massachusetts institute of technology — artificial intelligence laboratory

The Role of Programming in the Formulation of Ideas

Gerald Jay Sussman and Jack Wisdom

AI Memo 2002-018 November 2002

Abstract

Classical mechanics is deceptively simple. It is surprisingly easy to get the right answer with fallacious reasoning or without real understanding. To address this problem we use computational techniques to communicate a deeper understanding of Classical Mechanics. Computational algorithms are used to express the methods used in the analysis of dynamical phenomena. Expressing the methods in a computer language forces them to be unambiguous and computationally effective. The task of formulating a method as a computer-executable program and debugging that program is a powerful exercise in the learning process. Also, once formalized procedurally, a mathematical idea becomes a tool that can be used directly to compute results.

> Philosophy is written in that great book which ever lies before our eyes–I mean the Universe–but we cannot understand it if we do not learn the language and grasp the symbols in which it is written. This book is written in the mathematical language, and the symbols are triangles, circles, and other geometrical figures without whose help it is impossible to comprehend a single word of it, without which one wanders in vain through a dark labyrinth.
>
> *Galileo Galilei* [1]

Learning physics is hard. Part of the problem is that physics is naturally expressed in mathematical language. A student must simultaneously learn the mathematical language and the content that is expressed in that language. This is like trying to read *Les Misérables*, while struggling with French grammar.

One common proposal is to minimize the mathematical overhead, by replacing it with imprecise qualitative arguments. This is attractive, but it is inadequate to convey the actual power of the science. It is essential that aspects of the world be described with precise symbols and that quantitative consequences of those descriptions can be compared with experiment.

It is necessary to present the science in the language of mathematics. Unfortunately, when we teach science we use the language of mathematics in the same way that we use our natural language. We depend upon a vast amount of shared knowledge and culture, and we only sketch an idea using mathematical idioms. We are insufficiently precise to convey an idea to a person who does not share our culture. Our problem is that since we share the culture we find it difficult to notice that what we say is too imprecise to be clearly understood by a student new to the subject.

One way to become aware of the precision required to unambiguously communicate a mathematical idea is to program it for a computer. Rather than using canned programs purely as an aid to visualization or numerical computation, we use computer programming in a functional style to encourage clear thinking. Programming forces one to be precise and formal, without being excessively rigorous. The computer does not tolerate vague descriptions or incomplete constructions. Thus the act of programming makes one keenly aware of one's errors of reasoning or unsupported conclusions.[1]

[1] The idea of using computer programming to develop skills of clear thinking was originally advocated by Seymour Papert. An extensive discussion of this idea, applied to the education of young children can be found in Papert [3].

A Fuzzy Derivation

In traditional notation the Lagrange equations are written

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}^i} - \frac{\partial L}{\partial q^i} = 0. \tag{1}$$

where superscripts are used to index the coordinates and the velocities. We can restrict our attention here to systems with a single degree of freedom. Consider a particle with position x, velocity $\dot{x}$, and potential energy $V(x)$. A Lagrangian for this system is

$$L = \frac{1}{2}m\dot{x}^2 - V(x). \tag{2}$$

Derivation of the Lagrange equations is simple. First,

$$\frac{\partial L}{\partial \dot{x}} = m\dot{x}, \tag{3}$$

then, with

$$\frac{d}{dt}(m\dot{x}) = m\ddot{x} \tag{4}$$

and

$$\frac{\partial L}{\partial x} = -\frac{\partial V}{\partial x}, \tag{5}$$

the Lagrange equation is

$$m\ddot{x} = -\frac{\partial V}{\partial x}. \tag{6}$$

Mass times acceleration is minus the gradient of the potential energy.

Let's think about what we have just done.

The first thing we did was change the name of the variable from q to x. The Lagrange equations are a pattern for a way of deriving such equations, not an explicit rule. The details have to be filled in for particular Lagrangians. For instance, if there are many degrees of freedom there will be that many variables introduced and a Lagrange equation for each.

Now equation (2) says that for this system, if we know the coordinate x of the particle and its rate of change $\dot{x}$ then we can compute the value of the Lagrangian.

The partial derivatives are computed by holding all other variables fixed while taking the derivative with respect to the variable of interest. The value of such a partial derivative generally depends on the variables that the original function depends upon. So, $\partial L/\partial \dot{x}$ generally depends upon x and $\dot{x}$, just as L does. In this particular case, the partial derivatives are simpler. The partial derivative $\partial L/\partial \dot{x}$ depends only on $\dot{x}$ and the partial derivative $\partial L/\partial x$ depends only on x. In any case, the partials depend upon the same quantities as the original Lagrangian. And especially note that in thinking about these partial derivatives we are thinking of the coordinate x and the velocity $\dot{x}$ as being independent. We hold one fixed while we vary (differentiate with respect to) the other. Fine.

Now we perform the time derivative. We are so used to doing this sort of calculation that we hardly notice, but a bit of magic happens here. We read equation (4) quite naturally: the time derivative of mass times the velocity is the mass times the acceleration. We would have little trouble filling in the details for a more general Lagrangian. The time derivative d/dt acts like any other derivative, with product and chain rule, and acting on a variable adds a dot (transforms generalized coordinates into generalized velocities, and generalized velocities into generalized accelerations). So the time derivative of the position coordinate is the velocity

$$\frac{d}{dt}x = \dot{x}. \tag{7}$$

The position x and the velocity $\dot{x}$ are not after all independent: one is the time derivative of the other. But we assumed that they were independent when we computing the partial derivatives. Whoops! It should not be the case that interpretations change during mathematical derivations. But whether or not we understand this subtlety of how two variables can in one step of a derivation be independent and in the next dependent, we have derived the correct Lagrange equation.

Though such statements (and derivations that depend upon them) seem very strange to students, they are told that if they think about them harder they will understand. So the student must either come to the conclusion that he/she is dumb and just accepts it, or that the derivation is correct, with some appropriate internal rationalization. Students often learn to carry out these manipulations without really understanding what they are doing.

We believe that a major obstacle to the understanding and teaching of physics is the use of variables whose meaning depends upon and changes with context, as well as the sort of impressionistic mathematics that goes along with the use of such variables. Mathematical expressions are often presented which can be understood only after much reflection, and sometimes it can be quite difficult to decide whether a complicated statement is really true or not.

Why did it work?

The traditional use of Leibnitz's notation and Newton's notation is convenient in simple situations, but in more complicated situations it can be a serious handicap to clear reasoning. Much of the way we usually operate with the mathematical expressions is by rote manipulations, rather than by reference to the meanings of the expressions. This is fine for experts discussing among themselves, but it is fatal for teaching.

A mechanical system is described by a Lagrangian function of the system state (time, coordinates, and velocities). A motion of the system is described by a path that gives the coordinate for each moment of time. A path is allowed if and only if it satisfies the Lagrange equations. In traditional notation the Lagrange equation is written

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0. \tag{8}$$

What could this expression possibly mean?

Let's try to write a program that implements Lagrange equations. What are Lagrange equations for? Our program must take a proposed path and give a result that allows one to decide if the path is allowed. This is already a problem, the equation shown above does not have a slot for a path to be tested.

So we have to figure out how to insert the path to be tested. The partial derivatives do not depend on the path; they are derivatives of the Lagrangian function. They are functions with the same arguments as the Lagrangian. But the time derivative d/dt makes sense only for a function of time. Thus we must be intending to substitute the path (a function of time) and its derivative (also a function of time) into the coordinate and velocity arguments of the partial derivative functions.

So probably we meant something like the following (assume that w is a path through the coordinate configuration space, and so $w(t)$ is the configuration at time t):

$$\frac{d}{dt}\left(\left.\frac{\partial L(t,q,\dot{q})}{\partial \dot{q}}\right|_{\substack{q\,=\,w(t)\\ \dot{q}\,=\,\frac{dw(t)}{dt}}}\right)-\left.\frac{\partial L(t,q,\dot{q})}{\partial q}\right|_{\substack{q\,=\,w(t)\\ \dot{q}\,=\,\frac{dw(t)}{dt}}}=0. \tag{9}$$

In this equation we see that the partial derivatives of the Lagrangian function are taken, then the path and its derivative are substituted for the position and velocity arguments of the Lagrangian, resulting in an expression in terms of the time.

This equation is complete. It has meaning independent of the context and there is nothing left to the imagination. The earlier equations require the reader to fill in lots of detail that is implicit in the context. They do not have a clear meaning independent of the context.

By thinking operationally we have reformulated the Lagrange equations into a form that is explicit enough to specify a computation. We could convert it into a program for any symbolic manipulation program because it tells us *how* to manipulate expressions to compute the residuals of Lagrange's equations for a purported solution path.

Functional Abstraction

But this corrected use of Leibnitz notation is ugly. We had to introduce extraneous symbols (q and $\dot{q}$) in order to indicate the argument position specifying the partial derivative. Nothing would change here if we replaced q and $\dot{q}$ by a and b.[2] We can simplify the notation by admitting that the partial derivatives of the Lagrangian are themselves new functions, and by specifying the particular partial derivative by the position of the argument that is varied

$$\frac{d}{dt}((\partial_2 L)(t,w(t),\frac{d}{dt}w(t)))-(\partial_1 L)(t,w(t),\frac{d}{dt}w(t))=0, \tag{10}$$

where $\partial_i L$ is the function which is the partial derivative of the function L with respect to the ith argument.[3]

[2] That the symbols q and $\dot{q}$ can be replaced by other arbitrarily chosen non-conflicting symbols without changing the meaning of the expression tells us that the partial derivative symbol is a logical quantifier, like forall and exists ($\forall$ and $\exists$).

[3] The argument positions of the Lagrangian are indicated by indices starting with zero for the time argument.

Two different notions of derivative appear in this expression. The functions $\partial_2 L$ and $\partial_1 L$, constructed from the Lagrangian L, have the same arguments as L. The derivative d/dt is an expression derivative. It applies to an expression that involves the variable t. It gives the rate of change of the value of the expression as the value of the variable t is varied.

These are both useful interpretations of the idea of a derivative. But functions give us more power. There are many equivalent ways to write expressions that compute the same value. For example $1/(1/r_1 + 1/r_2) = (r_1 r_2)/(r_1 + r_2)$. These expressions compute the same function of the two variables r_1 and r_2. The first expression fails if $r_1 = 0$ but the second one gives the right value of the function. If we abstract the function, say as $\Pi(r_1, r_2)$, we can ignore the details of how it is computed. The ideas become clearer because they do not depend on the detailed shape of the expressions.

So let's get rid of the expression derivative d/dt and replace it with an appropriate functional derivative. If f is a function then we will write Df as the new function which is the derivative of f:[4]

$$(Df)(t) = \left.\frac{d}{dx} f(x)\right|_{x=t} \tag{11}$$

To do this for the Lagrange equation we need to construct a function to take the derivative of.

Given a configuration-space path, there is a standard way to make the

[4]The derivative of a function f is the function Df whose value for a particular argument is something that can be multiplied by an increment Δx in the argument to get a linear approximation to the increment in the value of f:

$$f(x + \Delta x) \approx f(x) + Df(x)\Delta x.$$

For example, let f be the function that cubes its argument ($f(x) = x^3$); then Df is the function that yields three times the square of its argument ($Df(y) = 3y^2$). So $f(5) = 125$ and $Df(5) = 75$. The value of f with argument $x + \Delta x$ is

$$f(x + \Delta x) = (x + \Delta x)^3 = x^3 + 3x^2\Delta x + 3x\Delta x^2 + \Delta x^3$$

and

$$Df(x)\Delta x = 3x^2\Delta x.$$

So $Df(x)$ multiplied by Δx gives us the term in $f(x + \Delta x)$ that is linear in Δx, providing a good approximation to $f(x + \Delta x) - f(x)$ when Δx is small.

state-space path, which we can abstract to a mathematical function Γ:

$$\Gamma[w](t) = (t, w(t), \frac{d}{dt}w(t)). \tag{12}$$

Given this Γ we can write:

$$\frac{d}{dt}((\partial_2 L)(\Gamma[w](t))) - (\partial_1 L)(\Gamma[w](t)) = 0. \tag{13}$$

If we now define composition of functions $(f \circ g)(x) = f(g(x))$, we can reexpress the Lagrange equations entirely in terms of functions:

$$D((\partial_2 L) \circ (\Gamma[w])) - (\partial_1 L) \circ (\Gamma[w]) = 0. \tag{14}$$

The functions $\partial_1 L$ and $\partial_2 L$ are partial derivatives of the function L. Composition with $\Gamma[w]$ evaluates these partials with coordinates and velocites appropriate for the path w, making functions of time. Applying D takes the time derivative. The Lagrange equation states that the difference of the resulting functions of time must be zero. This statement of the Lagrange equation is complete, unambiguous, and functional. It is not encumbered with the particular choices made in expressing the Lagrangian. For example, it doesn't matter if the time is named t or τ, and it has an explicit place for the path to be tested.

This expression is equivalent to a computer program:[5]

```
(define ((Lagrange-equations Lagrangian) q)
  (- (D (compose ((partial 2) Lagrangian) (Gamma q)))
     (compose ((partial 1) Lagrangian) (Gamma q))))
```

In the Lagrange equations procedure the formal parameter `Lagrangian` is a procedure that implements the Lagrangian function. The derivatives of the Lagrangian are also procedures, for example, `((partial 2) Lagrangian)`. The state-space path procedure `(Gamma q)` is constructed from the coordinate path procedure `q` by the procedure `Gamma`:

[5] The programs in this paper are written in Scheme, a dialect of Lisp. The details of the language are not germane to the points being made. What is important is that it is mechanically interpretable, and thus unambiguous. The particular language is chosen because it is easy to write programs that manipulate representations of mathematical functions. An informal description of Scheme and the use of it to represent the mathematical objects of mechanics can be found in [5]. A formal description of Scheme can be obtained in [4]. You can get the software from [6].

```
(define ((Gamma q) t)
  (up t (q t) ((D q) t)))
```

where `up` is a constructor for a data structure that represents a state of the dynamical system (time, coordinates, velocities). Note that we flexibly manipulate representations of mathematical functions in this language.

We started out thinking that the original statement of Lagrange's equations accurately captured the idea. But we really don't know until we try to teach it to a naive student. If the student is sufficiently ignorant, but is willing to ask questions, we are led to clarify the equations in the way that we did. There is no dumber but more insistent student than a computer. A computer will absolutely refuse to accept a partial statement, with missing parameters or a type error. In fact, the original statement of Lagrange's equations contained an obvious type error: the Lagrangian is a function of multiple variables, but the d/dt is applicable only to functions of one variable.

Total Time Derivative

Lagrangians are not in one to one correspondence with physical systems. It is well known that a total time derivative of a function of coordinates and time can be added to a Lagrangian without changing the Lagrange equations. The usual statement is that if L is a Lagrangian for a system then

$$L' = L + \frac{d}{dt}F, \tag{15}$$

is also a Lagrangian for the system.

Now we have found a way of understanding what the d/dt meant in the traditional statement of the Lagrange equations. We inserted paths into the partials of the Lagrangian so that they became functions of time, so d/dt could apply. Here we are making a new Lagrangian L', which must be a function of the state variables t, q, and $\dot{q}$. If we insert paths to make sense of the d/dt then we will have a function of time only. This is not something that can be added to a Lagrangian; it has the wrong arguments. So what does d/dt mean here?

Well, even if we do not understand what it means we can do it. Consider the motion of a particle of mass m with potential energy $V(x)$. Equation (2) gives a Lagrangian for this system:

$$L = \frac{1}{2}m\dot{x}^2 - V(x). \tag{16}$$

Now consider adding the total time derivative of, say, $F = x^2$ to this. The total time derivative is

$$\frac{d}{dt}F = 2x\dot{x}, \tag{17}$$

using the usual rules (stated above). So

$$L' = \frac{1}{2}m\dot{x}^2 - V(x) + 2x\dot{x}. \tag{18}$$

Let's compute the Lagrange equations to check that everything is ok. As usual,

$$\frac{\partial L}{\partial \dot{x}} = m\dot{x} + 2x \tag{19}$$

and

$$\frac{\partial L}{\partial x} = -\frac{\partial V}{\partial x} + 2\dot{x} \tag{20}$$

then

$$\frac{d}{dt}(m\dot{x} + 2x) = m\ddot{x} + 2\dot{x}, \tag{21}$$

and the Lagrange equation is

$$\ddot{x} = -\frac{\partial V}{\partial x}, \tag{22}$$

as before.

So what happened? If we had evaluated the function F along the coordinate path w we would have

$$(w(t))^2. \tag{23}$$

This depends on time, so we can take the time derivative to get

$$\frac{d}{dt}(w(t))^2 = 2w(t)\frac{dw(t)}{dt}. \tag{24}$$

But we need a state function to be able to add it to a Lagrangian, so we need that state function that gives this result when evaluated on the path w. The desired state function is

$$G(t, x, \dot{x}) = \frac{d}{dt}F = 2x\dot{x}. \tag{25}$$

So the total time derivative is computed by inserting paths, differentiating with respect to time, then abstracting back to a state function. Notice the particular path does not matter to the final answer. We see that d/dt means something different here than it did in the Lagrange equations.

To explain more clearly what this total time derivative is we can again think about how we might program a computer to find it. First, we can give a more precise definition of what this total time derivative is. The total time derivative d/dt of a function of state variables F gives a new function G of state variables such that if paths were inserted then one would be the time derivative of the other. Let's introduce a notation $D_t F$ for the total time derivative of the function F. Then

$$D(F \circ \Gamma[w]) = (D_t F) \circ \Gamma[w]. \tag{26}$$

where Γ was introduced earlier. Though this precisely describes what this total time derivative is, this does not give us a recipe to compute it. How do we abstract a function on a path back to a function of state variables? The particular path does not matter. So any path that agrees with the state given will do. Assume F depends on the time and the coordinates, then $D_t F$ will in general depend on the time, the coordinates, and the velocities. One coordinate path that has this state at time t is the polynomial $w(t) = x_0 + v_0(t - t_0)$, where the coordinates and velocities at time t_0 are x_0 and v_0. (If more state information needed to be matched we would add more terms.) Note that this w is a function of x_0 and v_0. So we should write

$$w(t) = W(t_0, x_0, v_0)(t) = x_0 + v_0(t - t_0). \tag{27}$$

Now we can form the total time derivative state function

$$\begin{aligned}(D_t F)(t, x, v) &= D(F \circ \Gamma[W(t, x, v)])(t) \\ &= \partial_0 F(t, x, v) + \partial_1 F(t, x, v) v,\end{aligned} \tag{28}$$

using the chain rule in the second step. This gives an explicit formula for the total time derivative.

We can write corresponding programs. The function $F(t, x, \dot{x}) = x^2$ used above can be defined

```
(define (F state)
  (square (coordinate state)))
```

The derivative of `F` along the literal path `w` is

```
(show-expression
 ((D (compose F (Gamma (literal-function 'w)))) 't))
```

$$2w(t)\,Dw(t)$$

The result of taking the derivative of `F` along the path `w` is an expression involving the path `w` and its derivative `(D w)`. This is not suitable for adding to a Lagrangian. Now we define the procedure for computing the total time derivative

```
(define ((Dt F) state)
  (let ((t0 (time state))
        (x0 (coordinate state))
        (v0 (velocity state)))
    (define (w t)
      (+ x0 (* v0 (- t t0))))
    ((D (compose F (Gamma w))) t0)))
```

The total time derivative of `F` is

```
(show-expression
 ((Dt F) (up 't_0 'x_0 'v_0)))
```

$$2v_0x_0$$

The result depends only on the state variables. This is suitable for adding to a Lagrangian.

Howlers and Fallacies

Traditional presentations of mechanics are full of fallacious derivations of true theorems. Consider the following "proof:"

The Legendre transformation allows us to transport our description of a physical situation from a Lagrangian representation to a Hamiltonian representation. The Hamiltonian function associated with a particular Lagrangian function is given as

$$H(t,q,p) = pv - L(t,q,v), \tag{29}$$

where the generalized velocity v is expressed in terms of the momentum p, where the momentum is defined as a derivative of the Lagrangian:

$$p = \frac{\partial L(t, q, v)}{\partial v}. \tag{30}$$

Now suppose we want to consider how the passive variable q is handled by the Legendre transformation. One might be led to write

$$\frac{\partial H(t, q, p)}{\partial q} = 0 - \frac{\partial L(t, q, v)}{\partial q} \tag{31}$$

because the variable q does not appear in pv and it only appears in the Lagrangian in the second argument. This is usually justified by an argument about "independence" of the variables. Let's examine this more carefully.

In the Legendre transformation generalized velocity v is expressed in terms of the generalized momentum p *and* the generalized coordinate q. Indeed,

$$v = \mathcal{V}(t, q, p), \tag{32}$$

where the function $\mathcal{V}$ is obtained by solving for v in the definition of the generalized momentum above. Thus there are hidden dependencies that are ignored by the handwaving argument. We are encouraged by the Leibnitz derivative notation to concentrate on the algebraic expressions: it is easy to assume that if a variable does not appear explicitly in an expression then the value of the expression does not depend on that variable.

The correct argument is almost as simple. First, we must be explicit about the functional dependencies:

$$H(t, q, p) = pv - L(t, q, v) = p\mathcal{V}(t, q, p) - L(t, q, \mathcal{V}(t, q, p)) \tag{33}$$

Now, when we take the partial derivative we need to use the chain rule, to get

$$\begin{aligned}\partial_1 H(t, q, p) \;\; = \;\; & p\partial_1 \mathcal{V}(t, q, p) \\ & -\partial_1 L(t, q, \mathcal{V}(t, q, p)) \\ & -\partial_2 L(t, q, \mathcal{V}(t, q, p))\partial_1 \mathcal{V}(t, q, p).\end{aligned}$$

We then use the definition of momentum to note that the first and third terms cancel, leaving

$$\partial_1 H(t, q, p) = -\partial_1 L(t, q, \mathcal{V}(t, q, p)), \tag{34}$$

which was the fallaciously derived true theorem arrived at earlier.

Does this matter? Probably not very much, if we are only interested in the answer. But it sure inhibits the progress of thoughtful students who want to understand everything deeply, or when we are working at the frontiers of understanding, where we really must be careful.

Fuzzy Independence

Mechanics is replete with arguments that depend upon whether variables are independent. Many of these arguments, upon close examination, make no sense whatsoever. Yet, we are assured that if we think about it a little harder we will understand. Maybe we do, or maybe we just become familiar.

In the traditional derivation of the Lagrange equations we saw that initially x and $\dot{x}$ are considered independent state variables, and later must become paths, where one is the derivative of the other. If we are careful to distinguish coordinate paths from coordinates these problems do not occur. If q is a coordinate path, the coordinates at time t are $q(t)$. The derivative of the coordinate function gives the velocity function Dq. The velocity for the path q at time t is $Dq(t)$. As long as we are speaking of paths the velocity is the derivative of the coordinate—they are not independent. We may also wish to speak of the initial state of a system in terms of the initial coordinates x_0 and velocities v_0 at time t_0. We may specify these independently, and find the dynamical path q that emanates from this initial data: $x_0 = q(t_0)$ and $v_0 = Dq(t_0)$.

The transition from the Lagrangian formulation to the Hamiltonian formulation of mechanics is particularly tricky. Is momentum independent of the velocity or not? There are many wrong statements made on this issue in standard texts. For example[6] we find

> It may be objected that q and p cannot be varied independently, because the defining equations [$p = \partial L/\partial \dot{q}$] link p with q and $\dot{q}$. One could not then have a variation of q (and $\dot{q}$) without a corresponding variation of p. . . . the entire objection is completely at variance with the intent and spirit of the Hamiltonian picture. Once the Hamiltonian formulation has been set up, [the defining equations] *form no part of it.* The momenta have been elevated

[6] This is a quotation from Goldstein's excellent book [2], p. 364. The emphasis here is Goldstein's.

> to the status of independent variables, on an equal basis with the coordinates and connected with them and the time *only through the medium of the equations of motion themselves* and not by any a priori defining relationship.

We are told that though we must initially understand momentum through its definition as the partial derivative of the Lagrangian with respect to the velocity and so is a function of the positions and velocities, once we make the transition to the Hamiltonian formulation of mechanics this relationship must be forgotten. The reason is that derivations are presented that depend on momentum being independent of positions and velocities. The absurdity of these statements is evident if we think of the motion of a particle with Lagrangian $L(t, x, v) = \frac{1}{2}mv^2 - V(x)$. The momentum is $p = mv$. Except for a scale factor momentum is the same as the velocity. We are told emphatically that we must forget this fact while we derive Hamilton's equations and only after deriving them are allowed to discover that one of Hamilton's equations gives $v = p/m$.

Given a coordinate path q and a Lagrangian L the momentum path p can be computed. It is not as simple as computing the velocity, but there is no more freedom here than in the calculation of the velocity. The momentum at time t is

$$p(t) = \partial_2 L(t, q(t), Dq(t)). \tag{35}$$

Given a path q, the coordinate at time t is $q(t)$, and the velocity is $Dq(t)$; the momentum can be computed from these. In specifying initial conditions at a moment, specifying the coordinates and the momenta at some particular time is equivalent to specifying the coordinates and the velocities at that time.

Hamilton's equation for the particle with potential energy V, considered above, are a pair of first-order equations

$$Dq(t) = p(t)/m \tag{36}$$

$$Dp(t) = -DV(q(t)). \tag{37}$$

The first equation restates the fact that the momentum and the velocity may be computed from one another, and that the velocity function is the derivative of the coordinate function. The momentum path p is no more independent of the position path q than was the velocity path Dq. The momentum does not take on a new status in the Hamiltonian formulation.

Goldstein argues that in the Hamiltonian case the coordinate path and the momentum path may be varied independently, and so apparently derives both of Hamilton's equations:

$$
\begin{aligned}
Dq(t) &= \partial_2 H(t, q(t), p(t)) && (38)\\
Dp(t) &= -\partial_1 H(t, q(t), p(t)). && (39)
\end{aligned}
$$

The correct argument is that the first equation is a consequence of the Legendre transformation: it is a restatement of the relationship between the velocities and the momenta. The variation of the momentum path can in fact be computed from the variation of the coordinate path; it cannot be independently varied.

We can also write the Lagrange equations, which are second order in time, as an analogous pair of first-order equations. The Lagrange equations for the coordinate path q are

$$mD^2q(t) = -DV(q(t)). \tag{40}$$

Introducing the velocity path $u = Dq$, this can be written as a pair of first order equations

$$
\begin{aligned}
Dq(t) &= u(t) && (41)\\
mDu(t) &= -DV(q(t)). && (42)
\end{aligned}
$$

The first equation restates the fact that the velocity path is the derivative of the coordinate path, as equation 38 relates the momentum path to the coordinate path. The second equation encodes the dynamics, as does equation 39.

The status of momentum does not change in the transition from Lagrangian to Hamiltonian mechanics. And the meaning of symbols (or their status of independence) does not change in any valid derivation. Carefully keeping track of when we are speaking of paths and the relationship of one type of path to another eliminates the confusion.

What did we learn?

Physics is often presented with impressionistic mathematics in which the meanings of symbols depends on context and often even changes as a derivation proceeds. In the traditional derivation of the Lagrange equations we saw that initially x and $\dot{x}$ are considered independent state variables, and later must become paths and one is the derivative of the other. The meaning changed. Similarly, we developed an interpretation of what d/dt must mean in the Lagrange equation: compute the time derivative after inserting the path being tested. But in thinking about adding total time derivatives to Lagrangians we came to a rather different understanding: after inserting paths and taking the time derivative, we had to abstract back to a state function. So again, the meaning of d/dt depends on context.

It is easy to get away with such nonsense when discussing things we already understand with our colleagues, especially when the fallacious reasoning does not affect the correctness of the result. It is worse when trying to work out novel problems with difficult interpretations. In this case we can be led down paths that yield wrong answers. However the use of vague or ambiguous context-dependent notation puts an unreasonable burden on students. It is not much harder to express ourselves precisely.

We are advocating precision of expression, not mathematical rigor. Our mathematical formulas must be complete enough to be interpreted mechanically. However, we need not obsess about the convergence of series or the properties of the real number system.

One way we have found to clarify the mathematical expressions that are found in mechanics is to write them in terms of explicit functions and to employ operators that manipulate functions. We avoid the Leibnitz derivative notation and restrict Newton's dots to the names of variables. Instead we use the functional derivatives (and partial derivatives) that are used in modern mathematical texts.[7]

By programing a computer to interpret our formulas we soon learn whether or not a formula is correct. If a formula is not clear it will not be interpretable. If it is wrong, we will get a wrong answer. In either case we are led to improve our program and as a result improve our understanding.

We have been teaching advanced classical mechanics at MIT for half a dozen years using this strategy. We use precise functional notation and we have students program in a functional language. The students enjoy this approach and we have learned alot ouselves. It is the experience of writing software for expressing the mathematical content of mechanics and the insights that we gain from doing it that we feel is revolutionary. We want others to have a similar experience.

[7] For a full presentation of modern mechanics using this functional and computational strategy see our textbook. [5]

Bibliography

[1] Galileo Galilei, *Il Saggiatore (The Assayer)*, 1623.

[2] Herbert Goldstein, *Classical Mechanics*, 2nd edition, Addison-Wesley, 1980.

[3] Seymour A. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, 1980.

[4] IEEE Std 1178-1990, *IEEE Standard for the Scheme Programming Language*, Institute of Electrical and Electronic Engineers, Inc., 1991.

[5] Gerald Jay Sussman and Jack Wisdom with Meinhard E. Mayer, *Structure and Interpretation of Classical Mechanics*, MIT Press, 2001.

[6] Free software is available at `http://www-mitpress.mit.edu/sicm`.