP into a Birkhoff-Witt basis of this algebra.
* To be sure, the order of basis elements on which this Birkhoff-
* Witt basis relies is Dt, vDx, aDx, eDx, aDv, eDv, eDa (see (i)) .

* The following commutation relations follow from the interpretation
* of the pDq as differential operators (see (ii)).

* Writing < for the linear order in the list, e.g. aDx < eDa, the following
* long list of id-statements simply says:
* For all pDq, rDs such that pDq > rDs
* replace pDq*rDs by rDs*pDq+c,
* where c=pDs for q=r and c=0 else.

id eDa*eDv=eDv*eDa;
id eDa*aDv=aDv*eDa+eDv;
id eDa*eDx=eDx*eDa;
id eDa*aDx=aDx*eDa+eDx;
id eDa*vDx=vDx*eDa;
id eDa*Dt=Dt*eDa;

id eDv*aDv=aDv*eDv;
id eDv*eDx=eDx*eDv;
id eDv*aDx=aDx*eDv;
id eDv*vDx=vDx*eDv+eDx;
id eDv*Dt=Dt*eDv;

id aDv*eDx=eDx*aDv;
id aDv*aDx=aDx*aDv;
id aDv*vDx=vDx*aDv+aDx;
id aDv*Dt=Dt*aDv;

id eDx*aDx=aDx*eDx;
id eDx*vDx=vDx*eDx;
id eDx*Dt=Dt*eDx;

id aDx*vDx=vDx*aDx;
id aDx*Dt=Dt*aDx;

```

```

id vDx*Dt=Dt*vDx;

endrepeat;
.sort

*** Programming Output ***

Bracket g;
Bracket tau;
* asks for expanding according to powers of tau, just what we want

Print S;
Print LHS;
Print DIFF;
.end;
***** Results *****

S =
+ tau * ( 2*L )
+ tau^2 * ( 2*Dt*L + 2*vDx*L + 2*aDv*L )
+ tau^3 * ( - 2/3*aF + 4/3*Dt*Dt*L + 8/3*Dt*vDx*L + 8/3*Dt*aDv*L + 4/3
*vDx*vDx*L + 8/3*vDx*aDv*L + 4/3*aDx*L + 4/3*aDv*aDv*L )
+ tau^4 * ( - 2/3*Dt*aF + 2/3*Dt*Dt*Dt*L + 2*Dt*Dt*vDx*L + 2*Dt*Dt*aDv
*L + 2*Dt*vDx*vDx*L + 4*Dt*vDx*aDv*L + 2*Dt*aDx*L + 2*Dt*aDv*aDv*L -
2/3*vDx*aF + 2/3*vDx*vDx*vDx*L + 2*vDx*vDx*aDv*L + 2*vDx*aDx*L + 2*
vDx*aDv*aDv*L + 2*aDx*aDv*L - 2/3*aDv*aF + 2/3*aDv*aDv*aDv*L )
+ tau^5 * ( - 2/5*Dt*Dt*aF + 4/15*Dt*Dt*Dt*Dt*L + 16/15*Dt*Dt*Dt*vDx*L
+ 16/15*Dt*Dt*Dt*aDv*L + 8/5*Dt*Dt*vDx*vDx*L + 16/5*Dt*Dt*vDx*aDv*L
+ 8/5*Dt*Dt*aDx*L + 8/5*Dt*Dt*aDv*aDv*L - 4/5*Dt*vDx*aF + 16/15*Dt*
vDx*vDx*vDx*L + 16/5*Dt*vDx*vDx*aDv*L + 16/5*Dt*vDx*aDx*L + 16/5*Dt*
vDx*aDv*aDv*L + 16/5*Dt*aDx*aDv*L - 4/5*Dt*aDv*aF + 16/15*Dt*aDv*aDv*
aDv*L - 2/5*vDx*vDx*aF + 4/15*vDx*vDx*vDx*vDx*L + 16/15*vDx*vDx*vDx*
aDv*L + 8/5*vDx*vDx*aDx*L + 8/5*vDx*vDx*aDv*aDv*L + 16/5*vDx*aDx*aDv*
L - 4/5*vDx*aDv*aF + 16/15*vDx*aDv*aDv*aDv*L - 2/5*aDx*aF + 4/5*aDx*
aDx*L + 8/5*aDx*aDv*aDv*L - 2/5*aDv*aDv*aF + 4/15*aDv*aDv*aDv*aDv*L )

LHS =
+ eF - Dt*eDv*L - vDx*eDv*L + eDx*L - aDv*eDv*L
(term manually moved from the last position to the first one)
+ tau * ( Dt*eF - Dt*Dt*eDv*L - 2*Dt*vDx*eDv*L + Dt*eDx*L - 2*Dt*aDv*
eDv*L + vDx*eF - vDx*vDx*eDv*L + vDx*eDx*L - 2*vDx*aDv*eDv*L - aDx*
eDv*L + eDx*aDv*L + aDv*eF - aDv*aDv*eDv*L )
+ tau^2 * ( 3/5*Dt*Dt*eF - 3/5*Dt*Dt*Dt*eDv*L - 9/5*Dt*Dt*vDx*eDv*L + 3/
5*Dt*Dt*eDx*L - 9/5*Dt*Dt*aDv*eDv*L + 6/5*Dt*vDx*eF - 9/5*Dt*vDx*vDx*
eDv*L + 6/5*Dt*vDx*eDx*L - 18/5*Dt*vDx*aDv*eDv*L - 9/5*Dt*aDx*eDv*L
+ 6/5*Dt*eDx*aDv*L + 6/5*Dt*aDv*eF - 9/5*Dt*aDv*aDv*eDv*L + 1/5*Dt*
eDv*aF + 3/5*vDx*vDx*eF - 3/5*vDx*vDx*vDx*eDv*L + 3/5*vDx*vDx*eDx*L

```

$$\begin{aligned}
& - 9/5*vDx*vDx*aDv*eDv*L - 9/5*vDx*aDx*eDv*L + 6/5*vDx*eDx*aDv*L + 6/ \\
& 5*vDx*aDv*eF - 9/5*vDx*aDv*aDv*eDv*L + 1/5*vDx*eDv*aF + 3/5*aDx*eF + \\
& 3/5*aDx*eDx*L - 9/5*aDx*aDv*eDv*L - 2/5*eDx*aF + 3/5*eDx*aDv*aDv*L + \\
& 3/5*aDv*aDv*eF - 3/5*aDv*aDv*aDv*eDv*L + 1/5*aDv*eDv*aF) \\
& + \text{higher order terms which were deleted manually in order to save space}
\end{aligned}$$

DIFF =

$$\begin{aligned}
& + \tau^2 * (1/10*Dt*Dt*eF - 1/10*Dt*Dt*Dt*eDv*L - 3/10*Dt*Dt*vDx*eDv*L \\
& + 1/10*Dt*Dt*eDx*L - 3/10*Dt*Dt*aDv*eDv*L + 1/5*Dt*vDx*eF - 3/10*Dt* \\
& vDx*vDx*eDv*L + 1/5*Dt*vDx*eDx*L - 3/5*Dt*vDx*aDv*eDv*L - 4/5*Dt*aDx* \\
& eDv*L + Dt*aDx*eDv*L*g + 1/5*Dt*eDx*aDv*L + 1/5*Dt*aDv*eF - 3/10*Dt* \\
& aDv*aDv*eDv*L + 1/5*Dt*eDv*aF + 1/10*vDx*vDx*eF - 1/10*vDx*vDx*vDx* \\
& eDv*L + 1/10*vDx*vDx*eDx*L - 3/10*vDx*vDx*aDv*eDv*L - 4/5*vDx*aDx*eDv* \\
& *L + vDx*aDx*eDv*L*g + 1/5*vDx*eDx*aDv*L + 1/5*vDx*aDv*eF - 3/10*vDx* \\
& aDv*aDv*eDv*L + 1/5*vDx*eDv*aF + 3/5*aDx*eF - aDx*eF*g + 3/5*aDx*eDx* \\
& L - aDx*eDx*L*g - 4/5*aDx*aDv*eDv*L + aDx*aDv*eDv*L*g - 2/5*eDx*aF + \\
& 1/10*eDx*aDv*aDv*L + 1/10*aDv*aDv*eF - 1/10*aDv*aDv*aDv*eDv*L + 1/5* \\
& aDv*eDv*aF) \\
& + \text{higher order terms which were deleted manually in order to save space}
\end{aligned}$$