

TODD D. MURPHEY

DYNAMICS FROM CALCULUS AND
GEOMETRY OF MOTION

Contents

<i>Preface: Really, Read This</i>	5
<i>Running ME 314</i>	8
<i>Calculus, Vectors, and Differential Equations</i>	13
<i>Calculus</i>	13
<i>Vectors and Matrix Operations</i>	20
<i>Derivatives, Gradients, and Hessians</i>	25
<i>Ordinary Differential Equations</i>	28
<i>Discussion of Complex Numbers</i>	34
<i>Dynamics and Calculus</i>	37
<i>An Example of Variational Principles at Work</i>	41
<i>Variational Principle for Unforced, Unconstrained Mechanics</i>	45
<i>External Forces and Forces of Constraint</i>	50
<i>Coordinate Independence</i>	55
<i>Coordinate Independence of the Euler-Lagrange Equations</i>	58
<i>Conservation Laws and Numerical Methods</i>	63
<i>Numerical Methods for Simulating ODEs</i>	63
<i>Conservation of Energy and Hamilton's Equations</i>	65
<i>Systems with Nontrivial Hamiltonians</i>	69
<i>Noether's Theorem</i>	71
<i>Aside: A pendulum using the configuration $q = z \in \mathbb{C}$</i>	77

<i>Leibniz Rule and Impacts</i>	81	
<i>Applying the Variational Principle to Impacts</i>	81	
<i>Numerical Aspects of Impact</i>	92	
<i>Inelastic and Plastic Impacts</i>	93	
<i>Class Exercise—Walking on a Treadmill</i>	96	
<i>Geometry of Lagrangians for Rigid Body Motion</i>	97	
<i>Rigid Body Motions</i>	98	
<i>Matrix Representations of Rigid Body Motions</i>	101	
<i>Euler's Theorem and Matrix Exponentials</i>	107	
<i>Euler Angles</i>	111	
<i>Rotational Kinetic Energy and Rotational Inertia</i>	115	
<i>Integral Expressions for Rotational Kinetic Energy</i>	115	
<i>The Rotational Inertia Tensor</i>	119	
<i>Body Velocities</i>	121	
<i>Inertia Tensor Transformation Properties and Principal Axes</i>	125	
<i>Advanced Topics</i>	129	
<i>Lagrangian Derivation of Euler's Equations</i>	129	
<i>Optimal control as an extension of dynamics and the Pontryagin Maximum Principle</i>		
<td>134</td>	134	

Preface: Really, Read This

First, read this section. In general, the preface to a technical book is the most useful thing to read because it tells you something about the mindset of the author, who they anticipated the audience being, and why they decided to write the notes rather than use other books. In the case of mechanics, and in particular finite dimensional mechanics—that is, the mechanics of a finite number of bodies interacting with each other—there are many wonderful books out there, and you should be asking yourself “Why isn’t Professor Murphey using one of those?” Why not use Marsden and Ratiu’s wonderful *Mechanics and Symmetry*¹ or Arnold’s terrific classic *Classical Mechanics*²? All of these are wonderfully written, and I—now—get tremendous pleasure out of reading—and rereading—their texts. These texts, however, are really intended for the graduate student and researcher; although an undergraduate can, in principle, read them without anything more than some calculus and linear algebra, doing so is a major task reserved for those sufficiently motivated by mathematical beauty. For the undergraduate, good texts are fewer. Morin’s witty *Introduction to Classical Mechanics*³ is certainly an example, and we will use it here as an optional text for those who want a secondary text to read. The problem is that it is simply too long and covers too much, making it difficult, I think, to ‘see the forest through the trees’. So the purpose of the present notes is to indoctrinate students in a particular view of mechanics—the variational view. This is not the only view, and indeed I will give the classical “force balance” view short shrift in my writing because I have not found force balance to be useful/insightful, either in industry or in my research. But that doesn’t mean others have not, and the student eager to learn about those techniques will find that it is easy to do so after this course. Regardless, *this* class largely ignores force balance techniques in favor of variational methods (which subsume force balance and are generally—but not always—easier to get correct). By making this choice we get to see impact equations, constrained systems, and integrals of motion. We get to avoid various notions of force balance in “moving frames”.⁴ And we get an appreciation of why coordinate-independent methods have real practical benefit. By including these subjects, the material is relevant to applications that one sees in engineering practice; most mechanisms are, after all, constrained in some way and often experience impacts.

The course is only 10 weeks long, meaning that one must choose between breadth and depth of material. Moreover, I am a big believer that computing a prediction in mechanics goes a long way to believing that those predictions mean something. So in addition to the core

¹ Jerrold E Marsden and Tudor S Ratiu. *Introduction to Mechanics and Symmetry*. Springer, 1994

² Vladimir Igorevich Arnol’d. *Mathematical Methods of Classical Mechanics*. Springer, 1974

³ David Morin. *Introduction to Classical Mechanics*. Cambridge University Press, 2008

⁴ which—trust me—is a good thing...

material, students will find themselves doing quite a bit of simple programming in *python*,⁵ a free programming language with a now mature scientific computing framework that will make much of what this class needs reasonable for someone at the undergraduate level to implement.

The basic outline of the course is to emphasize the role that “directional derivatives” play in variational methods. We use these to develop the Euler-Lagrange equations, the constrained and forced Euler-Lagrange equations, and Lagrangian view of impact laws (including elastic and plastic impacts). Coordinate invariance, integrals of motion, the Legendre transform, and Noether’s theorem are highlighted as essential ingredients of testing the efficacy of numerical methods. Importantly, each of these depend on introductory (but challenging!) ideas in calculus—differentiation, chain rule, product rule, Leibniz rule, and integration by parts show up over and over in the derivations—and one of the most important aspects of this class is to realize that fundamental calculus is one of the primary foundations of engineering analysis, perhaps along with linear algebra. Dynamics, mechanics, optimal control, machine learning, etc., are all applications of these more fundamental ideas. Hence, this is as much a class about ‘applied calculus’ as it is about dynamics.

Once these topics have been covered (roughly 14 50-minute lectures), I discuss rigid body geometry and how to use it to calculate Lagrangian dynamics for arbitrary rigid body systems. This means that rotational inertia does not get covered until near the end of the course, but when we do get to it the topic is quite straight forward.

These course notes are purposefully kept very short, though they have naturally grown over the years in response to interesting questions from students that deserve elaboration. The material is chosen carefully and the notes are intended to be read in detail. As mentioned, calculus plays an important role. The linear algebra requirements generally only require matrix multiplication, but the discussion of rotational inertia requires understanding eigenvalues and eigenvectors. Homework assignments are intended to both reinforce previous sections and anticipate upcoming ideas—they are an integral part of the text and the notes will not be useful without attempting the homework.

Lastly, this stuff is fun—it really is! Aside from being one of the most beautiful accomplishments of the human mind, it is also one of the best introductions to thinking mathematically and thinking, concretely, about the relationship between mathematical abstractions and the reality they are intended to represent. But this makes the material sound heavy and uninteresting. It is also the basis of video games and all computer visualization of dynamic bodies; it allows one to

⁵ A point I make in class is that just as a student cannot expect to be successful as an engineer without being able to write, that student can no longer expect to do so without being able to program. However, the student does not need to be a novelist or a software engineer—just competent at the nuts and bolts of daily programming. Python is an ideal environment for this type of programming.

simulate human bodies in motion (gymnasts, stroke survivors), satellites in space; it allows one to simulate toys (tops, cars, trebuchets, bells); it allows one to simulate profoundly silly things (godzilla knocking over the Eiffel tower—a class project from one of the first iterations of this class). In fact, all of the aforementioned examples have been projects in my class, and in my view the sillier the better—we learn better lighthearted. So I encourage my students to think of the most exciting, ridiculous, and profound mechanical systems and try and simulate them. In my experience, these projects have been done by the students who get the most out of the class and the material.

Running ME 314

Below is our overall strategy for making ME 314 work.

Course Resources

This course will use the following learning assets.

1. The course web page⁶
2. The course notes, available in PDF format on the course web page
3. Python and jupyter notebooks, for learning scientific computing capabilities in python, simulating mechanical dynamic systems, and for homework
4. Discussion forums in the course web page, for each week of material and for each homework
5. An optional textbook for the class is David Morin's wonderful book⁷ (available at Amazon in electronic format for around \$30, paperback for about the same price, hardback more expensive, also available for rent.⁸ I only recommend this if you know that you want a secondary text—we will be using course notes as the primary source and you will not need this book at any point.

⁶ <https://canvas.northwestern.edu/>

⁷ David Morin. *Introduction to Classical Mechanics*. Cambridge University Press, 2008

⁸ [https://www.amazon.com/](https://www.amazon.com/David-Morin-dp-0521876222/dp/0521876222/)
[David-Morin-dp-0521876222/dp/0521876222/](https://www.amazon.com/David-Morin-dp-0521876222/dp/0521876222/))

Support

There will be one TA and at least one grader helping, holding several hours of office hours per week (including a weekly python tutorial), all of which will be online and recorded so that other students can see what was discussed. We will also respond to online questions in the Canvas discussion forum, at scheduled times every day. When questions are short yes/no questions or very easy to answer in text, we will do so. Sometimes we will record short videos explaining the answer. However, there will be no outside meetings (e.g., tutoring) except to discuss personal matters impacting a student's participation in class (e.g., medical reasons for not being present).

Weekly Schedule

This class meets MTWF with a python tutorial hour every Thursday during classtime. Homework is generally posted on Monday and then due on the next Wednesday.⁹ Most Wednesdays there will be a 5-10 minute assessment. This will be very simple, and will be paper and pen/pencil.¹⁰ The cumulative grade on this weekly assessment will replace an exam in the class.

⁹ Solutions to homework are posted on the Friday after homework has been turned in. If students still have extensions at that point, I expect them to not look at solutions, as honor code and common sense both dictate.

¹⁰ If for any reason we find ourselves going remote, this will remain our assessment strategy and we will require students to write on paper and hold up the answer to their video.

Accommodations

Some of you will have accommodations. The only thing that will be affected will be the Wednesday assessments. Any student requesting accommodations related to a disability or other condition is required to register with ANU (accessiblenu@northwestern.edu; 847-467-5530) and provide professors with an accommodation notification from AccessibleNU, preferably within the first two weeks of class. All information will remain confidential.

Python and Colaboratory

This class will use python as the programming language. We will incorporate bits of python code throughout the notes as example code. This will both give students examples of syntax as well as prepare them for the upcoming problem sets. Moreover, students will be able to ask questions at office hours about python. These snippets might look something like this, which numerically solves a simple ordinary differential equation and then plots the resulting solution:

```

1 from scipy.integrate import solve_ivp
2 from numpy import arange
3 from matplotlib import pyplot
4 ttest=arange(0,10,0.1)
5 def exponential_decay(t, y): return -0.5 * y
6 sol = solve_ivp(exponential_decay, [0, 10], [1],t_eval=ttest)
7 print(sol.t)
8 print(sol.y)
9
10 plot1=pyplot.plot(sol.t,sol.y[0])
11 pyplot.show(plot1)
```

You will have the option of either using **Google's colaboratory**—available at colab.research.google.com—to run your code or using your own installation. Either way, you will need to turn in a PDF of your code each week.¹¹

Grading

The class will be graded based on 1) weekly homework (50%), 2) the weekly Tuesday assessment (25%), and 3) the final project (25%).

Homework Expectations and Policies

Homework is a huge part of learning dynamics well. Moreover, much of the homework in this class will be done in python, using jupyter notebooks. Python is the most popular and generally useful programming language right now, and jupyter notebooks provide a nice

¹¹ Saying ‘my code does not work’ will not be helpful to you or to the TA in terms of helping you. Whenever your code does not do what you expect, you should create a compact (i.e., a small amount of code) instance of what goes wrong to help the TA help you. The first week’s python help will provide some examples of this. **Run example code from the notes prior to asking for help.**

interface for it. We will help you with tutorials and installation instructions, but all the homework in this class can be also be done in Google's Colaboratory¹² if your installation of python does not work. We will provide a template jupyter notebook for each homework.

All homeworks will be submitted as a single PDF file in Canvas, except animations. Animations should be submitted as MP4 files either from screen capture or video taken with your phone or other camera. **If you do not submit your homework as a single PDF file, it will not be graded.**

The homework will start out with 'Homework 0'—essentially a test of whether you have python working correctly. In that homework we will give you nearly all the python code that you need. As the homework progresses to later in the term, we will expect more from students in terms of discovering how to develop the code they need.

Weekly Assessments

Every Wednesday at the beginning of class, we will have a short quiz. I will tell you the expected content of the quiz ahead of time, so there will not be any surprises. These will be on paper—you will need to have a writing implement every Wednesday. I will count down to zero and every student will turn in the quiz.¹³ If you are going to miss an assessment, e.g., because of being ill, you must let the TA and myself know prior to the time of the quiz (there are exceptions, of course, but these are likely to be extreme/unusual situations).

The assessments come in two forms: a) completing a proof of one of the main results in class; and b) solving a simple dynamics problem.

Schedule of Topics by Week

This schedule *will change* as the class evolves. This schedule reflects how the course went last year. Note that within this structure we have about four lectures that are not filled in. Whenever I find from our discussions in class that some part of the class is not clear or if the class does not have the background I am expecting, I will push back the lectures and make time for the topic. We always end up using all the lecture time.

1. Week 1

- (a) Tuesday: Introduction to class.
- (b) Wednesday: Introduction to python. Python introduction continued on Thursday.
- (c) Friday: Background on differentiation

¹² <http://colab.research.google.com>

¹³ If we ever have to go remote, after five minutes students will lift their piece of paper and so their webcam can see it. We will screenshot the entire class to confirm that students completed the assessment on time and then students will upload their own image to canvas—that will be what we grade. This avoids some of the issues associated with trying to proctor exams online, and it also enables us to give the quiz to someone who is sick (or needs to quarantine) and needs to take the quiz remotely.

2. Week 2

- (a) Monday: [An example of variational principles](#), the point mass in gravity
- (b) Tuesday: complete lecture on variational example, discuss perturbations and what they do or do not 'physically mean'. Discussion of integration by parts.
- (c) Wednesday: [Unforced, unconstrained mechanics and the Euler-Lagrange equations](#); HW 0 due
- (d) Friday: Python workflow example for mass in gravity

3. Week 3

- (a) Monday: Examples of the E-L Equations
- (b) Tuesday: [Constrained Euler-Lagrange Equations](#)
- (c) Wednesday: Finish Constrained E-L Equations; weekly quiz; HW 1 due
- (d) Friday: External Forces;

4. Week 4

- (a) Monday: [Coordinate independence](#)
- (b) Tuesday: Coordinate independence continued
- (c) Wednesday: [Conservation laws and Hamilton's equations](#); weekly quiz; HW 2 due
- (d) [Nontrivial Hamiltonians](#)

5. Week 5

- (a) Monday: [Noether's theorem](#)
- (b) Tuesday: Noether's theorem;
- (c) Wednesday: Noether's theorem code, weekly quiz; numerical methods; HW 3 due
- (d) Friday: finish up Noether, start discussion of complex numbers

6. Week 6

- (a) Monday: Complex Numbers; Linear algebra concepts, including complex eigenvalues and eigenvectors;
- (b) Tuesday: Dynamics on Complex Numbers
- (c) Wednesday: finish Dynamics on Complex Numbers, start [Impacts](#); weekly quiz; HW 4 due
- (d) Friday: Elastic impacts;

7. Week 7

- (a) Monday: Elastic impacts
- (b) Tuesday: Elastic impacts, time stepping
- (c) Wednesday: Inelastic impact and plastic impact; weekly quiz;
HW 5 due
- (d) Friday: simultaneous impact, and impact with open system
Hamiltonians;

8. Week 8

- (a) Monday: **Rigid Body Motion**
- (b) Tuesday: **Matrix representations**;
- (c) Wednesday: Matrix representations
- (d) Friday: Matrix representations

9. Week 9

- (a) Monday: Example of solving for dynamics data using geometry
- (b) Tuesday: **Euler's theorem**;
- (c) Wednesday: **Euler angles**; weekly quiz; start Rotational Kinetic Energy; HW6 due
- (d) Friday: **Rotational Kinetic Energy**

10. Week 10

- (a) Monday: **Rotational Inertia and Body velocities**
- (b) Tuesday: **Principal Axes of Inertia**
- (c) Wednesday: Overview of constructing dynamics, including bodies with volume; also open questions; Nothing due the day before Thanksgiving. **no weekly quiz**

11. Week 11

- (a) Monday: Overview of constructing dynamics, including bodies with volume
- (b) Tuesday: No class
- (c) Wednesday: HW7 due (no weekly quiz)
- (d) Friday: Help with final projects

Calculus, Vectors, and Differential Equations

THIS SECTION WILL HAVE all the different pieces of background mathematics that we anticipate will be important to the class.

Calculus

There are not that many things you need to remember from calculus, but the few that you do need are extremely important. As I will emphasize in the lectures, there are five ideas from calculus that play a role over and over in engineering analysis:

1. the directional derivative;
2. chain rule;
3. product rule;
4. Leibniz rule;
5. integration by parts.

It is astonishing to me how these are ignored in favor of *operational* aspects of calculus.¹⁴ Although operational aspects of calculus matter to performing calculations by hand—the practical taking of derivatives and integrals—they are not nearly as important as the five things in this list above.

These notes are intended to serve as a reminder of what you learned in calculus, and I will reference these facts throughout the rest of the class.¹⁵

¹⁴ e.g., students routinely learn calculus facts, like $\frac{d}{dx}x^2 = 2x$. Like other math facts that one learns as a child (e.g., $2 + 3 = 5$, $7 \times 4 = 28$), these facts really do help ground our understanding through exposure to operations. But calculus is also a set of powerful ideas that can be applied to settings where *there are no explicit formulas to manipulate*. The topic of this course is such a setting—we will almost never work with explicit expressions to which we apply the tools of calculus.

¹⁵ In fact, when I ask students questions in class, the student caught off guard daydreaming about something else is well served by guessing one of these five possibilities!

Derivatives, Derivative Notation, and Integrals

First, you have to know what a derivative of a function with respect to a scalar is. For instance, if I have a function $f(x)$ —where x is an unknown number—the *derivative* of $f(x)$ with respect to x is defined to be

$$\frac{d}{dx}f(x) = \frac{df}{dx}(x) = \frac{df(x)}{dx} = \lim_{dx \rightarrow 0} \frac{f(x + dx) - f(x)}{dx}.$$

This definition states that if I change x by a small amount dx ,¹⁶ then the value of the function will change as well, and the amount of that change is captured by the derivative if dx is small enough. That is, $f(x + dx) \approx f(x) + [\frac{d}{dx}f(x)]dx$, and this approximation becomes exact as $dx \rightarrow 0$. (We will not work rigorously with what the word “approximate” means mathematically in this setting, but rest assured that mathematicians have provided us a foundation where we can use the word roughly according to its intuitive meaning.)

That mental picture is also what motivates the *directional derivative*—the derivative that will be the workhorse of variational calculus and everything we do in this class. The *directional derivative* of $f(x)$ in the direction dx is defined to be:

$$\frac{d}{dx}f(x) \cdot dx = \frac{d}{d\epsilon}f(x + \epsilon dx)|_{\epsilon=0}.$$

Note that this yields the same thing as the derivative above, except that now we are multiplying the derivative $\frac{d}{dx}f(x)$ by the direction dx . Why bother with this definition? Although a little more abstract, this is the definition that generalizes nicely to the types of functions we will care about.^{17,18}

Sometimes we will specifically be taking derivatives of a function with respect to time t . In this case, special notation is sometimes used, called “dot” notation. If I have a variable y that is a function of t (that is, the function is of the form $y(t)$), then I can write the first derivative of y with respect to t as

$$\frac{d}{dt}y = \dot{y}.$$

Moreover, I can write the second derivative of y with respect to t as

$$\frac{d^2}{dt^2}y = \ddot{y}.$$

Since mechanical systems almost always involve two time derivatives—because mechanical systems are described in terms of their *acceleration*, the second time derivative of position—this notation is very common in engineering. The place where it plays the biggest role is

¹⁶ A note on the supposed difference between dx and Δx is warranted. Many textbooks prefer d to reference infinitesimals and Δ to reference finite differences. There is nothing wrong with this perspective, but context typically makes clear whether we are talking about finite (small) values or infinitesimals; I see no real value in distinguishing between them. However, I invite students to convince me!

¹⁷ I will also comment here that the directional derivative is also known as the Gâteaux derivative, which should not be confused with the Fréchet derivative. Do not worry about this—just be aware that the Fréchet derivative is the more mathematically rigorous notion of derivative while the Gâteaux derivative is the practical notion of derivative used in basically all engineering analysis.

¹⁸ Other notation for the derivative of $f(x)$ with respect to x includes the following:

1. $\frac{df}{dx}(x)$
2. $\frac{df(x)}{dx}$
3. $f'(x)$
4. $Df(x)$
5. $D_x f(x)$
6. $D_1 f(x)$
7. $f_x(x)$.

If this seems like a lot of things that all mean the same thing, you are correct! What motivates these is not a difference in meaning, but a difference in how complex a derivation *looks*. In class, we will see mathematical derivations where one notation would be really irritating while another notation is easier to keep track of. Nevertheless, these really are just notational choices; the definitions above are what you should pay attention to in terms of your understanding.

in the study of *ordinary differential equations (ODEs)*, something I will bring up shortly.

Lastly, if I differentiate a function with respect to multiple variables, I will need to be able to identify the variable in question. Let's say I have $f(x, y)$ and I want to take the derivative with respect to x and the derivative with respect to y , then I use the *partial derivative* notation (using the symbol ∂) to represent those derivatives:

$$\frac{\partial}{\partial x} f(x, y) \quad \frac{\partial}{\partial y} f(x, y).$$

Integrals are simply "the opposite" or the inverse of differentiation. If $\frac{d}{dx}g(x) = f(x)$, then $\int f(x)dx = g(x)$. We will discuss integrals in general more later in the class, but for the beginning of class integrals will always be with respect to time t .¹⁹

If $x(t)$ represents the position of an object as a function of time, its first derivative is velocity

$$v(t) = \frac{d}{dt}x(t) = \dot{x}(t) = \dot{x}.$$

Its second derivative is acceleration

$$a(t) = \dot{v}(t) = \ddot{v} = \ddot{x}(t) = \ddot{x}.$$

One of the reasons we use the python programming language is that it has a well-developed set of symbolic tools—including symbolic manipulation of calculus—available in the *Sympy* package. The code below uses *Sympy* to create the function $f(t) = x^t$ and then takes the partial derivative with respect to x and then with respect to t .

```

1 import sympy as sym
2 from sympy.abc import x, y, z, t
3
4 # define a function as a SymPy expression
5 f = x**t
6 print('original function f: ')
7 display(f)
8
9 # compute the derivative of f wrt to x
10 dfdx = f.diff(x)
11 print('derivative of f wrt x: ')
12 display(dfdx)
13
14 # compute the derivative of f wrt to t
15 dfdt = f.diff(t)
16 print('derivative of f wrt t: ')
17 display(dfdt)

```

Listing 1: SymPy is a powerful symbolic computing framework in python

¹⁹ Examples If n is an integer ($n = 1, 2, 3, \dots$) and h is a scalar parameter (it is a number), then

$$\begin{aligned}\frac{d}{dt} t^n &= nt^{n-1} \\ \frac{d}{dt} \sin(\omega t) &= \omega \cos(\omega t) \\ \frac{d}{dt} \cos(\omega t) &= -\omega \sin(\omega t) \\ \frac{d}{dt} e^{ht} &= he^{ht}\end{aligned}$$

Product Rule, Chain Rule, Leibniz Rule, and Integration by Parts

There are four rules beyond the definition of the directional derivative that you need to remember from calculus—*chain rule, product rule, Leibniz rule, and integration by parts*.

Product rule states that if f and g are both functions of x , then the derivative of the product $f(x)g(x)$ is equal to

$$\frac{d}{dx} (f(x)g(x)) = \left(\frac{d}{dx} f(x) \right) g(x) + f(x) \left(\frac{d}{dx} g(x) \right).$$

```

1 import sympy as sym
2 from sympy.abc import x, y, z, t
3 from sympy import sin
4
5 # define two functions as a SymPy expressions
6 f = x**2
7 g = sin(x)
8 print('original functions f and g: ')
9 display(f, g)
10
11 # compute the derivative of f
12 df = f.diff(x)
13 print('derivative of f: ')
14 display(df)
15
16 # compute the derivative of g
17 dg = g.diff(x)
18 print('derivative of g: ')
19 display(dg)
20
21 # compute the derivative of f*g directly
22 fg = f * g
23 dfg = fg.diff(x)
24 print('derivative of f*g: ')
25 display(dfg)
26
27 # compute the derivative of f*g using product rule
28 dfg = f * dg + g * df
29 print('derivative of f*g: ')
30 display(dfg)

```

Listing 2: SymPy Example for Product Rule

Chain rule states that if one composes $f(y)$ and $g(x)$ (i.e., has the function $h(x) = f(g(x))$), then the derivative is:

$$\frac{d}{dx} f(g(x)) = \frac{d}{dy} f(g(x)) \left(\frac{d}{dx} g(x) \right).$$

This is often very confusing because it isn't clear what the form of “ y ” is. The key is that $g(x)$ must have its output be the same type of variable as y is—so if $g(x)$ is a number, then y must be a number and if $g(x)$ is a vector, then y is a vector. Note that when x is a scalar, the particular order of multiplication doesn't matter. When dealing with

vector-valued functions, you will want to pay attention to the order of multiplication in both product rule and chain rule.²⁰

20

```

1 import sympy as sym
2 from sympy.abc import x, y
3 from sympy import sin
4
5 # define two functions as a SymPy expressions
6 f = x**2
7 g = sin(f) # note that g is function f, but in SymPy,
8         # an expression must be comprised of symbols.
9         # f is an expression, x is a symbol, thus in
10        # SymPy g is actually an expression of x
11 print('original functions f and g: ')
12 display(f, g)
13
14 # compute the derivative of g directly
15 dg = g.diff(x)
16 print('derivative of g: ')
17 display(dg)
18
19 #####
20 # then verify it with chain rule
21
22 # define a symbolic version of g
23 sym_g = sin(y) # now here new_g is function of a symbol y
24 sym_dg = sym_g.diff(y)
25
26 # use chain rule
27 dg = sym_dg.subs(y, f) * f.diff(x) #sym_dg.subs(y, f) substitutes f for
           symbol y
28 print('derivative of g: ')
29 display(dg)

```

Listing 3: SymPy Example for Chain Rule

Leibniz rule will play a role in understanding impacts, but it shows up everywhere and often confuses students (and, well, everyone really). People are frankly often a bit scared by Leibniz rule, but it is tremendously useful. Let's say that $f(t)$ is actually a function that involves an *integral*. For example

$$f(t) = \int_0^t \sin(\tau)d\tau$$

is a function that involves an integral. For any given t , this function will return a number, and that means that you can differentiate with respect to t . Leibniz rule tells us that the derivative can be found by:

$$\frac{d}{dt} \int_0^t h(\tau)d\tau = h(t).$$

This says that the derivative of this integral is equal to the function being integrated at the time t . Leibniz rule is sometimes known as the Fundamental Theorem of Calculus because it relates derivatives and integrals, showing that one inverts the other.²¹

Examples

$$\frac{d}{dx} \sin(\omega x) = \omega \cos(\omega x)$$

$$\frac{d}{dt} \int_0^t \sin(\tau)d\tau = \sin(t)$$

²¹ Example

Continuing with our example from earlier, let

$$h(b) = \int_a^b f(x)dx = \int_a^b x^2 dx.$$

Now we can ask what the derivative of h with respect to b is. By application of Leibniz rule one gets

$$\frac{\partial h}{\partial b} = x^2|_{x=b} = b^2.$$

If we generalize slightly to the problem

$$h(b) = \int_{a^2}^{b^3} f(x)dx = \int_{a^2}^{b^3} x^2 dx,$$

the derivative of h with respect to b can be determined using Leibniz rule and chain rule together. To see this, rewrite h as

$$h = h(c(b)),$$

where $h = \int_a^b x^2 dx$ and $c = b^3$. Then by Leibniz rule we get that

$$\frac{\partial h}{\partial c} = c^2 = b^6.$$

By chain rule

$$\frac{\partial h}{\partial b} = \frac{\partial h}{\partial c} \frac{\partial c}{\partial b}$$

and $\frac{\partial c}{\partial b} = 3b^2$. So

$$\frac{\partial h}{\partial b} = \frac{\partial h}{\partial c} \frac{\partial c}{\partial b} = 3b^8$$

Now check that this what you get if you have python do the symbolic calculations.

```

1 from sympy.abc import a, b
2 from sympy import integrate
3 #normally this would go at the top of your code, but I'm
4 #putting it here so you can see how to import a function
5 #you want to use
6
7 #define the definite integral of f from a squared to b cubed
8 h=integrate(f,(x,a**2,b**3))
9 display(h)
10
11 #now take the derivative of that, and it should give us the same thing as
12 # Leibniz rule
12 dhdb=h.diff(b)
13 display(dhdb)

```

Listing 4: SymPy Example for Leibniz Rule

Integration by Parts Integration by parts is commonly remembered using the form

$$\int u dv = uv - \int v du.$$

Although there is nothing wrong with this, there are two confusing parts. First, these are indefinite integrals, so at minimum one should be thinking of integration by parts as

$$\int_a^b u dv = uv|_a^b - \int_a^b v du,$$

where $uv|_a^b$ implies that you will evaluate uv between a and b (i.e., $uv|_a^b = u(b)v(b) - u(a)v(a)$). Moreover, if u and dv multiply like udv , then in general v and du will not multiply like vdu . This is because in general vector and matrix multiplication does not commute. Instead, v and du will multiply like dvv to be dimensionally consistent. So we finally arrive at my preferred way of writing integration by parts.

$$\int_a^b u dv = uv|_a^b - \int_a^b d(uv).$$

Example

Consider the integration below:

$$\begin{aligned}
\int x \cos(x) dx &= \int \underbrace{x}_u d(\underbrace{\sin(x)}_v) \\
&= x \sin(x) - \int \sin(x) dx \\
&= x \sin(x) + \cos(x)
\end{aligned}$$

```

1 import sympy as sym
2 from sympy import symbols, integrate
3 from sympy import sin, cos
4

```

```

5 # define function to be integrated
6 x, y = symbols('x, y')
7 func = x * cos(x)
8 print('function to be integrated: ')
9 display(func)
10
11 # direct symbolic integration
12 h = integrate(x * cos(x))
13 print('integration of the function: ')
14 display(h)
15
16 # extract u and v for integration by parts
17 u = x
18 v = sin(x)
19 print('u and v: ')
20 display(u, v)
21
22 # integration by parts
23 h = u * v - integrate(v, u)
24 print('integration of the function: ')
25 display(h)

```

Listing 5: SymPy Example for Leibniz Rule

Taylor Series

Yes, Taylor series are really, really important! The Taylor series of a function $f(x)$ is an approximation of the function f at a value of x that only involves derivatives of f evaluated at that value of x (in this case $x = 0$).

$$f(x) = f(0) + \left(\frac{d}{dx}f(x)|_{x=0} \right) x + \frac{1}{2!} \left(\frac{d^2}{dx^2}f(x)|_{x=0} \right) x^2 + \frac{1}{3!} \left(\frac{d^3}{dx^3}f(x)|_{x=0} \right) x^3 + \cdots + \frac{1}{n!} \left(\frac{d^n}{dx^n}f(x)|_{x=0} \right) x^n + \cdots$$

Examples of Taylor series that are particularly relevant to this class are derivatives of e^{ht} , $\sin(\omega t)$, and $\cos(\omega t)$.

$$\begin{aligned} e^{ht} &= 1 + ht + \frac{1}{2!}h^2t^2 + \frac{1}{3!}h^3t^3 + \frac{1}{4!}h^4t^4 + \frac{1}{5!}h^5t^5 + \frac{1}{6!}h^6t^6 \dots \\ \sin(\omega t) &= \omega t - \frac{1}{3!}\omega^3t^3 + \frac{1}{5!}\omega^5t^5 + \dots \\ \cos(\omega t) &= 1 - \frac{1}{2!}\omega^2t^2 + \frac{1}{4!}\omega^4t^4 - \frac{1}{6!}\omega^6t^6 + \dots \end{aligned}$$

Vectors and Matrix Operations

Vectors—and matrix operations on those vectors—play a critical role in *organizing* information about a dynamical system. We will use vectors *all the time* in this class when we are analyzing and simulating ordinary differential equations that represent mechanical system behavior. You should think of vectors and matrix operations as largely being useful *notation*; they primarily serve as a way of organizing information in a nice, convenient way that helps with computation.

Vectors are elements of vector spaces. This somewhat snarky definition of a vector—preferred by mathematicians everywhere—is irritating but helpful to keep in mind. What the mathematicians of the world are trying to say is that what makes a vector a vector are its *properties* rather than its numerical *representation*. The two properties that define a vector space \mathcal{X} are i) that you can add two vectors together and get another vector (i.e., $x_1 \in \mathcal{X}$ and $x_2 \in \mathcal{X}$ implies that $x_1 + x_2 \in \mathcal{X}$, where the notation \in should be read ‘is in’); and ii) you can multiply a vector by a scalar and still be in the vector space (i.e., $x \in \mathcal{X}$ and $\alpha \in \mathbb{R} \implies \alpha x \in \mathcal{X}$, where \mathbb{R} are the real numbers and \implies should be read ‘implies’).

This may feel pretty abstract, largely because in vector calculus you focused on the idea that ‘vectors are lists of numbers’—something that, while often true, misses the key point. The key point is that you may never have really thought about the fact that you are not always allowed to add numbers.²² Nevertheless, it is *often* true that vectors are lists of numbers, like $[1, 2, 3, 4]$. If $x_1 = [a, b]$ and $x_2 = [c, d]$, we can add them $x_1 + x_2 = [a + c, b + d]$. We can also scale them $\alpha x_1 = [\alpha a, \alpha b]$.

Moreover, *matrix multiplication of a vector* can be used to transform the vector. As an example, assume we have a vector w and matrix A :

$$w = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}. \text{The vector } w \text{ is a } 3 \times 1 \text{ vector and } A$$

is a matrix that has three rows and three columns, with a total of nine numbers.

We need some notion of multiplication between the matrix A and the vector w : $Aw = A \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$. The ‘right’ way to define multiplication between A and w is to take elements of the rows of A and multiply them by their corresponding elements of the state vector and sum them together.²³ For instance, if I take the A matrix I just

²² Just to develop a bit of intuition about this, think about ‘angle wrapping’—two angles can only be added together if you enable the angles to ‘wrap’, identifying 0 with 2π . Similarly, scaling an angle by a scalar also requires angle wrapping. But angle wrapping is the result of imposing vector space structure onto the angles; they do not have that structure naturally.

²³ This is not the only possible notion of multiplication, but it is the most useful one, again emphasizing the matrix operations and vectors are really notation for organizing computation.

defined and multiply A and w , I get

$$Aw = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 \cdot x + 2 \cdot y + 3 \cdot z \\ 4 \cdot x + 5 \cdot y + 6 \cdot z \\ 7 \cdot x + 8 \cdot y + 9 \cdot z \end{bmatrix}.$$

Note that the result is a vector again, so multiplying a matrix times a vector gives a vector. If A is a square matrix, the new vector Aw will be of the same dimension as w .

Matrix and vector multiplications are supported in both Python's SymPy and NumPy packages, but they can be a little tricky and confusing sometimes. Below is an example of doing Matrix multiplications using both packages:

```

1 ######
2 # Typically in SymPy and NumPy, vectors and matrices are under the same
3 # object,
4 # the only difference is their dimensions. Below are the examples:
5 #####
6 # Matrix multiplication in SymPy
7
8 import sympy as sym
9 from sympy.abc import x, y, z, a, b, c
10 from sympy import Matrix
11 print('----- SymPy Examples -----')
12
13 # define a "matrix" as SymPy's Matrix object
14 A = Matrix([[ 1, 2, 3],
15             [ 4, 5, 6],
16             [ 7, 8, 9]])
17
18 # define a "vector" also as SymPy's Matrix object
19 w = Matrix([x, y, z])
20
21 # display the matrix multiplication
22 print('A * w = ')
23 display(A * w)
24
25 # define a 1-by-3 vector using transpose
26 # (w is a 3-by-1 vector)
27 v = Matrix([a, b, c]).T
28 print('w = ')
29 display(w)
30 print('v = ')
31 display(v)
32
33 # w * v will be a 3-by-3 matrix
34 print('w * v = ')
35 display(w * v)
36
37 #####
38 # Matrix multiplication in NumPy
39 # Note:
40 #     Things are a little more tricky in NumPy, since
41 #     it provides two kinds of multiplications for
42 #     vectors/matrices. You will them below:
43
```

```

44 import numpy as np
45 print('----- NumPy Examples -----')
46
47 # define a "matrix" as NumPy's Array object
48 A = np.array([[ 1, 2, 3],
49               [ 4, 5, 6],
50               [ 7, 8, 9]])
51 # since NumPy doesn't support symbols, we give
52 # values to vector w, which is also a NumPy's
53 # Array object
54 w = np.array([1, 2, 3])
55
56 # if you simply multiply them like in SymPy ...
57 print('A * w = ', end='\n')
58 print(A*w)
59
60 # the result above means NumPy's default multiplication
61 # is not for matrices, instead, it's an element-wise
62 # multiplication:
63 #     A can be considered as a vector of vectors, whose
64 #     elements are [1, 2, 3], [4, 5, 6], and [7, 8, 9].
65 #     Thus, element-wise multiplication means w will be
66 #     multiplied to each vector one-by-one. That's why
67 #     the result is still a 3-by-3 matrix
68
69 # the real matrix multiplication is conducted by the "dot()"
70 # method in NumPy, there're two ways using that:
71
72 # method 1: the object is already a numpy array "np.array()", 
73 # not a default Python list "[]" (these two are totally different
74 # things, even though sometimes they can be mixed):
75 print('type of A and w: ', type(A), type(w))
76 print('A * w = ', end='\n')
77 print(A.dot(w))
78
79 # method 2: valid for both NumPy Array or Python's List
80 A = [[1, 2, 3],
81       [4, 5, 6],
82       [7, 8, 9]]
83 w = [1, 2, 3]
84 print('type of A and w: ', type(A), type(w))
85 print('A * w = ', end='\n')
86 print(np.dot(A, w))
87
88 # NumPy typically doesn't have explicit array objects
89 # distinguishing a 1-by-3 vector and 3-by-1 vector.
90 # The last example in SymPy section can be achieved through
91 # NumPy's "outer()" method
92 v = np.array([4, 5, 6])
93 print('w = ', w)
94 print('v = ', v)
95 print('w * v = ', end='\n')
96 print(np.outer(w, v))

```

Listing 6: Matrix multiplication in both SymPy and NumPy

When are vectors not lists of numbers? Think for a moment about the two functions $\sin(t)$ and $\cos(t)$. You probably find it intuitive that you are allowed to add these together. But these are *functions* that take on a different value for an infinite number of values of t . That

means that if we were to write $\sin(t)$ as a vector of values (where each element of the vector tells us the value of \sin at that particular t), we would need a vector that is infinitely long. Nevertheless, functions form a vector space—we can take two functions and add them to get another function (like $f(t) = \sin(t) + \cos(t)$, which is also a function t) and we can take a function and scale it to get another function (like $f(t) = 2\sin(t)$). We will use the fact that functions can be added together and scaled *a lot* in this class.

Other Stuff to Know: Matrix Commutation, Eigenvalues, Determinants

Matrix multiplication does not commute! That is, given a matrix A and a matrix B —both square matrices so that multiplication can be in either order—then in general $AB \neq BA$.

A square $n \times n$ matrix has n eigenvalues λ_i and eigenvectors v_i . The equation that these satisfy²⁴ is

$$Av_i = \lambda_i v_i.$$

Do not let the simplicity of this statement undermine how important it is—it says that every matrix acts on its eigenvectors by scaling them. Remember that if A is an $n \times n$ matrix, there are n eigenpairs (λ_i, v_i) .

The determinant of A roughly tells you how a matrix maps volume—does it shrink or expand a volume? This is specifically important when the determinant of A is zero ($\det A = 0$). This implies that a volume has been mapped to set of zero volume, indicating that some eigenvector has a zero eigenvalue.

Eigenvalues can be imaginary valued (where $\lambda = a \pm jb$ and $j = \sqrt{-1}$). We will discuss eigenvalues a bit later in the class. Python provides numerical (and symbolic) tools to evaluate eigenvalues.

Python's NumPy package supports eigenvalue and eigenvector decomposition through its "linalg.eig()" method, which could return a set of eigenvalues and eigenvectors given a matrix. Below is an example of how to use it.

```

1 import numpy as np
2
3 # define a square matrix
4 A = np.array([[3, 1, 1],
5               [1, 4, 2],
6               [1, 2, 5]])
7
8 # do eigen decomposition
9 w, v = np.linalg.eig(A)
10 print('list of eigenvalues:')
11 print(w)
12 print('list of eigenvectors (each column):')
13 print(v)
14
```

²⁴ Many students associate eigenvalues and eigenvectors with the equation $\det(A - \lambda I) = 0$. This equation shows up when solving for eigenvalues and eigenvectors, but is not very important in terms of defining them. The equation to the left says that eigenvalues and eigenvectors are the special case where a matrix operation only scales the vector. The fact that these vectors and scalars exist at all is part of the magic of linear algebra.

```
15 # let's verify the results
16 for i in range(len(w)):
17     wi = w[i] # i-th eigenvalue
18     vi = v[:,i] # i-th eigenvector
19     print('\nverify {}-th set of eigenvalue and eigenvector: '.format(i+1))
20     print('A * v = ', A.dot(vi))
21     print('w * v = ', wi * vi)
```

Listing 7: EigenPairs in NumPy

Derivatives, Gradients, and Hessians

For all that follows, assume the following.

1. We have a *function* $f : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $x \mapsto f(x) \in \mathbb{R}$;
2. We have a *norm* $\|x\|$ (also called a metric) on \mathbb{R}^n that gives us a measure of distance or size of a vector; For us this will often be the Euclidean metric where $\|x\|^2 = x^T x = \sum_i^n x_i^2$. But sometimes it will *not* be the Euclidean metric—sometimes it will be the kinetic energy—so don't assume that the norm is always the standard one.
3. Lastly, sometimes the norm comes from an *inner product* $\langle x, y \rangle$ for $x, y \in \mathbb{R}^n$, where $\|x\|^2 = \langle x, x \rangle$. The inner product additional gives us a notion of *orthogonality* defined by the cos function:

$$\cos \theta = \frac{\langle x, y \rangle}{\|x\| \|y\|}.$$

Note that changing the inner product changes whether two vectors are orthogonal or not.

The First Derivative and the Gradient

The derivative of f with respect to x is the linear map that is locally a best approximation to the function f . That is, for ϵ small,

$$\|f(x + \epsilon v) - (f(x) + \epsilon A \cdot v)\|$$

is smallest when $A = Df(x)$. If $n = 1$, this is saying that the slope of the tangent line is the best possible slope you can choose to approximate f .

The gradient $\nabla f(x)$ is *only* defined when an inner product is available. In particular, the gradient satisfies:

$$Df(x) \cdot v = \langle \nabla f(x), v \rangle.$$

A necessary condition for x^* to extremize f is that $Df(x^*) = 0$ (i.e., $Df(x^*) \cdot v = 0 \forall v \in \mathbb{R}^n$).²⁵ If an inner product exists, then $\nabla f(x^*) = 0$ (i.e., $\langle \nabla f(x^*), v \rangle = 0 \forall v \in \mathbb{R}^n$) is a necessary condition for x^* to extremize f .

²⁵ It being a necessary condition means that if $Df(x^*) \neq 0$, x^* is *not* an extremizer.

The Second Derivative and the Hessian

The second derivative of f with respect to x is the bilinear map that is locally a best approximation to the function f to second order. That is, for ϵ small,

$$\|f(x + \epsilon v) - \left(f(x) + \epsilon A \cdot v + \frac{1}{2} \epsilon^2 B \cdot (v, v) \right)\|$$

is smallest when $A = Df(x)$ and $B = D^2f(x)$. If $n = 1$, this is saying that the coefficient of the parabola with concavity $D^2f(x)$ is the best possible parabola you can choose to approximate f .

If one writes $D^2f \cdot (v, v)$ in matrix form, one gets $v^T Hv$, where H is the matrix representation of $D^2f(x)$ and is equal to the second partial derivatives of f evaluated at x . Note that since mixed partials commute, $H = H^T$ (i.e., H is symmetric). Lastly, symmetric matrices have *real* eigenvalues—there are no complex eigenvalues for $H = H^T$.

A sufficient condition for x^* to be a minimizer of f is that H be positive definite (noted by $H > 0$). A matrix A is positive definite if $x^T Ax > 0 \forall x \in \mathbb{R}^n \setminus \{0\}$. A test for H being positive definite is that all its eigenvalues are greater than zero.

Understanding the notation $D^2f(x) \cdot (v, v)$

There has been quite a bit of confusion about how to take the second derivative. Let us proceed formally from the first derivative $Df(x) \cdot v$. Then

$$\frac{d}{d\epsilon} Df(x + \epsilon v) \cdot v|_{\epsilon=0} = \left(D^2f(x + \epsilon v) \cdot v|_{\epsilon=0} \right) \cdot v$$

because v does not depend on ϵ . Moreover, $D^2f(x + \epsilon v) \cdot v|_{\epsilon=0} = D^2f(x) \cdot v$, so

$$\frac{d}{d\epsilon} Df(x + \epsilon v) \cdot v|_{\epsilon=0} = \left(D^2f(x) \cdot v \right) \cdot v.$$

To evaluate this, we note that first $D^2f(x)$ acts on one copy of v to produce a covector and then that covector acts on the second copy of v to produce a number. If H is the Hessian, then $D^2f(x) \cdot v = v^T H$ and $(D^2f(x) \cdot v) \cdot v = v^T Hv$. It is largely because of this that we use the notation $Df(x) \cdot (v, v)$ for $(D^2f(x) \cdot v) \cdot v$.

Some properties of the first derivative and the second derivative

The first derivative is *linear*—an operation $g(x)$ is linear if $g(\alpha x + \beta y) = \alpha g(x) + \beta g(y) \forall \alpha, \beta \in \mathbb{R}$ and $x, y \in \mathbb{R}^n$. That is, $Df(x)$ being linear means that $Df(x) \cdot (\alpha v + \beta w) = \alpha Df(x) \cdot v + \beta Df(x) \cdot w$ if $\alpha, \beta \in \mathbb{R}$ and $v, w \in \mathbb{R}^n$.

The second derivative is *bilinear*, just meaning that $Df(x) \cdot (v, w)$ is linear with respect to both v and w .

Examples

1. $(x, y) \in \mathbb{R}^2$, $f(x, y) = \frac{1}{2}(x^2 + y^2)$. Let \mathbb{R}^2 have the inner product $\langle x, y \rangle = x^T y$. Then $\nabla f(0) = (x, y)^T|_{x=0, y=0} = (0, 0)^T$ and

$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. H has two eigenvalues, both are 1, so it is positive definite and $(0, 0)$ is a minimizer.

2. $x \in \mathbb{R}^1$, $f(x) = \frac{1}{4}x^4$. $Df(0) \cdot v = x^3 \cdot v|_{x=0} = 0$ and $D^2f(x) \cdot (v, v) = 3x^2v_{x=0}^2 = 0$, so x^* is clearly an extremizer, but the second derivative tells us nothing (although it is a minimizer).

Ordinary Differential Equations

Ordinary differential equations (ODEs) are probably the single most common representation of physical systems that change over time. For instance, you are probably familiar with the statement that $\frac{d^2}{dt^2}x = \ddot{x} = -g$, a description of a particle falling in gravity (where we use the convention that $g > 0$ and ‘up’ is the positive direction of x). This is a second-order ODE because it involves two derivatives with respect to time. This is *not* a partial differential equation (PDE) because derivatives are only being taken with respect to a single independent variable (in this case t). For instance, the equation $\frac{\partial C(x)}{\partial t} = D \frac{\partial^2 C(x)}{\partial x^2}$ (called the diffusion equation or heat equation, describing how heat diffuses through a homogenous medium); this equation is a partial differential equation because there are derivatives *both* with respect to t and with respect to x .

In this class, we will use Python to simulate dynamic systems—simulate here means numerically integrate the equations of motion forward in time. More on numerical integration is coming in the next section, but, for now, it is important to point out a limitation of nearly every numerical solver available for ODEs. *Numerical solvers only work for first-order differential equations.* This means that they cannot even provide us with the position of the particle in gravity when we give them the equation $\ddot{x} = -g$. Luckily there is a very simple trick that we can use to make every differential equation first order. We make x a vector \mathbf{x} by including the derivatives of x in the vector. So, the ODE for the particle in gravity becomes:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -g \end{bmatrix} \text{ with } \mathbf{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}.$$

The ODE is now first-order because it involves only one derivative of \mathbf{x} with respect to time.

Numerical Integration of ODEs

Numerical integration is the computation strategy we use to take an ordinary differential equation and generate predictions based on a current state. This process is often called *time stepping* because it generates a prediction in the future based on taking steps from some initial condition.

The simplest numerical integration scheme is *Euler integration*. In it, we assume that we have a differential equation $\dot{x}(t) = f(x(t))$ (where I am including t explicitly to make understanding this a bit easier). On the left hand side we see the derivative of $x(t)$ with re-

spect to t , which *by definition* is equal to

$$\dot{x}(t) := \lim_{dt \rightarrow 0} \frac{x(t + dt) - x(t)}{dt} = f(x(t)) \quad (1)$$

(where $:=$ is the notation used to say that the left hand side is *defined* to be the right hand side). This says that “the derivative of x is equal to the limit as dt goes to zero of $x(t + dt)$ minus $x(t)$ divided by dt ”. So far I haven’t changed anything—this is still a differential equation and it is still equal to $f(x(t))$.

However, if I *do not* take the limit, and instead simply say that dt is some rather small number—where I’m not going to say exactly what I mean by “small”—we get an expression that only *approximates* \dot{x} and is therefore *approximately* equal to $f(x(t))$.

$$\frac{x(t + dt) - x(t)}{dt} \approx f(x(t)) \quad (2)$$

If I rearrange this approximate equation, I get that $x(t + dt)$ is approximately equal to $x(t)$ plus dt times $f(x(t))$.

$$x(t + dt) \approx x(t) + dt f(x(t))$$

This formula, called *Euler integration*, means that if we know x at some time t , we can approximate x at $t + dt$. This also means that you should expect to need to know x at *some* time in order to get the algorithm started—knowing the value of x at a particular time is called a *boundary condition*, typically an *initial condition*.

Euler integration can be applied to any first-order ODE; this is partly why it is so important to be able to convert a second-order ODE into a first-order ODE (with more states). In the case of a linear ODE with a vector state, such as

$$\dot{w} = Aw$$

one can use Euler integration directly on the state w . The initial condition for the differential equation needs to include initial conditions of all the states. For instance, we might have

$$w_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}.$$

Euler integration applied to this ODE—expressed in terms of vectors and matrices—will look like

$$w(t + dt) = w(t) + dt Aw(t).$$

The great thing about this formula is that once you have the matrix A , the code for your Euler integration will always look the same! As

an example, let's look at the vector and matrix notation for a spring-mass system (a object of mass m attached to a spring with spring constant k that is attached to a wall). You probably already know that this system has a second-order differential equation

$$\ddot{x} = -\frac{k}{m}x.$$

If we rewrite that equation in first order form we get two first-order differential equations with a state of position x and velocity v (here $v = \dot{x}$, but we give it a new name to avoid confusion in our code).

$$\begin{aligned}\dot{x} &= v \\ \dot{v} &= -\frac{k}{m}x\end{aligned}$$

If we set $w = \begin{bmatrix} x \\ v \end{bmatrix}$, and choose $A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix}$, then Aw gives us the right hand side of the differential equation.

Let's say that $x_0 = 0.5$ and $v_0 = 0$ and we want to use this vector notation in Euler integration with a time step of $dt = 0.1$. We set w_0 to be the vector of intial conditions of the two states and then

$$w_0 = \begin{bmatrix} x_0 \\ v_0 \end{bmatrix} \Rightarrow w(0.1) \approx \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} + \underbrace{0.1}_{dt} \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -\frac{k}{20m} \end{bmatrix}$$

If I multiply this out, I get an approximation of $w(0.1)$ —both the position and velocity of the mass at time $t = 0.1$.

Why don't we always use Euler integration, given that it is so easy to implement? Euler integration is not very *accurate*, and as dt becomes small it converges to the correct solution slowly. Instead, there are what are called *higher order* schemes, where the word 'order' tells you how fast the scheme converges as you make dt smaller and smaller. Two such schemes are Runge-Kutta (RK) schemes. The second-order RK (RK2) scheme is given by

$$x(t + dt) \approx x(t) + dt f\left(x(t) + \frac{dt}{2} f(x(t))\right).$$

This scheme is much more stable than Euler, as you will see in the python code below. The four-order RK scheme (RK4) is given by

$$x(t + dt) \approx x(t) + \frac{1}{6}(k_1 + k_2 + k_3 + k_4)$$

where

$$\begin{aligned}k_1 &= dt f(x(t)) \\k_2 &= dt f\left(x(t) + \frac{k_1}{2}\right) \\k_3 &= dt f\left(x(t) + \frac{k_2}{2}\right) \\k_4 &= dt f(x(t) + k_3).\end{aligned}$$

This (rather mysterious) formula provides an excellent integrator. You can see how to implement it below in python, and you will see later that often this will work as well as the built-in integrators in python when simulating mechanical systems.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def dynvec(x):
5     xret=np.dot(A,x)
6     return xret
7
8 def euler(f,x):
9     xnew=x+f(x)*dt
10    return xnew
11
12 def rk2(f,x):
13     xnew=x+dt*f(x+dt/2.*f(x))
14     return xnew
15
16 def rk4(f,x):
17     k1=dt*f(x)
18     k2=dt*f(x+k1/2.)
19     k3=dt*f(x+k2/2.)
20     k4=dt*f(x+k3)
21     xnew=x+(1/6.)*(k1+2*k2+2*k3+k4) #be sure to use float not int. The
22     period after the 6 matters!
23     return xnew
24
25 x0=np.array([1.0,0.0])
26 dt=0.5
27 T=100
28 N=int(T/dt)
29 print(int(N))
30 xarray=np.zeros((2,N))
31 xarray[:,0]=x0
32 t=np.arange(0,T,dt)
33 A=np.array([[0,1],[-1,-0.01]]) #very light damping to demonstrate Euler
34     instability
35
36 #replace rk4 with rk2func and eulerfunc and decide which scheme is most
37     physically plausible
38 i=0
39 while i<=N-2:
40     xarray[:,i+1]=rk4(dynvec,xarray[:,i])
41     i+=1

```

```

39 #print(xarray)
40 plot1=plt.plot(t,xarray[0,:N],t,xarray[1,:N])
41 plt.show(plot1)
42

```

Listing 8: Numerical integration routines in NumPy

Useful Properties of Linear ODEs

Although few mechanical systems are linear, it is helpful to remember a couple of key facts about linear ordinary differential equations.

1. *Linear, constant coefficient, first-order ODEs have exponential solutions.* You can think of this in terms of the ODE

$$\dot{z} = Az,$$

where z is the $n \times 1$ vector of state variables and A is an $n \times n$ matrix. This equation is first-order, and if all the elements of A are numbers—that is, not functions of time—then what you can conclude is that

$$z(t) = e^{rt}v.$$

However, r and v are unknown.

2. *Eigenvectors and eigenvalues are an important part of solving linear, constant-coefficient ODEs.* Given a system $\dot{z} = Az$ and solution $z(t) = e^{rt}v$, we know that $e^{rt}v = Ae^{rt}v$, so $re^{rt}v = Ae^{rt}v$. Dividing both sides by the nonzero e^{rt} gives us

$$Av = rv,$$

which is exactly the eigenvalue/eigenvector relationship for the matrix A . This means that if $z(t) = e^{rt}v$, the unknown r and v can be determined by solving for eigenpairs of A .

3. *Exponentials of complex numbers are `sin()` and `cos()`.* First we need to see the relationship between oscillation and the exponential functions. How do we get an exponential function like e^{ht} to give us $\cos()$ and $\sin()$? The idea is really simple and hinges on the Taylor series. We are going to Taylor expand e^{ht} and $\sin(\omega t)$ and $\cos(\omega t)$.

$$\begin{aligned}
 e^{ht} &= 1 + ht + \frac{1}{2!}h^2t^2 + \frac{1}{3!}h^3t^3 + \frac{1}{4!}h^4t^4 + \frac{1}{5!}h^5t^5 + \frac{1}{6!}h^6t^6 \dots \\
 \sin(\omega t) &= +\omega t - \frac{1}{3!}\omega^3t^3 + \frac{1}{5!}\omega^5t^5 \dots \\
 \cos(\omega t) &= 1 - \frac{1}{2!}\omega^2t^2 + \frac{1}{4!}\omega^4t^4 - \frac{1}{6!}\omega^6t^6 \dots
 \end{aligned}$$

The thing to note here is that the $\sin(\omega t)$ term has every other term—every odd power of t —of the exponential, and the $\cos(\omega t)$ terms have the left over terms—all the even powers of t . Moreover,

the signs change—sometimes they are plus signs and sometimes they are negative signs. If you think of this as being a requirement on h , it would imply that

$$h^0 = 1, h^1 = \omega, h^2 = -\omega^2, h^3 = -\omega^3, h^4 = \omega^4 \dots,$$

while keeping even powers separate from odd powers.

What function of ω could force these relationships to hold? Amazingly, the choice of

$$h = j\omega \quad \text{with} \quad j = \sqrt{-1}$$

accomplishes all of this! Plug $h = j\omega$ into the exponential and see what happens. We get the exponential with powers of j , ω , and t in the Taylor expansion.

$$e^{j\omega t} = 1 + j\omega t + \frac{1}{2!}(j\omega)^2 t^2 + \frac{1}{3!}(j\omega)^3 t^3 + \frac{1}{4!}(j\omega)^4 t^4 + \frac{1}{5!}(j\omega)^5 t^5 + \frac{1}{6!}(j\omega)^6 t^6 \dots$$

We know that $j = \sqrt{-1}$, so $j^2 = -1$, $j^3 = -j$, and $j^4 = 1$. This gives us a bunch of terms that alternate sign and half of them include a j .

$$e^{j\omega t} = 1 + j\omega t + \frac{1}{2!}(-1)\omega^2 t^2 + \frac{1}{3!}(-j)\omega^3 t^3 + \frac{1}{4!}(1)\omega^4 t^4 + \frac{1}{5!}(j)\omega^5 t^5 + \frac{1}{6!}(-1)\omega^6 t^6 \dots$$

If we split these into the terms that have j and the terms that don't have j ,

$$e^{j\omega t} = \left(1 - \frac{1}{2!}\omega^2 t^2 + \frac{1}{4!}\omega^4 t^4 - \frac{1}{6!}\omega^6 t^6 + \dots\right) + j \left(\omega t - \frac{1}{3!}\omega^3 t^3 + \frac{1}{5!}\omega^5 t^5 + \dots\right)$$

we get exactly the Taylor expansion of $\cos()$ and $\sin()$. This formula $e^{j\omega t} = \cos(\omega t) + j \sin(\omega t)$ is called *Euler's formula* and you will use it many times while taking this course. Sometimes it will have a \pm symbol, since complex numbers often come in *complex conjugate pairs*, and specifically in the case of eigenvalues/eigen-vectors always will. This version of Euler's formula is:

$$e^{\pm j\omega t} = \cos(\omega t) \pm j \sin(\omega t).$$

4. *Linear ODEs satisfy superposition.* Superposition is just a fancy way of saying that if we have one solution $x_1(t)$ and a second solution $x_2(t)$, then adding these solutions and/or scaling them will also be a solution.²⁶ This important property plays a pivotal role next.
5. *Complex eigenvalues/eigenvectors imply oscillation.* Assume that the matrix A has a pair of complex eigenvalues, so that $r = \pm j$.

²⁶That is, solutions to a particular linear ODE form a vector space because they can be scaled and added together and still be a solution.

Then by superposition we can add and subtract solutions in terms of Euler's equations. Note that if $x_1 = e^{j\omega t} = \cos(\omega t) + j \sin(\omega t)$ and $x_2 = e^{-j\omega t} = \cos(\omega t) - j \sin(\omega t)$, then $\frac{x_1+x_2}{2} = \cos(\omega t)$ and $\frac{x_1-x_2}{2j} = \sin(\omega t)$. That is, by applying superposition to exponential functions, we can get rid of the dependence on j and replace it with dependence on $\cos()$ and $\sin()$.²⁷

Discussion of Complex Numbers

What is a complex number?

1. a number that is of the form of $a + jb$ where $j = \sqrt{-1}$;
2. the number that makes $\sin(\omega t)$ and $\cos(\omega t)$ exponential functions.²⁸

Both of these are correct, and the problem is that most people think of the first one rather than the second one, whereas the first is a *consequence* of the second. To get a feel for this, let's look at a particular example of planar rotation of the vector $x = [1, 0]$ by $\frac{\pi}{4}$ (45°). This rotation is represented by a rotation matrix:

$$A = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

and the rotation of x is achieved through matrix multiplication Ax . How can we see that A is a rotation from its eigenvalues and eigenvectors? The eigenvalues and eigenvectors of A are:

$$\lambda_1 = \frac{\sqrt{2}}{2}(1 - j), v_1 = \begin{bmatrix} -j \\ 1 \end{bmatrix}$$

$$\lambda_2 = \frac{\sqrt{2}}{2}(1 + j), v_2 = \begin{bmatrix} +j \\ 1 \end{bmatrix}$$

By the definition of the eigenvalue-eigenvector pairs, Ax can be rewritten in terms of the eigenvectors v_i , using the fact that

$$Av_i = \lambda_i v_i.$$

Specifically, we know that

$$Av_1 = \lambda_1 v_1$$

and

$$Av_2 = \lambda_2 v_2$$

and that²⁹

²⁷ There is no greater injustice in mathematics than calling numbers involving $j = \sqrt{-1}$ *imaginary*. They are in no way imaginary, and everyone would be very well served by simply thinking of them as indicating oscillation or rotation is involved.

²⁸ If sin and cos were *not* exponential functions, think of how hard it would be to solve ordinary differential equations!

²⁹ I recommend students check this!

$$x = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{v_1 - v_2}{-2j}$$

allowing us to conclude that

$$Ax = A \left(\frac{v_1 - v_2}{-2j} \right) = \frac{\lambda_1 v_1 - \lambda_2 v_2}{-2j}.$$

We already know that A is a rotation, so somehow these complex numbers need to all lead to a real-valued vector that is a rotation from $x = [1, 0]$. To see that this is, in fact, the case, look at Fig. 1.

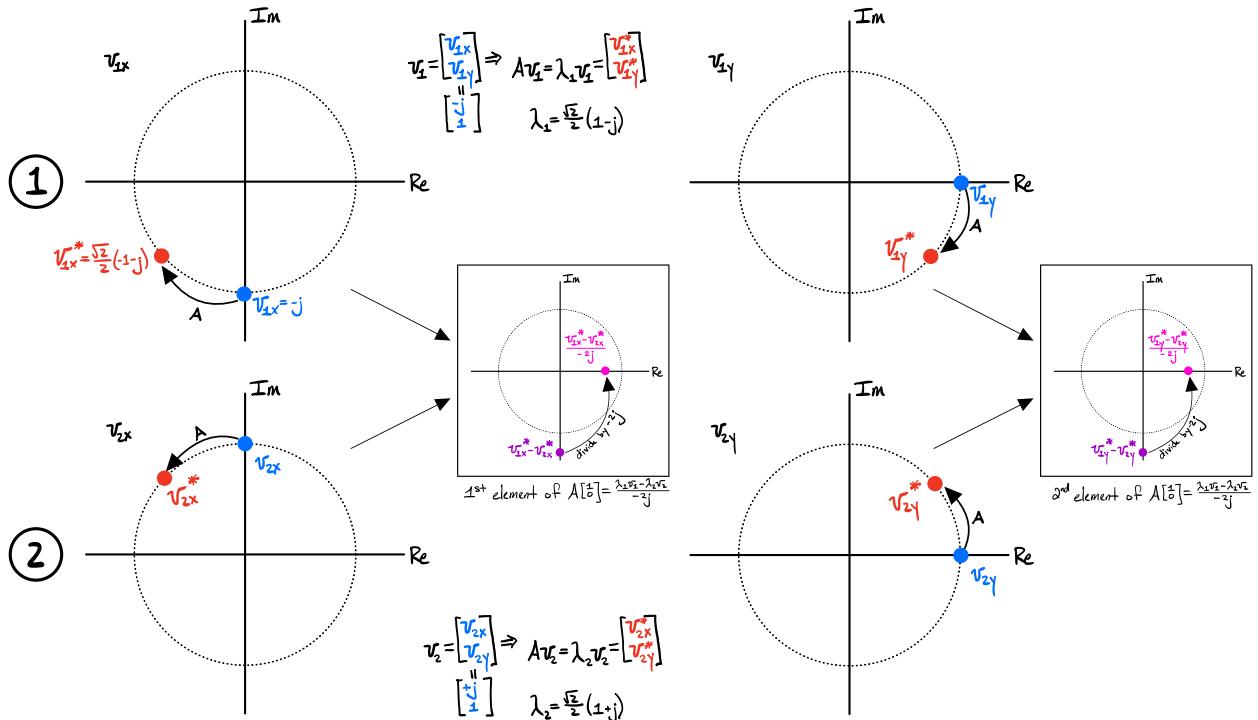


Fig. 1 shows the geometry of how A and its eigenvalues act on v_1 and v_2 , as well as how they then combine back together³⁰ to give the real-valued vector that is the rotated vector Ax . This figure goes through component by component of v_1 and v_2 , e.g., showing how $\lambda_1 = \frac{\sqrt{2}}{2}(1 - j)$ affects the first component v_{1x} of v_1 .

Figure 1: Complex eigenvalues imply rotation.

³⁰ in the formula $Ax = \frac{\lambda_1 v_1 - \lambda_2 v_2}{-2j} \dots$

Other Useful Facts

Lastly, here are some facts worth knowing:

1. *Exponents of Exponentials:* $e^{a+b} = e^a e^b$
2. *The Quadratic Formula:* If we have three coefficients α , β , and γ , then

$$\alpha h^2 + \beta h + \gamma = 0$$

implies that

$$h = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}.$$

Symbolic and Numerical Computation in Python

In this class you will use the programming language Python for both symbolic and numerical computations, the computations depend on the SymPy and NumPy packages respectively, and here are some notes about using these two packages:

1. Python (especially in NumPy) supports built-in data type conversion, which means the Python will decide the data type of each variable based your code. This is a great feature in most of the time, but things could go wrong. When you're doing numerical computation, it's highly recommended to explicitly define or initialize your variables (no matter it's just a single number or an array) as float numbers. In the examples we provide in these notes, we try to always define variables as float numbers.
2. SymPy and NumPy have some built-in "constants" and "functions" that have the same name but cannot be mixed. The most common error is from "pi" — you can import it from both SymPy (`from sympy import pi`) and NumPy (`from numpy import pi`). But "pi" in SymPy is a symbolic variable and in NumPy is a numerical value. Similarly, the functions "sin()" and "cos()" cannot be mixed between the two packages.

Dynamics and Calculus

THE STUDY OF MECHANICAL OBJECTS evolving in time (e.g., ballistic motion, planetary motion, vehicles, robots, et cetera) essentially comes down to the study of calculus. Even in Newton's laws, $F = ma$ requires one to be happy with the idea of taking the position $x(t)$ and differentiating it twice to obtain acceleration $a(t)$. These notes are about the application of calculus to problems in predicting mechanical behavior. This means that the student will need to know some calculus.³¹

Fortunately, one does not need *much* calculus to survive dynamics, but the calculus you do need is critical. In particular, there are five things from calculus you have to know:

- **Directional derivatives:**

$$Df(x) \cdot v = \frac{d}{d\epsilon} f(x + \epsilon v)|_{\epsilon=0},$$

- **Chain rule:**

$$\text{if } h(x) = f(g(x)), \text{ then } Dh(x) \cdot v = Df(g(x)) \cdot Dg(x) \cdot v,$$

- **Product rule:**

$$\text{if } h(x) = f(x)g(x), \text{ then } Dh(x) \cdot v = [Df(x)g(x) + f(x)Dg(x)] \cdot v,$$

- **Leibniz rule (the Fundamental Theorem of Calculus):**

$$\frac{d}{dt} \int_0^t f(\tau) d\tau = f(t),$$

- **Integration by parts:**

$$\int_{t_0}^{t_f} u dv = uv|_{t_0}^{t_f} - \int_{t_0}^{t_f} du v. \quad ^{32}$$

You will use these over and over and over, so trying to avoid them or simply survive them is pointless.³³ For a broad overview of these topics, please see the previous chapter on mathematical background.

One of the problems with most undergraduate exposure to calculus is that these main tools get lost in a sea of particular examples.³⁴ But the *value* of calculus has little to do with the evaluation of calculus operations on particular functions; examples serve to improve understanding but rarely represent anything of importance in a real

³¹ One could argue that this class is essentially an advanced calculus class; regardless of whether that is your view, I take the approach in this class that "learning dynamics" is only a peripheral goal, and may not be that important in its own right (e.g., with all the software tools available for simulating systems, it is not clear what role traditional dynamics has to play for a practicing engineer). Dynamics is an exceptionally concrete setting for understanding tools from calculus that play a role in simulation, machine learning, reinforcement learning, optimal control, and other areas. Moreover, numerical methods discussed in this class will give engineers a healthy dose of skepticism about how well computers can make predictions.

³² Note, as discussed earlier, that we keep the multiplication order in case these are vectors or matrices and therefore may not commute.

³³ A little linear algebra will come in handy as well, but it really is minimal and ultimately comes down to knowing how to multiply a matrix and a vector and a vector and a vector (e.g., the "dot" product).

³⁴ such as $\frac{d}{dx} x^3 = 3x^2$ and $\int \cos(x) dx = \sin(x)...$

application. Instead, almost everything in modern engineering depends on numerical simulation, and the same tools from calculus can play a very important role in setting up concrete calculations—in the case of this class, calculations for making predictions about mechanical behavior over time.

To get us started, and to remind ourselves of our initial exposures to differential calculus, how does one differentiate these three different functions?

$$\begin{aligned}f(x) &= \frac{1}{2}x^2 \quad x \in \mathbb{R} \\f(x) &= \frac{1}{2}x^T x \quad x \in \mathbb{R}^n \\f(x) &= \frac{1}{2} \int_{t_0}^{t_f} x^T x dt \quad x \in \mathcal{L}(\mathbb{R}^n)\end{aligned}$$

The first one probably feels familiar enough, but even the second one is probably a little alien. Most of us do not know how to differentiate the third one from our undergraduate experience with calculus. Nevertheless, the idea of differentiation is the same in each of the three cases, and our goal is to recognize all three as being essentially the same problem.

As you can see below, SymPy does a pretty good job in each one of these cases (be sure to inspect this code to see how it relates to what is above!).

```

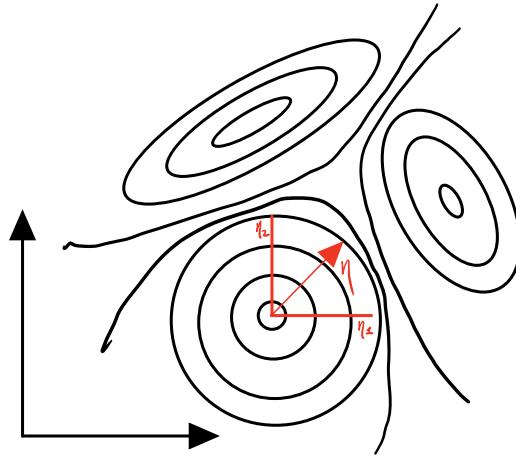
1 import sympy as sym
2 from sympy.abc import x, y, z, t, s
3 from sympy import Matrix
4 from sympy import Function, Symbol
5 from sympy import diff, integrate
6 t=Symbol('t')
7 s=Symbol('s')
8 f=Function('f')(s)
9 g=Function('g')(s)
10
11 # define a vector
12 w = Matrix([ x, y, z])
13 #define a function X of the vector w
14 X=w.T*w
15 diffX=(w.T*w).jacobian(w)
16 print('The derivative of X is',diffX)
17
18 #define a function Y of the function f
19 Y=(f**2)
20 diffY=Y.diff(s)
21 print('The derivative of Y is',diffY)
22
23 #define a function Z of the integral of a function and see if sympy applies
24     Leibniz rule
24 Z=integrate(f**2,[s,0,t])
25 diffZ=Z.diff(t)
26 print('The derivative of Z is',diffZ)
27

```

```

28 #define a function W of the integral of two functions and differentiate with
  respect to one of the functions
29 W=integrate(f*g,[s,0,1])
30 diffW=W.diff(f)
31 print('The derivative of W is',diffW)

```



To help with the task of making differentiation a little more manageable, the definition of differentiation we use is the **Gâteaux derivative** (also called the **directional derivative**):

$$\frac{\partial f(x)}{\partial x} \cdot v = \frac{d}{d\epsilon} f(x + \epsilon v)|_{\epsilon=0}. \quad (3)$$

It is worth taking a moment to think about what the Gâteaux derivative requires. First note that ϵ is a scalar parameter. It does not depend on time or on any spatial dimension. As a result, it is relatively easy to procedurally calculate a directional derivative (and this is why we use them everywhere in engineering analysis instead of the more-mathematically-correct-but-less-manageable Fréchet derivative). The second thing to note is that x and v do not need to be scalars—they can be basically anything. Both x and v can be scalars, vectors, matrices, curves, et cetera. *But* one must be able to *add* x and v for the Gâteaux derivative to make sense operationally (that is, that if I gave you a function of x and a direction v , would you be able to calculate the directional derivative). As a result, if x is a vector, v must be a vector; if x is a curve, v must be a curve.

In general, for “nice” functions differentiability and directional derivatives are equivalent.³⁵ More importantly, Gâteaux derivatives give us a *practical* way to differentiate things that are unintuitive. Going back to the three examples mentioned earlier, we can now *evaluate* the derivatives even if they are unintuitive.

1. $f(x) = \frac{1}{2}x^2 \quad x \in \mathbb{R}$

Figure 2: The directional derivative evaluates the derivative of a function at x in a particular direction η . The total number of possible directions η is the same dimension as the dimension of x , so if x is a function of an independent variable like time, η will also be so.

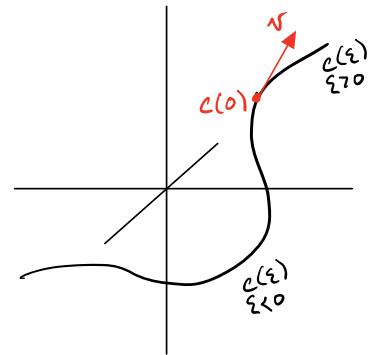


Figure 3: To see that Eq. (3) makes sense, imagine evaluating f along some differentiable curve $c(\epsilon) : \mathbb{R} \rightarrow \mathbb{R}^n$ that satisfies $c(0) = x$ and $c'(0) = v$; $c(\epsilon) = x + \epsilon v$ is one such curve. Then by chain rule we know that $\frac{d}{d\epsilon} f(c(\epsilon)) = Df(c(\epsilon)) \cdot c'(\epsilon)$, which implies $\frac{d}{d\epsilon} f(c(\epsilon))|_{\epsilon=0} = Df(x) \cdot v$. The geometric interpretation of this is that the directional derivative is the derivative of f along *all possible* differentiable curves $c(\epsilon)$.

³⁵ From vector calculus, we know that there is something wrong with this notion of differentiability. Sometimes it implies a mapping is differentiable when it is not. For instance, $f(x, y) = \frac{2xy}{x^2+y^2}$ with $f(0, 0) = 0$ satisfies

$$\lim_{x=y \rightarrow 0} f(x, y) = \frac{2x^2}{2x^2} = 1$$

and

$$\lim_{x=0, y \rightarrow 0} f(x, y) = 0$$

which means that the function is not differentiable at $(0, 0)$, but the directional derivatives *do* exist.

$$\Rightarrow \frac{\partial f}{\partial x} \cdot v = \frac{d}{d\epsilon} (f(x + \epsilon v)|_{\epsilon=0}) = \frac{d}{d\epsilon} \frac{1}{2} (x + \epsilon v)^2|_{\epsilon=0} = (x + \epsilon v) \cdot v = xv$$

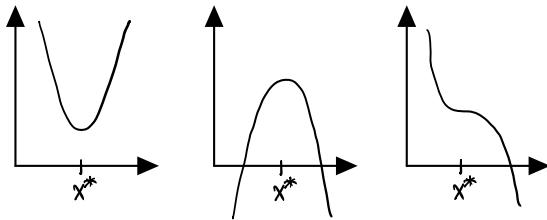
2. $f(x) = \frac{1}{2} x^T x \quad x \in \mathbb{R}^n$

$$\Rightarrow \frac{\partial f}{\partial x} \cdot v = \frac{d}{d\epsilon} (f(x + \epsilon v)|_{\epsilon=0}) = \frac{d}{d\epsilon} \frac{1}{2} (x + \epsilon v)^T (x + \epsilon v)|_{\epsilon=0} = (x + \epsilon v)^T \cdot v = x^T v$$

3. $f(x) = \frac{1}{2} \int_{t_0}^{t_f} x^T x dt \quad x \in \mathcal{L}(\mathbb{R}^n)$ ³⁶

$$\Rightarrow \frac{\partial f}{\partial x} \cdot v = \frac{d}{d\epsilon} (f(x + \epsilon v)|_{\epsilon=0}) = \frac{d}{d\epsilon} \frac{1}{2} \int_{t_0}^{t_f} (x + \epsilon v)^T (x + \epsilon v) dt|_{\epsilon=0} = \int_{t_0}^{t_f} x^T v dt$$

If we were extremizing these functions, we would want to “set the derivative to zero” which means that the directional derivative should be zero for all possible perturbations v . This implies that the extrema occur at $x = 0 \in \mathbb{R}$, $x = 0 \in \mathbb{R}^n$ and $x = 0(t) \in \mathcal{L}(\mathbb{R}^n)$. This last idea will play prominently in our development of the *least action principle*. Consider the plot below—if x^* were the point of interest in each case, how would you give me a principle by which one could identify x^* ?



³⁶ The notation $\mathcal{L}(\mathbb{R}^n)$ should be thought of as “the vector space of curves that have well defined integrals”.

Figure 4: If x^* is the point we are looking for in each of these three cases, what principle can we use to implicitly define x^* ?

An Example of Variational Principles at Work

Suppose we have a particle in gravity of mass m with its position measured by the variable x (where the positive direction is up), seen in Fig. 5. Then Newton's equations (force balance) say

$$\begin{aligned} F &= ma \\ \Rightarrow -mg &= ma \\ \Rightarrow -mg &= m\ddot{x} \\ \Rightarrow \ddot{x} &= -g \end{aligned}$$

which is a differential equation in x . Note that this means—rightly so—that x is dependent on time t (that is, x is allowed to change over time), and that as a result we often write $x(t)$ to indicate that x can change in time. To compute $x(t)$ using the differential equation $\ddot{x} = -g$, one additionally needs *boundary conditions*, typically the *initial position* $x(t_0)$ and *initial velocity* $\dot{x}(t_0)$.

In this case, force balance is very easy—and looks a lot like the force balance systems we have evaluated before, such as the block on an incline plane seen in Fig. 6—but we know that force balance can be quite difficult if the system becomes complex. Moreover, questions that we might answer with force balance for the block on the incline plane are often questions of statics—will the block start slipping, etc. We are not interested in statics but are instead interested in dynamics—*how* will the block move when it slips and *when* will it arrive at the bottom of the incline? Dynamic analysis of this type can be evaluated using Newton's laws, which are reasonably simple for the particle in gravity (and the inclined block). Dynamic analysis can also be done using a *variational principle*—a fancy name for taking the derivative of something and setting it equal to zero—which will be harder for the particular systems of a particle in gravity or an inclined block, but easier in general for more complex systems. So help gain some intuition, we will spend the rest of this section deriving $\ddot{x} = -g$ as a consequence of a variational principle. In the next section we will see that insights into the particle in gravity generalize extraordinarily well.

Applying a Variational Principle to the Particle in Gravity

There quantity

$$A(x(t)) = \int_{t_0}^{t_f} KE(x(t)) - V(x(t)) dt$$

is called the *action* or *action integral*. The variables KE and V are kinetic energy³⁷ and potential energy³⁸ respectively. Note that the

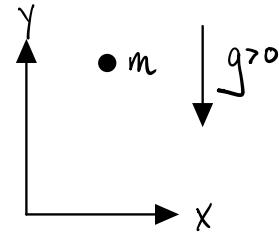


Figure 5: A particle in gravity x units above the ground in gravity with gravity constant $g > 0$.

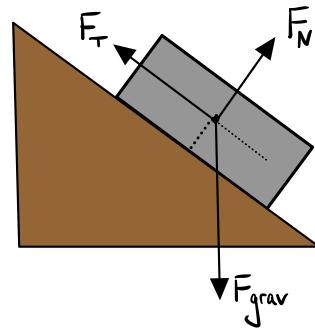


Figure 6: Free body diagram for a block on an inclined plane with gravity constant $g > 0$.

³⁷ The sum of all the energy from each particle in the system, $\frac{1}{2}m||v||^2$ for each particle, where v is the velocity of the particle in \mathbb{R}^3 .

³⁸ How much energy is stored due to the configuration of the particles—e.g., the compression of spring or height in a gravitational field.

action integral is the integral of the difference between the two, not the sum. In the case of the particle in Fig. 5, we would have $KE(x(t)) = \frac{1}{2}mv^2 = \frac{1}{2}m\dot{x}^2$ and $V = mgh = mgx$ (assuming $g > 0$). Note that even though $x(t)$ is a *curve*, $A(x(t))$ is a scalar— KE and V are both scalars, and the time integral of a scalar is a scalar. Lastly, again note that $A(x(t))$ is *not* the total energy (that would be $KE(x(t)) + V(x(t))$ and there would be no integral), so even though this definition involves familiar concepts, it is probably not itself familiar.

The *variational principle*—sometimes called the *action principle*—states that the trajectory of a particle (in this case a trajectory that is only allowed to move up and down with the x coordinate measuring its motion) is the one that *extremizes the action integral with respect to perturbations to the trajectory*.³⁹ We know that functions (in this case $A(x(t))$) are extremized when their derivatives are “equal to zero”, so the variational principle is often written using the δ to signify differentiation of a function (in this case A) with respect to another function (in this case $x(t)$):

$$\delta A = 0$$

which should be read as (roughly) “the derivative of the action integral A is zero”.⁴⁰

However, it might be more than a little unclear what δA means precisely; instead we say that the directional derivative has to be zero for all possible directions $\eta(t)$. This means that $\eta(t)$ is a mathematical perturbation to $x(t)$.⁴¹ That is, with the formalism of the directional derivative, we now state a substantially more *procedurally* precise notion of the variational principle:

$$\underbrace{\frac{\partial A(x(t))}{\partial x(t)} \cdot \eta(t)}_{DA(x(t)) \cdot \eta(t)} = \frac{d}{d\epsilon} A(x(t) + \epsilon\eta(t))|_{\epsilon=0} = 0 \quad \forall \eta(t).$$

Note that since $x(t)$ and $\eta(t)$ have to add together, they must be of the same dimension— $\eta(t)$ must be a curve, just like $x(t)$.⁴²

To see how this will create a principle that yields the same differential equation we found using force balance, we now take the derivative of A with respect to $x(t)$ in the direction of $\eta(t)$ and set the result equal to zero.

³⁹ It is technically better to say extremize, but this almost always means minimize.

⁴⁰ We will actually have to amend the principle slightly, as we will see.

⁴¹ **Very Important:** $\eta(t)$ is most emphatically *not* an actual physical perturbation—it has no physical interpretation at all.

⁴² This can cause confusion, since sometimes it seems like we add things of different dimension. For instance, a student might reasonably say that $\sin(t) + 0 = \sin(t)$. But what is meant by 0 in this case is $0(t)$ —a signal that is identically zero at all times t .

$$\begin{aligned}
A(x(t)) &= \int_{t_0}^{t_f} KE(x(t)) - V(x(t)) dt \\
&= \int_{t_0}^{t_f} \frac{1}{2} m \dot{x}^2 - mgx dt \\
\Rightarrow A(x(t) + \epsilon \eta(t)) &= \int_{t_0}^{t_f} \frac{1}{2} m(\dot{x} + \epsilon \dot{\eta})^2 - mg(x + \epsilon \eta) dt \\
\Rightarrow \frac{d}{d\epsilon} A(x(t) + \epsilon \eta(t)) &= \int_{t_0}^{t_f} m(\dot{x} + \epsilon \dot{\eta}) \dot{\eta} - mg \eta dt \\
\Rightarrow \frac{d}{d\epsilon} A(x(t) + \epsilon \eta(t))|_{\epsilon=0} &= \int_{t_0}^{t_f} m \dot{x} \dot{\eta} - mg \eta dt
\end{aligned}$$

Now, we want to be able to factor the perturbation $\eta(t)$ out so that we can determine what expression must be equal to zero in order for the action integral to be extremized, but the first term $m \dot{x} \dot{\eta}$ has an $\dot{\eta}$ instead of η . Hence, we need to rewrite the term

$$\int_{t_0}^{t_f} m \dot{x} \dot{\eta} dt$$

so that there is only an η on the right hand side under the integral. To do so, we need *integration by parts*.⁴³ Applying integration by parts yields

$$\int_{t_0}^{t_f} m \dot{x} \dot{\eta} dt = m \dot{x} \eta|_{t_0}^{t_f} - \int_{t_0}^{t_f} m \ddot{x} \eta dt.$$

Adding the term due to potential energy back in we get

$$m \dot{x} \eta|_{t_0}^{t_f} - \int_{t_0}^{t_f} m \dot{x} \eta dt - \int_{t_0}^{t_f} mg \eta dt = \underbrace{m \dot{x} \eta|_{t_0}^{t_f}}_{\text{problem!!!}} - \int_{t_0}^{t_f} \underbrace{[m \ddot{x} + mg]}_{\text{exciting}} \eta dt$$

and one notices that the terms in the square brackets look a lot like the terms that we saw showing up from force balance!

Just to regroup, we now know that

$$\frac{d}{d\epsilon} A(x(t) + \epsilon \eta(t))|_{\epsilon=0} = \underbrace{m \dot{x} \eta|_{t_0}^{t_f}}_{\text{problem!!!}} - \int_{t_0}^{t_f} \underbrace{[m \ddot{x} + mg]}_{\text{exciting}} \eta dt.$$

Moreover, we would like to set that right hand side expression equal to zero and conclude that $\ddot{x} = -g$, since that's what we expect from evaluating Newton's laws. This means that we need to get rid of the "problem" terms somehow. However, if $\eta(t_0)$ and $\eta(t_f)$ happened to be zero, that term vanishes entirely (regardless of what \dot{x} is at t_0 and t_f). This is indeed what we do.⁴⁴

With the above in mind, we are now motivated to modify the action principle as:

⁴³ Again, that

$$\int_a^b u dv = uv|_a^b - \int_a^b du v$$

where I have preserved the order of multiplication in case u and v happen to be vectors.

⁴⁴ **WHAT!?! We cannot just run around setting terms equal to zero for no reason!** Or at least so says any student paying attention at this point! But here is where some physical intuition can be helpful. First, consider $\eta(t_0)$: it is a perturbation to the configuration x at its initial condition, but the question we are trying to answer is how it *evolves* from its initial condition as time progresses—the initial condition is part of the problem specification. This, in my mind, justifies disallowing perturbations at time t_0 . Disallowing perturbations at time t_f is more subtle, but since Newtonian systems are *reversible*, going backwards in time also is well-posed and one may answer the question "if the configuration is x at t_f , how did it get there?". In that case, the specification assumes that the configuration at time t_f is specified. What is *not* specified is how we get between the two.

The *action principle* states that the trajectory of the particle is the one that extremizes the action with respect to perturbations in the trajectory that are zero at t_0 and t_f .

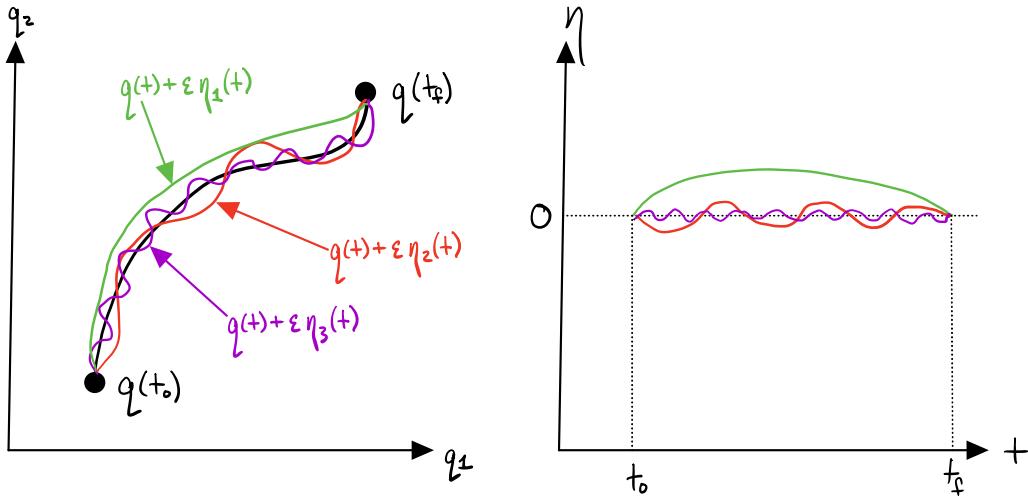
This updates the above equation so that we have

$$\delta A \cdot \eta = \int_{t_0}^{t_f} [m\ddot{x} + mg]\eta dt$$

for all possible η where $\eta(t_0) = \eta(t_f) = 0$. Moreover, from the statement of the principle, we know that we are looking for $x(t)$ where $\delta A \cdot \eta = 0$. Looking at the integral, the only way that integral will be equal to zero for any choice of η is if the term on the left happens to always be equal to zero. This means that $m\ddot{x} + mg = 0$, which implies that $\ddot{x} = -g$, which is indeed what we find from force balance.

The upshot of all this is that the variational principle reproduces the same result as we get when applying force balance. And if one was only ever faced with modeling the dynamics of a particle in gravity, this would seem to be a long-way-around approach. But we will see that variational approaches have other properties that will be very helpful when modeling more complex systems, particularly those with constraints or experiencing impacts.

Variational Principle for Unforced, Unconstrained Mechanics



Last time we saw that the variational principle leads to the same equations of motion as Newton's laws—at least for a particle in gravity. For that specific example, Newton's laws would be the way to go. But as systems become more complex, and as one wants to be able to *choose* how one measures coordinates, Newton's laws can become increasingly challenging to use. Here we formalize the variational principle for general mechanical systems described by kinetic energy and potential energy, but (for now) assuming no constraints and no external forces.

Define q to be a set of variables that uniquely describe the system (called the *configuration*) at any moment in time.^{45,46}

Remember that

$$A(q(t)) = \int_{t_0}^{t_f} KE(q(t), \dot{q}(t)) - V(q(t)) dt = \int_{t_0}^{t_f} L(q(t), \dot{q}(t)) dt,$$

where $L(q(t), \dot{q}(t))$ is the *Lagrangian*—a scalar-valued function made up of the difference between kinetic and potential energy. We want to extremize $A(q(t))$ with respect to $q(t)$. In shorthand, this is “setting the derivative equal to zero” or

$$\delta A = 0$$

but a more useful way of thinking about it is that the directional derivative has to be zero in all directions $\eta(t)$, keeping in mind the lesson from the point mass in gravity that the perturbations η need to be zero at t_0 and t_f . That is, we require that $\delta A(q(t)) \cdot \eta(t) = 0 \forall \eta(t)$ such that $\eta(t_0) = \eta(t_f) = 0$.

Figure 7: **Visualization of the variational principle:** (left) a trajectory $x(t)$ along with three perturbations—keeping in mind that there are an *infinite* number of them since x is a function of time t —to the trajectory $x(t) + \varepsilon \eta(t)$ and (right) the perturbations by themselves are zero at t_0 and t_f .

⁴⁵ For instance, the position $(x, y) \in \mathbb{R}^2$ and the orientation θ of a vehicle describe both where it is on a map and its compass orientation, in which case $q = (x, y, \theta)$.

⁴⁶ Another example—one students will grow to both love and hate throughout this course—is that of a n -link pendulum, where the configuration of each revolute joint i is described by an angle θ_i , so that $q = (\theta_1, \theta_2, \dots, \theta_n)$.

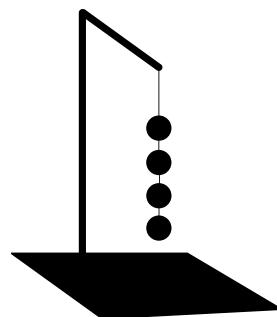


Figure 8: An n link spherical pendulum.

To facilitate taking this derivative, define the quantity $q_\epsilon(t) = q(t) + \epsilon\eta(t)$ to make the notation a little simpler while evaluating the Gâteaux derivative. Then the action principle states that

$$\begin{aligned}\delta A(q(t)) \cdot \eta(t) &= \frac{d}{d\epsilon} A(q(t) + \epsilon\eta(t))|_{\epsilon=0} \\ &= \frac{d}{d\epsilon} A(q_\epsilon(t))|_{\epsilon=0} = 0 \quad \forall \eta(t) \text{ s.t. } \eta(t_0) = \eta(t_f) = 0.\end{aligned}$$

Derivation of the Euler-Lagrange Equations⁴⁷

Our strategy for deriving the equations of motion governing the behavior of a mechanical system is to differentiate $A(q(t))$ and set the result equal to zero, looking for the generalization of $\ddot{x} = -g$ that we found in the previous section. We start by evaluating the directional derivative $\frac{d}{d\epsilon} A(q_\epsilon(t))|_{\epsilon=0}$.⁴⁸

$$\begin{aligned}\frac{d}{d\epsilon} A(q_\epsilon(t)) &= \int_{t_0}^{t_f} \frac{\partial L}{\partial q_\epsilon} \eta(t) + \frac{\partial L}{\partial \dot{q}_\epsilon} \dot{\eta}(t) dt \\ &\quad \text{and by integration by parts} \\ \int_{t_0}^{t_f} \frac{\partial L}{\partial \dot{q}_\epsilon} \dot{\eta}(t) dt &= \frac{\partial L}{\partial \dot{q}_\epsilon} \eta(t)|_{t_0}^{t_f} - \int_{t_0}^{t_f} \left(\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_\epsilon} \right) \eta(t) dt.\end{aligned}$$

Since the endpoints of the variation are zero

$$\begin{aligned}\frac{\partial L}{\partial \dot{q}_\epsilon} \eta(t)|_{t_0}^{t_f} &= 0, \\ &\quad \text{so} \\ \frac{d}{d\epsilon} A(q_\epsilon(t)) &= \int_{t_0}^{t_f} \left(\frac{\partial L}{\partial q_\epsilon} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_\epsilon} \right) \eta(t) dt \\ \frac{d}{d\epsilon} A(q_\epsilon(t))|_{\epsilon=0} &= \int_{t_0}^{t_f} \left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) \eta(t) dt = 0 \quad \forall \eta(t) \\ &\quad \text{which implies} \\ 0 &= \frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}}.\end{aligned}$$

The equations

$$\boxed{\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0}$$

are called the *Euler-Lagrange* equations.⁴⁹ Keep in mind that q is a vector of configuration variables, so these are partial derivatives. As a result, if q is n dimensional—that is, $q = \{q_1, q_2, \dots, q_n\}$ —one should end up with n scalar equations as part of “the” differential equation. So one could write the above equation as:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0 \quad \forall i$$

⁴⁷ My exams include a requirement that students memorize several derivations—as assessed by me removing a single line and students filling it in. The derivation of the Euler-Lagrange Equations is the most important of these proofs/derivations.

⁴⁸ Students are often surprised by the addition sign $+$ in the first line showing up. This is just chain rule. For instance, if I have a function $f(x(\sigma), y(\sigma))$ and I want to evaluate $\frac{\partial f}{\partial \sigma}$, then I use chain rule to find

$$\frac{\partial f}{\partial \sigma} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial \sigma} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial \sigma}.$$

The variable $\dot{\eta}(t)$ showing up in the first line of the derivation implies that we are *assuming* that perturbations/variations applied to $q(t)$ must be differentiable in time. Later we will relax this assumption when we study impacts.

⁴⁹ I am multiplying both sides by -1 in order to keep these terms in the standard order, but—think about this for a moment—this does not change the solution $q(t)$ to the ordinary differential equation. We will see later that this order is the standard convention when one includes external forces.

or, equivalently,

$$\begin{aligned}\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_1} - \frac{\partial L}{\partial q_1} &= 0 \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_2} - \frac{\partial L}{\partial q_2} &= 0 \\ &\vdots \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_n} - \frac{\partial L}{\partial q_n} &= 0.\end{aligned}$$

Row Vectors Versus Column Vectors A note on the dimensions of the Euler-Lagrange Equations is in order. As written, our notation suggests that that $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q}$ is a column vector; this is for the convenience of stacking equations on top of each other as above, when writing out the ODE describing the dynamics of a particular q_i . Nevertheless, just as $\frac{\partial f}{\partial x}$ is a row vector (that multiplies a column vector v to give the scalar directional derivative), both $\frac{\partial L}{\partial q}$ and $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}}$ are *row* vectors. Every once in a while this will matter to us during a derivation, and I will point it out.

Examples

1. A particle in gravity:

$q = x \in \mathbb{R}^1$ is the vertical height, the kinetic energy is $KE(x(t), \dot{x}(t)) = \frac{1}{2}m\dot{x}^2$, and the potential energy is $V = mgx$. This means we can write the Lagrangian as $L = KE - V = \frac{1}{2}m\dot{x}^2 - mgx$. Evaluating the Euler-Lagrange Eqs yields $\frac{d}{dt}(m\dot{x}) + mg = 0$, which implies that $\ddot{x} = -g$.

2. A particle in \mathbb{R}^3 subject to potential $V(x, y, z)$:

This potential could be due to a spring (see Fig. 9), gravity, electro-magnetic fields, etcetera. This gives us a Lagrangian of the form $L = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) - V(x, y, z)$. Evaluating the Euler-Lagrange Eqs. yields

$$\begin{aligned}\frac{d}{dt}(m\dot{x}) + \frac{\partial V}{\partial x} &= 0 & m\ddot{x} + \frac{\partial V}{\partial x} &= 0 \\ \Rightarrow \frac{d}{dt}(m\dot{y}) + \frac{\partial V}{\partial y} &= 0 & \Rightarrow m\ddot{y} + \frac{\partial V}{\partial y} &= 0 \\ \frac{d}{dt}(m\dot{z}) + \frac{\partial V}{\partial z} &= 0 & m\ddot{z} + \frac{\partial V}{\partial z} &= 0.\end{aligned}$$

3. A particle in \mathbb{R}^3 subject to potential $V(x, y, z)$ using vector notation:

Set $q = (x, y, z)$. Then

$$L = \frac{1}{2}\dot{q}^T J \dot{q} - V(q) \text{ where } J = m \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The Euler-Lagrange Eqs give us $\ddot{q}^T J + \frac{\partial V}{\partial q} = 0$, and taking the transpose of both sides yields

$$J\ddot{q} + \nabla V = 0$$

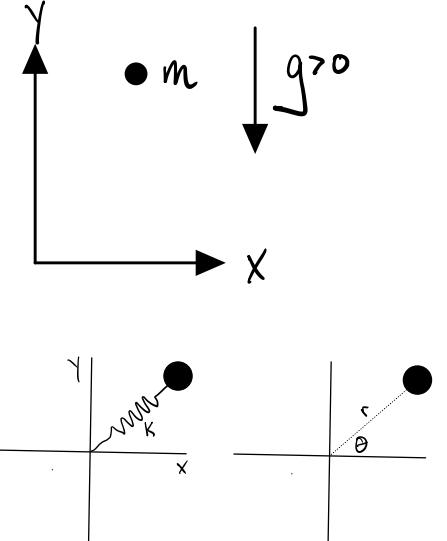


Figure 9: A particle in gravity with $q = x$ and a particle with a spring potential with $q = (x, y, z)$.

which are the vector notation for Newton's equations for a particle in a potential field.

To really see the power of the Lagrangian method, you have to look at examples where the “natural” geometric description is not in Cartesian coordinates. In particular, the Euler-Lagrange equations are coordinate independent,⁵⁰ so you can choose the coordinates you want to use, like in the case of the pendulum.

The “natural” choice of coordinates for the pendulum is the angle θ . This yields a kinetic energy of $KE = \frac{1}{2}mv^2 = \frac{1}{2}m(L\dot{\theta})^2$ and a potential energy of $V = mgh = mgL(1 - \cos(\theta))$, resulting in a Lagrangian

$$L(\theta, \dot{\theta}) = \frac{1}{2}mL^2\dot{\theta}^2 - mgL(1 - \cos(\theta))$$

yielding the Euler-Lagrange equations

$$\begin{aligned} \frac{d}{dt}(mL^2\dot{\theta}) - (-mgL\sin(\theta)) &= 0 \\ mL^2\ddot{\theta} + mgL\sin(\theta) &= 0 \\ \ddot{\theta} &= -\frac{g}{L}\sin(\theta). \end{aligned}$$

It is worth looking at the equation $\ddot{\theta} = -\frac{g}{L}\sin(\theta)$ and asking whether this differential equation is *plausible*.⁵¹

The Lagrangian and the Euler-Lagrange equations can be easily computed and solved using Python's SymPy package—below is example code for computing the acceleration of a particle falling in gravity.

```

1 ######
2 # Solve Euler-Lagrange equations for particle falling in gravity
3 #####
4
5 import sympy as sym
6 from sympy.abc import t
7 from sympy import symbols, Function, Eq, solve
8
9 # define "constants" as symbols
10 g, m = symbols('g, m') # r'' means "raw string" in Python
11
12 # define particle state (as a Function)
13 x = Function('x')(t)
14
15 # define velocity and acceleration
16 xdot = x.diff(t)
17 xddot = xdot.diff(t)
18
19 # compute kinetic energy and potential energy
20 ke = 0.5 * m * xdot**2
21 pe = m * g * x
22
23 # compute Lagrangian
24 L = ke - pe
25 print('Lagrangian: ')

```

⁵⁰ We will discuss this property in detail in a few sections from now.

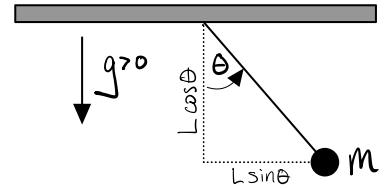


Figure 10: A pendulum of length L hanging θ away from vertical in gravity

⁵¹ This is an undervalued skill when students are taking mathematics-oriented course. How can this ODE be reasoned about? First, one can “linearize” it by replacing $\sin()$ with its small angle approximation, leading to $\ddot{\theta} \approx -\frac{g}{L}\theta$. If the student has taken a linear algebra course, they will recognize this as a harmonic oscillator. If not, consider the case where $\dot{\theta} = 0$ and θ is small and positive. This corresponds to the pendulum in the figure above being slight “up” to the right—and the ODE predicts acceleration to the left. This makes sense, suggesting that our result is plausible. In general, this class is about developing the skills needed to *increase confidence* in a simulation without blindly accepting it.

```
26 display(L)
27
28 # compute derivative of L wrt x
29 dLdx = L.diff(x)
30
31 # compute derivative of L wrt xdot
32 dLdxdot = L.diff(xdot)
33
34 # keep computing the time derivative of that
35 d_dLdxdot_dt = dLdxdot.diff(t)
36
37 # now we have the Euler-Lagrange equations
38 el_eqns = Eq(d_dLdxdot_dt-dLdx, 0)
39 print('Euler-Lagrange equation:')
40 display(el_eqns)
41
42 # solve the equation ... for *xddot*!
43 soln = solve(el_eqns, xddot)
44 print('solution for xddot:')
45 for sol in soln:
46     display(Eq(xddot, sol))
```

External Forces and Forces of Constraint

Last time we saw that if a system is completely described by its kinetic energy and potential energy, we can get an ordinary differential equation that predicts the evolution of the *unforced, unconstrained* mechanical system. We have not yet determined if that prediction is any good, or if it relates to Newton's Laws.⁵² But most engineered systems—and indeed most systems in general—have external forces acting on them.⁵³ Moreover, mechanical systems are often subject to constraints—a foot cannot go through the floor, a slider has to run along a track, et cetera. The upshot of this section is that the effects due to external forces and constraints can also be modeled using variational methods by modifying our previous derivation of the unforced, unconstrained case.⁵⁴

External Forces in the Euler-Lagrange Equations

External forces are surprisingly simple to incorporate into a variational perspective using the extreme action principle. The Lagrange d'Alhembert Principle states that in the presence of external forces F the curve $q(t)$ that the physical system follows satisfies

$$\frac{d}{d\epsilon} \int_{t_0}^{t_f} L(q_\epsilon, \dot{q}_\epsilon) dt|_{\epsilon=0} + \int_{t_0}^{t_f} F \eta dt = 0 \quad \forall \eta \text{ such that } \eta(t_0) = \eta(t_f) = 0.$$

This principle yields the *forced Euler-Lagrange Equations*.

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = F,$$

As an exercise, modify what we did in the previous section to obtain these equations.

What the principle does not tell you is how to *compute* F . That is, you have to be able to express the external forces F in coordinates q in order to use it here. This often requires changing coordinates on F from a choice of coordinates where you know how to model F (e.g., friction at a point of contact produces a force that is opposed to the velocity of that point along the contact surface) to the coordinates you are using (e.g., you are probably not using the *location* of the point of contact as your generalized coordinates). We will discuss this more later, but the way we change coordinates on forces is to use the *equivalent instantaneous work principle*⁵⁵, which states that the mechanical instantaneous work $F \cdot V$ —the Euclidean inner product of a force (or “wrench” consisting of a force and a torque) F and velocity V —must be the same in any coordinate system. This statement should seem reasonable; power is an energetic concept, and one does not expect its value to change depending on the geometry with which we observe it.⁵⁶

⁵² We would expect any theory to agree with Newton's Laws so long as the system is at normal macroscopic scales at normal speeds.

⁵³ E.g., forces in the form of control authority from motors or other actuators, forces from friction and other dissipative effects, et cetera.

⁵⁴ This turns out *not* to be true, or at least not easily true, of the Hamiltonian perspective on dynamics. Nevertheless, we will connect to Hamiltonians a few sections from now.

F is a row vector, so the pairing with η in the expression $\int_{t_0}^{t_f} F \eta dt$ is dimensionally consistent and will produce a real number, not a vector or a matrix.

⁵⁵ Richard M Murray, Zexiang Li, and S Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC press, 1994

⁵⁶ Normally, at this point in describing something, I would want to provide an example. The problem is that this idea is pretty trivial in simple examples, leading to confusion about why we need it at all. So I delay further discussion of external forces until we have the tools we need to incorporate them into more complex mechanical systems.

The Constrained Euler-Lagrange Equations

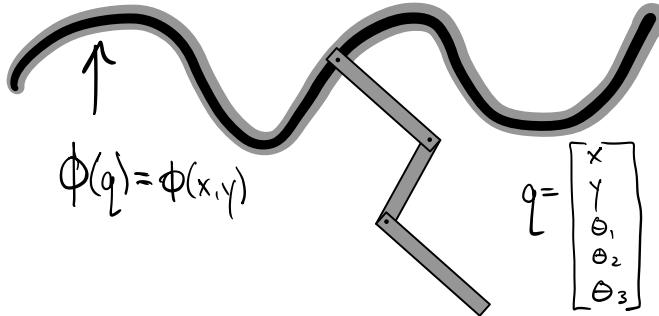


Figure 11: This triple pendulum is completely described by the (x, y) coordinates of its base and the $(\theta_1, \theta_2, \theta_3)$ coordinates of each joint. But if the base is constrained—think of this as a hapless triple pendulum riding a rollercoaster—then the pendulum cannot fall to the ground under the influence of gravity $g > 0$. The constraint $\phi(x, y) = 0$ describes the fact that the base of the pendulum cannot fall off the track.

Constraints are among the common mechanical features. Gears are constrained to move together, feet do not fall through the floor, roller coaster slide along rails—constraints are everywhere and belong front and center in any discussion of mechanical modeling. Here we examine what happens we further modify the extreme action principle to take into account the presence of a constraint. That is, we need to examine what happens if trajectories $q(t)$ (and their variations $\eta(t)$) are constrained to some subset of Q (technically called a submanifold of Q). We will only consider constraints that can be specified by a function $\phi(q) = 0$; this omits important classes of constraints, particularly those that are *nonholonomic*⁵⁷, but will be sufficiently general for much of what an undergraduate might want to do.

We again are going to derive equations of motion for constrained mechanical systems by setting the derivative of the action integral equal to zero

$$\delta A(q(t)) \cdot \eta(t) = \frac{d}{d\epsilon} A(q_\epsilon(t))|_{\epsilon=0} = 0 \quad \forall \eta(t) \text{ s.t. } \eta(t_0) = \eta(t_f) = 0.$$

However, in this case we will subject the variations to the condition that $\phi(q_\epsilon(t)) = 0 \forall \eta(t) \text{ s.t. } \eta(t_0) = \eta(t_f) = 0$. That is, we require that *all* the variations η satisfy the constraint requirements.⁵⁸

Derivation of the Constrained Euler-Lagrange Equations

We proceed as before, taking the derivative using the Gâteaux derivative and setting it equal to zero:

$$\frac{d}{d\epsilon} \int_0^T L(q_\epsilon, \dot{q}_\epsilon) dt \Big|_{\epsilon=0} = 0 \Rightarrow \int_0^T \left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) \eta(t) dt = 0.$$

So far, nothing has changed and one might, incorrectly, conclude that the equations of motion are the same. However, we are *restricting* the variations $\eta(t)$ to satisfy $\phi(q_\epsilon) = \phi(q + \epsilon\eta) = 0$. If $\phi(q + \epsilon\eta) = 0$, then

⁵⁷ Constraints specified by $\phi(q) = 0$ are called *holonomic*, and those that cannot be specified this way are called *nonholonomic*. Nonslip constraints, such as those encountered when driving a car—the wheels cannot slip sideways—are important examples of nonholonomic constraints.

⁵⁸ This requirement is not at all obvious, and as an exercise the student should try to derive equations of motion from the constrained variational principle without this added requirement to see what goes wrong.

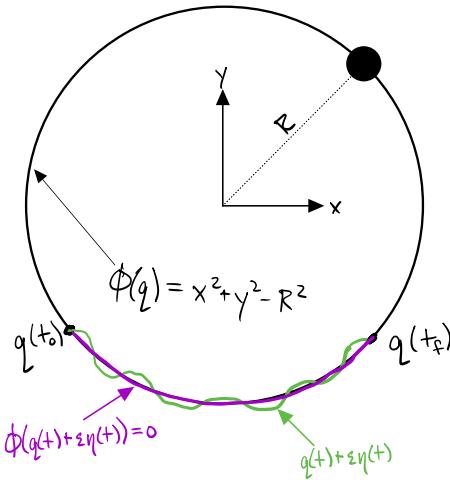


Figure 12: A mass constrained to a circle in the presence of gravity $g > 0$. Note that the variations/perturbations applied to $q(t)$ must also lie on the circle in order for $\phi(q_\epsilon(t)) = 0$ to hold.

certainly its derivative with respect to ϵ evaluated at $\epsilon = 0$ is also 0:

$$\frac{d}{d\epsilon} \phi(q + \epsilon\eta)|_{\epsilon=0} = \frac{\partial \phi}{\partial q}\eta = 0.$$

As result we have simultaneous equations to consider.

$$\begin{aligned} \int_0^T \left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right) \eta(t) dt &= 0 \\ \frac{\partial \phi}{\partial q}\eta &= 0 \end{aligned}$$

Geometrically, these two equations each imply that $\left(\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right)$ —which is just a row vector for any given choice of t —is orthogonal to every $\eta(t)$ satisfying the constraint $\phi(q + \epsilon\eta) = 0$ at every t and $\frac{\partial \phi}{\partial q}$ —also a row vector—is orthogonal to every $\eta(t)$ satisfying the constraint $\phi(q + \epsilon\eta) = 0$ at every t . As a result, we have two n -dimensional vectors that are each orthogonal to the same $n - 1$ vectors—this implies they must be collinear.⁵⁹

That is:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \lambda \frac{\partial \phi}{\partial q},$$

typically referred to as the *constrained Euler-Lagrange equations*. Notice that there are n equations but $n + m$ unknowns ($n \dot{q}$ terms and $m \lambda$ terms for each of the m constraints). However, we additionally know that $\phi(q) = 0$, which gives us $n + m$ equations and $n + m$ unknowns, leading to a (nominally) well-posed problem. To illustrate, Fig. 13 shows a mass constrained to a circle. The gradient of ϕ is orthogonal to the tangent to the circle—these are the directions the variations $\eta(t)$ can point. Similarly, the vector $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q}$ would also be orthogonal to those same vectors.⁶⁰

⁵⁹ That is, they must lie on the same line, and must therefore be scalar multiples of each other, where we will call the scalar λ . This statement is not terribly obvious. Think this through in the case where $\phi(q)$ is scalar, so that if $q \in \mathbb{R}^n$, there will be $n - 1$ independent η that satisfy $\phi(q_\epsilon) = 0$. In this case, we have two vectors that are orthogonal to the same $n - 1$ vectors in an n -dimensional space. As a result, there is only one dimension “left over” for the vectors to exist in, and as a result they are scalar multiples of each other. The same reasoning holds for $\phi(q)$ multi-dimensional, which the student can verify as an exercise.

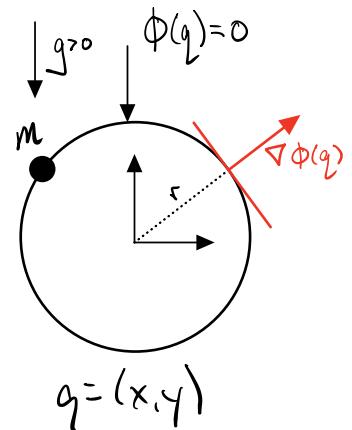


Figure 13: The forces F_N that keep a system constrained are orthogonal the constraint $\phi(q) = 0$.

⁶⁰ Though not necessarily pointing away from the origin—it could be pointing towards the origin!

Note that the term $\lambda \frac{\partial \phi}{\partial q}$ shows up in the exact same way that external forces show up—for this reason, we call them *forces of constraint* or *constraint forces*. If we have a Lagrangian $L(q, \dot{q})$ and constraint $\phi(q) = 0$, constraint forces are the forces that keep q from leaving the set that satisfies $\phi(q) = 0$.⁶¹

The constraint forces F_N are orthogonal to the surface $\phi(q) = 0$, as depicted in Fig. 13. Moreover, $\nabla \phi(q)$ is orthogonal to the level set $\phi(q) = 0$, so we are looking for a force that is a scalar multiple of $\nabla \phi(q)$; we call this scalar factor λ (called the *multiplier*). We solve for \ddot{q} and λ independently using one set of $n + m$ equations. In particular, we have

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} &= \lambda \nabla \phi(q) \\ \phi(q) &= 0. \end{aligned}$$

Now, typically $\phi(q) = 0$ is not be sufficient to solve for \ddot{q} and λ independently of each other, because $\phi(q) = 0$ does not depend on either \ddot{q} or λ . But if $\phi(q) = 0$, its time derivatives must as well, and taking the second derivative of $\phi(q) = 0$ should, by chain rule, give us a dependence on \ddot{q} . So we differentiate the constraint to get

$$\frac{d}{dt} \phi(q) = \frac{\partial \phi}{\partial q} \dot{q} = 0$$

and

$$\frac{d}{dt} \left(\frac{\partial \phi}{\partial q} \dot{q} \right) = \frac{\partial^2 \phi}{\partial q^2} (\dot{q}, \ddot{q}) + \frac{\partial \phi}{\partial q} \ddot{q} = 0.$$

This last equation can be shown to always provide enough information to solve for \ddot{q} and λ independently.⁶² This gives us the *constrained Euler-Lagrange Equations*:

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} &= \lambda \nabla \phi(q) \\ \frac{d^2}{dt^2} \phi(q) &= \frac{\partial^2 \phi}{\partial q^2} (\dot{q}, \ddot{q}) + \frac{\partial \phi}{\partial q} \ddot{q} = 0. \end{aligned}$$

Lastly, if there are external forces F , they are simply added to the constraint forces.

Example—Bead on a wire

As an example of constrained dynamics, consider again the mass constrained to a circle in Fig. 13. This is a point mass pendulum, but represented in $q = (x, y)$ coordinates—along with a constraint—instead of $q = \theta$ coordinates. The Lagrangian for this system is

⁶¹ In a static situation of an object resting on the ground, the constraint force is the opposing force that keeps the object from falling through the floor.

⁶² The amount of differentiation required to generate this guarantee is called the *differential index* of a differential-algebraic equation. A set of equations is called a differential algebraic equation when they involve a differential equality, e.g.,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \lambda \nabla \phi(q)$$

and an algebraic equality, e.g.,

$$\phi(q) = 0.$$

$L = KE - V = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy$ and the constraint is $\phi(q) = x^2 + y^2 - R^2$. The resulting equations of motion are $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \lambda \nabla \phi(q)$, which yields

$$\begin{aligned} m\ddot{x} &= \lambda(2x) \\ m\ddot{y} &= -mg + \lambda(2y) \end{aligned}$$

and

$$\begin{aligned} \phi(x, y) &= x^2 + y^2 - R^2 = 0 \\ \Rightarrow x\dot{x} + y\dot{y} &= 0 \\ \Rightarrow \dot{x}^2 + \dot{y}^2 + x\ddot{x} + y\ddot{y} &= 0. \end{aligned}$$

Solving for $\lambda, \dot{x}, \dot{y}$ —either by hand or in using a symbolic software package such as *python* using *SymPy*—one gets

$$\begin{aligned} \lambda &= \frac{mgy - m(\dot{x}^2 + \dot{y}^2)}{2R^2} \\ \dot{x} &= \frac{x(gy - (\dot{x}^2 + \dot{y}^2))}{R^2} \\ \dot{y} &= \frac{-gx^2 - y(\dot{x}^2 + \dot{y}^2)}{R^2} \end{aligned}$$

and the *constraint forces* are $\lambda(t)\nabla\phi(q(t))$.⁶³

```

1 import numpy as np
2 import sympy as sym
3 from sympy.abc import t, m, g
4 from sympy import solve, Eq
5
6 x=sym.Function(r'x')(t)
7 y=sym.Function(r'y')(t)
8 lam=sym.Function(r'lam')(t)
9 xdot=x.diff(t)
10 ydot=y.diff(t)
11 xddot=x.diff(t,t)
12 yddot=y.diff(t,t)
13
14 eq1=Eq(-m*xddot, lam*2*x)
15 eq2=Eq(-m*g-m*yddot, lam*2*y)
16 eq3=Eq(2*(xdot**2+ydot**2+x*xddot+y*yddot),0)
17
18 sol=solve([eq1,eq2,eq3],xddot,yddot, lam)
19 sol=sym.simplify(sol)
20 print(f'The Euler-Lagrange Equations are: {sol}')

```

⁶³ These constraint forces evolve over time, so the force of constraint explicitly depends on the configuration of the system and its velocity—a fact that is hopefully intuitive! Moreover, if one only wants to know the evolution of $x(t)$ and $y(t)$ given an initial condition (that satisfies the constraint), then one may at this point ignore the equation for λ ; it is now completely taken into account in \dot{x} and \dot{y} . However, if one needs to know the constraint forces—say, for stress analysis—then one should compute λ after simulating $x(t)$ and $y(t)$.

Coordinate Independence

What expectations should one have of a physical principle or of physical law? It is easy to jump to things like conservation principles and other ideas we will see play important roles later, but some principles are much more fundamental. For instance, *continuity* is a principle of motion we generally do not like to see violated; except for very small quantum phenomena, we generally do not see system states $q(t)$ and $\dot{q}(t)$ jumping discontinuously from one value to another, so we probably expect a physical principle to agree with that observation.⁶⁴ Another fundamental principle, nearly as fundamental as continuity, is that of *coordinate independence*.

Coordinate independence, as a principle of motion, implies that it should not matter how we choose coordinates, we will “get the same answer” regardless. That should seem reasonable—whatever the principle, if I imagine it in one set of abstract coordinates versus another, nothing about the thing I am modeling has changed.⁶⁵ Coordinate independence formalizes this notion, by insisting that if I

1. apply a principle in one set of coordinates x to obtain a differential equation in x , and
2. apply the same principle in another set of coordinates y to obtain a differential equation in y , then
3. performing a coordinate transformation on the differential equation in x to y coordinates should give me the same equation as the one I obtained from applying the principle in y coordinates.

To illustrate this idea, consider the system in Fig. 14. It has perhaps one of the simplest mechanical system one can imagine—two masses moving horizontally. Three coordinate choices are defined— $x = (x_1, x_2)$, $y = (y_1, y_2)$, and $\theta = (\theta_1, \theta_2)$. The coordinate choices reflect different preferences one might have about the mechanical system, often motivated by how one *measures* its behavior.⁶⁶

Here we go through the process of showing that the Euler-Lagrange equations are indeed coordinate independent. Then, in the next section, we will formally show that is so.

The Euler-Lagrange Equations in relative coordinates

This first example is the most natural coordinate choice if there are sensors measuring the *relative* distance between the two masses.⁶⁷ First, choose $m = m_1 = m_2$, but have $k_1 \neq k_2$. The kinetic energy is then $KE = \frac{1}{2}m(\dot{x}_1^2 + (\dot{x}_1 + \dot{x}_2)^2)$ and the potential energy is $V =$

⁶⁴ Naturally, we will see a special case where discontinuities *do* occur—namely mechanical impacts, discussed later.

⁶⁵ That is, its behavior does not depend on my attitude towards it.

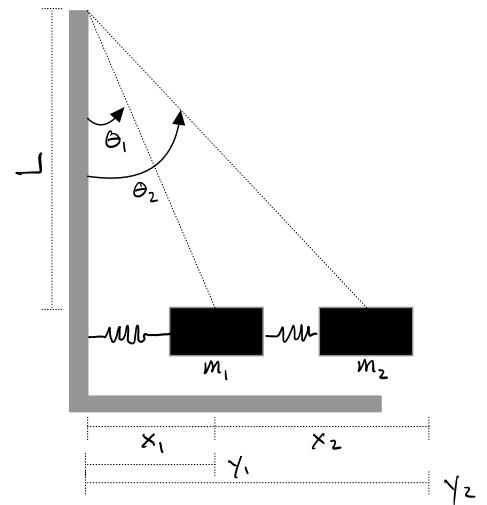


Figure 14: Two masses on a line with mass m , spring of constant k_1 between mass 1 and the wall, spring of constant k_2 between mass 2 and mass 1, both springs have natural length of zero.

⁶⁶ If using range measurements (say, between cars), y coordinates might make sense. If using inertial measurement, x coordinates might make sense. If using overhead measurements, θ coordinates might make sense.

⁶⁷ As in the case of vehicles measuring relative distance in order to avoid collision.

$\frac{1}{2}k_1x_1^2 + \frac{1}{2}k_2x_2^2$. Therefore, the Lagrangian is

$$L = \frac{1}{2}m(\dot{x}_1^2 + (\dot{x}_1 + \dot{x}_2)^2) - \frac{1}{2}k_1x_1^2 - \frac{1}{2}k_2x_2^2$$

and the Euler-Lagrange equations are:

$$\begin{aligned} \frac{d}{dt}(m\dot{x}_1 + m(\dot{x}_1 + \dot{x}_2)) - (-k_1x_1) &= 0 \\ \frac{d}{dt}(m(\dot{x}_1 + \dot{x}_2)) - (-k_2x_2) &= 0 \\ \Rightarrow 2m\ddot{x}_1 + m\ddot{x}_2 + k_1x_1 &= 0 \\ m\ddot{x}_1 + m\ddot{x}_2 + k_2x_2 &= 0 \\ \Rightarrow m\ddot{x}_1 + k_1x_1 - k_2x_2 &= 0 \\ m\ddot{x}_2 - k_1x_1 + 2k_2x_2 &= 0. \end{aligned}$$

The Euler-Lagrange Equations in inertially-fixed coordinates

This second choice of coordinates is most natural if there is a “ground-based” measurement that is measuring both masses relative to an absolute frame of reference.⁶⁸ The kinetic energy is $KE = \frac{1}{2}m(\dot{y}_1^2 + \dot{y}_2^2)$ and the potential energy is $V = \frac{1}{2}k_1y_1^2 + \frac{1}{2}k_2(y_2 - y_1)^2$. Therefore, the Lagrangian is

$$L = \frac{1}{2}m(\dot{y}_1^2 + \dot{y}_2^2) - \frac{1}{2}k_1y_1^2 - \frac{1}{2}k_2(y_2 - y_1)^2$$

and the Euler-Lagrange equations are:

$$\begin{aligned} \frac{d}{dt}(m\dot{y}_1) - (-k_1y_1 - k_2(y_2 - y_1)(-1)) &= 0 \\ \frac{d}{dt}(m\dot{y}_2) - (-k_2(y_2 - y_1)) &= 0 \\ \Rightarrow m\ddot{y}_1 + k_1y_1 - k_2(y_2 - y_1) &= 0 \\ m\ddot{y}_2 + k_2(y_2 - y_1) &= 0. \end{aligned}$$

The Euler-Lagrange Equations in angular coordinates

This is what an overhead radar might look like in terms of its coordinates. The transformation from x coordinates to θ coordinates is

$$\begin{aligned} x_1 &= L \tan \theta_1 \\ x_2 &= L \tan(\theta_1 + \theta_2) - L \tan \theta_1 \end{aligned}$$

and these equations can be plugged into the equations for the kinetic energy and the potential energy for the x coordinates to find the Euler-Lagrange equations in terms of θ_1 and θ_2 .⁶⁹

⁶⁸ Here, the vehicles may be using absolute measurements to make sure they are not over the speed limit, which is expressed in absolute terms rather than relative terms.

⁶⁹ Do this in python as an exercise!

Equivalence of x and y coordinate representations

The coordinate transformation from the x coordinates to the y coordinates is

$$\begin{aligned}y_1 &= x_1 \\y_2 &= x_1 + x_2\end{aligned}$$

which implies—by the equations of motion expressed in y —that

$$\begin{aligned}m\ddot{x}_1 + k_1x_1 - k_2x_2 &= 0 \\m(\ddot{x}_1 + \ddot{x}_2) + k_2x_2 &= 0 \\\Rightarrow m\ddot{x}_2 - k_1x_1 + 2k_2x_2 &= 0.\end{aligned}$$

(The third line comes from subtracting the first line from the second line.)

Hence, the equations in x and the equations in y are equivalent even though the equations in x are not written in Cartesian coordinates.

Note: Lastly, it is worth pointing out that Newton's Laws ($\Sigma F = ma$) are *not* coordinate independent—at least not in a meaningful way.⁷⁰ This point is obvious if one thinks about evaluating $\Sigma F = ma$ in x coordinates (where a would need to be interpreted as $a = \ddot{x}$) and y coordinates (again with $a = \ddot{y}$). Applying $\Sigma F = ma$ in each of these coordinate systems yields completely different results—and one of them is indeed incorrect, because Newton's Laws are only intended for an inertially fixed, Cartesian reference frame.⁷¹ As a result, only the inertially-fixed y coordinates can be used for evaluating Newton's laws.⁷²

⁷⁰ Some scientists—in my experience often physicists at conferences—sometimes say that Newton's Laws are coordinate independent, but what they mean by this is that Newton's laws can be applied to a system in inertially-fixed coordinates and then transformed into any other coordinate system. That is true, but not helpful conceptually since it is also true of any possible principle that yields a differential equation in inertially-fixed Euclidean space. We want the notion of coordinate independence to mean something non-vacuous, so we will insist on the stronger interpretation of the phrase. *Never trust a physicist!*

⁷¹ I.e., a Euclidean coordinate system using rectangular coordinates.

⁷² Evaluate $\Sigma F = ma$ for both the x coordinates and the y coordinates (in both cases treating the acceleration as the second time derivative of the coordinate $a = \ddot{x}$ or $a = \ddot{y}$) and confirm that the equations one gets in x coordinates cannot be correct.

Coordinate Independence of the Euler-Lagrange Equations

In the previous lecture we saw that—in the context of a specific example—the Euler-Lagrange equations are coordinate independent. That is, we demonstrated that if one computes the equations of motion in one coordinate system x and in another coordinate system q , changing coordinates on the differential equations from q to x will yield the same answer as if one had originally computed them in x . As this last sentence indicates, sometimes these ideas are challenging to say in words, so we represent coordinate independence using a *commutation diagram* like that seen in Fig. 15.

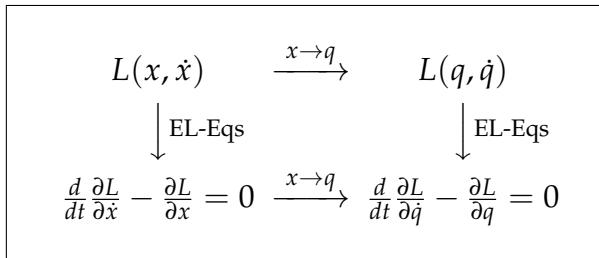


Figure 15: Coordinate independence means given a Lagrangian L in x coordinates computing a) the change of coordinates and then evaluating the Euler-Lagrange Equations should be the same as computing b) the Euler-Lagrange Equations and then changing coordinates. This equivalence is illustrated in the diagram above.

We know that the Euler-Lagrange equations hold when $q = x = \{x_i \in \mathbb{R}^3\}$ is the collection of Cartesian (rectangular) coordinates for each point mass because that is what Newton's Laws $\Sigma F = ma$ tell us. So, for that choice of coordinates, we have

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = 0.$$

Coordinate independence says that the *exact* same equation should hold in any choice of coordinates q , not just the Cartesian choice.

That is, if the above equation is true, we want to *show* that it is necessarily true that

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

Proof of Coordinate Independence for the Euler-Lagrange Equations

To prove that

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = 0 \implies \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0,$$

we assume that we know two things:

1. we *assume* that $\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = 0$
2. we *assume* that we have *another* choice of coordinates q and that there is a change of coordinates between x and q .⁷³

⁷³ That is, we can rewrite x in terms of q and perhaps t so that $x = x(q, t)$.

As a result of these two assumptions, we have a Lagrangian of the form $L(x, \dot{x})$ where x depends on q . Hence, by chain rule

$$\frac{\partial L}{\partial q} = \frac{\partial L}{\partial x} \frac{\partial x}{\partial q} + \frac{\partial L}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial q}$$

and

$$\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial x} \frac{\partial x}{\partial \dot{q}} + \frac{\partial L}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial \dot{q}}.$$

However, note that $\frac{\partial x}{\partial \dot{q}} = 0$ because $x = x(q, t)$,⁷⁴ so

$$\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial \dot{q}}.$$

Moreover,

$$\dot{x} = \frac{d}{dt}(x(q, t)) = \frac{\partial x}{\partial q} \frac{dq}{dt} + \frac{\partial x}{\partial t} = \frac{\partial x}{\partial q} \dot{q} + \frac{\partial x}{\partial t},$$

which implies

$$\frac{\partial \dot{x}}{\partial \dot{q}} = \frac{\partial}{\partial \dot{q}} \left(\frac{\partial x}{\partial q} \dot{q} + \frac{\partial x}{\partial t} \right) = \frac{\partial x}{\partial q}.$$

Therefore, we have $\frac{\partial \dot{x}}{\partial \dot{q}} = \frac{\partial x}{\partial q}$, which implies

$$\frac{\partial L}{\partial \dot{q}} = \frac{\partial L}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial q}$$

which, in turn, implies

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial q} \right) = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) \frac{\partial x}{\partial q} + \frac{\partial L}{\partial \dot{x}} \frac{d}{dt} \left(\frac{\partial x}{\partial q} \right).$$

Now, by the fact that mixed partials commute, we get

$$\frac{d}{dt} \frac{\partial x}{\partial q} = \frac{\partial}{\partial q} \dot{x}$$

implying

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) \frac{\partial x}{\partial q} + \frac{\partial L}{\partial \dot{x}} \frac{\partial}{\partial q} \dot{x} = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) \frac{\partial x}{\partial q} + \frac{\partial L}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial q}.$$

Lastly, we know that

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} = \frac{\partial L}{\partial x}$$

by Newton's equations, which gives us

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = \frac{\partial L}{\partial x} \frac{\partial x}{\partial q} + \frac{\partial L}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial q} = \frac{\partial L}{\partial q}.$$

So we have shown that

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

for any choice of coordinates q . This ends the proof. \square

⁷⁴ Students sometimes find it unintuitive that q and \dot{q} are *independent* of each other. After years of this question coming up, it finally occurred to me that this is linguistic confusion— q and \dot{q} are *related* to each other since one of the time derivative of the other. But they are *independent* because the position can be specified separately from the velocity.

Example: Cartesian and polar coordinates for a planar particle

Assume that we have a coordinate transformation between (x, y) coordinates and (r, θ) coordinates:

$$(x, y) = (r \cos \theta, r \sin \theta).$$

We can write the kinetic energy and the potential energy in both these coordinates and obtain equations of motion using the Euler-Lagrange equations. Then we can coordinate transform one of those ordinary differential equations—in this case the ordinary differential equation expressed in (x, y))—into the other coordinate system—in this case, the (r, θ) coordinates.

$$\begin{aligned} KE &= \frac{1}{2}mv^2 = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) = \frac{1}{2}m\left(\left(\frac{d}{dt}(r \cos \theta)\right)^2 + \left(\frac{d}{dt}(r \sin \theta)\right)^2\right) \\ &= \frac{1}{2}m\left((\dot{r} \cos \theta - r\dot{\theta} \sin \theta)^2 + (\dot{r} \sin \theta + r\dot{\theta} \cos \theta)^2\right) \\ V &= mgy = mgr \sin \theta \end{aligned}$$

In Cartesian coordinates this gives us:

$$\begin{aligned} \ddot{x} &= 0 \\ \ddot{y} &= -g \end{aligned}$$

and in polar coordinates we get:

$$\begin{aligned} \ddot{r} &= -g \sin \theta + r\dot{\theta}^2 \\ \ddot{\theta} &= -\frac{1}{r}(g \cos \theta + 2\dot{r}\dot{\theta}) . \end{aligned}$$

Are these equivalent?⁷⁵ Plug $x = r \cos \theta$ and $y = r \sin \theta$ into \ddot{x} and \ddot{y} and solve for \ddot{r} and $\ddot{\theta}$.

$$\begin{aligned} \ddot{x} &= \frac{d^2}{dt^2}(r \cos \theta) = \frac{d}{dt}(\dot{r} \cos \theta - r\dot{\theta} \sin \theta) \\ &= \ddot{r} \cos \theta - \dot{r}\dot{\theta} \sin \theta - \dot{r}\dot{\theta} \sin \theta - r\dot{\theta}^2 \cos \theta - r\ddot{\theta} \sin \theta \\ &= (\ddot{r} - r\dot{\theta}^2) \cos \theta - (\dot{r}\dot{\theta} + \dot{r}\dot{\theta} + r\ddot{\theta}) \sin \theta \\ \ddot{y} &= \frac{d^2}{dt^2}(r \sin \theta) = \frac{d}{dt}(\dot{r} \sin \theta + r\dot{\theta} \cos \theta) \\ &= \ddot{r} \sin \theta + \dot{r}\dot{\theta} \cos \theta + \dot{r}\dot{\theta} \cos \theta - r\dot{\theta}^2 \sin \theta + r\ddot{\theta} \cos \theta \\ &= (\ddot{r} - r\dot{\theta}^2) \sin \theta + (\dot{r}\dot{\theta} + \dot{r}\dot{\theta} + r\ddot{\theta}) \cos \theta \end{aligned}$$

and all of this equals $(0, -g)^T$. We can rewrite the equation above using matrix notation:

$$\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \ddot{r} - r\dot{\theta}^2 \\ 2\dot{r}\dot{\theta} + r\ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \end{bmatrix}$$

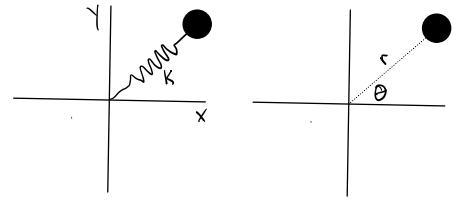


Figure 16: A particle with a spring potential represented using both $q = (x, y)$ and $q = (r, \theta)$ coordinates. This example is worked out using a gravitational potential, and as an exercise the student can check that exchanging the gravity potential with a spring potential will not change the coordinate independence.

⁷⁵ They better be!

This 2×2 matrix is a *rotation matrix*⁷⁶ that can be inverted by transposing it. This gives us:

$$\begin{bmatrix} \ddot{r} - r\dot{\theta}^2 \\ 2r\dot{\theta} + r\ddot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}^T \begin{bmatrix} 0 \\ -g \end{bmatrix} = \begin{bmatrix} -g \sin \theta \\ -g \cos \theta \end{bmatrix}$$

so the two equations are *equivalent* under change of coordinates.

⁷⁶ These types of matrix operations will be very important to us when we get to geometry of motion.

Conservation Laws and Numerical Methods

The relationship between conservation laws and symmetries of motion play an important role in our modern understanding of physics⁷⁷; we will ignore beauty and instead focus on the somewhat less aesthetically satisfying goal of critically assessing numerical capabilities, particularly in the context of simulation. This is a skill worth having—simulation capabilities are common,⁷⁸ but how much should an engineer *trust* these, particularly if the engineer does not have access to source code or if the source code is necessarily so complex it cannot be directly evaluated? Conservation laws provide one possible answer to this conundrum.

Numerical Methods for Simulating ODEs

Here we recap the background section on numerical simulation. For more information, see the first chapter on background mathematics.

The simplest numerical integration scheme is *Euler integration*. In it, we assume that we have a differential equation $\dot{x}(t) = f(x(t))$. On the left hand side we see the derivative of $x(t)$ with respect to t , which *by definition of the derivative* is equal to

$$\dot{x}(t) := \lim_{dt \rightarrow 0} \frac{x(t + dt) - x(t)}{dt} = f(x(t)).$$

If I set dt to be some rather small number, we get an expression that approximates \dot{x} and is therefore *approximately* equal to $f(x(t))$.

$$\frac{x(t + dt) - x(t)}{dt} \approx f(x(t))$$

Rearranging, I get:

$$x(t + dt) \approx x(t) + dt f(x(t)).$$

This formula, called *Euler integration*, means that if we know x at some time t ,⁷⁹ we can approximate x at $t + dt$.

Euler integration can be applied to any first-order⁸⁰ ODE. As

⁷⁷ Leon M Lederman and Christopher T Hill. *Symmetry and the beautiful universe*. Prometheus books, 2011

⁷⁸ For instance, one can download robotics simulators and physics engines (e.g., PyBullet) used for games. Even more commonly, one finds numerical integration schemes that are built into scientific computing platforms (such as python's NumPy).

⁷⁹ called a boundary condition or initial condition...

⁸⁰ ODEs involving only a single derivative with respect to the independent variable, in our case t .

discussed in the background section, Euler integration (and other integrators) can be directly applied to vector-valued ODEs such as

$$\dot{w} = Aw.$$

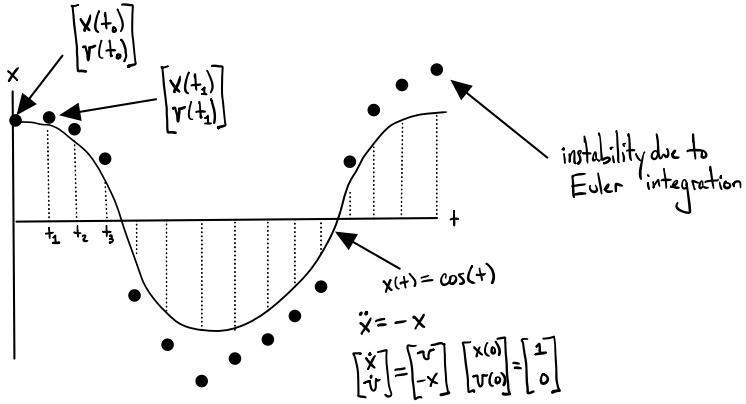


Figure 17: Numerical integrators will diverge in *nonphysical* ways from the true solution. In this case, an Euler-like integrator adds energy to the solution over time.

Euler integration converges slowly to the exact solution as dt goes to zero. Instead, there are what are called *higher order* schemes, where the word ‘order’ tells you how fast the scheme converges as you make dt smaller and smaller. The fourth-order RK scheme (RK4) is given by

$$x(t + dt) \approx x(t) + \frac{1}{6}(k_1 + k_2 + k_3 + k_4)$$

where

$$\begin{aligned} k_1 &= dt f(x(t)) \\ k_2 &= dt f\left(x(t) + \frac{k_1}{2}\right) \\ k_3 &= dt f\left(x(t) + \frac{k_2}{2}\right) \\ k_4 &= dt f(x(t) + k_3). \end{aligned}$$

This provides an excellent integrator. But how can we tell which integrator is providing sensible results, and how can we choose a dt ? Conserved quantities provide an objective measure of integrator performance, as we will see.

Conservation of Energy and Hamilton's Equations

There are two standard types of conservation laws in mechanical systems: *Energy* and *Momentum*. To understand energy and momentum, we first introduce a transform that will play a role in defining them. Assume that we have $L(q, \dot{q})$ and the Euler-Lagrange Equations hold. The *Legendre* transform is defined to be the mapping

$$(q, \dot{q}) \mapsto (q, p) \quad \text{where} \quad p = \frac{\partial L}{\partial \dot{q}}.$$

The variable p is called the *generalized momentum*.⁸¹ Hence, \dot{q} and p are related by the Legendre transformation, and if the Legendre transformation is invertible, then one can go back and forth between them.

The *Hamiltonian* H is defined using the generalized momentum and the Legendre transform.

$$H = p\dot{q} - L(q, \dot{q})$$

This Hamiltonian is a conserved quantity—that is, it is constant when evaluated along solutions to the unforced Euler-Lagrange Equations—and often⁸² corresponds to the *total energy* of the system $T = KE + V$.⁸³

Conservation laws such as conservation of the Hamiltonian give us analytical information about the qualitative behavior of a system and can be extremely helpful in *evaluating* numerical methods. Specifically, consider obtaining the Hamiltonian numerically by a) first numerically integrating an ordinary differential equation to approximate $q(t)$ and $\dot{q}(t)$, and then b) plugging the result in to the Hamiltonian expression $H(q, \dot{q})$. The result, for a Hamiltonian system, should be perfectly constant. But it will not be. Common numerical methods such as Euler integration and Runge-Kutta schemes distort the Hamiltonian behavior, like the sketch in Fig. 18 suggests. Integration schemes such as explicit and implicit Euler integration distort the integration quite a lot. High-order integration schemes like fourth and fifth order Runge-Kutta schemes will distort it less. But all numerical schemes will generate some distortion of the energy unless they are specifically designed not to.⁸⁴

Proof that the Hamiltonian is conserved along trajectories of the unforced Euler-Lagrange Equations

To prove that the Hamiltonian H is conserved along solutions of the unforced Euler-Lagrange Equations, we first look at the partial

⁸¹ This comes from the fact that if $L(x, \dot{x}) = 1/2m\dot{x}^2 + V(x)$, we have $p = m\dot{x}$ —the standard momentum that we all learn about in high school.

⁸² but *not* always....

⁸³ Note that the total energy only differs from the Lagrangian by having a “+” sign instead of a “−” sign; nevertheless, they are different and should not be confused.

⁸⁴ something I will discuss at the end of these notes...

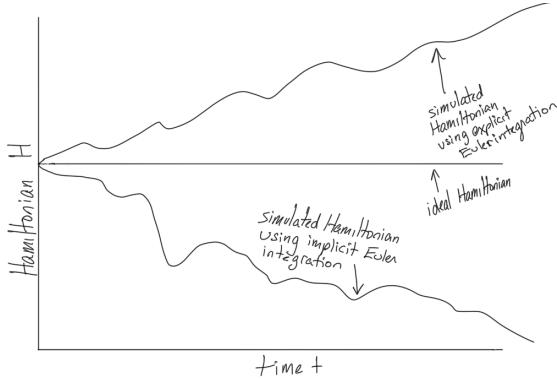


Figure 18: Sketch of what the “true” Hamiltonian might look like compared to the approximate Hamiltonian obtained from explicit and implicit Euler integration

derivatives of $H = p\dot{q} - L(q, \dot{q})$.

$$\begin{aligned}\frac{\partial H}{\partial p} &= \underbrace{\dot{q} + p \frac{\partial \dot{q}}{\partial p}}_{\text{product rule}} - \underbrace{\frac{\partial L}{\partial q} \frac{\partial \dot{q}}{\partial p}}_{=p} = \dot{q} \\ \frac{\partial H}{\partial q} &= -\underbrace{\frac{\partial L}{\partial q}}_{\text{by EL Eqs}} = -\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) = -\dot{p}.\end{aligned}$$

This gives us what are called *Hamilton’s equations*.⁸⁵

⁸⁵ Note that because p is a row vector, $\frac{\partial H}{\partial p}$ is a column vector (just like q).

$$\dot{q} = \frac{\partial H}{\partial p} \quad \dot{p} = -\frac{\partial H}{\partial q}$$

To see that H is conserved, take its time derivative:

$$\dot{H} = \underbrace{\frac{\partial H}{\partial q} \dot{q}}_{\text{scalar}} + \underbrace{\frac{\partial H^T}{\partial p} \dot{p}^T}_{\text{scalar}} = \frac{\partial H}{\partial q} \frac{\partial H}{\partial p} - \frac{\partial H^T}{\partial p} \frac{\partial H^T}{\partial q} = 0$$

where $\frac{\partial H^T}{\partial p} \frac{\partial H^T}{\partial q} = \frac{\partial H}{\partial q} \frac{\partial H}{\partial p}$ ⁸⁶ because $\frac{\partial H^T}{\partial p} \frac{\partial H^T}{\partial q}$ is a scalar and is therefore equal to its own transpose.

⁸⁶ That is, it commutes.

Examples

1. A particle in gravity:

Again, a particle in gravity with coordinate x measuring vertical

motion.

$$\begin{aligned}
 KE &= \frac{1}{2}m\dot{x}^2 & V &= mgx \\
 L &= \frac{1}{2}m\dot{x}^2 - mgx \\
 p &= \frac{\partial L}{\partial \dot{x}} = m\dot{x} \\
 H &= p\dot{x} - L(x, \dot{x}) = p\dot{x} - \frac{1}{2}m\dot{x}^2 + mgx \\
 &= \frac{1}{m}p^2 - \frac{1}{2m}p^2 + mgx \\
 &= \frac{1}{2m}p^2 + mgx
 \end{aligned}$$

which is the total energy of the system. Moreover, $\dot{H} = 0$, so total energy is conserved and

$$\begin{aligned}
 \dot{q} &= \frac{\partial H}{\partial p} = \frac{1}{m}p \\
 \dot{p} &= -\frac{\partial H}{\partial q} = -mg
 \end{aligned}$$

are Hamilton's equations for this system.

2. A particle in \mathbb{R}^3 subject to potential $V(x, y, z)$:

In this case,

$$L = \frac{1}{2}\dot{q}^T \mathcal{I} \dot{q} - V(q)$$

where \mathcal{I} is some constant, symmetric matrix.⁸⁷ The $p = \frac{\partial L}{\partial \dot{q}} = \dot{q}^T \mathcal{I}$.⁸⁸ Therefore $p\mathcal{I}^{-1} = \dot{q}^T \Rightarrow \dot{q} = \mathcal{I}^{-1}p^T$. This implies that

$$\begin{aligned}
 H &= p\dot{q} - L(q, \dot{q}) = p\mathcal{I}^{-1}p^T - \frac{1}{2}(\mathcal{I}^{-1}p^T)^T \mathcal{I}(\mathcal{I}^{-1}p^T) + V(q) \\
 &= p\mathcal{I}^{-1}p^T - \frac{1}{2}p\mathcal{I}^{-1}p^T + V(q) \\
 &= \frac{1}{2}p\mathcal{I}^{-1}p^T + V(q)
 \end{aligned}$$

which is total energy for this system as well.

Aside: Using the Legendre transform to compute the conserved quantity in terms of q and \dot{q} only works for systems where the Legendre transform is invertible. That is, one needs to be able to find \dot{q} in terms of p . If the Legendre transform of a Lagrangian is *not* invertible, it is called a *degenerate* Lagrangian. For example, we could in principle define a Lagrangian using coordinates $q = (x, y)$ with kinetic energy $KE = \frac{1}{2}m\dot{x}^2$ and potential energy $V = x^2 + y^2$. This could be interpreted as a mass connected to the origin by a spring, but one that for some reason only has inertia in the x coordinates and

⁸⁷ often just the identity matrix times the mass m ...

⁸⁸ which again means that p is a row vector while \dot{q} is a column vector

has no inertia in the y coordinates. In this case, $p = \frac{\partial L}{\partial \dot{q}} = (m\dot{x}, 0)$, so you cannot find \dot{q} from p .

Lastly, these examples follow the standard choice of using systems where the total energy is the Hamiltonian. Although this is common in physics, where physical systems under consideration are often *closed*,⁸⁹ in engineering settings systems are *open*⁹⁰ and the Hamiltonian can be something other than total energy. Next time we will see a system where the Hamiltonian is *not* the total energy.

⁸⁹ That is, the system is energetically isolated from its environment

⁹⁰ where the system is energetically coupled to its environment....

Systems with Nontrivial Hamiltonians

In this section we investigate a system with a Hamiltonian that is *not* the total energy. Such systems are often not discussed in physics classes, but are nearly ubiquitous in engineered systems because of geometric relationships between external power sources and the internal dynamics of a mechanism.⁹¹ The example—a hoop in three dimensions spinning about the z axis with a bead that is allowed to move freely along the hoop—is based on the flyball governor used originally to regulate steam engines.

The flyball governor is kinematically coupled to an axle—the faster the train goes, the faster the governor spins. A mass is attached to the spinning axis via a revolute joint; it is essentially a pendulum, the base of which is itself spinning about the vertical axis. The mass is in gravity, so when the train is stopped its equilibrium is at the configuration that describes the mass at its lowest hanging position.⁹² A simple version of this system is shown in Fig. 19.

First we need to figure out the Lagrangian. To do so, we calculate the velocity tangent to the sphere that the mass is on. In particular, we are looking for the velocity $v = v_1 + v_2$ where $v_1 \perp v_2$. We get

$$v_1 = R\dot{\theta} \quad v_2 = \omega R \sin \theta$$

which gives us kinetic energy and potential energy

$$\begin{aligned} KE &= \frac{1}{2}mv^2 = \frac{1}{2}mR^2\dot{\theta}^2 + \frac{1}{2}m\omega^2R^2 \sin^2 \theta \\ V &= mgh = mgR(1 - \cos \theta) \end{aligned}$$

where $g > 0$. So, the Lagrangian is

$$L = \frac{1}{2}mR^2\dot{\theta}^2 + \frac{1}{2}m\omega^2R^2 \sin^2 \theta - mgR(1 - \cos \theta)$$

which gives us the Euler-Lagrange equations:

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} &= \frac{d}{dt} (mR^2\dot{\theta}) - (m\omega^2R^2 \sin \theta \cos \theta - mgR \sin \theta) = 0 \\ \implies mR^2\ddot{\theta} - m\omega^2R^2 \sin \theta \cos \theta + mgR \sin \theta &= 0. \end{aligned}$$

This is a second-order nonlinear differential equation in θ , as you hopefully anticipate.

What does the Hamiltonian H look like for this system? We find the generalized momentum by computing the Legendre transformation

$$p = \frac{\partial L}{\partial \dot{\theta}} = mR^2\dot{\theta} \quad (\Rightarrow \dot{\theta} = \frac{1}{mR^2}p)$$

⁹¹ thus violating the “closed system” condition required for conservation of energy...

⁹² The position with the lowest potential energy.

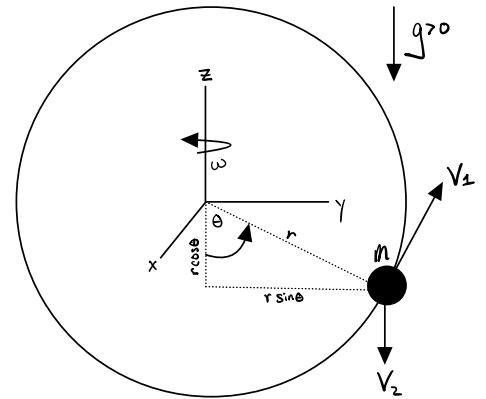


Figure 19: A hoop that spins around the z axis at rate ω with configuration θ measured from the negative z axis. The mass m is radius R away from the origin.

which then gives us the Hamiltonian

$$\begin{aligned} H(\theta, p) &= p\dot{\theta} - L(\theta, \dot{\theta}) \\ &= \frac{1}{mR^2}p^2 - \left[\frac{1}{2}\frac{1}{mR^2}p^2 + \frac{1}{2}m\omega^2R^2\sin^2\theta - mgR(1 - \cos\theta) \right] \\ &= \frac{1}{2}\frac{1}{mR^2}p^2 - \frac{1}{2}m\omega^2R^2\sin^2\theta + mgR(1 - \cos\theta). \end{aligned}$$

But this is *not* the total energy: the total energy of the system is

$$\begin{aligned} KE + V &= \frac{1}{2}mR^2\dot{\theta}^2 + \frac{1}{2}m\omega^2R^2\sin^2\theta + mgR(1 - \cos\theta) \\ &= \frac{1}{2}\frac{1}{mR^2}p^2 + \frac{1}{2}m\omega^2R^2\sin^2\theta + mgR(1 - \cos\theta). \\ H(\theta, p) &= \frac{1}{2}\frac{1}{mR^2}p^2 - \frac{1}{2}m\omega^2R^2\sin^2\theta + mgR(1 - \cos\theta). \end{aligned}$$

These two expressions only differ by the sign (indicated in red above), but that difference means that only one of the quantities can be conserved. As a result, in this case, total energy is *not* conserved—something else is!

To gain some intuition about why this might be, note that ω term indicates that the system is being driven at a fixed velocity, which can only happen if it is coupled to some external physical system. That is, this system is not a closed system, so conservation of energy is not necessarily expected.⁹³ This is often how we build mechanical systems—by geometrically coupling them to each other—and you likely do not want to have to model the entire train just to determine the behavior the flyball governor.

⁹³ even without friction...

Noether's Theorem

The Hamiltonian, typically Total Energy, is not the only conserved quantity. Momentum, typically denoted p , can also be a conserved, but this depends on how the Lagrangian depends on the configuration. We will look at momentum in two stages—the first is by inspection, just “looking” at the Lagrangian and realizing instances where there must be a conserved quantity. This will motivate our deeper look at Noether’s theorem, the most simple version of which states that conserved quantities are associated with geometric invariance of the Lagrangian.⁹⁴

To see it is reasonable that p is sometimes conserved, supposed that we have a coordinate q_k such that $\frac{\partial L}{\partial q_k} = 0$.⁹⁵ We are assuming that we are talking about a mechanical system, so we are assuming that the Euler-Lagrange equations hold:

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

Looking at each coordinate q_i individually, this is equivalent to

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0 \quad \forall i.$$

If $\frac{\partial L}{\partial q_k} = 0$, then

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} = 0$$

which implies that $\frac{\partial L}{\partial \dot{q}_k} = p_k$ is constant.⁹⁶

If we happen to write things down using coordinates such that we have $\frac{\partial L}{\partial q_k} = 0$, then we can find the conserved quantity easily—it is just $p = \frac{\partial L}{\partial \dot{q}_k}$. But we could potentially write the system using a different set of coordinates where the Lagrangian depends on all the coordinates but there still exists a conserved quantity.

Example

As an example, let’s go back (again) to the particle in gravity. In (x, y) coordinates we know that

$$L = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy,$$

and, in these coordinates, we have $\frac{\partial L}{\partial x} = 0$. The analysis above implies

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} = \frac{d}{dt} (m\dot{x}) = 0,$$

so $p = m\dot{x}$ is a conserved quantity.⁹⁷ In (r, θ) coordinates, however,

⁹⁴ Don’t worry for now about what this phrase *geometric invariance* means—I will make that precise soon.

⁹⁵ That is, L does not depend on q_k .

⁹⁶ That is, $\frac{\partial L}{\partial \dot{q}_k} = p_k$ is *conserved*.

⁹⁷ And indeed, this is just the linear momentum “in the horizontal direction” where gravity cannot cause acceleration.

we get

$$L = \frac{1}{2}m(r^2\dot{\theta}^2 + \dot{r}^2) - mgr\sin\theta.$$

Here we do *not* have $\frac{\partial L}{\partial q_k} = 0$ for any choice of q_k . ⁹⁸ To demonstrate that $p = m\dot{x}$ is still conserved, we derive the dynamics in polar coordinates and express p in those coordinates to show that $\dot{p} = 0$. The Euler-Lagrange equations in (r, θ) coordinates are

$$\begin{aligned} 0 &= \frac{d}{dt} \frac{\partial L}{\partial \dot{r}} - \frac{\partial L}{\partial r} = \frac{d}{dt} (m\dot{r}) - mr\dot{\theta}^2 + mg\sin\theta \\ 0 &= \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = \frac{d}{dt} (mr^2\dot{\theta}) + mgr\cos\theta. \end{aligned}$$

Now look at \dot{p} : $m\dot{x} = m\frac{d}{dt}(r\cos\theta)$ as

$$\begin{aligned} \frac{d}{dt} \left(m \frac{d}{dt} (r\cos\theta) \right) &= \frac{d}{dt} (m\dot{r}\cos\theta - mr\dot{\theta}\sin\theta) \\ &= m\ddot{r}\cos\theta - m\dot{r}\dot{\theta}\sin\theta - m\dot{r}\dot{\theta}\sin\theta - mr\ddot{\theta}\sin\theta - mr\dot{\theta}^2\cos\theta \\ &= m\ddot{r}\cos\theta - 2m\dot{r}\dot{\theta}\sin\theta - mr\ddot{\theta}\sin\theta - mr\dot{\theta}^2\cos\theta, \end{aligned}$$

keeping track of the fact that from the dynamics we know \ddot{r} and $\ddot{\theta}$.

First, notice that

$$\frac{d}{dt} (mr^2\dot{\theta}) = 2mr\dot{r}\dot{\theta} + mr^2\ddot{\theta}$$

so we can make the simplification

$$\frac{d}{dt} \left(m \frac{d}{dt} (r\cos\theta) \right) = m\ddot{r}\cos\theta - \frac{d}{dt} (mr^2\dot{\theta}) \frac{\sin\theta}{r} - mr\dot{\theta}^2\cos\theta.$$

Using the second Euler-Lagrange equation we make a substitution and continue as

$$\begin{aligned} \frac{d}{dt} \left(m \frac{d}{dt} (r\cos\theta) \right) &= m\ddot{r}\cos\theta - (-mgr\cos\theta) \frac{\sin\theta}{r} - mr\dot{\theta}^2\cos\theta \\ &= \cos\theta [m\ddot{r} + mg\sin\theta - mr\dot{\theta}^2] \end{aligned}$$

Notice the bracketed expression is the first Euler-Lagrange equation, and must be zero. Thus we have demonstrated the conservation law $\dot{p} = 0$,

$$\frac{d}{dt} \left(m \frac{d}{dt} (r\cos\theta) \right) = 0,$$

for this particular example of a particle in gravity expressed in polar coordinates.⁹⁹

As this example suggests, verifying momentum conservation in this coordinate system (where we are without the $\frac{\partial L}{\partial q_k} = 0$ condition) is not easy, even for a very simple system. Furthermore, it is unlikely that we would have been able to predict this conserved quantity without some intuition from the (x, y) coordinate representation. Rather than relying on coordinates that easily reveal conserved momenta, we would rather have a tool that efficiently determines conserved momenta in any coordinate system we desire.¹⁰⁰

⁹⁸ Stating the obviously, the system dynamics have not changed physically and the conserved momentum *still exists*, but its conservation is less apparent in this coordinate system.

⁹⁹ Unlike the Hamiltonian, if it is going to be this much work to find a conserved quantity, it frankly may not be worth it. This is why Noether's theorem, discussed next, is so important.

¹⁰⁰ Enter Emily Noether ...

Noether's Theorem

Theorem (Pseudo-Noether's Theorem). *There exists a conserved quantity p for each transformation $G(q)$ that locally leaves $L(q, \dot{q})$ invariant.*

The first thing for us to do is parse some of the words above into precise mathematical statements. A transformation on q is a mapping that takes an n dimensional vector q and returns an n -dimensional vector. But since we are concerned with *local* transformations, $G(q)$ only describes something happening *near* q . So the transformation should be thought of as starting from q and then moving $G(q)$ away from q —the notation for this is sometimes $q \mapsto q + \epsilon G(q)$.¹⁰¹ Now what does it mean for L to be *locally invariant* to the transformation $G(q)$? In words, this should mean that L *does not change* if we change q a small amount.¹⁰² What does this remind you of? It is the same thing as asking if the derivative of L with respect to ϵ is zero—that is, it is the same as requiring that the directional derivative of L in the direction G is zero. We can now restate Noether's theorem in a more mathematically precise way.

Theorem (Noether's Theorem). *There exists a conserved quantity p for each transformation $G(q)$ where $\frac{d}{d\epsilon} L(q_\epsilon, \dot{q}_\epsilon)|_{\epsilon=0} = 0$ and $q_\epsilon = q + \epsilon G(q)$.*

Proof of Noether's Theorem

Define $L_\epsilon = L\left(q_\epsilon, \frac{d}{dt}(q_\epsilon)\right)$ and, from our assumption in the theorem, we know that $\frac{\partial L_\epsilon}{\partial \epsilon}|_{\epsilon=0} = 0$. Moreover,

$$\begin{aligned}\frac{\partial L_\epsilon}{\partial \epsilon} &= \frac{\partial L}{\partial q_\epsilon} \frac{\partial q_\epsilon}{\partial \epsilon} + \frac{\partial L}{\partial \dot{q}_\epsilon} \frac{\partial \dot{q}_\epsilon}{\partial \epsilon} \\ &= \frac{\partial L}{\partial q_\epsilon} G(q) + \frac{\partial L}{\partial \dot{q}_\epsilon} \dot{G}(q)\end{aligned}$$

which implies

$$\frac{\partial L_\epsilon}{\partial \epsilon}|_{\epsilon=0} = \frac{\partial L}{\partial q} G(q) + \frac{\partial L}{\partial \dot{q}} \dot{G}(q).$$

Use the Euler-Lagrange Equations to rewrite the first term

$$= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) G(q) + \frac{\partial L}{\partial \dot{q}} \dot{G}(q)$$

By product rule this is equivalent to

$$= \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} G(q) \right).$$

So, we know that $\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} G(q) \right) = 0$, implying that $p = \frac{\partial L}{\partial \dot{q}} G(q)$ is constant. \square

Not only does the proof of Noether's theorem tell us that a conserved quantity exists, it also tells us a formula for what it is—the conserved quantity is the scalar $p = \frac{\partial L}{\partial \dot{q}} G(q)$.¹⁰³

¹⁰¹ This can be read as “ q maps to $q + \epsilon G(q)$.”

¹⁰² I.e., $q \mapsto q + \epsilon G(q)$ for ϵ very small.

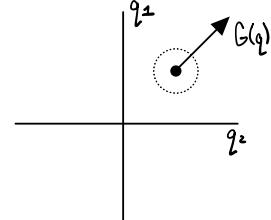


Figure 20: Noether's Theorem requires the invariance to hold *locally* around q , indicating that a derivative will be used.

¹⁰³ Note that we are overloading the variable p here. Earlier, the vector-valued quantity $p = \frac{\partial L}{\partial \dot{q}}$ in the Legendre transform was called the generalized momentum—a vector of terms, with no guarantees that any of them are conserved. Here we also call the conserved scalar p , because it is a momentum-like quantity. (And p really is a scalar because $\frac{\partial L}{\partial \dot{q}}$ is a row vector and $G(q)$ is a column vector, so they combine to form a scalar.)

Example

Particle in rectangular coordinates with gravitation potential:

Let's again return to the particle in gravity, first represented in $q = (x, y)$ coordinates. In the context of Noether's theorem, consider the transformation given by $G(q) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$,

$$q_\epsilon = \begin{bmatrix} x_\epsilon \\ y_\epsilon \end{bmatrix} = q + \epsilon \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} x + \epsilon \\ y \end{bmatrix}$$

This choice of a constant $G(q)$ does not explicitly depend on q , but that is allowed. Look at the evaluation of the directional derivative of L in the direction $G(q)$ —that is $\frac{d}{d\epsilon} L_\epsilon$.

$$\begin{aligned} L_\epsilon &= \frac{1}{2}m \left(\left(\frac{d}{dt}(x + \epsilon) \right)^2 + \left(\frac{d}{dt}y \right)^2 \right) - mgy \\ &= \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy \end{aligned}$$

This means that $\frac{d}{d\epsilon} L_\epsilon = 0$, since L_ϵ has no dependence on ϵ , and therefore $L(q, \dot{q})$ is locally invariant under the transformation q_ϵ .¹⁰⁴ Noether's theorem then gives us the ability to determine the conservation quantity:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} G(q) \right) = \frac{d}{dt} \left(\begin{bmatrix} m\dot{x} & m\dot{y} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right) = \frac{d}{dt} (m\dot{x}) = 0.$$

This confirms our earlier analysis of this problem, that $p = m\dot{x}$ is conserved.¹⁰⁵

Particle in rectangular coordinates with spring potential:

Now we do the same analysis with a different potential $V(x, y) = \frac{1}{2}k(x^2 + y^2)$. This gives us the following Lagrangian.

$$L = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - \frac{1}{2}k(x^2 + y^2)$$

What types of transformations locally leave L invariant? Rotation about the origin keeps the Lagrangian constant, so we focus on rotation.

Let

$$R(\epsilon) \cdot (x, y)^T = \begin{bmatrix} \cos \epsilon & -\sin \epsilon \\ \sin \epsilon & \cos \epsilon \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \cos \epsilon - y \sin \epsilon \\ x \sin \epsilon + y \cos \epsilon \end{bmatrix}$$

¹⁰⁴ The transformation that corresponds physically to a translation of the x coordinate.

¹⁰⁵ In fact, we can make a more general connection between Noether's theorem and systems satisfying the $\frac{\partial L}{\partial \dot{q}_k} = 0$ condition. Notice that choosing any constant G yields $\dot{q}_\epsilon = \frac{d}{dt}(q + \epsilon G) = \dot{q}$. For a system with $\frac{\partial L}{\partial \dot{q}_k} = 0$ for some q_k , consider $G = e_k$ where e_k is a unit vector in the direction of the k -th coordinate. This yields

$$\begin{aligned} L_\epsilon &= L(q_\epsilon, \dot{q}) \\ &= L(q, \dot{q}) \end{aligned}$$

where the first equality comes from $\dot{q}_\epsilon = \dot{q}$ and the second comes from the $\frac{\partial L}{\partial \dot{q}_k} = 0$ condition (L is independent of q_k , and the only change between q and q_ϵ is a difference in that coordinate). Thus for all systems satisfying $\frac{\partial L}{\partial \dot{q}_k} = 0$, the Lagrangian remains fixed under translations in the q_k direction and the transformation given by $G = e_k$ is valid in the application Noether's theorem. With $G = e_k$ we have the conservation law

$$0 = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} G(q) \right) = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} e_k \right) = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_k} \right)$$

indicating the familiar result $\frac{\partial L}{\partial \dot{q}_k} = p_k$ is constant. We've demonstrated Noether's theorem gives reliable results in this familiar case, what can it tell us when we don't have $\frac{\partial L}{\partial \dot{q}_k} = 0$?

with ϵ fixed (constant) and small. This changes L in the following manner:

$$\begin{aligned}
 L_\epsilon & \left(R(\epsilon) \cdot (x, y)^T, \frac{d}{dt} \left(R(\epsilon) \cdot (x, y)^T \right) \right) \\
 &= \frac{1}{2} m \left(\left(\frac{d}{dt} (x \cos \epsilon - y \sin \epsilon) \right)^2 + \left(\frac{d}{dt} (x \sin \epsilon + y \cos \epsilon) \right)^2 \right) \\
 &\quad - \frac{1}{2} k \left((x \cos \epsilon - y \sin \epsilon)^2 + (x \sin \epsilon + y \cos \epsilon)^2 \right) \\
 &= \frac{1}{2} m \left((\dot{x} \cos \epsilon - \dot{y} \sin \epsilon)^2 + (\dot{x} \sin \epsilon + \dot{y} \cos \epsilon)^2 \right) \\
 &\quad - \frac{1}{2} k \left((x \cos \epsilon - y \sin \epsilon)^2 + (x \sin \epsilon + y \cos \epsilon)^2 \right) \\
 &= \frac{1}{2} m \left((\sin^2 \epsilon + \cos^2 \epsilon) \dot{x}^2 + (\sin^2 \epsilon + \cos^2 \epsilon) \dot{y}^2 \right) \\
 &\quad - \frac{1}{2} k \left((\sin^2 \epsilon + \cos^2 \epsilon) x^2 + (\sin^2 \epsilon + \cos^2 \epsilon) y^2 \right) \\
 &= \frac{1}{2} m (\dot{x}^2 + \dot{y}^2) - \frac{1}{2} k (x^2 + y^2).
 \end{aligned}$$

This means that $L = L_\epsilon$ for any choice of ϵ — $L(q, \dot{q})$ remains fixed under the rotation $R(\epsilon)$, implying that it must be locally invariant.¹⁰⁶ Note that in this case, the transformation *globally* keeps L fixed, but we only have to find one that *locally* does.¹⁰⁷

However, even though we know that there must be a momentum that is conserved, the transformation $R(\epsilon) \cdot (x, y)^T$ does not have the structure that our proof of Noether's theorem requires.¹⁰⁸ To find $G(q)$, we can take the derivative of $R(\epsilon) \cdot (x, y)^T$ to approximate it locally.¹⁰⁹

$$q_\epsilon = \begin{bmatrix} x_\epsilon \\ y_\epsilon \end{bmatrix} = \begin{bmatrix} 1 & -\epsilon \\ \epsilon & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\epsilon \begin{bmatrix} -y \\ x \end{bmatrix}}_{G(q)}$$

To apply Noether's theorem, we now evaluate $\frac{d}{d\epsilon} L_\epsilon|_{\epsilon=0}$, where $L_\epsilon = L(q_\epsilon, \dot{q}_\epsilon)$.

$$\begin{aligned}
 L_\epsilon &= L(q_\epsilon, \dot{q}_\epsilon) = \frac{1}{2} m \left(\left(\frac{d}{dt} (x - \epsilon y) \right)^2 + \left(\frac{d}{dt} (y + \epsilon x) \right)^2 \right) \\
 &\quad - \frac{1}{2} k \left((x - \epsilon y)^2 + (y + \epsilon x)^2 \right) \\
 &= \frac{1}{2} m \left((\dot{x} - \epsilon \dot{y})^2 + (\dot{y} + \epsilon \dot{x})^2 \right) - \frac{1}{2} k \left((x - \epsilon y)^2 + (y + \epsilon x)^2 \right) \\
 &= \frac{1}{2} m \left(\dot{x}^2 + \dot{y}^2 + \epsilon^2 (\dot{x}^2 + \dot{y}^2) \right) - \frac{1}{2} k \left(x^2 + y^2 + \epsilon^2 (x^2 + y^2) \right) \\
 &= L(q, \dot{q}) + \epsilon^2 L(q, \dot{q}).
 \end{aligned}$$

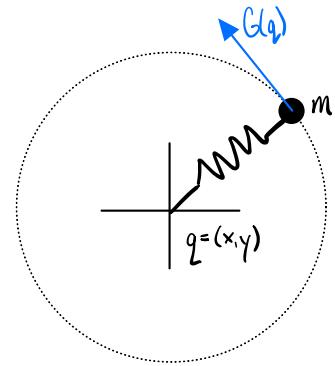


Figure 21: For point mass with a radial spring potential, symmetries are rotational.

¹⁰⁶ When a transformation exists that keeps L fixed, we call it a *symmetry*.

¹⁰⁷ In general, finding the transformation is a big challenge!

¹⁰⁸ It is not of the form $q \mapsto q + \epsilon G(q)$.

¹⁰⁹ This is the same thing as using a “small angle” approximations ($\sin \epsilon \approx \epsilon$ and $\cos \epsilon \approx 1$).

$$\implies \frac{d}{d\epsilon} L_\epsilon|_{\epsilon=0} = 0$$

So the choice $G(q) = \begin{bmatrix} -y \\ x \end{bmatrix}$ that defines the transformation q_ϵ is valid in Noether's theorem and the corresponding conservation law is

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} G(q) \right) = \frac{d}{dt} \left(\begin{bmatrix} m\dot{x} & m\dot{y} \end{bmatrix} \begin{bmatrix} -y \\ x \end{bmatrix} \right) = \frac{d}{dt} (m(x\dot{y} - y\dot{x})) = 0.$$

One might recognize the conserved quantity as the system's *angular momentum* about the origin. Physically this conservation law is intuitive in the sense that the spring force always points into the origin and thus can never exert a torque about the origin. There are no other potential sources of torque about this point, and thus the angular momentum about it is constant. Nevertheless, there are instances of conserved quantities that are *not* intuitive.¹¹⁰

The triple cart pendulum

As a last example, what would $G(q)$ be for the triple cart-pendulum shown in Fig. 23?

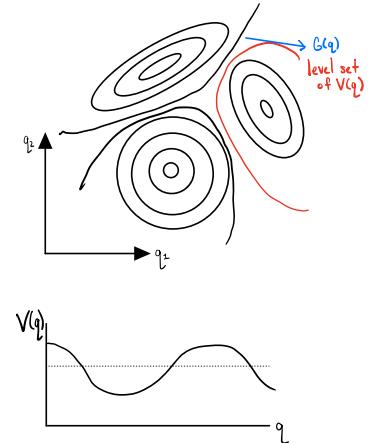


Figure 22: Why would any differentiable $V(q)$ necessarily have a conserved quantity?

¹¹⁰ As an example, think about conditions on V in $L = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - V(x, y)$ that would guarantee a conserved momentum. Hint: so long as V is differentiable in x , a conserved quantity must exist. Why?

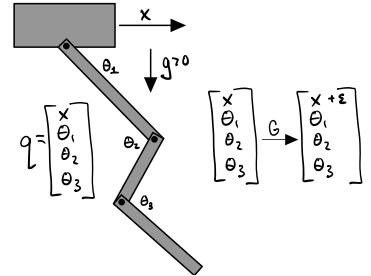


Figure 23: Consider the triple pendulum attached to a cart. What does $G(q)$ look like?

Aside: A pendulum using the configuration $q = z \in \mathbb{C}$

The discussion of Noether's theorem natural brings up *rotation* as a very important type of motion, both in terms of degrees of freedom¹¹¹ and in terms of symmetries/consered quantitites. One of the topics we will dig deeper into later is how to *represent* rotation. We have already seen two representations, one using angles as generalized coordinate and hte other using Cartesian coordinates with constraints.¹¹² Both of these choices are reasonably intuitive—we have experience with both angles and positions. But neither of these choices are particularly good—angles ‘wrap’, so that $\theta = \pi$ is the same configuration as $\theta = 3\pi$, and Cartesian coordinates are ‘not natural’ for rotation and therefore require a constraint. It turns out that complex numbers $z \in \mathbb{C}$ are natural for describing rotation, and in this section I'm going to explain why and derive some equations for a pendulum using complex numbers *as the configuration*.

First, some facts about complex numbers.

$$z = x + jy \text{ where } j = \sqrt{-1}$$

$$\bar{z} = x - jy$$

The \bar{z} is called the *complex conjugate* of z . The unit circle on \mathbb{C} is described by

$$x^2 + y^2 = z\bar{z} = 1$$

so \bar{z} is the inverse of z when constrained to the unit circle. That is,

$$\bar{z} = z^{-1} \text{ if } \|z\| = 1.$$

Note that $z^{-1}\dot{z}$ is a pure imaginary number. You can see this by direct computation.

$$z\bar{z} = 1 \implies \dot{z}\bar{z} + z\dot{\bar{z}} = 0 \implies \dot{z}\bar{z} = -z\dot{\bar{z}}$$

$$\dot{z}\bar{z} = -z\dot{\bar{z}}$$

$$(\dot{x} + j\dot{y})(x - jy) = -(x + jy)(\dot{x} - j\dot{y})$$

$$(x\dot{x} + y\dot{y}) + j(x\dot{y} - y\dot{x}) = -(x\dot{x} + y\dot{y}) - j(y\dot{x} - x\dot{y})$$

Note that this implies $(x\dot{x} + y\dot{y}) = -(x\dot{x} + y\dot{y})$, so $(x\dot{x} + y\dot{y}) = 0$, implying that $z^{-1}\dot{z}$ is a pure imaginary number. Therefore $\bar{z}\dot{z}$ and $\dot{z}\bar{z}$ are both pure imaginary numbers as well. Moreover, $z^{-1}\delta z$, where δz is a variation to z , must also be pure imaginary.¹¹³

Define

$$\nu = z^{-1}\dot{z} \text{ and } \eta = z^{-1}\delta z.$$

Note that

$$KE = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2)$$

¹¹¹ manhy mechanical systems have rotational degrees of freedom...

¹¹² e.g., for a pendulum a) $q = \theta$ gives us

$$\ddot{\theta} = -\frac{g}{L} \sin(\theta)$$

and b) (x, y) with $\phi(x, y) = x^2 + y^2 - r^2$ gives us

$$\begin{aligned} \lambda &= \frac{mgy - m(\dot{x}^2 + \dot{y}^2)}{2R^2} \\ \ddot{x} &= \frac{x(gy - (\dot{x}^2 + \dot{y}^2))}{R^2} \\ \ddot{y} &= \frac{-gx^2 - y(\dot{x}^2 + \dot{y}^2)}{R^2}. \end{aligned}$$

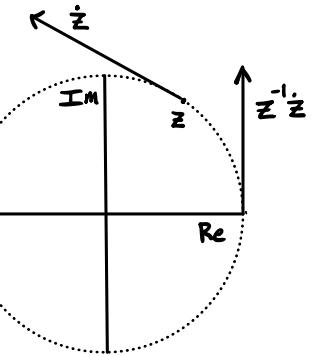


Figure 24: The mapping $z^{-1}\dot{z}$ maps the complex vector \dot{z} back to tangent to the unit circle at $z = 1$, making it a pure imaginary number. This mapping can take place at every z , so as the pendulum evolves, its accelerations can always be related (through z^{-1} to a pure imaginary number).

¹¹³ Remember that when one applies the operation δ to the product $z^{-1}z = 1$, all the same outcomes as differentiation with respect to time will apply.

is the kinetic energy of the mass. Moreover,¹¹⁴

$$\begin{aligned} -\frac{1}{2}mv^2 &= +\frac{1}{2}mv\bar{v} \\ &= \frac{1}{2}mz^{-1}\dot{z}\overline{z^{-1}\dot{z}} \\ &= \frac{1}{2}m\underbrace{z^{-1}\overline{z^{-1}}}_{=1}\underbrace{\dot{z}\bar{z}}_{||\dot{z}||^2} \\ &= \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) \\ &= KE, \end{aligned}$$

implying we can write the kinetic energy using $KE = -\frac{1}{2}mv^2$. So, to evaluate the pendulum dynamics¹¹⁵ we look at

$$A = \int_{t_0}^{t_f} \underbrace{-\frac{1}{2}mv^2}_{\ell(v)} dt \implies \delta A = \int_{t_0}^{t_f} -mv\delta v dt.$$

Moreover, we can determine δv from the definitions $v = z^{-1}\dot{z}$ and $\eta = z^{-1}\delta z$ above by differentiating the first expression using δ ¹¹⁶ and the second with respect to time.

$$\begin{aligned} \delta v &= -z^{-2}\delta z\dot{z} + z^{-1}\delta\dot{z} \\ \dot{\eta} &= -z^{-2}\dot{z}\delta z + z^{-1}\delta\dot{z} \end{aligned}$$

To get rid of the right hand terms, subtract the bottom equation from the top equation.¹¹⁷

$$\delta v - \dot{\eta} = -z^{-2}\delta z\dot{z} - (-z^{-2}\dot{z}\delta z) = 0 \implies \delta v = \dot{\eta}$$

This is equal to zero because complex number multiplication commutes.¹¹⁸ So we get

$$\delta A = \int_{t_0}^{t_f} -mv\delta v dt = \underbrace{-mv\eta|_{t_0}^{t_f}}_{=0} - \int_{t_0}^{t_f} -m\dot{v}\eta dt = \int_{t_0}^{t_f} m\dot{v}\eta dt$$

Finally, we have

$$\delta A = \int_{t_0}^{t_f} m\dot{v}\eta dt = 0 \quad \forall \eta \text{ s.t. } \eta(t_0) = \eta(t_f) = 0$$

Just as we did before at the very beginning of this class, this implies that

$$m\dot{v} = 0.$$

This, together with the equation for z gives us the ordinary differential equation governing the pendulum with no gravity.

¹¹⁴ This first minus sign is here because v is pure imaginary, so v^2 will have a minus sign that needs to be canceled.

¹¹⁵ without gravity, for now...

¹¹⁶ remember that δ is the derivative with respect to a perturbation that satisfies $\eta(t_0) = \eta(t_f) = 0$

¹¹⁷ I promise this is going somewhere!

¹¹⁸ When we do this later in \mathbb{R}^3 using $g \in SE(3)$, this commutation will not hold.

$$\begin{aligned}\dot{z} &= z\nu \\ m\dot{\nu} &= 0\end{aligned}$$

Notice that the circular constraint is taken into account just by the multiplication properties of complex numbers, rather than having to enforce a constraint or using angles. Compare these equations to the equations in the side note above using $q = \theta$ and $q = (x, y)$ —these equations are *much* simpler precisely because complex multiplication ($z\nu$) “naturally” encodes rotation. We will see a similar story later when we look at Euler’s Equations—using a representation of rotation matrices; indeed, these equations for the pendulum *are* Euler’s equations in \mathbb{R}^2 .¹¹⁹

¹¹⁹ A great exercise is to show that if you add gravity—in $q \in \mathbb{C}$ a potential dependent on the imaginary value of z —you will get equations of motion

$$\begin{aligned}\dot{z} &= z\nu \\ \dot{\nu} &= -gjRe(z).\end{aligned}$$

(This should make sense given that $Re(z)$ is $\sin(\theta)$ in polar coordinates for complex numbers.)

Leibniz Rule and Impacts

Mechanical impacts play an important role in how a mechanism (e.g., a robot) interacts with its environment. Impacts can be incidental—e.g., two objects colliding unintentionally—or purposeful—e.g., feet impacting the ground during locomotion. Most of the time we learn about impacts in the context of a single particle impacting a surface or two particles impacting each other.

For instance, a student may have seen something like what is shown in Fig. 25, where a point mass is impacting the wall, and ‘conservation of momentum’ implies that ‘angles of equal incidence’—that is, that the incoming angle θ_1 is the same as the outgoing angle θ_2 . This is a disservice to students, since it builds up very incorrect intuition about what happens during impact. Indeed, it is obvious that the linear momentum in the direction orthogonal to the wall is *not* conserved, making the statement about momentum and energy conservation heuristic at best. In Fig. 26, a double pendulum is swinging so that its second link will hit the floor—how will its motion change post impact? Hopefully it is intuitive that no ad hoc reasoning about ‘angles of equal incidence’ or conservation of linear momentum can be naively applied here. And if those ideas cannot generalize to a two link pendulum, we need more general ideas, mathematical tools, and intuition. In this chapter, we will remedy the situation.

Applying the Variational Principle to Impacts

Suppose we have a system with Lagrangian $L(q, \dot{q})$ that is evolving over a trajectory $q(t)$ from time $t = 0$ to T . Assume, also, that the trajectory is continuous everywhere and differentiable everywhere, except at a point $\tau \in [0, T]$, at which a collision occurs, and a discontinuity in \dot{q} is allowed. The action is

$$A(q, \dot{q}, \tau) = \int_0^T L(q, \dot{q}) dt = \int_0^\tau L(q, \dot{q}) dt + \int_\tau^T L(q, \dot{q}) dt,$$

where, for reasons soon to become evident, we split the integral in the terms before and after the collision, keeping track of the ‘time of

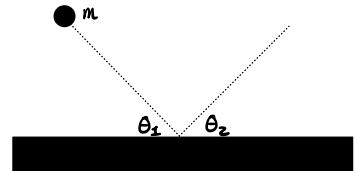


Figure 25: The elastic impact of a point mass with a surface leads to “angles of equal incidence”—that is, $\theta_1 = \theta_2$. This is misleading.

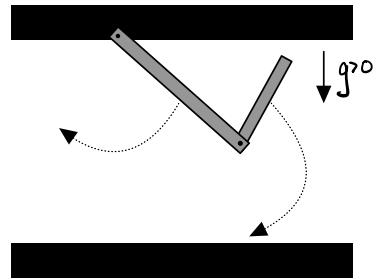
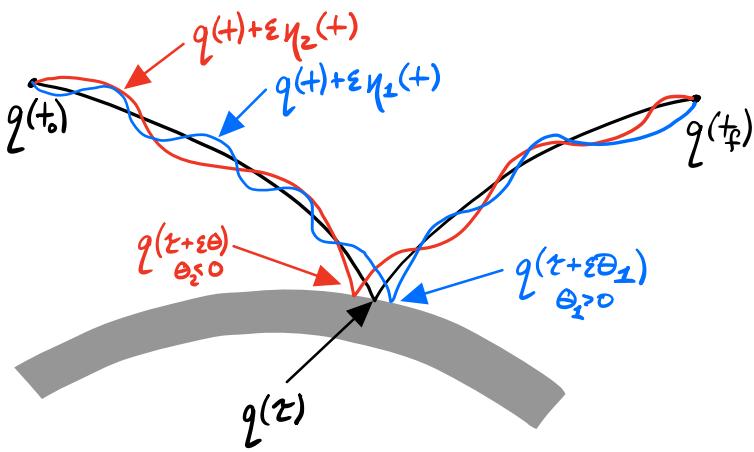


Figure 26: What will happen with this double pendulum experiences an impact?

impact' τ .



To find the equations of motion, we want to take the *exact* same approach as we did in the non-impacting case—set δA to zero for all possible variations in the trajectory. To do this, we consider variations $\eta(t)$, as before, and introduce variations θ^{120} to the time of impact τ .

$$A(\underbrace{q + \varepsilon\eta}_{q_\varepsilon}, \underbrace{\dot{q} + \varepsilon\dot{\eta}}_{\dot{q}_\varepsilon}, \underbrace{\tau + \varepsilon\theta}_{\tau_\varepsilon}) = \int_0^{\tau_\varepsilon} L(q_\varepsilon, \dot{q}_\varepsilon) dt + \int_{\tau_\varepsilon}^T L(q_\varepsilon, \dot{q}_\varepsilon) dt$$

Looking at Fig. 27, it should be clear that $\eta(t)$ and θ are related—some $\eta(t)$ will force θ to be positive¹²¹ and some $\eta(t)$ will force θ to be negative.¹²² Thus we have (where we will constrain η and θ to vary in a physically meaningful way later)

$$\begin{aligned} \delta A \cdot (\eta, \theta)^T &= \frac{d}{d\varepsilon} \left[\int_0^{\tau_\varepsilon} L(q_\varepsilon, \dot{q}_\varepsilon) dt + \int_{\tau_\varepsilon}^T L(q_\varepsilon, \dot{q}_\varepsilon) dt \right]_{\varepsilon=0} \\ &= \left[\int_0^{\tau^-} \left(\frac{\partial L}{\partial q_\varepsilon} \cdot \eta + \frac{\partial L}{\partial \dot{q}_\varepsilon} \cdot \dot{\eta} \right) dt + L(q_\varepsilon, \dot{q}_\varepsilon) \theta \Big|_{\tau^-} + \int_{\tau^+}^T \left(\frac{\partial L}{\partial q_\varepsilon} \cdot \eta + \frac{\partial L}{\partial \dot{q}_\varepsilon} \cdot \dot{\eta} \right) dt - L(q_\varepsilon, \dot{q}_\varepsilon) \theta \Big|_{\tau^+} \right]_{\varepsilon=0} \end{aligned}$$

where the non-integral terms comes from a combination of Leibniz rule and chain rule

$$\begin{aligned} &= \left[\int_0^{\tau^-} \left(\frac{\partial L}{\partial q_\varepsilon} \cdot \eta - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_\varepsilon} \cdot \eta \right) dt + \left(\frac{\partial L}{\partial \dot{q}_\varepsilon} \cdot \eta + L(q_\varepsilon, \dot{q}_\varepsilon) \theta \right) \Big|_{\tau^-} \right]_{\varepsilon=0} \\ &\quad + \left[\int_{\tau^+}^T \left(\frac{\partial L}{\partial q_\varepsilon} \cdot \eta - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_\varepsilon} \cdot \eta \right) dt - \left(\frac{\partial L}{\partial \dot{q}_\varepsilon} \cdot \eta + L(q_\varepsilon, \dot{q}_\varepsilon) \theta \right) \Big|_{\tau^+} \right]_{\varepsilon=0} \end{aligned}$$

where the $\frac{\partial L}{\partial \dot{q}_\varepsilon} \cdot \eta$ terms *not* under the integral come from integration by parts and the fact

that $\eta(\tau)$ need not be zero

$$= \int_0^{\tau^-} \left(\frac{\partial L}{\partial q} \cdot \eta - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \cdot \eta \right) dt - \left[\frac{\partial L}{\partial \dot{q}} \cdot \eta + L(q, \dot{q}) \theta \right]_{\tau^-}^{\tau^+} + \int_{\tau^+}^T \left(\frac{\partial L}{\partial q} \cdot \eta - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \cdot \eta \right) dt.$$

Figure 27: $q(t)$ and $\eta(t)$ have to both be in the free space where $\phi(q) \geq 0$, so the impact time and the impact location of the variations are constrained.

¹²⁰ This θ is not a configuration, it is just a variation/perturbation to τ .
This overloading of θ is a cumbersome aspect of choosing another greek letter that is familiar to be paired with τ .

¹²¹ indicating a variation to impact that is forcing the impact to occur *later* than τ ...

¹²² indicating a variation to impact that is forcing the impact to occur *earlier* than τ ...

Notice that the two integrals must be zero,¹²³ as both are over intervals on which the trajectory is not experiencing an impact, a case where we have already applied the variational principle, concluding that the Euler-Lagrange equations must hold on those intervals. We have, then, *at the time of impact τ*

$$\delta A \cdot (\eta, \theta)^T = - \left[\frac{\partial L}{\partial \dot{q}} \cdot \eta + L(q, \dot{q}) \theta \right]_{\tau^-}^{\tau^+},$$

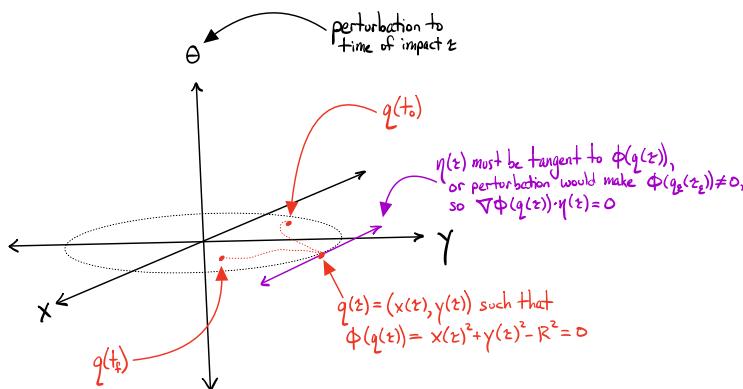
which has to vanish for all possible (η, θ) pairs at time τ .¹²⁴

The equation is evaluated only at the time of impact, for which we know that $\phi[q(\tau)] = 0$ —the system is experiencing contact with the surface at the time of impact. The derivative of this relation must also hold,¹²⁵ $\delta\phi[q(\tau)] = 0$. We calculate just as before

$$\begin{aligned} \frac{d}{d\varepsilon} \phi[q_\varepsilon(\tau_\varepsilon)] \Big|_{\varepsilon=0} &= 0, \\ \frac{d}{d\varepsilon} \phi[q(\tau + \varepsilon\theta) + \varepsilon\eta(\tau + \varepsilon\theta)] \Big|_{\varepsilon=0} &= 0, \\ \frac{\partial \phi}{\partial q_\varepsilon(\tau_\varepsilon)} \cdot \left[\frac{d}{d\varepsilon} (q(\tau + \varepsilon\theta) + \varepsilon\eta(\tau + \varepsilon\theta)) \right] \Big|_{\varepsilon=0} &= 0, \end{aligned}$$

where the “.” symbol has been included to indicate that a row vector is being composed with a column vector.

$$\begin{aligned} \frac{\partial \phi}{\partial q_\varepsilon(\tau_\varepsilon)} \cdot \left[\dot{q}(\tau_\varepsilon)\theta + \underbrace{\eta(\tau_\varepsilon) + \varepsilon\dot{\eta}(\tau_\varepsilon)\theta}_{\text{result of product rule}} \right] \Big|_{\varepsilon=0} &= 0, \\ \frac{\partial \phi}{\partial q} \cdot [\dot{q}(\tau)\theta + \eta(\tau)] &= 0 \end{aligned}$$



It is worth regrouping at this point and thinking about what we have shown so far. At this point, at the time of impact τ , the varia-

¹²³ The integral over $[0, \tau)$ and the integral $[\tau, T]$.

¹²⁴ The student may have been wondering up until now why I am using the notation τ_- and τ_+ . Both of these evaluate to τ , but in this equation it is more clear that they are *labels* that allow us to distinguish between the state *before* impact and the state *after* impact, both of which are evaluated at τ .

¹²⁵ We used this same trick when evaluating the constrained Euler-Lagrange Eqs.

Figure 28: τ is related to where the impact occurs.

tional principles tells us the following.

$$\delta A \cdot (\eta, \theta)^T = - \left[\frac{\partial L}{\partial \dot{q}} \cdot \eta + L(q, \dot{q}) \theta \right]_{\tau^-}^{\tau^+}$$

such that the variations $\eta(t)$ and θ satisfy the following equation.

$$\frac{\partial \phi}{\partial q} \cdot [\dot{q}(\tau)\theta + \eta(\tau)] = 0$$

The interpretation of this constraint can be seen in Fig. 28, where a circular impact condition in $q = (x, y)$ is shown, along with variations $\eta(t)$ and θ . At the time of impact, $\eta(t)$ must be tangent to the impact condition and perturbations θ will vary as well. How can we see this geometric intuition in the formula?

Note now that the space of all possible $(\eta(\tau), \theta)$ has $n + 1$ dimensions, since we are only considering η at time τ , $\eta(\tau)$ is n dimensional, and θ is a scalar. Imposing the condition

$$\frac{\partial \phi}{\partial q} \cdot [\dot{q}(\tau)\theta + \eta(\tau)] = 0$$

reduces the dimensionality of combinations of $\eta(t)$ and τ by one.¹²⁶ What we do now is split the variations $(\eta(\tau), \theta)$ into two cases.

1. First, look at the pairs $(\eta(\tau), 0)$ with $\frac{\partial \phi}{\partial q} \cdot \eta(\tau) = 0$.¹²⁷ The space of $\eta(\tau)$ is n -dimensional, which implies we have exactly $n - 1$ linear independent pairs that satisfy $\frac{\partial \phi}{\partial q} \cdot \eta(\tau) = 0$.
2. It follows that *any* other pair of $(\eta(\tau), \theta)$ that is linearly independent of the constrained pair in the above case is sufficient to describe the space of allowable perturbations. Choose that pair be $(\dot{q}(\tau), -1)$, which is indeed linearly independent of $(\eta(\tau), 0)$ and satisfies the equation $\frac{\partial \phi}{\partial q} \cdot [\dot{q}(\tau)\theta + \eta(\tau)] = 0$.

So the set of vectors in the space of $(\eta(\tau), \theta)$ pairs that spans the subspace that satisfy the constraint $\frac{\partial \phi}{\partial q} [\dot{q}(\tau)\theta + \eta(\tau)] = 0$ are below

$$(\eta(\tau), \theta) = \begin{cases} (\eta, 0) \text{ such that } \frac{\partial \phi}{\partial q} \cdot \eta(\tau) = 0 \\ (\dot{q}, -1) \end{cases}$$

The action principle can now be evaluated, using these variations to incorporate the relationship $\frac{\partial \phi}{\partial q} [\dot{q}(\tau)\theta + \eta(\tau)] = 0$.

$$0 = \delta A \cdot (\eta(\tau), 0)^T = - \frac{\partial L}{\partial \dot{q}} \cdot \eta \Big|_{\tau^-}^{\tau^+}, \quad \forall \eta(\tau) \text{ s.t. } \frac{\partial \phi}{\partial q} \cdot \eta(\tau) = 0,$$

$$0 = \delta A \cdot (\dot{q}, -1)^T = - \left[\frac{\partial L}{\partial \dot{q}} \cdot \dot{q} - L(q, \dot{q}) \right]_{\tau^-}^{\tau^+}$$

¹²⁶ unless $\nabla \phi = 0$, which doesn't make sense if there is a constraint...

¹²⁷ This constraint is $\frac{\partial \phi}{\partial q} \cdot [\dot{q}(\tau)\theta + \eta(\tau)] = 0$ with $\theta = 0$.

We can read the first equation as requiring that the operator $\frac{\partial L}{\partial \dot{q}}$ be orthogonal to all $\eta(\tau)$ that are orthogonal to $\frac{\partial \phi}{\partial q}$.¹²⁸ The equations then become

$$\begin{aligned}\frac{\partial L}{\partial \dot{q}}\Big|_{\tau^-}^{\tau^+} &= \lambda \frac{\partial \phi}{\partial q} \\ \left[\frac{\partial L}{\partial \dot{q}} \cdot \dot{q} - L(q, \dot{q}) \right]_{\tau^-}^{\tau^+} &= 0.\end{aligned}$$

¹²⁸ As was the case with the constrained Euler-Lagrange Equations, this is equivalent to saying that the two operators lie in the same direction.

This can also be written in terms of p and H .

$$\begin{aligned}p\Big|_{\tau^-}^{\tau^+} &= \lambda \frac{\partial \phi}{\partial q} \\ H\Big|_{\tau^-}^{\tau^+} &= 0.\end{aligned}$$

The first equation can be interpreted as restricting the change of momentum due to impact to lie perpendicular to the contact surface. The second equation simply states that the Hamiltonian (which is the energy in most cases) is conserved through the impact.

Example

Again, we first look at a point mass in gravity, in this case experiencing an impact with the ground. The configuration variables are x and y , and the Lagrangian is

$$L = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy.$$

Let the point hit a horizontal surface at $y = h$ from above, at time τ . We describe this surface by $\phi(x, y) = y - h$. The impact equations become

$$\begin{aligned}m\dot{x}(\tau^+) &= m\dot{x}(\tau^-) \\ m\dot{y}(\tau^+) &= m\dot{y}(\tau^-) + \lambda, \\ m\dot{x}^2(\tau^+) + m\dot{y}^2(\tau^+) &= m\dot{x}^2(\tau^-) + m\dot{y}^2(\tau^-).\end{aligned}$$

Here, $\dot{x}(\tau^-)$ and $\dot{y}(\tau^-)$ are the velocity of the point right before impact (known) and $\dot{x}(\tau^+)$ are $\dot{y}(\tau^+)$ the velocity right after impact (unknown). The equations have two solutions, but only one of them with a positive λ . Note that we restrict $\lambda \geq 0$ since the wall can only act away from its surface, and a negative λ would correspond to the

wall sucking the point in. The solution is,

$$\begin{aligned}\dot{x}(\tau^+) &= \dot{x}(\tau^-), \\ \dot{y}(\tau^+) &= -\dot{y}(\tau^-), \\ \lambda &= -2m\dot{y}(\tau^-).\end{aligned}$$

Algorithm Implementation for Elastic Impacts

However, the above are analytical and symbolic solutions for the impact update, when we simulate the system, how do we include impact in the simulation? First let's simulate the particle falling in gravity without impact (which you should be familiar with now):

```

1 import numpy as np
2 import sympy as sym
3 from sympy.abc import t
4 from sympy import Function, solve, Matrix, symbols
5 # define states
6 q = Matrix([Function('x')(t), Function('y')(t)])
7 qdot = q.diff(t)
8 qddot = qdot.diff(t)
9 # compute Lagrangian
10 m, g = 1, 9.8
11 ke = 0.5 * m * (qdot[0]**2 + qdot[1]**2)
12 pe = m * g * q[1]
13 L = ke - pe
14 print('Lagrangian:')
15 display(L)
16 # compute EL-equations (here we use SymPy's built-in method for simplicity)
17 from sympy.calculus.euler import euler_equations
18 el_eqns = euler_equations(L, func=[q[0], q[1]], vars=[t])
19 el_eqns = sym.Eq(Matrix([-el_eqns[0].lhs, -el_eqns[1].lhs]), Matrix([el_eqns
    [0].rhs, el_eqns[1].rhs]))
20 print('EL-equation(s):')
21 display(el_eqns)
22 # numerical evaluation
23 qddot_soln = solve(el_eqns, [qddot[0], qddot[1]], dict=True)[0]
24 xdot_func = sym.lambdify([q[0], q[1], qdot[0], qdot[1]], qddot_soln[qddot
    [0]])
25 ydot_func = sym.lambdify([q[0], q[1], qdot[0], qdot[1]], qddot_soln[qddot
    [1]])
26 # define dynamics for the particle
27 def particle_dyn(s):
28     return np.array([s[2], s[3], xdot_func(*s), ydot_func(*s)]) # argument
        '*s' is same as 's[0], s[1], s[2], s[3]'
29 s0 = np.array([1, 1, 0, 0])
30 print('[xdot, ydot, xdot, ydot] at initial condition [x={}, y={}, xdot={}, ydot={}]: [{} , {} , {} , {}]'.format(*s0, *particle_dyn(s0)))

```

If there is no impact, then next step is to simulate the trajectory based on the dynamics function, below is the pseudo-code for simulation without impact:

However, if there is impact, then for some iterations of the loop in Algorithm 1, we can not simply obtain s_{t+1} as $\text{integrate}(dyn, s_t)$ —before the integration, there should be an instantaneous update to s_t caused by the impact. This update is directly related to our previous

Algorithm 1: Simulating Trajectory without Impact

Input : $dyn(s)$: first-order dynamics of the system
 s_0 : initial condition
 N : length of simulation

Output: $traj$: vector/matrix representing the simulated trajectory

```

1  $t \leftarrow 0$  // initialize current time step
2  $traj \leftarrow [s_t]$  // initialize simulated trajectory
3 while  $t < N$  do
4    $s_{t+1} \leftarrow \text{integrate}(dyn, s_t)$ 
5    $t \leftarrow t + 1$ 
6   append( $traj$ ,  $s_t$ ) // insert the current state at the end of  $traj$ 
7 end while
8 return  $traj$  // return results
  
```

results:

$$\begin{aligned}\dot{x}(\tau^+) &= \dot{x}(\tau^-), \\ \dot{y}(\tau^+) &= -\dot{y}(\tau^-).\end{aligned}$$

Based on this, we can write the pseudo-code for simulation with impact:

Algorithm 2: Simulating Trajectory with Impact

Input : $dyn(s)$: first-order dynamics of the system
 s_0 : initial condition
 N : length of simulation

Output: $traj$: vector/matrix representing the simulated trajectory

```

1  $t \leftarrow 0$  // initialize current time step
2  $traj \leftarrow [s_t]$  // initialize simulated trajectory
3 while  $t < N$  do
4   if  $s_t$  will cause impact then
5      $s_t \leftarrow \text{impact\_update}(s_t)$  // instantaneous impact update:  $s_t$  on the
        right represents  $s(\tau^-)$ , the one on the left represents  $s(\tau^+)$ 
6     while  $s_t$  will cause impact do
7        $s_t \leftarrow \text{integrate}(dyn, s_t)$  // in practice one integration should end
          the loop
8     end while
9      $s_{t+1} \leftarrow s_t$ 
10   else
11      $s_{t+1} \leftarrow \text{integrate}(dyn, s_t)$ 
12   end if
13    $t \leftarrow t + 1$ 
14   append( $traj$ ,  $s_t$ ) // insert the current state at the end of  $traj$ 
15 end while
16 return  $traj$  // return results
  
```

In practice, for the above pseudo-code to work, we first need to write a function to decide when the impact would happen based on current state s_t , in order to do it, we need to define the constraint related to the impact:

```

1 # define impact constraint (assuming the particle hits the ground at y=0)
2 phi = q[1]-0
3 # numerically evaluate the constraint
  
```

```

4 # note that even phi is not in terms of qdot, we still include them in
5 # lambdify
6 phi_func = sym.lambdify([q[0], q[1], qdot[0], qdot[1]], phi)
7 # function for impact condition
8 # since we are numerically simulating the system, when impact happens, phi(
9     s)
10 # will not be exactly zero, thus we need to either catch the change of sign
11 # in
12 # phi(q) or set a threshold for that
13 def impact_condition(s, threshold=1e-1):
14     phi_val = phi_func(*s)
15     if phi_val > -threshold and phi_val < threshold:
16         return True
17     return False
18 # test the impact condition function
19 s_test = [1, 0.01, 1, 1]
20 print('impact will happen at [x={}, y={}, xdot={}, ydot={}]: {}'.format(*
21     s_test, impact_condition(s_test, 1e-1)))

```

After that, we need to have a (numerical) function which takes in a s_t and treats it as $s(\tau^-)$ to return $s(\tau^+)$. Recall that our equations for impact update are:

$$\frac{\partial L}{\partial \dot{q}} \Big|_{\tau^-}^{\tau^+} = \lambda \frac{\partial \phi}{\partial q},$$

$$\left[\frac{\partial L}{\partial \dot{q}} \cdot \dot{q} - L(q, \dot{q}) \right]_{\tau^-}^{\tau^+} = 0.$$

where $H(q, \dot{q}) = \left[\frac{\partial L}{\partial \dot{q}} \cdot \dot{q} - L(q, \dot{q}) \right]$ is the Hamiltonian of the system.
We can explicitly re-write the equations as:

$$\frac{\partial L}{\partial \dot{q}}(q(\tau^+), \dot{q}(\tau^+)) - \frac{\partial L}{\partial \dot{q}}(q(\tau^-), \dot{q}(\tau^-)) = \lambda \cdot \frac{\partial \phi}{\partial q}(q(\tau^-)),$$

$$H(q(\tau^+), \dot{q}(\tau^+)) - H(q(\tau^-), \dot{q}(\tau^-)) = 0.$$

Based on the equations above, let's first evaluate the related expressions at τ^- and τ^+ :

```

1 # define dummy variables for q(tau-), qdot(tau-), qdot(tau+)
2 # note that we don't need to define q(tau+), why?
3 xMinus, xdotMinus, yMinus, ydotMinus = symbols(r'x^-, \dot{x}^-, y^-, \dot{y}^-')
4 xdotPlus, ydotPlus = symbols(r'\dot{x}^+, \dot{y}^+')
5 # define substitution dictionary to evaluate expressions at tau- and tau+
6 # note that in "subs_plus", we replace q[0] and q[1] with xMinus and yMinus
    (why?)
7 subs_minus = {q[0]:xMinus, q[1]:yMinus, qdot[0]:xdotMinus, qdot[1]:ydotMinus
    }
8 subs_plus = {q[0]:xMinus, q[1]:yMinus, qdot[0]:xdotPlus, qdot[1]:ydotPlus}
9 # compute dLdqdot and evaluate it at tau- and tau+
10 dLdqdot = Matrix([L]).jacobian(qdot)
11 dLdqdot_Minus = dLdqdot.subs(subs_minus)
12 dLdqdot_Plus = dLdqdot.subs(subs_plus)
13 # compute dPhidq and evaluate it ONLY at tau- (why?)
14 dPhidq = Matrix([phi]).jacobian(q)
15 dPhidq_Minus = dPhidq.subs(subs_minus)

```

```

16 # compute Hamiltonian and evaluate it at tau- and tau+
17 H = (dLdqdot*qdot)[0] - L
18 H_Minus = H.subs(subs_minus)
19 H_Plus = H.subs(subs_plus)

```

Note that here we use dummy symbolic variables to substitute the state variables previously defined as symbolic functions. After evaluation at τ^- and τ^+ , we can easily write down the equations and solve them for $\dot{q}(\tau^+)$, the results will be the impact update rules:

```

1 # define equations for impact update
2 lamb = symbols('\lambda')
3 impact_eqns_lhs = Matrix([ dLdqdot_Plus[0]-dLdqdot_Minus[0], dLdqdot_Plus
    [1]-dLdqdot_Minus[1], H_Plus-H_Minus ])
4 impact_eqns_rhs = Matrix([ lamb*dPhidq_Minus[0], lamb*dPhidq_Minus[1], 0 ])
5 impact_eqns = sym.Eq(impact_eqns_lhs, impact_eqns_rhs)
6 print('equations for impact update:')
7 display(impact_eqns)
8 # solve them for qdot(tau+) and lamb
9 impact_solns = solve(impact_eqns, [xdotPlus, ydotPlus, lamb], dict=True)[0]
10 xdot_update_sol = impact_solns[xdotPlus]
11 ydot_update_sol = impact_solns[ydotPlus]
12 lamb_update_sol = impact_solns[lamb]
13 print('impact update solutions:')
14 display(sym.Eq(xdotPlus, xdot_update_sol))
15 display(sym.Eq(ydotPlus, ydot_update_sol))
16 display(sym.Eq(lamb, lamb_update_sol))

```

With the symbolic solutions for impact update, we can numerically evaluate them and define a function for the impact update in the simulation loop:

```

1 # numerically evaluate impact update solutions
2 xdot_update_func = sym.lambdify([xMinus, yMinus, xdotMinus, ydotMinus],
    xdot_update_sol)
3 ydot_update_func = sym.lambdify([xMinus, yMinus, xdotMinus, ydotMinus],
    ydot_update_sol)
4 # define impact update function
5 def impact_update(s):
6     return np.array([
7         s[0],
8         s[1], # q will be the same after impact
9         xdot_update_func(*s),
10        ydot_update_func(*s)
11    ])
12 # test impact update function
13 s_test = [1, 0.01, 1, 1]
14 print('impact update at [x={}, y={}, xdot={}, ydot={}]: {}, {}, {}, {}'.
    format(*s_test, *impact_update(s_test)))

```

Finally, everything is ready and all we need to do, is to combine the impact condition function and impact update function into the simulation as in Algorithm 2. Below is an example, but you can try your own implementation:

```

1 # define integrate function
2 def integrate(f, xt, dt):
3     """
4     RK4 integration
5     """

```

```

6     k1 = dt * f(xt)
7     k2 = dt * f(xt+k1/2.)
8     k3 = dt * f(xt+k2/2.)
9     k4 = dt * f(xt+k3)
10    new_xt = xt + (1/6.) * (k1+2.0*k2+2.0*k3+k4)
11    return new_xt
12 # define simulate function
13 def simulate_with_impact(f, x0, tspan, dt, integrate):
14     """
15     simulate with impact
16     """
17     N = int((max(tspan)-min(tspan))/dt)
18     x = np.copy(x0)
19     tvec = np.linspace(min(tspan), max(tspan), N)
20     xtraj = np.zeros((len(x0), N))
21     for i in range(N):
22         if impact_condition(x) is True:
23             x = impact_update(x)
24             xtraj[:, i] = integrate(f, x, dt)
25         else:
26             xtraj[:, i] = integrate(f, x, dt)
27         x = np.copy(xtraj[:, i])
28     return xtraj
29 # simulate the particle falling in gravity with impact
30 s0 = [0, 1, 0, 1]
31 traj = simulate_with_impact(particle_dyn, s0, tspan=[0, 5], dt=0.01,
32                             integrate=integrate)
33 import matplotlib.pyplot as plt
34 plt.plot(np.arange(traj.shape[1]), traj[0:2].T)
35 plt.show()

```

Example

Let a pendulum have configuration θ measured from vertical on a wall, so that $\theta = 0$ at its stable equilibrium. Then we have

$$L = \frac{1}{2}mR^2\dot{\theta}^2 - gmR(1 - \cos(\theta))$$

where R is the length of the pendulum, m is its mass, and $g > 0$ is the gravitational constant. From these we get the continuous time Euler-Lagrange equations before the impact

$$\ddot{\theta} = -\frac{g}{R} \sin(\theta)$$

and momentum

$$p = \frac{\partial L}{d\dot{q}} = mR^2\dot{\theta}$$

and Hamiltonian

$$H = p\dot{q} - L = \frac{1}{2}mR^2\dot{\theta}^2 + gmR(1 - \cos(\theta)).$$

So, the impact law tells us that

$$\left[\frac{\partial L}{d\dot{q}} \right]_{\tau^-}^{\tau^+} = \lambda(1)$$

$$\left[\frac{\partial L}{\partial \dot{q}} \cdot \dot{q} - L(q, \dot{q}) \right]_{\tau^-}^{\tau^+} = \left[mR^2\dot{\theta} \cdot \dot{\theta} - \frac{1}{2}mR^2\dot{\theta}^2 + gmR(1 - \cos(\theta)) \right]_{\tau^-}^{\tau^+} = 0.$$

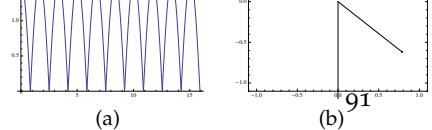


Figure 29: (a) simulation of θ versus t
(b) a snapshot of the animation

With these, we get an impact update law:

$$\dot{\theta}(\tau^+) = -\dot{\theta}(\tau^-)$$

when $\lambda \neq 0$. Simulating this system, we get the figures below.

Reasoning About Impacts

In this section we go through some examples that illustrate why impact mechanics can be unintuitive and why the engineer has to think carefully about what modeling decisions are sensible. Below are three examples that illustrate different outcomes in impact.

Example

Note that the impact equations only reference the Hamiltonian, not the total energy. Just as before, the Hamiltonian *may not be total energy!* Fig. 30 shows a double pendulum with the base revolute joint driven so that it moves at a specified velocity $\dot{\theta} = \omega$ —it is swinging the second link into the wall at the bottom. By definition, since $\dot{\theta}_1 = \omega$, the dynamics of θ_1 will be unaffected by the moment of impact, but θ_2 will certainly be affected by the impact. The update will reflect that the Hamiltonian $H(\theta_2, \dot{\theta}_2)$ is conserved, which will lead to a different post-impact prediction than if $H(\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2)$ were the conserved quantity.

Example

Even if the Hamiltonian *is* total energy, the kinematics of impact can lead to unintuitive outcomes. Consider Fig. 31, where three situations are shown. In the first, a single pendulum is experiencing an impact with the wall. Because it has only one degree of freedom, there really is no option but for the velocity to be “reflected” during impact—that is, $\dot{\theta}(\tau^+) = -\dot{\theta}(\tau^-)$. This is notably independent of the mass/geometry of the pendulum.

In the second image, a triple pendulum with $q = (\theta_1, \theta_2, \theta_3)$ is about to experience an impact, with the end of its third link impacting the wall. Because of the momentum terms, the third link may have to spin quite fast in order for the mass of the second link to continue moving and for total energy to be conserved—all three elements of \dot{q} will be affected *instantly* during impact. This is an unintuitive result since this means that energy is being instantly transferred through the entire body, but this is a consequence of the rigid body assumption. To model the time it takes for energy to move through the body, the elastic mechanical properties of each body would need to be taken into account in order to model the shock wave moving through the body. So here we can see the way in which our finite dimensional

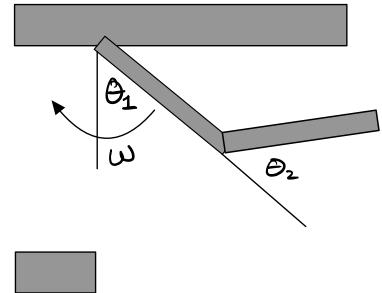


Figure 30: A driven double pendulum where $\dot{\theta}_1 = \omega$ and the second link will impact a wall. The Hamiltonian for this system is not total energy!

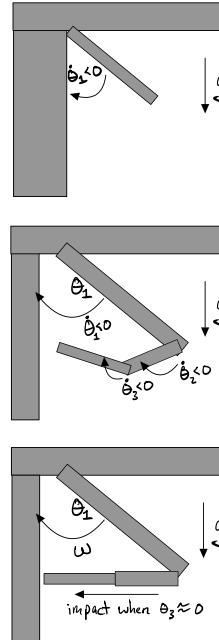


Figure 31: Configuration variables q stay constant during impact but their velocities \dot{q} change. How they change can depend significantly on the geometry of the body during impact.

model of impact starts to miss some conceptually important¹²⁹ facets of impacting bodies.

In the third image, impact is about to occur with θ_3 approximately zero. This will lead to $\dot{\theta}_3$ changing very little, and instead in this configuration almost all the change will be in $\dot{\theta}_1$. Here, the kinematics happen to be such that the impact behaves very much like it would in the first image. *This is why we need the impact equations.* We cannot make educated guesses about what will happen in impact, since the outcomes vary so much as a function of the kinematics at time of impact.

Example

The geometry of the bodies also matters to impact, because (obviously) *where* the impact occurs depends on the *shape* of the body. Up until now we have really been able to think in terms of point masses, but even for a two-link pendulum such as that shown in Fig. 32, the shape of the rectangle and the motion of the body will determine which part of the second link experiences impact. In the next sections we will see how geometry and dynamics can be used together, including in situations like this.

Numerical Aspects of Impact

Numerically integrating through impact is challenging. Fig. 33 shows an illustration of time stepping through an impact. Assume that $\phi(q) > 0$ in the free set and that $\phi(q) < 0$ in the impact set and that $\phi(q) = 0$ on the boundary. The condition $\phi(q) = 0$ will generally never be exactly satisfied by the set $\{\phi(q(t_0), \phi(q(t_1), \phi(q(t_2), \dots, \phi(q(t_8), \phi(q(t_9), \dots\})$. Instead, detecting impact has to be accomplished by looking for a *sign change* in $\phi(q(t_i))$ at every time step. Then the question is where should the impact be updated? In the figure, $q(t_6) > 0$ and $q(t_7) < 0$, so $q(t_7)$ is the first time one can tell an impact has occurred. But if the impact is evaluated there, in the *infeasible* set where $\phi(q) < 0$, the predictions might also be infeasible.¹³⁰ In some sense, the most correct solution would be to solve for the (purple) $q(t_7)$ that satisfies $\phi(q(t_7)) = 0$, but this would require changing dt until the condition is met. Alternatively, I recommend taking a step back to $q(t_6)$, where $\phi(q(t_6)) = \epsilon > 0$ and evaluate the impact there. That will be feasible for dt small enough and typically leads to reasonable simulations. Nevertheless, this solution is clearly heuristic.¹³¹

¹²⁹ but potentially irrelevant to a particular problem...

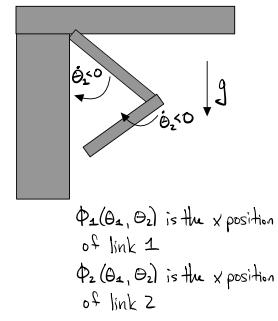


Figure 32: The geometry of a body determines which parts will experience impact—and which parts experience impact affects the impact update rule.

¹³⁰ E.g., $\phi(q(t_8)) < 0$ and $\phi(q(t_9)) < 0$, which would imply a physically implausible prediction.

¹³¹ And like all heuristics, should imply to the engineer that if the simulation predicts something implausible, this is a first place to consider.

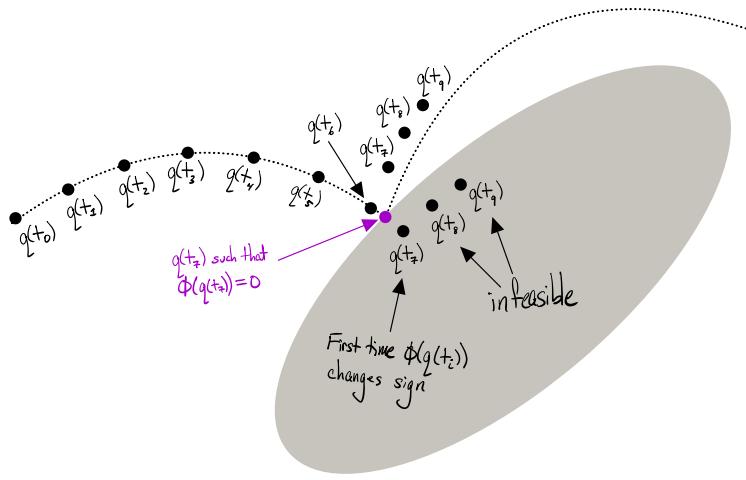


Figure 33: Numerically simulating impacts means *detecting* impact numerically and then evaluating *feasible* predictions post impact.

Inelastic and Plastic Impacts

With all the formalism of perfectly elastic impacts behind us, we can discuss inelastic and plastic impacts. Both cases are essentially heuristic in their approach—the actual mechanics of energy dissipation are complex and involve heat generation, material properties, and other factors that are outside the scope of what we discuss in this class. Nevertheless, quite reasonable predictions can be made, and the engineer just needs to keep track of them.¹³²

Plastic Impacts: The key thing that happens in plastic impacts is that post impact the impact condition $\phi(q(t)) = 0$. That is, after the impact, the two bodies stay in contact with each other, as illustrated in the left hand side of Fig. 34. Clearly the Hamiltonian *will not be conserved*—imagine a plastic impact between a particle and the ground, where post impact the particle just sits there with no kinetic energy at all. So we lose one of the equations and, at least algebraically, need to replace it with something else? What do we know after impact? Really, what we have to work with is the statement that $\phi(q(t)) = 0$ for $t \geq \tau$. As we did with constrained systems, the way to work with this is to take its time derivative, which must also be equal to zero, to conclude that $D\phi(q(t)) \cdot \dot{q}(\tau^+) = 0$. This gives us one more equation, one that can replace the previous conservation of the Hamiltonian. This gives us impact equations at time of impact τ of the following form.

¹³² This is part of a larger point that is worth making—the engineer will inevitably rely on a combination of a) rigorous methods; b) semi-rigorous methods well supported by a combination of theory and experimental evidence; and c) things the engineer simply has to make up in order to get a system to “work”. Academically, we tend to discourage b) and actively frown upon c), but my view is that the engineer simply needs to know which things are which—don’t confuse heuristics with rigor, use rigorous methods whenever possible to reduce the set of issues that need to be analyzed when a failure occurs, etc. That is, even in dynamics and simulation, the engineer’s mind matters if anything useful is being pursued.

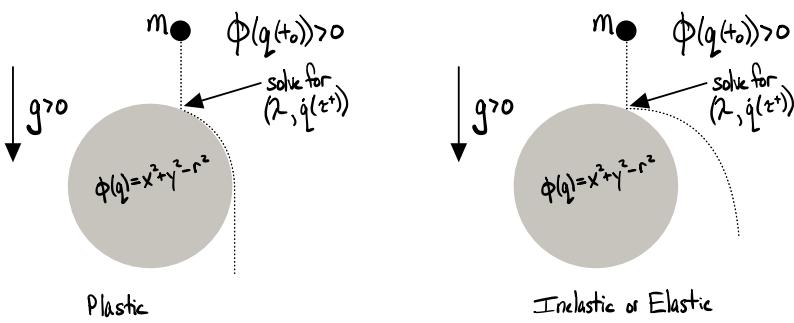


Figure 34: Non-elastic impacts are generally described as either plastic or inelastic. Plastic impacts stay in contact after the impact, while inelastic impacts are described in terms of how they reduce energy.

$$\boxed{\begin{aligned} p|_{\tau^-}^{\tau^+} &= \lambda \frac{\partial \phi}{\partial q} \\ D\phi(q(t)) \cdot \dot{q}(\tau^+) &= 0. \end{aligned}}$$

These equations, for a single impact equation, have $n + 1$ equations with $n + 1$ unknowns $(\dot{q}(\tau^+), \lambda)$. These equations are remarkable, given their heuristic nature, in that they are *very* good at predicting many impact situations.

Lastly, note that the plastic impact illustrated in Fig. 34 indicates that the particle will eventually lose contact with the surface. How should we detect that loss of contact? The scalar $\lambda(t)$ has a sign after impact that “pushes” back on the mass to prevent it from entering the interior of the surface. At some point, this sign *changes*, indicating that the surface would need to “pull” on the mass in order to keep the particle on the surface. Detecting these sign changes will allow the engineer the opportunity to automate the transition from constrained motion¹³³ to unconstrained motion¹³⁴ modeled by the unconstrained Euler-Lagrange Equations.

Inelastic impacts: Inelastic impacts are more challenging. Let’s say that some amount of energy is lost during impact—say, 80%. This idea is often called the *coefficient of restitution* α .¹³⁵ The most basic version of a coefficient of restitution suggests that the Hamiltonian will be scaled by the coefficient $H(\tau^+) = \alpha H(\tau^-)$.

$$\begin{aligned} p|_{\tau^-}^{\tau^+} &= \lambda \frac{\partial \phi}{\partial q} \\ H(\tau^+) &= \cancel{\alpha H(\tau^-)}. \end{aligned}$$

¹³³ after the plastic impact...

¹³⁴ indicating by the change in the sign of $\lambda(t)$...

¹³⁵ A quick survey of internet searches finds many informal discussions of this idea.

But in Fig. 34, what if the mass hits the impact surface at an angle, barely skimming its surface. It seems implausible for the amount of energy removed to not depend on the geometry of impact.¹³⁶ How do we incorporate this into an impact update. Instead of scaling the energy directly, we can scale the impulse $\lambda \frac{\partial \phi}{\partial q(\tau)}$, so first solving for the elastic impact and then scaling the impulse vector by α .

$$\begin{aligned}\tilde{p}(\tau^+) - p(\tau^-) &= \tilde{\lambda} \frac{\partial \phi}{\partial q} \\ \tilde{H}(\tau^+) &= H(\tau^-). \\ p(\tau^+) - p(\tau^-) &= \alpha \tilde{\lambda} \frac{\partial \phi}{\partial q}\end{aligned}$$

Solve the first two equations for $p(\tilde{\tau}^+)$ and $\tilde{\lambda}$ —once these temporary variables are known, the third equation yields $p(\tau^+)$. Then $p(\tau^+)$ with the property that the impulse has removed “energy orthogonal to the surface” without affecting “energy parallel to the surface”.¹³⁷ To be clear, this only succeeds in being better than directly manipulating the Hamiltonian—this is not a well grounded calculation.¹³⁸

¹³⁶ If the mass has nearly zero velocity normal to the surface and extremely high velocity tangent to the surface, the energy should be barely affected.

¹³⁷ As the scare quotes suggest, this is running the risk of being nonsense, but at least it is qualitatively correct.

¹³⁸ and the engineer should therefore be deeply cynical about any predictions it makes!

Class Exercise—Walking on a Treadmill

Take the above biped as a model of a human walking on a treadmill. Assume that the subject has a fracture in the femur (the thigh). You want to calculate what forces/torques are being applied to the fracture so that you can design a therapy routine (e.g., how fast the treadmill should make the subject walk). Of course, measurements cannot be taken directly at the fracture, so how do you estimate the forces and choose a walking speed that keeps them below a desired threshold?

1. How do you model the walking (assuming that each bone is a point mass—which, of course, it isn't)?
 - (a) What types of quantities will you need to be able to compute?

2. Based on your model, what measurements should you take?
Think of all the different types of sensors one might have in a well-equipped lab; you are allowed to use any of them.

3. With a model and these measurements in hand, how do you compute the forces and torques acting on the fracture?

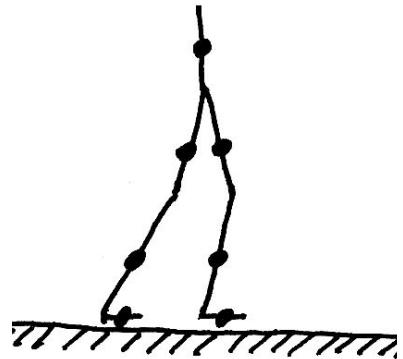


Figure 35: A planar biped walking on a treadmill

Geometry of Lagrangians for Rigid Body Motion

Mathematicians love geometry, and it is true that there are many aesthetically pleasing things about using geometric tools to represent mechanical systems. We will, however, focus on the practical aspects of geometry—when to use it, when to ignore it, when to ignore it while keeping it in the back of your mind in case something goes wrong. Roughly speaking, if you want your code to be very, very fast, you will want to use geometric representations. Also, if a mechanical body is free-falling (e.g., a satellite), geometric representations will be essential.

The rest of the time, geometry plays a more boring role—allowing us to algorithmically construct the Lagrangians that we have, until now, constructed out of geometric insight and labor-intensive and error-prone trigonometry. It's worth taking a moment to ask the following question.

What is the Euler-Lagrange *Algorithm*?

Your experience thus far splits the modeling of mechanical systems into distinct phases—each of these correspond, hopefully, to the organization of your code, similar to the Algorithm 3.

So what do we not know how to do at this point? If we know the Lagrangian, all of this can be automated in software. What we don't know is how to represent all this data describing the mechanical system in a reasonably efficient manner! Geometry tools help with this.

Algorithm 3: Euler-Lagrange Algorithm

Input : q : configuration
 $s(t_0) = (q(t_0), \dot{q}(t_0))$: initial condition for state s
 (t_0, t_f, dt) : simulation start and end times
 $KE(q, \dot{q})$: kinetic energy
 $V(q)$: potential energy
 $F(q)$: external forces
 $\phi_{con}(q)$: constraints
 $\phi_{im}(q)$: constraints

Output: $traj$: Numerical values of all states between t_0 and t_f in increments of dt .

1 Evaluate

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \lambda \nabla \phi_{con} + F$$

$$\frac{d^2}{dt^2} \phi_{con}(q(t)) = 0$$

symbolically // using SymPy

```

2 solve for dyn symbolically //  $\ddot{q}$  in the  $n+m$ -dim vector  $(\ddot{q}, \lambda)$ 
3 convert dyn to a numerical function // using lambdify
4 traj = {s(t_0)}
5 while  $t_i < t_f$  do
6   s(t_{i+1}) ← integrate(dyn, s(t_i))
7   if  $\phi_{im}(q(t_{i+1})) \leq 0$  then
8     s(t_i) ← impact_update(s(t_i)) // instantaneous impact update: s(t_i) on
      the right represents  $s(\tau^-)$ , the one on the left represents  $s(\tau^+)$ ,
      note that this is  $i$  not  $i+1$  to make  $s(t_i)$  feasible
9   s(t_{i+1}) ← integrate(dyn, s(t_i))
10  end if
11  t_{i+1} ← t_i
12  append s(t_{i+1}) to traj
13 end while
14 return traj // return results

```

Rigid Body Motions

Rigid body motion analysis is the workhorse of simulation of mechanical bodies when one can ignore elastic-body effects (e.g., flexibility in bodies). The rigidity assumption allows us to abstract a collection of point masses as a single mass—a mass that will eventually have both linear inertia (momentum due to linear motion of the center of mass) and rotational inertia (momentum due to rotation about the center of mass).¹³⁹

First, what properties should rigid body motion satisfy? Focusing on rotation, such as that seen in Fig. 36, consider the action of rotation Rot on points in the (y, z) plane in \mathbb{R}^3 , where rotation is taking place about the x axis. We will temporarily call these points r and s . As drawn, $Rot(r_0) = r_1$ and $Rot(s_0) = s_1$. But what happens if I scaled r_0 by a factor of $\frac{1}{2}$? The rotation $Rot(\frac{1}{2}r_0)$ would be on the same vector as r_1 , just half as far from the origin $\frac{1}{2}Rot(r_0)$. More generally $Rot(\alpha r) = \alpha Rot(r)$. Secondly, what would happen if I apply Rot to

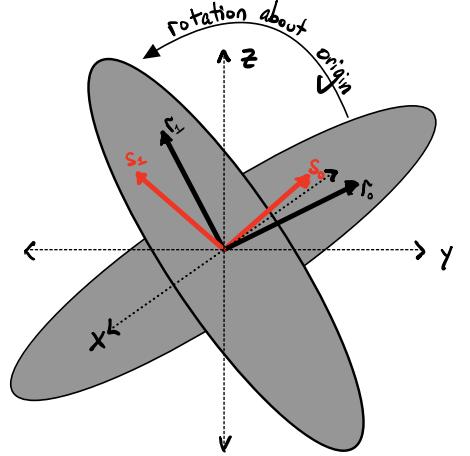


Figure 36: A body at two different orientations with points r_0 and r_1 . The body has rotated around the x axis

¹³⁹ This way of distinguishing between linear and rotational inertia is not universal, but it leads to the sensible outcome that individual point masses *do not have rotational inertia*. This way of thinking about linear/rotational inertia is explicitly independent of the coordinate choice one uses, consistent with the earlier section on coordinate independence of the Euler-Lagrange Equations.

$\frac{r_0+s_0}{2}$? The result would be a vector that is the average of the two, both algebraically and geometrically in the figure. That point would also be rotated to $\frac{r_1+s_1}{2}$, the average of the two rotated points. That is, $\text{Rot}(r_0 + s_0) = \text{Rot}(r_0) + \text{Rot}(s_0)$. These two properties combine to tell us that $\text{Rot}(\alpha r_0 + \beta s_0) = \alpha \text{Rot}(r_0) + \beta \text{Rot}(s_0)$ —**rotation satisfies the definition of a linear operation**. Moreover, as the figure suggests, distances between points must be preserved during rotation—this is the sense in which rotation satisfied the rigidity constraint.

Using the above observations as motivation, assume that we have a linear map A relating r_0 and r_1 so that $Ar_0 = r_1$. A rigid body motion should satisfy the requirement that the magnitude of r_0 should equal r_1 . The combination of linearity and distance preservation leads to a precise statement about how to describe A .

$$\begin{aligned} \|r_0\| &= \|r_1\| \\ \Rightarrow \|r_0\|^2 &= \|r_1\|^2 \\ \Rightarrow \langle r_0, r_0 \rangle &= \langle r_1, r_1 \rangle \\ \Rightarrow r_0^T r_0 &= r_1^T r_1 \\ \Rightarrow r_0^T r_0 &= (Ar_0)^T (Ar_0) \\ \Rightarrow r_0^T r_0 &= r_0^T A^T A r_0 \quad \forall r_0 \\ \Rightarrow A^T A &= I. \end{aligned}$$

Hence, the inverse of A is the transpose of A . An $n \times n$ matrix¹⁴⁰ that satisfies this inverse property belongs to the set $O(n)$, called the *Orthogonal Group*.¹⁴¹ This set is defined to be all such matrices of dimension $n \times n$.

$$O(n) = \left\{ A \in \mathbb{R}^{n \times n} \mid A^T A = I_{n \times n} \right\}$$

Here a matrix identity can help us: $\det(A^T A) = \det(A)^2$. This, along with the fact that $\det(I) = 1$, means that $\det(A) = \pm 1$. The fact that there are only two possible determinants of the matrix A suggests that we should ask whether we are interested in all cases of $O(n)$. The identity matrix $I_{n \times n}$ has determinant $+1$, and is an example of a rotation.¹⁴² So we are interested in the case where $\det(A) = +1$. This set is called the *Special Orthogonal Group*.

$$SO(n) = \left\{ A \in \mathbb{R}^{n \times n} \mid A^T A = I_{n \times n} \text{ and } \det A = +1 \right\}$$

$SO(n)$ provides a description of all the rigid body rotations.¹⁴³

Example

The rotation matrix discussed earlier in class during the section on Noether's theorem

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

¹⁴⁰ since the inverse exists, such a matrix must be square....

¹⁴¹ The word *group* should not be derailing here. The student can safely think of this as a *set*. What distinguishes a group from a set is that there is an operation, in this case matrix multiplication, such that the set is *closed* under the operation—two matrices A and B in $O(n)$ will yield AB in $O(n)$.

¹⁴² by an angle of $0\dots$

¹⁴³ If $\det(A) = -1$, A is a reflection, something we do not consider in this class, but something that is very interesting and (obviously, as the word 'reflection' suggests) relevant to optics.

is an element of $SO(2)$ because

$$\det(R(\theta)) = \cos^2 \theta + \sin^2 \theta = 1$$

and

$$\begin{aligned} R^T R &= \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \\ &= \left[\begin{array}{c|c} \cos^2 \theta + \sin^2 \theta & \cos \theta \sin \theta - \sin \theta \cos \theta \\ -\sin \theta \cos \theta + \cos \theta \sin \theta & \cos^2 \theta + \sin^2 \theta \end{array} \right] \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

Lastly, note that rigid body motions also preserve *angles* in addition to preserving distance.¹⁴⁴

¹⁴⁴ Exercise: show that rigid body motions also preserve angles between r_0 and r_1 by using the fact that the angle θ between r_0 and r_1 is defined by

$$\cos \theta = \frac{\langle r_0, r_1 \rangle}{\|r_0\| \|r_1\|}$$

Matrix Representations of Rigid Body Motions

Rigid body motion analysis would not be useful if it did not also enable us to use computational tools that *automate* much of what we were doing manually earlier in the class.¹⁴⁵ What we will see over the course of the next several lectures is that rigid body motion can be represented using matrices and matrix operations, particularly matrix multiplication. Near the end of the course, we will see that one can represent the configuration directly in terms of the matrix representations if desired. For now, we will spend some time looking at the standard examples and think about what they operationally achieve.

¹⁴⁵ Trigonometry, though pleasant for some, is generally not my preferred way to spend my time thinking about a mechanical system.

Matrix Representations of Rigid Body Motion in \mathbb{R}^2 and \mathbb{R}^3

We will consider four fundamental sets (groups), the *Special Orthogonal Group* and the *Special Euclidean Group*:

1. $SO(2)$ planar rotations (parameterized by θ);
2. $SE(2)$ planar rotations and translations (parameterized by x, y, θ);
3. $SO(3)$ 3D rotations (locally parameterized by θ, ψ, ϕ —roll, pitch, yaw);
4. $SE(3)$ 3D rotations and translations (locally parameterized by $x, y, z, \theta, \psi, \phi$).

It is hopefully clear that $SO(2)$, $SE(2)$ and $SO(3)$ are all subsets of $SE(3)$, so we will often focus on $SE(3)$ when doing computations.

Planar Rotation: $SO(2)$

The Special Orthogonal Group $SO(2)$ consists of all matrix representations of rotation in the plane \mathbb{R}^2 and is defined either by the property that the transpose is the rotation matrix R 's inverse and the rotation matrix R has determinant 1, or by the *parameterization* by θ .¹⁴⁶

$$SO(2) = \left\{ R \mid R^T R = I \& \det(R) = 1 \right\} = \left\{ \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \mid \theta \in \mathbb{R} \right\}.$$

How can we interpret a matrix $R \in SO(2)$? There are several reasonable interpretations.

¹⁴⁶ The ability to parameterize an operation is important—we like to give things coordinates—but it can also be misleading because having a *formula* only makes us more comfortable rather than actually helping with theory or calculation.

1. R takes a point $r \in \mathbb{R}^2$ and *rotates it*. That is, R is a representation of literally moving stuff (e.g., point masses) around.
2. R transforms coordinates that describe the location of a point mass $r \in \mathbb{R}^2$ from one frame to another. That is, R doesn't represent *doing* anything—it represents how we *measure* what is happening.
3. R is the coordinate system and can be the way we *represent* the motion of $r \in \mathbb{R}^2$. That is, R could be a component of the configuration q .

Consider Fig. 37, where a point $r \in \mathbb{R}^2$ is illustrated. The point r has no *coordinate representation* until we define the *frame* in which that representation will be made. Here we label one frame W (for *world*) and another frame B (for *body*).¹⁴⁷ In these frames, r has a particular coordinate value—we label these r_W and r_B . As is hopefully visually apparent, B and W are related by a rotation, so we can ask how r_W and r_B are related.

$$r_W = \begin{bmatrix} r_x \\ r_y \end{bmatrix}_W = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} r_x \\ r_y \end{bmatrix}_B = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} r_B$$

e.g., we can choose $\theta = \frac{\pi}{4} = 45^\circ$, $r_B = (1, 0)^T$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

One can check this calculation using trigonometry and for the case of $\theta = \frac{\pi}{4}$ can see in Fig. 37 that the outcome makes sense. The rotation matrix R transforms the coordinate representation of r_B to the coordinate representation r_W . Consistent with this interpretation, we can add indices to the rotation notation: $R = R_{WB}$. Consequently, given $\theta(t)$ and r_B , we can get $r_W(t)$. Moreover, since we know that $r_W = R(\theta)r_B$, we can find the velocity of the point r in inertial coordinates in frame W by looking at $\frac{d}{dt}r_W = \frac{d}{dt}(R(\theta))r_B$ ($r_B = 0$ because r_B is constant, since it is attached to the body). This means we could calculate the kinetic energy using the rigid body rotation $R(\theta(t))$ instead of doing trigonometry.

Lastly, it is worth noting that $SO(2)$ is equivalent to the unit-magnitude complex numbers, with the rotation matrix R replaced by the complex number z and matrix multiplication replaced by complex number multiplication. We saw that complex numbers can be used as configurations a few sections ago, and will examine the

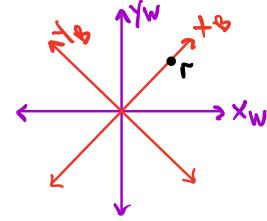


Figure 37: Frame B and frame W are related by a rotation of θ . If $r = (0, 1)$ in frame B , what will its coordinate be in frame W ?

¹⁴⁷ When there are many frames we might use a index such as $\{A, B, C, D, \dots\}$ or $\{B_1, B_2, B_3, \dots\}$ to help organize the relationship between frames.

possibility that rotations R can be used as configurations later.

Planar Rotation and Translation: $SE(2)$

Planar rotation and translation are typically parameterized by (p_x, p_y, θ) , where (p_x, p_y) is the translation and θ is the orientation. However, we can also parameterize it by (p_x, p_y, R) , where R is the rotation matrix describing rotation. The set of planar rigid body rotations is called the *Special Euclidean Group* $SE(n)$, defined below.

$$SE(n) = \left\{ (R, p) \mid R^T R = I \& \det(R) = 1 \& p \in \mathbb{R}^n \right\}$$

$$SE(2) = \left\{ \left(\begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}, \begin{bmatrix} p_x \\ p_y \end{bmatrix} \right) \mid (p_x, p_y, \theta) \in \mathbb{R}^3 \right\}.$$

We can transform points r_B in the plane by multiplying them by R and adding $p = (p_x, p_y)$. For instance, in Fig. 38 one might want to transform $r_B = (1, 0)$ to r_W using the rigid body transformation $(R(\theta = \frac{\pi}{4}), (1, 1)^T)$.

$$r_W = \begin{bmatrix} r_x \\ r_y \end{bmatrix}_W = R \left(\theta = \frac{\pi}{4} \right) \begin{bmatrix} 1 \\ 0 \end{bmatrix}_B + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + \frac{1}{\sqrt{2}} \\ 1 + \frac{1}{\sqrt{2}} \end{bmatrix}_W$$

Moreover, we might want to be able to rewrite this operation so that it only uses matrix multiplication. Define the matrix $g_{WB}(R_{WB}, p_{WB})$ to be

$$g_{WB}(R_{WB}, p_{WB}) = \begin{bmatrix} R_{WB} & p_{WB} \\ 0 & 1 \end{bmatrix}$$

$$R_{WB} \in \mathbb{R}^{2 \times 2} \quad p_{WB} \in \mathbb{R}^{2 \times 1} \quad 0 \in \mathbb{R}^{1 \times 2} \quad 1 \in \mathbb{R}^{1 \times 1}$$

and define

$$\bar{r}_B = \begin{bmatrix} r_x \\ r_y \\ 1 \end{bmatrix}_B.$$

What happens when we multiply \bar{r}_B by $g(R, p)$?¹⁴⁸

$$g(R, p)\bar{r} = \begin{bmatrix} Rr + p \\ 1 \end{bmatrix}$$

which, if we “unbar” \bar{r} gives us the rigid body transformation $Rr + p$. This g is called the *homogeneous representation* of the rigid body transformation and the \bar{r} is called the *homogeneous representation* of the point r .

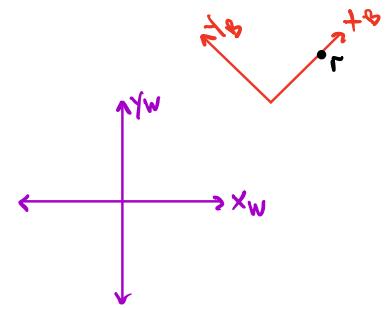


Figure 38: Two frames in the plane

¹⁴⁸ dropping the indices now, to avoid irritation....

Let's rewrite the transformation of $r_B = (1, 0)$ to r_W using the rigid body transformation ($R(\theta = 45^\circ), (1, 1)^T$).

$$\bar{r}_W = \begin{bmatrix} r_x \\ r_y \\ 1 \end{bmatrix}_W = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 1 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 + \frac{1}{\sqrt{2}} \\ 1 + \frac{1}{\sqrt{2}} \\ 1 \end{bmatrix}$$

We aren't yet seeing any major advantages of rigid body transformations, but we will. In particular, composition of rigid body transformations is achieved through matrix multiplication. To see this, assume that we have two transformations (like the multiple link pendulum) related frame A to frame C , with a frame B in between. Then the transformation

$$\begin{aligned} g_{AC} = g_{AB}g_{BC} &= \begin{bmatrix} R_{AB} & p_{AB} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{BC} & p_{BC} \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R_{AB}R_{BC} & R_{AB}p_{BC} + p_{AB} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_{AC} & p_{AC} \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

This means that elements of $SE(2)$ compose (through matrix multiplication) to form other elements of $SE(2)$ —assuming that multiplying two rotation matrices together yields another rotation matrix.¹⁴⁹ This is the key property that makes the set $SE(n)$ a *group*, and is why it is called the special orthogonal group.¹⁵⁰

Example: multiple transformations

Assume we have multiple transformations as indicated in Fig. 40, with the location of each translated frame L units in the x coordinates of the previous frame. Based on the illustration, these rigid body transformations can be written as follows.

$$\begin{aligned} g_{AB} &= \begin{bmatrix} R(\theta_1) & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} & g_{BC} &= \begin{bmatrix} I_{2 \times 2} & \begin{bmatrix} L \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \\ g_{CD} &= \begin{bmatrix} R(\theta_2) & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} & g_{DE} &= g_{BC} \\ g_{EF} &= \begin{bmatrix} R(\theta_3) & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} & g_{FG} &= g_{BC}. \end{aligned}$$

Moreover, for a point r with its coordinates known in frame G , we

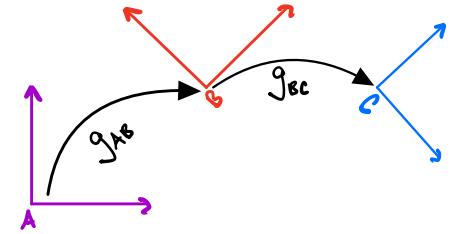


Figure 39: multiple transformations in the plane

¹⁴⁹ something you will have to prove on the homework...

¹⁵⁰ We could have just as easily been looking at the case of $n = 3$. The only things that change are that that R is no longer parameterized by a single variable θ and $p \in \mathbb{R}^3$.

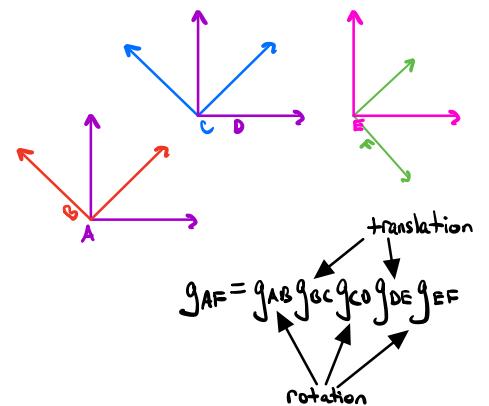


Figure 40: multiple transformations for a robot arm

can solve for its coordinates in frame A , and indeed any other frame.

$$r_A = g_{AB} g_{BC} g_{CD} g_{DE} g_{EF} \underbrace{g_{FG} r_G}_{r_F} \\ \underbrace{\quad\quad\quad}_{r_E} \\ \underbrace{\quad\quad\quad}_{r_D} \\ \underbrace{\quad\quad\quad}_{r_C} \\ r_B$$

If we wanted to calculate the velocity of the point q relative to A ,¹⁵¹ we differentiate both sides.¹⁵²

$$\frac{d}{dt} r_A = \frac{d}{dt} (g_{AB} g_{BC} g_{CD} g_{DE} g_{EF} g_{FG}) r_G$$

Example: compute Lagrangian of single pendulum using $SE(2)$

Here we $SE(2)$ transformations to compute the Lagrangian of a single pendulum.¹⁵³ The frames are defined as shown in the Figure 41.

```

1 import sympy as sym
2 from sympy.abc import t
3
4 # define constants
5 L, m, g = sym.symbols('L, m, g')
6 # define the system state theta
7 theta = sym.Function(r'\theta')(t)
8 thetadot = theta.diff(t)
9 thetaddot = thetadot.diff(t)
10
11 # we are interested the position of the pendulum represented in {W} frame
12 # thus we need to the SE(2) transformation from {W} to {B}, to make things
13 # easier, we start with the transformation from {W} to {A}, which is only a
14 # rotation for theta
15 g_wa = sym.Matrix([
16     [ sym.cos(theta), -sym.sin(theta), 0.],
17     [ sym.sin(theta),  sym.cos(theta), 0.],
18     [ 0., 0., 1.]
19 ])
20 print('SE(2) transformation from {W} to {A}')
21 display(g_wa)
22
23 # from {A} to {B} is just a translation long y-axis for -L
24 g_ab = sym.Matrix([
25     [ 1., 0., 0.],
26     [ 0., 1., -L],
27     [ 0., 0., 1.]
28 ])
29 print('SE(2) transformation from {A} to {B}')
30 display(g_ab)
31
32 # then the transformation from {W} to {B} is straightforward
33 g_wb = g_wa * g_ab
34 print('SE(2) transformation from {W} to {B}')
35 display(g_wb)

```

¹⁵¹ For instance, if we want to calculate its kinetic energy!

¹⁵² How do product rule and chain rule play a role here?

¹⁵³ Once the Lagrangian is computed, the rest is the same as in previous examples and assignments.

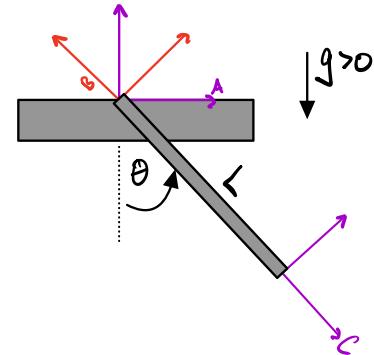


Figure 41: Simple Pendulum and Frames

```

36
37 # now, given the transformation, what's the position of the pendulum
   represented in {W}?
38 # well, the simple answer is the "p" vector in matrix g_wb
39 pend_x = g_wb[0,2]
40 pend_y = g_wb[1,2]
41 print('pendulum position in {W}:')
42 display(sym.Matrix([pend_x, pend_y]))

```

The above code might be confusing—why can we simply extract the p vector as the coordinates of the pendulum mass?¹⁵⁴ Below is another solution:

```

1 # we first think about the position in {A}
2 pend_x_B, pend_y_B = 0., 0. # position of pendulum in {B}
3 print('Pendulum position in {B}:')
4 display(sym.Matrix([pend_x_B, pend_y_B]))
5 pend_B = sym.Matrix([pend_x_B, pend_y_B, 1.]) # homogeneous representation
6
7 # transfer the position to {A}
8 pend_A = g_ab * pend_B
9 pend_x_A = pend_A[0]
10 pend_y_A = pend_A[1]
11 print('Pendulum position in {A}:')
12 display(sym.Matrix([pend_x_A, pend_y_A]))
13
14 # now we have the coordinates in {A}, we transfer it back to {B}
15 pend_W = g_wa * pend_A
16 pend_x_W = pend_W[0]
17 pend_y_W = pend_W[1]
18 print('Pendulum position in {W}:')
19 display(sym.Matrix([pend_x_W, pend_y_W]))

```

If we combine the multiplications, we will have $\text{pend}_W = g_{wa} * g_{ab} * \text{pend}_B$, note that $g_{wa} * g_{ab}$ is exactly the SE(2) transformation from W to A , and $\text{pend}_B = [0., 0., 1.]$, any matrix in $\mathbb{R}^{3 \times 3}$ times this vector is equal to extracting the third column out, which will be p vector in our case. At last, once we have the coordinates of the pendulum in W , we can compute the Lagrangian same as in previous assignments.

¹⁵⁴ Remember that the mass is at the origin of the C frame, but is also at the point $(0, -L)$ in the B frame.

Euler's Theorem and Matrix Exponentials

Euler's theorem is the workhorse of generalizing rotation from planar rotation to three dimensional rotation. Euler's theorem states that any rotation can be represented as a rotation of θ around a fixed axis ω . To make this as concrete as possible, for now we will let $\omega \in \mathbb{R}^3$ and $\|\omega\| = 1$, so that the magnitude of the vector ω is always 1.¹⁵⁵

Now some notation comes in handy. For $\omega \in \mathbb{R}^3$ define the "hat operator" $\hat{\omega}: \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$.

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

Note that for any vector ω , $\hat{\omega} = -\hat{\omega}^T$.¹⁵⁶ Moreover, we can also "unhat" a skew symmetric matrix, which we call the "unhat operator": $\hat{\cdot}: \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^3$. This "unhat" takes a skew symmetric 3×3 matrix and returns a 3×1 vector.

Next, it is useful to introduce **matrix exponentials**. Matrix exponentials allow us to characterize rotations in terms of the hat operator above.¹⁵⁷ If X is a square matrix, we can generalize from the scalar argument case e^x to the matrix argument case e^X using the Taylor expansion of the exponential.

$$e^X = \mathbb{I}_{n \times n} + X + \frac{1}{2!}X^2 + \frac{1}{3!}X^3 + \frac{1}{4!}X^4 + \dots$$

¹⁵⁸

Example: Planar rotation

Let $\omega = (0, 0, 1)^T \in \mathbb{R}^3$ and

$$\hat{\omega} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Then

$$\begin{aligned} e^{\hat{\omega}t} &= \mathbb{I}_{3 \times 3} + \hat{\omega}t + \frac{1}{2!}(\hat{\omega}t)^2 + \frac{1}{3!}(\hat{\omega}t)^3 + \frac{1}{4!}(\hat{\omega}t)^4 + \dots \\ &= \mathbb{I}_{3 \times 3} + \hat{\omega}t + \frac{1}{2!}\hat{\omega}^2 t^2 + \frac{1}{3!}\hat{\omega}^3 t^3 + \frac{1}{4!}\hat{\omega}^4 t^4 + \dots \end{aligned}$$

and

$$\hat{\omega}^2 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \hat{\omega}^3 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \hat{\omega}^4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

¹⁵⁵ Later it will be convenient to allow ω to be an arbitrary vector, and to interpret its magnitude as the angle θ .

¹⁵⁶ i.e., it is skew symmetric.

¹⁵⁷ Most students are familiar with the function e^x when $x \in \mathbb{R}$, but how do we define the exponential if $x \in \mathbb{R}^{n \times n}$?

¹⁵⁸ Note that this formally looks almost identical to the Taylor series for the scalar case, but instead of a "1" at the beginning, we have an $n \times n$ identity matrix $\mathbb{I}_{n \times n}$.

This implies that

$$\begin{aligned} e^{\hat{\omega}t} &= \begin{bmatrix} 1 - \frac{1}{2!}t^2 + \frac{1}{4!}t^4 \dots & -t + \frac{1}{3!}t^3 \dots & 0 \\ t - \frac{1}{3!}t^3 \dots & 1 - \frac{1}{2!}t^2 + \frac{1}{4!}t^4 \dots & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos t & -\sin t & 0 \\ \sin t & \cos t & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

Hence, $e^{\hat{\omega}}$ is the rotation of a body about the z axis, rotating at constant angular velocity. Alternatively, we could replace t with θ and $e^{\hat{\omega}\theta}$ would be a rotation about the vector ω by an angle θ .

More generally, if $\omega \in \mathbb{R}^3$ then $e^{\hat{\omega}\theta} \in SO(3)$. In this, ω is the axis of rotation and θ is the angle of rotation. Note we could have just as easily defined a rotation about a different axis, with a different angular velocity, for example $\omega = [\Omega, 0, 0]^T$ and found $\hat{\omega}$ to be

$$\hat{\omega} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\Omega \\ 0 & \Omega & 0 \end{bmatrix}$$

and then we would get that¹⁵⁹

$$e^{\hat{\omega}t} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \Omega t & -\sin \Omega t \\ 0 & \sin \Omega t & \cos \Omega t \end{bmatrix}.$$

¹⁵⁹ This would be a rotation in the (y, z) plane around the x axis—therefore the x component of a transformed point stays constant.

Using Matrix Exponentials for Differentiation

How do we use this? First, let's say that we want to take the derivative of $R = e^{\hat{\omega}\theta}$ with respect to θ . Then we get that

$$\begin{aligned} \frac{\partial R}{\partial \theta} &= \frac{\partial}{\partial \theta} \left(\mathbb{I} + \hat{\omega}\theta + \frac{1}{2!}(\hat{\omega}\theta)^2 + \frac{1}{3!}(\hat{\omega}\theta)^3 + \frac{1}{4!}(\hat{\omega}\theta)^4 + \dots \right) \\ &= \hat{\omega} + \frac{1}{1!}\hat{\omega}^2\theta + \frac{1}{2!}\hat{\omega}^3\theta^2 + \frac{1}{3!}\hat{\omega}^4\theta^3 + \dots \\ &= \hat{\omega} \left(\mathbb{I} + \hat{\omega}\theta + \frac{1}{2!}(\hat{\omega}\theta)^2 + \frac{1}{3!}(\hat{\omega}\theta)^3 + \dots \right) \\ &= \hat{\omega}R (= R\hat{\omega}). \end{aligned}$$

Moreover, if we want to differentiate R with respect to time, we just apply chain rule:

$$\frac{d}{dt}R = \frac{d}{dt}e^{\hat{\omega}\theta} = \frac{\partial e^{\hat{\omega}\theta}}{\partial \theta} \frac{\partial \theta}{\partial t} = \hat{\omega}R\dot{\theta}.$$

Using Matrix Exponentials for Calculating Lagrangians

How do we use this to calculate the dynamics in term of the configuration q ?

1. First, to find velocities of point masses based on their geometry , we need to be able to evaluate the time derivatives of rigid body transformations
2. Second, we'll need to take derivatives with respect to \dot{q} based on derivatives of the rigid body transformations (because those are the only things that depend on q).
3. Lastly, we'll need to take derivatives with respect to q based on derivatives of the rigid body transformations (again, because those are the only things that depend on q).

Example: single pendulum

$$\begin{aligned} q &= \theta \\ (x, y)^T &= R\left(-\frac{\pi}{2}\right)R(\theta)(L, 0) \\ (\dot{x}, \dot{y})^T &= R\left(-\frac{\pi}{2}\right)\dot{R}(\theta)(L, 0) \\ &= R\left(-\frac{\pi}{2}\right)R(\theta)\hat{\omega}\dot{\theta}(L, 0) \end{aligned}$$

So, the kinetic energy and potential energy can be described by

$$\begin{aligned} KE &= \frac{1}{2}m(\dot{x}, \dot{y}) \cdot (\dot{x}, \dot{y})^T \\ &= \frac{1}{2}m(R\left(-\frac{\pi}{2}\right)R(\theta)\hat{\omega}\dot{\theta}(L, 0))^T R\left(-\frac{\pi}{2}\right)R(\theta)\hat{\omega}\dot{\theta}(L, 0) \\ &= \frac{1}{2}m(L, 0)^T \dot{\theta} \hat{\omega}^T R(\theta)^T R\left(-\frac{\pi}{2}\right)^T R\left(-\frac{\pi}{2}\right)R(\theta)\hat{\omega}\dot{\theta}(L, 0) \\ &= \frac{1}{2}m(L, 0)^T \dot{\theta} \hat{\omega}^T \hat{\omega}\dot{\theta}(L, 0) \\ &= \frac{1}{2}m(L, 0)^T \hat{\omega}^T \hat{\omega}(L, 0) \dot{\theta}^2 \\ &= \frac{1}{2}mL^2\dot{\theta}^2 \end{aligned}$$

and

$$V = mgy = mg \left[R\left(-\frac{\pi}{2}\right)R(\theta)(L, 0) \right]_2 = -mgL \cos \theta$$

where the subscript 2 indicates that we are looking at the second element of the vector.

Notice that we are never writing down sin and cos expressions, so we are entirely avoiding the trigonometry associated with this problem. So, at minimum, this is really helpful in terms of evaluating the kinetic energy and the potential energy in a way that doesn't involve trigonometric identities.

Now, to evaluate the Euler-Lagrange Equations, we simply need to apply chain rule again to these expressions.

Let

$$\begin{aligned} a &= R\left(-\frac{\pi}{2}\right)R(\theta) \begin{bmatrix} L \\ 0 \end{bmatrix} \\ a' &= \frac{\partial a}{\partial \theta} = R\left(-\frac{\pi}{2}\right)R(\theta)\hat{\omega} \begin{bmatrix} L \\ 0 \end{bmatrix} \\ a'' &= \frac{\partial^2 a}{\partial \theta^2} = R\left(-\frac{\pi}{2}\right)R(\theta)\hat{\omega}^2 \begin{bmatrix} L \\ 0 \end{bmatrix} \end{aligned}$$

where $\hat{\omega} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. With these definitions, we have $\dot{a} = a'\dot{\theta}$. Hence, we get

$$L = KE - V = \frac{1}{2}m(a'\dot{\theta})^T a'\dot{\theta} - mg a_y = \frac{1}{2}m\dot{\theta}^T a'^T a'\dot{\theta} - mg a_y$$

where a_y is the "y" component of the vector a —that is, $a_y = [0 \ 1] \cdot a$.

Now, evaluate the Euler-Lagrange Equations:

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} &= 0 \\ \frac{d}{dt} \left(m(a'\dot{\theta})^T a' \right) - \left[\frac{1}{2}m\dot{\theta} \left(a''^T a' + a'^T a'' \right) \dot{\theta} - mg a'_y \right] &= 0 \\ m \left[\left[\dot{\theta}^T a'' \dot{\theta} + a' \ddot{\theta} \right]^T a' + (a'\dot{\theta})^T a'' \dot{\theta} \right] - \left[\frac{1}{2}m\dot{\theta} \left(a''^T a' + a'^T a'' \right) \dot{\theta} - mg a'_y \right] &= 0 \end{aligned}$$

after simplifying by multiplying out matrices

$$mgL \sin \theta + mL^2 \ddot{\theta} = 0$$

Again, we did not take any explicit derivatives of sin and cos terms.

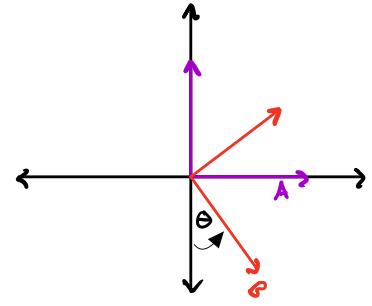


Figure 42: Simple rotation

Euler Angles

How do we typically parameterize $SO(3)$ —the space of rigid body rotations? Typically, particularly with mechanisms,¹⁶⁰ we use Euler angles, defined by:

$$R_Z = e^{\theta \hat{\omega}_Z} \quad R_X = e^{\psi \hat{\omega}_X} \quad R_Y = e^{\phi \hat{\omega}_Y}$$

where

$$\omega_Z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \omega_X = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \omega_Y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}.$$

Note that Euler angles do not commute:

$$R_X R_Y R_Z \neq R_Y R_X R_Z$$

but both of these are common choices for parameterizing $SO(3)$.

Example:

$$g_{AI} = g_{AB} g_{BC} g_{CD} g_{DE} g_{EF} g_{FG} g_{GH} g_{HI}$$

$$= \begin{bmatrix} I & \begin{bmatrix} 0 \\ 0 \\ h \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_Z(\theta_1) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & \begin{bmatrix} L_1 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_Z(\theta_2) & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} I & \begin{bmatrix} L_2 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_Z(\theta_3) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & \begin{bmatrix} L_3 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix} \\ 0 & 1 \end{bmatrix}$$

Also, $q = (\theta_1, \theta_2, \theta_3, z)$ and $\frac{d}{dt} \begin{bmatrix} I & \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & \begin{bmatrix} 0 \\ 0 \\ \dot{z} \end{bmatrix} \\ 0 & 0 \end{bmatrix}$ and

$$\frac{d}{dz} \begin{bmatrix} I & \begin{bmatrix} 0 \\ 0 \\ z \end{bmatrix} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ 0 & 0 \end{bmatrix}$$

¹⁶⁰In other instances, like modeling satellites in orbit, Euler angles can be disastrous because of singularities in their parameterization of $SO(3)$, but for our purposes they will be fine for almost everything we might come up with.

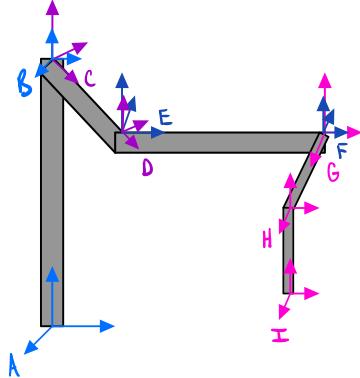


Figure 43: This example mechanism has three revolute joints and one prismatic joint. Each transformation from frame A to frame I can be decomposed into just a rotation or just a translation, and these are typically Euler angles because revolute joints are common.

ME 314 Project Proposals.

Project Proposals

Your project in this class will be due online by the Thursday of finals week. You need to choose a system to model that is somehow *interesting* to you. Last year, students modeled simple walking, ringing a bell, trebuchets, children's toys, and other examples. Other possibilities include modeling a ratchet or choose some other mechanical device. The general guidelines are that your project must:

1. involve at least two bodies and be more than 2 but not more than 5 degrees of freedom (unless something makes the extra degrees of freedom straight forward);
2. include rotational inertia in at least one body;
3. include impacts;
4. include some sort of external forcing (but this could be friction—not necessarily control forces/torques).
5. be planar; we will be limiting ourselves to projects in 2D.
6. animate the resulting simulation to show that it “works”;
7. no spheres—anything like a “ball” must be modeled as a polygon (typically a triangle or rectangle) so that impacts have some sort of nontrivial update.

You additionally have the option of choosing a “default” project—described below. This project has two bodies, six degrees of freedom,

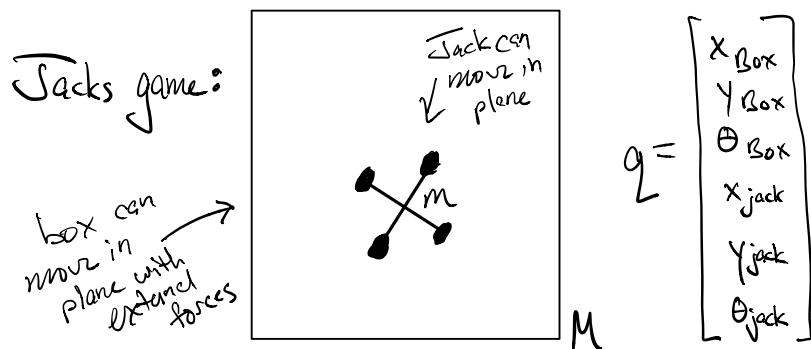


Figure 44: A “jack” or “dice” in a cup: The cup can be moved by external forces, causing the jack to bounce inside the cup.

includes impacts, has external forces (for shaking the cup), and is planar; it satisfies all the requirements and is a good indication of what is a reasonable level of complexity.

If you want to do a final project different from this default project, you will turn in a single page PDF that describes how your project meets all the requirements. Moreover, you will turn in a diagram indicating the configuration variables and the frames you will use to compute the equations of motion.

Rotational Kinetic Energy and Rotational Inertia

Integral Expressions for Rotational Kinetic Energy

Take home messages:

- Geometry shows us that rotational kinetic energy can be calculated with a volume integral using density, body angular velocity, and particle positions. This calculation happens in the body frame.
- Spatial distribution of mass about the axis of rotation influences rotational kinetic energy, distribution of mass along the axis does not
- Velocity is coordinate dependent, energy is not

A rigid body may possess kinetic energy due to both translational and rotational motions. For translational motions finding the associated kinetic energy doesn't present too much difficulty, as all the mass in the body is traveling in the same direction. As we will see, computing rotational kinetic energy requires us to pay attention to how mass is spatially distributed in the body. Fortunately, our understanding of geometry and rigid body transformations helps in this regard.

Rotational Kinetic Energy as a Volume Integral

Consider a rigid body in pure rotation about its center of mass. To track it as it rotates, let's affix a body coordinate frame to the center of mass of the rigid body. Since we're considering pure rotation, we can specify that the origin of the stationary frame is also at this center of mass such that the origins of the two frames will always be coincident. Suppose a rotation $R(t)$ describes the transformation from the body frame to our specified stationary frame. That is, for a particle located at r_B in the body frame, its position in the stationary frame is

$$r_W = R r_B$$

Then the velocity of that particle is

$$\frac{d}{dt} r_W = \dot{R} r_B$$

where there is no second term from the product rule since r_B is constant. But what is \dot{R} ? If we express the rotation as

$$R(t) = e^{\hat{\omega}\theta(t)}$$

for some normalized ω and a scalar $\theta(t)$, then

$$\dot{R} = R \hat{\omega} \dot{\theta}(t)$$

where $\omega \dot{\theta}(t)$ is the body angular velocity. Moving forward, we drop the assumption that ω is normalized, such that

$$\dot{R} = R \hat{\omega}$$

and ω denotes the body angular velocity ($\dot{\theta}$ will no longer appear). In this notation

$$\frac{d}{dt} r_W = R \hat{\omega} r_B$$

Now that we know how a single particle in the body moves, we proceed to calculating the rotational kinetic energy of the entire rigid body. Calculating this requires integrating the kinetic energies of differential volume elements over the entire volume of the body. This integral views the rigid body as an infinite sum of infinitely small particles (with associated infinitely small mass). The integral is simply stated

$$KE_{\text{rot}} = \frac{1}{2} \int_V \rho ||v||^2 dV = \frac{1}{2} \int_V \rho v^T v dV$$

where V is the volume of the rigid body, ρ is the density of the body (for now assumed constant), v signifies the magnitude of the velocity of a differential element dV , and v signifies the velocity vector of a differential element dV . Using our analysis above, we can plug in for v as

$$\begin{aligned} KE_{\text{rot}} &= \frac{1}{2} \int_V \rho (R \hat{\omega} r_B)^T (R \hat{\omega} r_B) dV \\ &= \frac{1}{2} \int_V \rho (\hat{\omega} r_B)^T R^T R (\hat{\omega} r_B) dV \\ &= \frac{1}{2} \int_V \rho (\hat{\omega} r_B)^T (\hat{\omega} r_B) dV \end{aligned}$$

Given the geometry and density of a rigid body, as well as its body angular velocity, this integral provides its rotational kinetic energy. Notice that our expression for the velocity of a particle depends R , implying some dependence on the choice of stationary frame. However, the $R^T R$ cancellation above yields an expression for kinetic energy that is independent of R . This should fit our intuition, that velocity is coordinate dependent but energy is not.

Example:

Rotation of a rigid block (a.k.a. rectangular prism). Consider a block with constant density ρ and dimensions 2ℓ , $2w$, and $2h$ aligned respectively with the x , y , and z axes of a coordinate frame placed at the center of the block. We will compute the kinetic energy of the block as it rotates about the z axis with an angular velocity of ω_1 . More specifically, the body angular velocity of the block is $\omega_1 = \begin{bmatrix} 0 & 0 & \omega_1 \end{bmatrix}^T$. Noting that in rectangular coordinates $r_B = \begin{bmatrix} x & y & z \end{bmatrix}^T$, the body's rotational kinetic energy is

$$\begin{aligned} \text{KE}_{\text{rot}} &= \frac{1}{2} \int_V \rho (\hat{\omega}_1 r_B)^T (\hat{\omega}_1 r_B) dV \\ &= \frac{1}{2} \int_{-h}^h \int_{-w}^w \int_{-\ell}^{\ell} \rho \begin{bmatrix} -y\omega_1 & x\omega_1 & 0 \end{bmatrix} \begin{bmatrix} -y\omega_1 \\ x\omega_1 \\ 0 \end{bmatrix} dx dy dz \\ &= \frac{1}{2} \int_{-h}^h \int_{-w}^w \int_{-\ell}^{\ell} \rho \omega_1^2 (x^2 + y^2) dx dy dz \end{aligned}$$

Crunching this integral by hand or with Mathematica produces

$$\begin{aligned} \text{KE}_{\text{rot}} &= \frac{1}{2} \rho \omega_1^2 \frac{8}{3} \ell w h (\ell^2 + w^2) \\ &= \frac{1}{2} m_{\text{tot}} \omega_1^2 \frac{1}{3} (\ell^2 + w^2) \\ &= \frac{1}{2} m_{\text{tot}} \omega_1^2 \frac{1}{12} ((2\ell)^2 + (2w)^2) \end{aligned}$$

where m_{tot} is the total mass of the block, $m_{\text{tot}} = \rho(2\ell)(2w)(2h) = 8\rho\ell wh$. The last expression is included to express the answer in terms of the side lengths of the block, the way it may appear in other references that solve this problem (say, a physics book).

Notice that, if we fix m_{tot} , this answer is independent of h . That is, for this geometry the kinetic energy of a very tall block rotating will be the same as that for a very short block as long as they have the same total mass and angular velocity. Physically this is true because the dimension h does not influence how mass is distributed about the axis of rotation. Is this always true?

Example:

Rotation of a rigid block again. Using the same block and notation from the previous example, let's see how the computations are affected by changing the axis of rotation. This time, let's rotate the block about an axis in the yz plane that passes through an edge of the block. That is, use a body angular velocity of $\omega_2 = \begin{bmatrix} 0 & \omega_2 \cos \theta & \omega_2 \sin \theta \end{bmatrix}^T$, where $\theta = \arctan \frac{h}{w}$. Moving forward

we make use of the shorthand $s_\theta = \sin \theta$ and $c_\theta = \cos \theta$. Keep in mind s_θ and c_θ are constants in this problem. Now

$$\begin{aligned}
\text{KE}_{\text{rot}} &= \frac{1}{2} \int_{\mathcal{V}} \rho (\hat{\omega}_2 r_B)^T (\hat{\omega}_2 r_B) dV \\
&= \frac{1}{2} \int_{-h}^h \int_{-w}^w \int_{-\ell}^{\ell} \rho \begin{bmatrix} \omega_2(z c_\theta - y s_\theta) & \omega_2 x s_\theta & -\omega_2 x c_\theta \end{bmatrix} \begin{bmatrix} \omega_2(z c_\theta - y s_\theta) \\ \omega_2 x s_\theta \\ -\omega_2 x c_\theta \end{bmatrix} dx dy dz \\
&= \frac{1}{2} \int_{-h}^h \int_{-w}^w \int_{-\ell}^{\ell} \rho \omega_2^2 (z^2 c_\theta^2 - 2zyc_\theta s_\theta + y^2 s_\theta^2 + x^2) dx dy dz \\
&= \frac{1}{2} \rho \omega_2^2 \frac{8}{3} \ell w h (h^2 c_\theta^2 + w^2 s_\theta^2 + \ell^2) \\
&= \frac{1}{2} m_{\text{tot}} \omega_2^2 \frac{1}{3} (h^2 c_\theta^2 + w^2 s_\theta^2 + \ell^2) \\
&= \frac{1}{2} m_{\text{tot}} \omega_2^2 \frac{1}{12} ((2h)^2 c_\theta^2 + (2w)^2 s_\theta^2 + (2\ell)^2)
\end{aligned}$$

In this case our result depends on all three of the block dimensions, ℓ , w , and h . This might have been expected, as changing any one of these dimensions does change the distribution of mass about the axis of rotation.

Also, this example gives some insight into the complexity of the general case. If the body angular velocity ω is an arbitrary vector (not aligned with any particular axis or plane), then every entry in the vector $\hat{\omega} r_B$ will be a binomial, and the number of terms in the volume integral will grow to 9. Even with python to help us, we'd rather not repeatedly perform these integrals every time the body angular velocity changes. Can we use some geometric properties to simplify the dependence of rotational kinetic energy on ω ?

The Rotational Inertia Tensor

Take home messages:

- Body angular velocity can be isolated from the last lecture's volume integral. The remaining integral gives the body's inertia tensor.
- Placing the body frame at the center of mass simplifies the expression of total kinetic energy when both translations and rotations are present
- With m and \mathcal{I} in hand, total kinetic energy is just the sum of a quadratic in \dot{p} and a quadratic in ω .

Defining Rotational Inertia

Let's return to our integral expression of rotational kinetic energy.

$$\text{KE}_{\text{rot}} = \frac{1}{2} \int_{\mathcal{V}} \rho (\hat{\omega} r_B)^T (\hat{\omega} r_B) d\mathcal{V}$$

We can manipulate this integral by utilizing the following property of the hat operator, $\hat{\omega} r_B = -\hat{r}_B \omega$. If this is not immediately apparent, it can easily be verified by hand. We can make use of this fact to produce

$$\begin{aligned} \text{KE}_{\text{rot}} &= \frac{1}{2} \int_{\mathcal{V}} \rho (\hat{\omega} r_B)^T (\hat{\omega} r_B) d\mathcal{V} \\ &= \frac{1}{2} \int_{\mathcal{V}} \rho (-\hat{r}_B \omega)^T (-\hat{r}_B \omega) d\mathcal{V} \\ &= \frac{1}{2} \int_{\mathcal{V}} \rho (\hat{r}_B \omega)^T (\hat{r}_B \omega) d\mathcal{V} \\ &= \frac{1}{2} \int_{\mathcal{V}} \rho \omega^T \hat{r}_B^T \hat{r}_B \omega d\mathcal{V} \end{aligned}$$

Notice the body angular velocity ω is independent of the volume integral over $d\mathcal{V}$. Thus the ω 's can be pulled outside the integral producing

$$\text{KE}_{\text{rot}} = \frac{1}{2} \omega^T \left(\int_{\mathcal{V}} \rho \hat{r}_B^T \hat{r}_B d\mathcal{V} \right) \omega$$

The quantity inside the parentheses is known as the inertia tensor \mathcal{I} for a rigid body. If computed once, it can be used for all time to relate body angular velocity and rotational kinetic energy. In the Cartesian coordinates of our two examples, $r_B = [x \ y \ z]^T$ and thus

$$\mathcal{I} = \int_{\mathcal{V}} \rho \hat{r}_B^T \hat{r}_B d\mathcal{V} = \int_{\mathcal{V}} \rho \begin{bmatrix} (y^2 + z^2) & -xy & -xz \\ -xy & (x^2 + z^2) & -yz \\ -xz & -yz & (x^2 + y^2) \end{bmatrix} dx dy dz$$

Combining Rotations and Translations

Suppose we have a rigid body in both translational and rotational motion relative to the stationary frame. Keeping the body frame affixed to the center of mass, it is now the pair (p, R) that gives the position and orientation of the body relative to the stationary frame. For a particle at r_B , its velocity is given by $\dot{p} + \dot{R}r_B = \dot{p} + R\hat{\omega}r_B$. Noting this, the body's total kinetic energy is

$$\begin{aligned}\text{KE}_{\text{tot}} &= \frac{1}{2} \int_V \rho (\dot{p} + R\hat{\omega}r_B)^T (\dot{p} + R\hat{\omega}r_B) dV \\ &= \frac{1}{2} \int_V \rho \left(\dot{p}^T \dot{p} + 2\dot{p}^T R\hat{\omega}r_B + (R\hat{\omega}r_B)^T (R\hat{\omega}r_B) \right) dV \\ &= \frac{1}{2} \int_V \rho \left(\dot{p}^T \dot{p} + 2\dot{p}^T R\hat{\omega}r_B + \omega^T \hat{r}_B^T \hat{r}_B \omega \right) dV\end{aligned}$$

Examining specifically the middle term

$$\frac{1}{2} \int_V \rho 2\dot{p}^T R\hat{\omega}r_B dV = (\dot{p}^T R\hat{\omega}) \int_V \rho r_B dV = (\dot{p}^T R\hat{\omega}) 0 = 0$$

where the integral vanishes because we have placed the body frame at the center of mass. Simplifying the remaining nonvanishing terms

$$\begin{aligned}\text{KE}_{\text{tot}} &= \frac{1}{2} \int_V \rho \left(\dot{p}^T \dot{p} + \omega^T \hat{r}_B^T \hat{r}_B \omega \right) dV \\ &= \frac{1}{2} (\dot{p}^T \dot{p}) \int_V \rho dV + \frac{1}{2} \omega^T \left(\int_V \rho \hat{r}_B^T \hat{r}_B dV \right) \omega \\ &= \frac{1}{2} m \dot{p}^T \dot{p} + \frac{1}{2} \omega^T \mathcal{I} \omega\end{aligned}$$

Body Velocities

Without any real motivation except for the fact that $R^{-1}\dot{R}$ is important in calculating the kinetic energy for pure rotation, look at $g^{-1}\dot{g}$, called the *body velocity* for $SE(n)$.

$$g^{-1}\dot{g} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \dot{R} & \dot{p} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{R} & \dot{p} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \hat{\omega} & R^T \dot{p} \\ 0 & 0 \end{bmatrix}$$

Now look at kinetic energy.

$$KE = \frac{1}{2}m\|\dot{p}\|^2 + \frac{1}{2}\omega^T \mathcal{I}\omega$$

and note that since R preserves distances (and therefore R^T preserves distances), we can also write the kinetic energy as:

$$KE = \frac{1}{2}m\|R^T \dot{p}\|^2 + \frac{1}{2}\omega^T \mathcal{I}\omega.$$

So how do we write the kinetic energy in terms of the body velocity? Define the “hat” operator of a vector of the form $[v, \omega] \in \mathbb{R}^6$ to be

$$\begin{bmatrix} v \\ \omega \end{bmatrix} \hat{=} \widehat{\begin{bmatrix} v \\ \omega \end{bmatrix}} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}.$$

Then we can take the “unhat” of a 4×4 matrix to get an element of \mathbb{R}^6 . In particular define the body velocity to be the vector $V^b = \mathbb{R}^6$ that, when hatted, is $g^{-1}\dot{g}$,

$$V^b = (\hat{g^{-1}\dot{g}}) = \begin{bmatrix} R^T \dot{p} \\ \omega \end{bmatrix}$$

which gives us

$$KE = \frac{1}{2}(V^b)^T \begin{bmatrix} mI_{n \times n} & 0 \\ 0 & \mathcal{I} \end{bmatrix} V^b$$

The terrific thing about this formulation of the kinetic energy is that you can compute the dynamics of a set of general interconnected rigid bodies by just looking at their inertial properties in the body frame and looking at the g transformations that interconnect them.

Transforming \mathcal{I}

Look at the body velocity V^b of a particular body defined in terms of two different frames, the first to one part of a body and the second to another part of the body.

$$\begin{aligned}\widehat{V}_B^b &= g_{WB}^{-1} \dot{g}_{WB} = \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \\ \widehat{V}_C^b &= g_{WC}^{-1} \dot{g}_{WC} = (g_{WB} g_{BC})^{-1} \frac{d}{dt} (g_{WB} g_{BC}) = g_{BC}^{-1} \widehat{V}_B^b g_{BC}.\end{aligned}$$

(Note that $\dot{g}_{BC} = 0$ because g_{BC} is a transformation within the body that does not vary with time.) Looking at the right hand side, \widehat{V}_C^b is a coordinate transformation of \widehat{V}_B^b . Let's try to rewrite this without the "hat" operator.

$$\begin{aligned}\widehat{V}_C^b &= \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \widehat{\omega} & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \widehat{\omega} R & \widehat{\omega} p + v \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} R^T \widehat{\omega} R & R^T(\widehat{\omega} p + v) \\ 0 & 0 \end{bmatrix}\end{aligned}$$

Now, using the fact that $\widehat{R^T \omega} = R^T \widehat{\omega} R$ and $\widehat{\omega} p = -\widehat{p} \omega$, we get

$$\begin{aligned}\widehat{V}_C^b &= \begin{bmatrix} \widehat{R^T \omega} & R^T(-\widehat{p} \omega + v) \\ 0 & 0 \end{bmatrix} \\ \Rightarrow V_C^b &= \begin{bmatrix} R^T(-\widehat{p} \omega + v) \\ R^T \omega \end{bmatrix} \\ &= \begin{bmatrix} R^T & -R^T \widehat{p} \\ 0 & R^T \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \underbrace{\begin{bmatrix} R^T & -R^T \widehat{p} \\ 0 & R^T \end{bmatrix}}_{\text{Adjoint}} V_B^b.\end{aligned}$$

Example

Assume that g_{WB} is of the form

$$g_{WB} = \begin{bmatrix} R_Z(\theta) & \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \\ 0 & \end{bmatrix}$$

and g_{BC} is of the form

$$g_{BC} = \begin{bmatrix} R_Z(\psi) & \begin{bmatrix} p_x \\ p_y \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}$$

where ψ , p_x and p_y are constants. Then you can verify that

$$V_C^b = \begin{bmatrix} R^T & -R^T \hat{p} \\ 0 & R^T \end{bmatrix} V_B^b$$

Back to transforming \mathcal{I}

Why is this useful for transforming \mathcal{I} ? Now look at kinetic energy (which we know is coordinate invariant).

$$\begin{aligned} KE &= \frac{1}{2}(V_B^b)^T \mathcal{I}_B V_B^b \\ &= \frac{1}{2}(V_C^b)^T \mathcal{I}_C V_C^b \\ &= \frac{1}{2} \left(\begin{bmatrix} R^T & -R^T \hat{p} \\ 0 & R^T \end{bmatrix} V_B^b \right)^T \mathcal{I}_C \begin{bmatrix} R^T & -R^T \hat{p} \\ 0 & R^T \end{bmatrix} V_B^b \\ &= \frac{1}{2}(V_B^b)^T \begin{bmatrix} R^T & -R^T \hat{p} \\ 0 & R^T \end{bmatrix}^T \mathcal{I}_C \begin{bmatrix} R^T & -R^T \hat{p} \\ 0 & R^T \end{bmatrix} V_B^b. \end{aligned}$$

Hence, since this must be true for all possible V_B^b , we get that

$$\mathcal{I}_B = \begin{bmatrix} R^T & -R^T \hat{p} \\ 0 & R^T \end{bmatrix}^T \mathcal{I}_C \begin{bmatrix} R^T & -R^T \hat{p} \\ 0 & R^T \end{bmatrix}$$

Example

Consider the simple situation where g_{WB} is of the form

$$g_{WB} = \begin{bmatrix} R_Z(\theta) & \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}$$

and g_{BC} is of the form

$$g_{BC} = \begin{bmatrix} I_{3 \times 3} & \begin{bmatrix} L \\ 0 \\ 0 \end{bmatrix} \\ 0 & 1 \end{bmatrix}$$

where L is constant. If the body is a point mass and frame C is located at the mass, then in the C frame $\mathcal{I}_C = \text{diag}(m, m, m, 0, 0, 0)$. However, in the B frame (the point around which the mass is moving), we get

$$\begin{aligned}\mathcal{I}_B &= \left[\begin{array}{cc} I_{3 \times 3} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & L \\ 0 & -L & 0 \end{bmatrix} \\ 0_{3 \times 3} & I_{3 \times 3} \end{array} \right]^T \mathcal{I}_C \left[\begin{array}{cc} I_{3 \times 3} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & L \\ 0 & -L & 0 \end{bmatrix} \\ 0_{3 \times 3} & I_{3 \times 3} \end{array} \right] \\ &= \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & Lm \\ 0 & 0 & m & 0 & -Lm & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -Lm & 0 & L^2m & 0 \\ 0 & Lm & 0 & 0 & 0 & L^2m \end{bmatrix}\end{aligned}$$

The kinetic energy calculation in both instances yields the same kinetic energy of $KE = \frac{1}{2}mL^2\theta^2$.

Inertia Tensor Transformation Properties and Principal Axes

Take home messages:

- If we have the rotation R relating two frames, we immediately can relate representations of the inertia tensor in the two frames
- The inertia tensor can be diagonalized for special choices of the body frame. To find the frames for which this is true, you need only to solve an eigenvalue problem

Inertia Tensor Transformation Properties

Sometimes the need arises to change the choice of body frame. Perhaps solving for the rotational inertia is easy in one frame, but the connection of the body to some larger mechanical system is easier to view in another frame. Maybe the rotational inertia has some special structure under a different choice of body frame (more on this later). Whatever the motivation, we'd like the freedom to change the body frame. How does this affect the rotational inertia? Well, suppose the new choice of body frame b' is related to the old body frame b by a rotation $R_{b'b}$. Given the position of a particle in the old frame r_b , its position in the new frame is

$$r_{b'} = R_{b'b} r_b$$

Using this relationship, the inertia tensor in the b' frame, which we'll denote \mathcal{I}' , is

$$\begin{aligned} \mathcal{I}' &= \int_V \rho \hat{r}_{b'}^T \hat{r}_{b'} dV \\ &= \int_V \rho (\widehat{R_{b'b} r_b})^T (\widehat{R_{b'b} r_b}) dV \\ &= \int_V \rho (R_{b'b} \hat{r}_b R_{b'b}^T)^T (R_{b'b} \hat{r}_b R_{b'b}^T) dV \\ &= \int_V \rho (R_{b'b} \hat{r}_b R_{b'b}^T)^T (R_{b'b} \hat{r}_b R_{b'b}^T) dV \\ &= \int_V \rho R_{b'b} \hat{r}_b^T R_{b'b}^T R_{b'b} \hat{r}_b R_{b'b}^T dV \\ &= R_{b'b} \left(\int_V \rho \hat{r}_b^T \hat{r}_b dV \right) R_{b'b}^T \\ &= R_{b'b} \mathcal{I} R_{b'b}^T \end{aligned}$$

If the identity $\widehat{(R r)} = R \widehat{r} R^T$ (used to move from the second to the third equality) is unfamiliar, it can easily be proven by hand for both 2D and 3D cases. To check the transformation of inertia above, consider the rotational kinetic energy in the b' frame. In that frame

the body angular velocity is $\omega' = R_{b'b} \omega$, where ω signifies the body angular velocity in the b frame (as it has all along). Now rotational kinetic energy is

$$\begin{aligned}\text{KE}_{\text{rot}} &= \frac{1}{2} (\omega')^T \mathcal{I}' \omega' \\ &= \frac{1}{2} (R_{b'b} \omega)^T (R_{b'b} \mathcal{I} R_{b'b}^T) (R_{b'b} \omega) \\ &= \frac{1}{2} \omega \mathcal{I} \omega\end{aligned}$$

This shows that though rotational inertia and body angular velocity are frame dependent, rotational kinetic energy is a frame-independent quantity as we intuitively would expect.

Rigid Body Principal Axes

If we return to the rigid block examples, notice that the inertia tensor is diagonal for the coordinate frame chosen in these examples.

Though we didn't calculate \mathcal{I} explicitly, we did show

$$\mathcal{I} = \int_V \rho \hat{r}_b^T \hat{r}_b dV = \int_V \rho \begin{bmatrix} (y^2 + z^2) & -xy & -xz \\ -xy & (x^2 + z^2) & -yz \\ -xz & -yz & (x^2 + y^2) \end{bmatrix} dx dy dz$$

and for the block the off-diagonal integrals all vanish due to symmetry of the block with respect to the chosen coordinate axes. A diagonal inertia tensor may be useful to simplify calculations of energy, but does one necessarily exist for all rigid bodies (even those with geometries more complex than that of the rigid block)? Actually, every rigid body has at least one set of orthogonal coordinate axes that diagonalizes \mathcal{I} , known as the principal axes of the body. To find these axes, consider that we've already calculated \mathcal{I} in some frame b . From frame b we just need to identify the rotation R_{b^*b} that produces

$$\mathcal{I}^* = R_{b^*b} \mathcal{I} R_{b^*b}^T$$

with \mathcal{I}^* diagonal. We can manipulate this to produce

$$\mathcal{I} R_{b^*b}^T = R_{b^*b}^T \mathcal{I}^*$$

In this form we can see that the i^{th} column of $R_{b^*b}^T$, which we'll denote u_i^* , satisfies

$$\mathcal{I} u_i^* = \mathcal{I}_{ii}^* u_i^*$$

where \mathcal{I}_{ii}^* is the i^{th} term on the diagonal of \mathcal{I} . Since \mathcal{I}_{ii}^* is a scalar, the equality above has the form of an eigenvalue problem. The elements on the diagonal of \mathcal{I}^* , commonly referred to as the principal moments of inertia, are just the eigenvalues of \mathcal{I} . The columns of $R_{b^*b}^T$,

which are identically the rows of R_{b^*b} , are simply the eigenvectors associated with the eigenvalues \mathcal{I}_{ii}^* . Hence, solving the eigenvalue problem above provides the information necessary to calculate the transformation R_{b^*b} that produces the principal axes, as well defines the diagonalized inertia tensor \mathcal{I}^* .

As a final comment, we return to the comment that a rigid body has at least one set of principal axes. It turns out that a body with some symmetry properties may have more than one set. For instance, a body with an axis of rotational symmetry will have an infinite number of principal axes, corresponding to the infinite set of axes that have one axis aligned with the body's axis of symmetry. This situation causes the eigenvalue problem posed above to have a repeated eigenvalue, for which there is not just an associated eigenvector but an associated eigenspace. Any orthogonal unit basis vectors in this eigenspace will suffice in constructing the R_{b^*b} transformation.

Advanced Topics

Lagrangian Derivation of Euler's Equations

In this section we derive Euler's equations for a rotating body.

$$\begin{aligned}\dot{R} &= R\hat{\omega} \\ \mathcal{I}\dot{\omega} &= -\omega \times (\mathcal{I}\omega)\end{aligned}$$

The first equation should look familiar. It is the same as pointing out that $R^T \dot{R}$ is skew symmetric. The second equation is less intuitive, but it is certainly simple. These equations for a rotating body get rid of all the dependence on Euler angles, and hence get rid of the dependence on $\sin()$ and $\cos()$ terms and other transcendental functions.

So how should we derive this equation using Lagrangian dynamics? We know that one can write the Lagrangian in terms of R —we did this in an earlier lecture—so we will start there, assuming for this exercise that $V = 0$. As a reminder $L(R, \dot{R}) = KE(R, \dot{R})$ can be written as

$$L(R, \dot{R}) = \frac{1}{2} \int_V \rho(\dot{R}r)^T \dot{R} r dV.$$

Based on Euler's equations above, we should expect to see ω show up naturally. Indeed, we can rewrite the Lagrangian directly in terms of ω (here we are following¹⁶¹). That is, write $\ell(\omega) = \frac{1}{2} \omega^T \mathcal{I} \omega$. Both of these are equally good representations of the kinetic energy, as we saw before.

However, if we want to use the action principle expressed in terms of $\ell(\omega)$, we will need to relate variations in ω , labeled $\delta\omega$, to variations in R , labeled δR . That is, we want to use a variational principle of the form:

$$\delta \int \ell(\omega) dt = 0$$

possibly with some boundary conditions, but the Euler-Lagrange equations are in terms of $L(R, \dot{R})$, so we need to change coordinates to ω coordinates.

First, note that $R^{-1} \delta R$ is skew symmetric (for the same reason $R^{-1} \dot{R}$ is skew symmetric!). Let us use, for convenience, the notation

¹⁶¹ Jerrold E Marsden and Tudor S Ratiu. *Introduction to Mechanics and Symmetry*. Springer, 1994

$\hat{\Sigma} = R^{-1}\delta R$. So $\hat{\Sigma}$ is skew symmetric and is the variation δR mapped back to the identity $\mathbb{I}_{3 \times 3}$. Moreover, we know that $R^{-1}\dot{R} = \hat{\omega}$, which implies—by taking the derivative of $R^{-1}\dot{R} = \hat{\omega}$ with respect to R —that

$$-\underbrace{R^{-1}\delta R}_{\hat{\Sigma}}\underbrace{R^{-1}\dot{R}}_{\hat{\omega}} + R^{-1}\delta\dot{R} = \widehat{\delta\omega}.$$

(One can also get this using $R^T\dot{R} = \hat{\omega} \implies \delta(R^T\dot{R}) = \widehat{\delta\omega} \implies (\delta R)^T\dot{R} + R^T\delta\dot{R} = \widehat{\delta\omega} \implies (\delta R)^T R^T\dot{R} + R^T\delta\dot{R} = \widehat{\delta\omega} \implies (R^T\delta R)^T R^T\dot{R} + R^T\delta\dot{R} = \widehat{\delta\omega} \implies -R^T\delta R R^T\dot{R} + R^T\delta\dot{R} = \widehat{\delta\omega}$. However, using $R^{-1}\dot{R}$ generalizes to $g \in SE(3)$.) Since we have defined $\hat{\Sigma} = R^{-1}\delta R$, we can take the time derivative of this expression and get

$$\dot{\hat{\Sigma}} = -\underbrace{R^{-1}\dot{R}}_{\hat{\omega}}\underbrace{R^{-1}\delta R}_{\hat{\Sigma}} + R^{-1}\delta\dot{R}.$$

Combining these two equations, we get

$$\widehat{\delta\omega} = \dot{\hat{\Sigma}} + \hat{\omega}\hat{\Sigma} - \hat{\Sigma}\hat{\omega}.$$

Lastly, an identity that is easy to check is that $(\widehat{\omega \times \Sigma}) = \hat{\omega}\hat{\Sigma} - \hat{\Sigma}\hat{\omega}$, allowing us to “unhat” the entire expression to get

$$\delta\omega = \dot{\Sigma} + \omega \times \Sigma.$$

Lastly, note that because $\hat{\Sigma} = R^{-1}\delta R$, we know that

$$\delta R(t_0) = \delta R(t_f) = 0 \implies \Sigma(t_0) = \Sigma(t_f) = 0.$$

Getting back to the action principle, now we can evaluate the derivative with respect to variations in ω .

$$\begin{aligned} \delta \int \ell(\omega) dt &= \int_{t_0}^{t_f} \omega^T \mathcal{I} \delta \omega dt \\ &= \int_{t_0}^{t_f} \omega^T \mathcal{I} [\dot{\Sigma} + \omega \times \Sigma] dt \\ &= \int_{t_0}^{t_f} \omega^T \mathcal{I} \dot{\Sigma} + (\mathcal{I}\omega)^T (\omega \times \Sigma) dt \\ &= \int_{t_0}^{t_f} -\dot{\omega}^T \mathcal{I} \Sigma + ((\mathcal{I}\omega) \times \omega)^T \Sigma dt \\ &= \int_{t_0}^{t_f} [-\mathcal{I}\dot{\omega} + (\mathcal{I}\omega) \times \omega]^T \Sigma dt. \end{aligned}$$

The middle line is a combination of integration by parts (to get rid of $\dot{\Sigma}$ using the fact that $\Sigma(t_0) = \Sigma(t_f) = 0$) and the cross product identity $a \cdot (b \times c) = c \cdot (a \times b)$ for $a, b, c \in \mathbb{R}^3$. Just as in the variational

principle near the beginning of class, we get that term multiplying Σ in the integral must be zero.

$$\mathcal{I}\dot{\omega} = (\mathcal{I}\omega) \times \omega$$

These equations, known as Euler's equations, are a special case of the *Euler-Poincaré Equations*.

Generalization to SE(3)

Now let us look at what happens when one generalizes the approach for deriving Euler's Equations to $SE(3)$. Again, we will assume there is no potential energy and no external forces for now. We know that

$$L(g, \dot{g}) = \frac{1}{2} \int_{\mathcal{V}} \rho(\dot{g}\bar{r})^T \dot{g}\bar{r} d\mathcal{V}$$

where \bar{r} is the homogeneous representation of a point $r \in \mathbb{R}^3$ (though the integral is still over a volume in \mathbb{R}^3). This Lagrangian can be equivalently written as

$$\ell(V_b) = \frac{1}{2} V_b^T \mathcal{I} V_b$$

where V_b is the body velocity that satisfies $\hat{V}_b = g^{-1}\dot{g}$. Moreover, $\delta\ell = V_b^T \mathcal{I} \delta V_b$. Just as before, the question we face is how δV_b is related to δg .

We will use the fact that $\hat{V}_b = g^{-1}\dot{g}$ and define $\hat{\Sigma} = g^{-1}\delta g$, just as we did for Euler's equations. Differentiating the former with respect to g and the latter with respect to time, we get:

$$\begin{aligned} \widehat{\delta V_b} &= -\underbrace{g^{-1}\delta g}_{\hat{\Sigma}} \underbrace{g^{-1}\dot{g}}_{\hat{V}_b} + g^{-1}\delta\dot{g} \\ \dot{\hat{\Sigma}} &= -\underbrace{g^{-1}\dot{g}}_{\hat{V}_b} \underbrace{g^{-1}\delta g}_{\hat{\Sigma}} + g^{-1}\delta\dot{g} \end{aligned}$$

which implies that

$$\widehat{\delta V_b} = \dot{\hat{\Sigma}} + \hat{V}_b \hat{\Sigma} - \hat{\Sigma} \hat{V}_b.$$

Let $\Sigma = (\sigma_\omega, \sigma_v)$ and $V_b = (\omega, v)$. Then,

$$\hat{V}_b \hat{\Sigma} - \hat{\Sigma} \hat{V}_b = \begin{bmatrix} \widehat{\omega \times \sigma_\omega} & \hat{\omega} \sigma_v - \hat{\sigma}_\omega v \\ 0 & 0 \end{bmatrix}.$$

If we "unhat" the entire equation, we get

$$\delta V_b = \dot{\Sigma} + \begin{bmatrix} \omega \times \sigma_\omega \\ \omega \times \sigma_v - \sigma_\omega \times v \end{bmatrix}.$$

Now we substitute into the action principle expressed in terms of $\ell(V_b)$.

$$\begin{aligned} \int_{t_0}^{t_f} \delta\ell(V_b) dt &= \int_{t_0}^{t_f} V_b^T \mathcal{I} \left[\dot{\Sigma} + \begin{bmatrix} \omega \times \sigma_\omega \\ \omega \times \sigma_v - \sigma_\omega \times v \end{bmatrix} \right] dt \\ &= \int_{t_0}^{t_f} \left[-V_b^T \mathcal{I} \Sigma + V_b^T \mathcal{I} \begin{bmatrix} \omega \times \sigma_\omega \\ \omega \times \sigma_v - \sigma_\omega \times v \end{bmatrix} \right] dt \end{aligned}$$

This is by integration by parts, with $\Sigma(t_0) = \Sigma(t_f) = 0$.

Now assume \mathcal{I} is block diagonal $\mathcal{I} = \text{diag}(J, m\mathbf{I}_{3 \times 3})$

$$\begin{aligned} &= \int_{t_0}^{t_f} \left[-[\dot{\omega}^T J, m\dot{v}^T] \Sigma + [\omega^T J, mv^T] \begin{bmatrix} \omega \times \sigma_\omega \\ \omega \times \sigma_v - \sigma_\omega \times v \end{bmatrix} \right] dt \\ &= \int_{t_0}^{t_f} \left[-[\dot{\omega}^T J, m\dot{v}^T] \Sigma + [(J\omega) \times \omega, mv \times \omega] \begin{bmatrix} \sigma_\omega \\ \sigma_v \end{bmatrix} \right] dt \\ &= \int_{t_0}^{t_f} \left[-[\dot{\omega}^T J, m\dot{v}^T] + [(J\omega) \times \omega, mv \times \omega] \right] \Sigma dt \\ &= \int_{t_0}^{t_f} \left[-\dot{\omega}^T J + (J\omega) \times \omega, -m\dot{v}^T + mv \times \omega \right] \Sigma dt \\ &= 0 \quad \forall \Sigma. \end{aligned}$$

As a result, we get

$$\begin{aligned} \dot{g} &= g \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} \\ J\dot{\omega} &= (J\omega) \times \omega \\ \dot{v} &= v \times \omega \end{aligned}$$

Consequences for Numerical Methods

It is worth pointing out that the above equations, paired with $\dot{R} = R\hat{\omega}$, are tricky to simulate well. This is primarily due to the fact that numerical integration of $\dot{R} = R\hat{\omega}$ typically leads to a prediction step where $R(t_k) \notin SO(3)$. Lots of numerical methods try to address this, with the most common method being to represent R as the exponential of an element of the Lie algebra, and then model the evolution of elements of the Lie algebra.

Exercises

- Derive the equations of motion for a rigid body in $SE(3)$ with $V = mgh = mge_3g\bar{0}$.
- Derive the equations of motion for a rigid body in $SE(3)$ with $V = mgh = mge_3g\bar{0}$ and external thrusters.

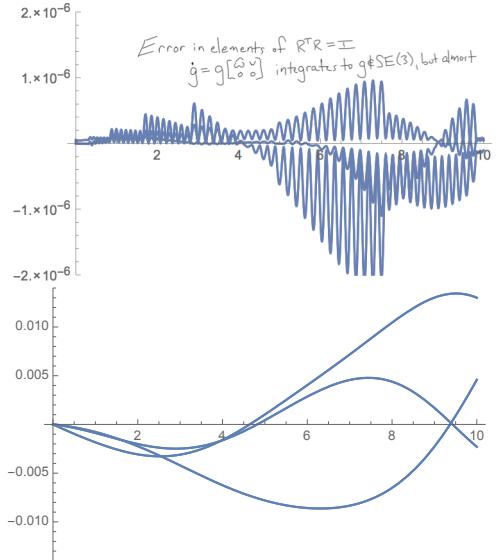


Figure 45: Applying an integrator directly to \dot{g} slowly leads to solutions that do not satisfy the requirements of the group. In this case, the elements of $R^T R$ slowly drift away from the identity matrix over ten seconds. The integrator has a huge impact on this, of course, and on the left an adaptive integrator is being used while Euler integration (step size $dt = 0.01$) is being used on the right. As a consequence of Euler integration, the size of the body can implicitly expand as a function of time.

- Derive the equations of motion for *two* rigid bodies in $SE(3)$ interconnected by a spherical joint.

Optimal control as an extension of dynamics and the Pontryagin Maximum Principle

It may be surprising to hear that optimal control—the study of decision-making over time subject to constraints—is a natural extension of the study of dynamics. This is partly for historical reasons, since the derivation of dynamics predictions based on an optimization principle was itself based on a desire to make optimality a universal principle—for *studying* optics and mechanics and dynamics and for *designing* outcomes in these areas. In this regard science and engineering used to be much more entangled culturally than they are now.

The key point from our perspective is that we will be shifting from using optimization to *predict* natural phenomena to using optimization to *design* processes. There are lots of design optimizations one might want to do, but we will focus on decision-making, where a control vector can be chosen at every instant in time to influence the state of a system. For instance, one might have a motor that can exert a torque f_τ on a mechanical system at every time t , so that the goal becomes to choose $f_\tau(t)$ to achieve some goal.

To get from the ideas in variational principles as applied to dynamics to optimal control, a few key operations are needed. First, we will need to introduce an *objective*, which will look suspiciously like the action integral, and an *input* that will give us something to optimize over. Second, we need to introduce some operations that assist with differentiation—the *integral representation of an ordinary differential equation*, the *state transition matrix* for a linear ordinary differential equation, and the *convolution equation* for an affine linear ordinary differential equation (this will be how we talk about a linear system having an ‘input’ that can be controlled). These components, along with differentiation, will yield a relatively general form of the Pontryagin Maximum Principle, which is a set of ordinary differential equations governing optimality.

It is worth noting that the formulation we present here is not complete nor is it general. However, in a few pages one can get a relatively general view of optimal control as well as numerical methods that are genuinely useful. What one does not get is a) mechanisms for telling whether a given problem even has an optimal solution at all, and b) generalizations that allow one to take into account practical limitations (e.g., input saturation), not to mention other interesting things. These other interesting facets of optimal control are what make it a field of study, rather than simply an extension of dynamics like the view taken here.

Integral representations of ordinary differential equations

We will talk about derivatives of integral equations so that we can apply chain rule to an objective function $J(x(t), u(t))$, where $u(t)$ is the *input* to the system. First, we need to be able to represent an ordinary differential equation as an integral equation, to make it easier to differentiate with respect to state and control variables.

Let us remind ourselves of some things. Suppose we have a system described by the nonlinear ordinary differential equation:¹⁶²

$$\dot{x} = f(x(t), u(t)) \quad x(0) = x_0.$$

An equivalent representation of that ordinary differential equation is:

$$x(t) = x_0 + \int_0^t f(x(\tau), u(\tau)) d\tau.$$

This latter equation is the *integral* form of the differential equation. To see that the two are equivalent, we can check if $x(t)$ satisfies the differential equation. By the Leibniz rule¹⁶³ we know that

$$\frac{d}{dt} \left[x_0 + \int_0^t f(x(\tau), u(\tau)) d\tau \right] = f(x(t), u(t)).$$

So the expression for $x(t)$ satisfies the differential equation. Note that the differential equation incorporates the initial condition in a way that makes sense—at $t = 0$ the expression evaluated to x_0 .

This integral representation of the ordinary differential equation is straight forward to differentiate using the same approaches we used before for differentiating the action integral, though this time we will differentiate with respect to the input variable $u(t)$. As a result, we can apply chain rule to a cost function that depends on $x(t)$ and $u(t)$ and evaluate the first derivative of $J(x(t), u(t))$ with respect to $u(t)$. When that derivative is equal to zero (for all possible directions $v(t)$, the perturbations to $u(t)$), we know that we have a local minimizer, a local maximizer, or possibly a saddle point.

Derivatives of Objective Functions with Dynamic Constraints

Suppose we have an optimal control system and we want to optimize the system by setting the derivative of a cost J equal to zero. We will minimize an objective function J with respect to the input u .¹⁶⁴ For reasons that largely have to do with the nice mathematical outcome it ensures, we will choose the cost function to be of the form:¹⁶⁵

$$J(u) = \int_0^T \ell(x, u) dt + m(x(T))$$

Moreover, we will assume that the state $x(t)$ and the control $u(t)$ are constrained by their dynamic relationship

$$\dot{x} = f(x, u) \quad x(0) = x_0.$$

¹⁶² We will typically suppress the dependence on t to avoid too much notation.

¹⁶³ We haven't used this since we discussed impacts, but here it is again!

¹⁶⁴ What we should minimize with respect to is a key decision; we could also minimize with respect to the pair (x, u) that satisfies the dynamics, or, possibly, a pair (x, μ) that does not satisfy the dynamics. Each of these leads to different algorithms, but we will focus here on minimizing solely with respect to u —the most common choice.

¹⁶⁵ The function under the integral sign ℓ will also depend explicitly on time t , but we will not worry about that here; it does not impact the analysis.

To differentiate J with respect to u , we will take the same approach as we did when developing the least action principle, and differentiate using the directional derivative.

$$\frac{d}{d\epsilon} J(u + \epsilon v)|_{\epsilon=0} = \int_0^T \frac{d}{d\epsilon} \ell(x, u + \epsilon v) dt + \frac{d}{d\epsilon} m(x(T))|_{\epsilon=0}$$

This would be easy except for the fact that x depends on u , so we have to figure out how they are related, and the first bit of that includes carefully applying chain rule. The above equation is rewritten below adjusting the notation to make the dependence on u and its perturbations $v(t)$ more clear.

$$\begin{aligned} \frac{d}{d\epsilon} J(u + \epsilon v)|_{\epsilon=0} &= \int_0^T \frac{d}{d\epsilon} \ell(x(u + \epsilon v), u + \epsilon v) dt + \frac{d}{d\epsilon} m(x(u + \epsilon v)|_{t=T})|_{\epsilon=0} \\ &= \int_0^T D_1 \ell(x(u + \epsilon v), u + \epsilon v) \frac{\partial x}{\partial u} v + D_2 \ell(x(u + \epsilon v), u + \epsilon v) v dt + Dm(x(u + \epsilon v)) \frac{\partial x(T)}{\partial u}|_{\epsilon=0} \\ &= \int_0^T D_1 \ell(x(t), u(t)) \frac{\partial x}{\partial u}(t) + D_2 \ell(x(t), u(t)) v(t) dt + Dm(x) \frac{\partial x}{\partial u}(T) \end{aligned}$$

This notation is pretty unwieldy, but we are going to need it below. The D_1 notation just means we are differentiating with respect to the first variable and the D_2 means we are differentiating with respect to the second variable. If there is only one variable, we just use D to denote differentiation. Moreover, the expression $\frac{\partial x}{\partial u}$ is a function of time—it tell us how x is affected, to first order, when u is changed...anywhere in the time evolution of the input u . This is important conceptually, because in a dynamical system (like a pendulum), changing u at any point will generally have an effect on all subsequent states x later in time. So, we don't yet know how to concretely compute $\frac{\partial x}{\partial u}(t)$, but this first-order sensitivity will be important for analyzing the derivatives of J with respect to u .

To simplify the notation further, define $a^T = D_1 \ell(x(t), u(t))$ and $b^T = D_2 \ell(x(t), u(t))$, and $z = \frac{\partial x}{\partial u}(t)$. This gives us

$$\frac{d}{d\epsilon} J(u + \epsilon v)|_{\epsilon=0} = \int_0^T a(t)^T z(t) + b^T v(t) dt + Dm(x) z(T).$$

This really is just simplifying the notation to make the equation less visually complex—nothing mathematical has happened yet. However, it does let us interpret each term. The first two terms are the derivative of ℓ with respect to the state (evolving in time) multiplying the perturbation of the state due to a perturbation in control (also evolving in time). The second term is the derivative of ℓ with respect to the control (evolving in time) multiplying the perturbation to the control. So the derivative of the objective has two parts—the sensitivity to the state and the sensitivity to the control. They add together as a consequence of chain rule.

Now what we need is to understand the sensitivity of the state to the control $z(t) = \frac{\partial x}{\partial u}(t)$. This is where the integral form of the ordinary differential equation comes in. Assume that $\dot{x} = f(x, u)$ so that

$$x(t) = x_0 + \int_0^t f(x(\tau), u(\tau)) d\tau.$$

We can differentiate this with respect to u .¹⁶⁶

$$\frac{\partial x}{\partial u}(t) \cdot v(t) = \frac{d}{d\epsilon} \left[x_0 + \int_0^t f(x(\tau), u(\tau) + \epsilon v(\tau)) d\tau \right]_{\epsilon=0}$$

Evaluating this expression using chain rule, we get:

$$\begin{aligned} &= \left[\frac{d}{d\epsilon} \int_0^t f(x(\tau), u(\tau)) d\tau \right]_{\epsilon=0} \\ &= \int_0^t D_1 f(x(\tau), u(\tau)) \frac{\partial x}{\partial u}(\tau) e + D_2 f(x(\tau), u(\tau)) v(\tau) d\tau \end{aligned}$$

Looking at this can make one's head spin a bit the first time, so, as we did above with the derivative of the objective function, relabeling some variables can help. Setting $A(t) = D_1 f(x(t), u(t))$, and $B(t) = D_2 f(x(t), u(t))$ the equation above becomes

$$\frac{\partial x}{\partial u}(t) \cdot v(t) = \int_0^t A(\tau) z(\tau) + B(\tau) v(\tau) d\tau.$$

This also is just relabeling to make the equation a little easier to look at. We see that the sensitivity of the trajectory $x(t)$ to perturbations to $u(t)$ is a combination of how the dynamics evolve the sensitivity (the matrix A multiplies the sensitivity z so that changes in u early in time get integrated to future times) as well as the immediate effect of v on the trajectory.

Moreover, the equation above is the integral representation of the ordinary differential equation

$$\dot{z} = A(t)z(t) + B(t)v(t) \quad z(0) = 0.$$

That is, the function $z = \frac{\partial x}{\partial u}(t)$ satisfies a linear differential equation, which will help with analysis and will help identify what appears in the derivative of $J(u)$. Since this equation is linear, we need two ideas from linear systems theory: the *state transition matrix* and the *convolution equation*.

State transition matrices

The linear (time-varying) system above simplifies if we set $B(t) = 0$, which we do for the moment. This gives us a linear (time-varying) system

$$\dot{x} = A(t)x(t) \quad x(t_0) = x_0.$$

¹⁶⁶ Note the slight abuse of notation here, since I am allowing x to still take t as its argument.

The *state transition matrix* for this system is defined to be the matrix $\Phi(t, t_0)$ that satisfies the *matrix-valued* differential equation

$$\dot{\Phi} = A(t)\Phi \quad \Phi(t_0, t_0) = Id_{n \times n}.$$

A remarkable and useful property of this matrix is that

$$x(t) = \Phi(t, t_0)x_0$$

holds for any linear system. In the special case where $A(t) = A$ —the system is linear time-invariant (LTI)—it turns out that $\Phi(t, t_0) = e^{A(t-t_0)}$, the matrix exponential of $A(t - t_0)$.¹⁶⁷

¹⁶⁷ So matrix exponentials show up in both linear systems theory and geometry. This isn't a coincidence, but I won't get into that here.

Convolution equations

Now we add $B(t)$ back in and consider the linear time-varying system

$$\dot{x} = A(t)x(t) + B(t)u(t) \quad x(t_0) = x_0.$$

This does not have the same linear structure as the previous system, so the state transition matrix associated with $A(t)$ is not quite enough to compute the solution. Instead, we get the *convolution* equation

$$x(t) = \Phi(t, t_0)x_0 + \int_{t_0}^t \Phi(t, \tau)B(\tau)u(\tau)d\tau.$$

This equation is not quite as complex as it looks. Roughly speaking, it says that the control is coupled to what the system would have done based on its initial condition through the integral equation that weighs the input against the state transition matrix operating on the input. If $A = 0$, then we expect to be able to directly control the state; this is exactly what is predicted because for $A = 0$ we have $\Phi(t, t_0) = Id_{n \times n} \forall t$.

The Pontryagin Maximum Principle as a Variational Principle

$$\begin{aligned} J(u) &= \int_{t_0}^{t_f} \ell(x, u)dt + m(x(t_f)) \\ \dot{x} &= f(x, u) \\ x(t_0) &= x_0 \\ DJ(u) \cdot v &= \int_{t_0}^{t_f} \underbrace{D_1 \ell(x, u)}_{a(t)^T z(t)} \underbrace{\frac{\partial x}{\partial u}}_{z(t)} + \underbrace{D_2 \ell(x, u)}_{b(t)^T v} v dt + \underbrace{Dm(x(t_f))}_{p_1^T} \underbrace{\frac{\partial x(t_f)}{\partial u}}_{z(t_f)} \end{aligned}$$

We know that $z(t)$ and $v(t)$ are related through the linear differential equation

$$\dot{z} = \underbrace{D_1 f(x, u)}_{A(t)} z + \underbrace{D_2 f(x, u)}_{B(t)} v$$

and

$$z(t_0) = 0$$

because x_0 is given, so cannot be variable.

Using this notation,

$$\begin{aligned} DJ(u) &= \int_{t_0}^{t_f} a^T z + b^T v dt + p_1^T z(t_f) \\ &= \int_{t_0}^{t_f} a^T \left(\int_{t_0}^t \phi(t, \tau) B(\tau) v(\tau) d\tau \right) + b^T v dt + p_1^T \left(\int_{t_0}^{t_f} \phi(t_f, \tau) B(\tau) v(\tau) d\tau \right) \end{aligned}$$

now, since $a(t)^T$ does not depend on τ , we can bring it in the second integral.

$$\begin{aligned} &= \int_{t_0}^{t_f} \int_{t_0}^t a^T \phi(t, \tau) B(\tau) v(\tau) d\tau dt + \int_{t_0}^{t_f} b^T v dt + p_1^T \left(\int_{t_0}^{t_f} \phi(t_f, \tau) B(\tau) v(\tau) d\tau \right) \end{aligned}$$

Now change the order of integration, noting in Fig. 48 the change in boundary conditions on the integrals.

Figure 46: Changing order of integration.

$$= \int_{t_0}^{t_f} \int_{\tau}^{t_f} a^T \phi(t, \tau) B(\tau) v(\tau) dt d\tau + \int_{t_0}^{t_f} b^T v dt + p_1^T \left(\int_{t_0}^{t_f} \phi(t_f, \tau) B(\tau) v(\tau) d\tau \right)$$

factor out the terms that depend on τ to the right

$$= \int_{t_0}^{t_f} \left[\int_{\tau}^{t_f} a^T \phi(t, \tau) dt \right] B(\tau) v(\tau) d\tau + \int_{t_0}^{t_f} b^T v dt + p_1^T \left(\int_{t_0}^{t_f} \phi(t_f, \tau) B(\tau) v(\tau) d\tau \right)$$

now combine the left term with the right term

$$= \int_{t_0}^{t_f} \left[\underbrace{\int_{\tau}^{t_f} a^T \phi(t, \tau) dt}_{p(\tau)^T} + p_1^T \phi(t_f, \tau) \right] B(\tau) v(\tau) d\tau + \int_{t_0}^{t_f} b^T v dt$$

and, ignoring how we actually compute $p(\tau)$ for a moment...and replacing τ with t because it doesn't matter what we *label* the integration variable, we get

$$= \int_{t_0}^{t_f} \left[p(t)^T B(t) + b(t)^T \right] v(t) dt = 0$$

This lets us conclude that the term in the brackets must be equal to zero.

$$p(t)^T B(t) + b(t)^T = 0$$

Substituting in our notation, we get

$$p(t)^T D_2 f(x, u) + D_2 \ell(x, u) = 0$$

But we still need to figure out how to evaluate $p(\tau)$. Coming back to the expression for $p(\tau)$.

$$p(\tau)^T = \int_{\tau}^{t_f} a^T \phi(t, \tau) dt + p_1^T \phi(t_f, \tau)$$

This is similar to the convolution equation we saw before, but it isn't quite the same. The time variable for p is τ and τ appears at the beginning of the integral instead of the end of the integral, suggesting that this system is running backwards in time. This is at least consistent with the fact that $\phi(T, T) = I_{n \times n}$, and that p_1 is evaluated at time T .

Now, $p(t)$ satisfies a nice but somewhat unexpected differential equation. You can think of the integral expression that defines $p(s)^T$ as being the convolution equation, except that in this case the state transition matrix $\Phi(\tau, s)$ operates on a boundary condition at the final time T instead of the initial time 0. So let's examine $p(\tau)$ and differentiate it to find out what ordinary differential equation describes it.¹⁶⁸

$$p(\tau)^T = \int_{\tau}^{t_f} a(s)^T \Phi(s, \tau) ds + p_1^T \Phi(t_f, \tau)$$

so

$$\begin{aligned} \frac{d}{d\tau} p(\tau)^T &= \frac{d}{d\tau} \left[\int_{\tau}^{t_f} a(s)^T \Phi(s, \tau) ds + p_1^T \Phi(t_f, \tau) \right] \\ &= -a(\tau)^T \phi(\tau, \tau) + p_1^T \phi(t_f, \tau)(-A) + \int_{\tau}^{t_f} a(s)^T \Phi(s, \tau)(-A) ds \\ &= -a(\tau)^T + \left[\int_{\tau}^{t_f} a(s)^T \Phi(s, \tau) ds + p_1^T \phi(t_f, \tau) \right](-A) \\ &= -a(\tau)^T - p^T A. \end{aligned}$$

So we get that

$$\dot{p}^T = -p^T A - a^T$$

or, in column format,

$$\dot{p} = -A^T p - a = -D_1 f(x, u)^T p - D_1 \ell(x, u)^T.$$

What boundary condition do we know for p ? Only what its value is as t_f , so instead of an initial condition we have the terminal condition

$$p(t_f) = p_1$$

Let's collect what we know so far.

$$\begin{aligned} \dot{x} &= f(x, u) \quad x(t_0) = x_0 \\ \dot{p} &= -D_1 f(x, u)^T p - D_1 \ell(x, u) \quad p(t_f) = Dm(x(t_f)) \\ 0 &= p^T D_2 f(x, u) + D_2 \ell(x, u) \end{aligned}$$

¹⁶⁸ Explain here that $\frac{d}{d\tau} \phi(t, \tau) = \phi(t, \tau)(-A)$, and why it is true.

These three equations are a version of *Pontryagin's Maximum Principle*. If one can solve the third equation for u and substitute into the first two equations, what results is a two point boundary value problem that can be numerically solved (including in python).

Example Consider a linear spring-mass system with a force applied. The ODE governing this system is

$$\ddot{x} = -kx + u,$$

where u is the force applied. In first-order form, this is

$$\begin{bmatrix} \dot{x} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \quad \begin{bmatrix} x(t_0) \\ w(t_0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Now let's assume that we want the state to move from the initial condition to some desired state, such as $(x_{ref}, w_{ref}) = (2, 0)$. Then we could create an objective function

$$J = \int_{t_0}^{t_f} \ell(x, u) dt + m(x(t_f))$$

where

$$\ell = \frac{d_1}{2}(x(t) - 2)^2 + \frac{d_2}{2}u(t)^2$$

and

$$m(x(t_f)) = \frac{c_1}{2}(x(t_f) - 2)^2 + \frac{c_2}{2}w(t)^2.$$

Then the equation governing p would be

$$\begin{bmatrix} \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = - \begin{bmatrix} 0 & -k \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} - \begin{bmatrix} d_1(x(t) - 2) \\ 0 \end{bmatrix} \quad \begin{bmatrix} p_1(t_f) \\ p_2(t_f) \end{bmatrix} = \begin{bmatrix} c_1(x(t_f) - 2) \\ 0 \end{bmatrix}.$$

Lastly, looking at

$$p^T D_2 f + D_2 \ell = 0$$

we get that

$$u(t) = -p_2(t).$$

This gives us the following two point boundary value problem to solve.

$$\begin{aligned} \dot{x} &= -w \\ \dot{w} &= -kx - p_2 \\ \dot{p}_1 &= -d_1 x + k p_2 + 2 \\ \dot{p}_2 &= -p_1 \end{aligned}$$

with the boundary conditions

$$\begin{aligned}x(t_0) &= 1 \\w(t_0) &= 0 \\p_1(t_f) &= c_1(x(t_f) - 2) \\p_2(t_f) &= 0\end{aligned}$$

Solving this ODE will give us p_2 which in turn gives us the optimal $u(t)$.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.integrate import solve_bvp # import the TPBVP solver
4
5 # initialize parameters
6 N = 100
7 T = 10.0
8 k = 1.6
9 d1 = 2.0
10 d2 = 0.1
11 c1 = 100.0
12 c2 = 0.0 #weight on terminal velocity equals zero in all code below #not
     included in any equations of motion#
13 xref = 2.0 #desired position at the end of the time horizon
14 x0 = np.array([1., 0.])
15 t_axis = np.linspace(T, 0, N) # forward time steps
16 t_axis2 = np.linspace(0, T, N) # backward time steps
17
18 # initial guess for the time series (x,p)
19 z = np.zeros((4, N))
20
21 # TPBVP system dynamics (in vector form)
22 def fun(t, z):
23     return np.vstack((z[1],
24                       -k*z[0]-(1/d2)*z[3],
25                       -d1*z[0]+k*z[3]+d1*xref,
26                       -1.0*z[2]))
27
28 # boundary condition specification based on usage of solve_bvp, see scipy
     documentation
29 def bc(z_a, z_b):
30     xT = z_b[0:2]
31     p1 = np.vstack((c1*(xT[0]-xref), c2*xT[1]))
32     return np.array([z_a[0]-x0[0], z_a[1]-x0[1], z_b[2]-p1[0][0], z_b[3]-p1
     [1][0]])
33
34 # solve TPBVP
35 res = solve_bvp(fun, bc, t_axis2, z)
36 x1 = res.sol(t_axis2)[0:2]
37 p1 = res.sol(t_axis2)[2:4]
38 u1 = -1/d2 * p1[1]
39 print("Solver success: ", res.success)
40
41 # plot
42 fig = plt.figure()
43 ax = plt.gca()
44 ax.plot(t_axis2, x1[0,:], 'r')
```

```

45 ax.plot(t_axis2, x1[1,:], 'b')
46 ax.plot(t_axis2, u1.reshape(N), 'g')
47 ax.legend(['$x_1$(t)', '$x_2$(t)', 'u(t)'])
48 ax.set_title("Solution From TPBVP")
49 ax.set_xlabel("Time Steps")
50 ax.set_ylabel("System State")
51 plt.grid()
52 plt.show()
53 plt.close()

```

Note that this code can remain nearly unchanged if one replaces the system with a nonlinear pendulum, and the scipy functions will work just as well if one defines the dynamics to be the following.

```

1 # TPBVP system dynamics for nonlinear pendulum (treat k as the gravity term,
2 # also in vector form)
3 def fun(t, z):
4     return np.vstack((z[1],
5                     -k*np.sin(z[0])-c1*z[3],
6                     -d1*z[0]+k*np.cos(z[0])*z[3]+xref,
7                     -1.0*z[2]))

```

Exercise Use the code above to find an optimal control that controls the bead on the hoop to a particular angle away from vertical.

How do we tell if a solver has given us a good solution?

As the code above indicates, one can *state* the conditions for optimality for a general nonlinear system easily, and *specifying code* that nominally solves for the optimal solution is not much more challenging. However, unless the solver explicitly returns an error, how do we assess the numerically-obtained solution? One possibility is to evaluate the directional derivative of J in sample directions $v(t)$, though this is going to numerically sensitive, hard to interpret (how close to zero are we expecting?), and the set of $v(t)$ one chooses may not capture the worst-case directions. Alternatively one can look at the structure of the optimality conditions, which will lead to a surprising¹⁶⁹ connection to dynamics.

Look again at the conditions in Pontryagin's Maximum Principle.

$$\begin{aligned}\dot{x} &= f(x, u) \quad x(t_0) = x_0 \\ \dot{p} &= -D_1 f(x, u)^T p - D_1 \ell(x, u) \quad p(t_f) = Dm(x(t_f)) \\ 0 &= p^T D_2 f(x, u) + D_2 \ell(x, u)\end{aligned}$$

A notational restatement of these can be achieved by using the Hamiltonian formalism.

$$H = \ell + p^T f$$

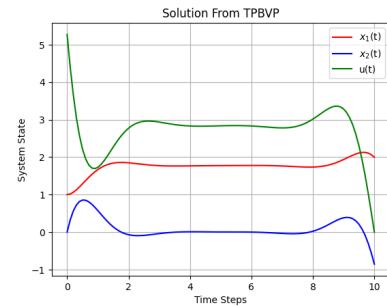


Figure 47: The solution to the TPBVP, including the optimal u , and states x and \dot{x} .

¹⁶⁹ At least to me!

The conditions for optimality are equivalent to Hamilton's equations, which we saw earlier in class.

$$\begin{aligned}\dot{x} &= D_p H^T \\ \dot{p} &= -D_x H^T \\ 0 &= D_u H^T\end{aligned}$$

Note that these equations have nothing to do with mechanical energy. Instead, from a purely formal perspective, they provide a convenient way of remembering the maximum principle. Moreover, one now sees why the variable p is labeled with a p —it shares the same role as momentum in the mechanical setting. But despite *not* being a mechanical Hamiltonian representing total energy¹⁷⁰, this Hamiltonian is conserved. After all, the only thing the proof of conservation relied on is Hamilton's equations, and the specifics of energy never played a role.¹⁷¹ The upshot is that when we obtain an optimal $u(t)$, it should be a conserved quantity.

For the spring-mass-force control problem above,

$$H = \frac{1}{2}d_1(x - x_{ref})^2 + \frac{1}{2}d_2u^2 + [p_1, p_2] \cdot \begin{bmatrix} w \\ -kx + u \end{bmatrix}$$

where $u = \frac{-1}{d_2}p_2$. Plotting this from the solution to the TPBVP gives the plot below, which is indeed very close to conserved, suggesting that for this problem the solver did a good job approximating the solution.

```

1 # plot Hamiltonian
2 H = 0.5*d1*(x1[0]-xref)**2+0.5*(1/d2)*p1[1]**2+p1[0]*x1[1]-(1/d2)*p1[1]**2-k
   *x1[0]*p1[1]
3 fig = plt.figure()
4 ax = plt.gca()
5 ax.plot(t_axis2, H[:]-sum(H)/len(H), 'r')
6 ax.legend(['$H(t)$'])
7 ax.set_title("Control Hamiltonian From TPBVP")
8 ax.set_xlabel("Time Steps")
9 ax.set_ylabel("Control Hamiltonian $H$")
10 plt.grid()
11 plt.show()
12 plt.close()
```

You will not be surprised to hear that as systems become more complex,¹⁷² black-box solvers struggle to provide good solutions.

¹⁷⁰ ...or something like total energy

¹⁷¹ This is not *quite* true—where did *an* aspect of mechanics play a role in proving that the Hamiltonian is conserved?

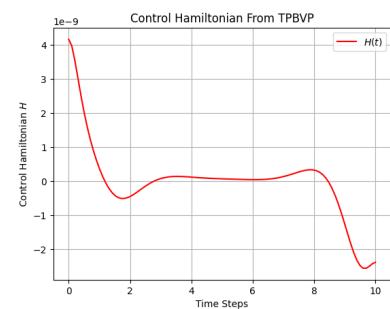


Figure 48: The control Hamiltonian H for the spring-mass-force system. I have plotted the difference from the overall average value of H .

¹⁷² higher dimensional, nonlinear, and even non-differentiable...

References

- [1] Vladimir Igorevich Arnol'd. *Mathematical Methods of Classical Mechanics*. Springer, 1974.
- [2] Leon M Lederman and Christopher T Hill. *Symmetry and the beautiful universe*. Prometheus books, 2011.
- [3] Jerrold E Marsden and Tudor S Ratiu. *Introduction to Mechanics and Symmetry*. Springer, 1994.
- [4] David Morin. *Introduction to Classical Mechanics*. Cambridge University Press, 2008.
- [5] Richard M Murray, Zexiang Li, and S Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC press, 1994.