Lecture 19. The Method of Zs

When problems get complicated numerical complexity makes computation SLOW

The method of Zs speeds the computation up

We will eventually look at the third problem of problem set 5
which is computationally impractical (at least in Mathematica) without Zs
We set this in the context of Hamilton’s equations with nonholonomic constraints

I generally replace connectivity constraints with pseudononholonomic constraints

When I do this we recall that we can write \( \dot{q}^i = S_j^i u^j \)

The momentum can be written in terms of \( u^j \).

\[
p_i = \frac{\partial L}{\partial \dot{q}^i} \{ q^i \rightarrow S_j^i u^j \}
\]

where I am borrowing Mathematica’s substitution operator
The game is to isolate the $u$s, writing the derivatives as coefficients times $u$s.

I will denote the coefficients in the momentum (Hamilton and reduced Hamilton) equations by $Z$s.

This is going to seem a wee bit weird. Bear with me. It’ll make sense eventually.
We note that $p_i$ is linear in $u^j$, which we can write as

$$p_i = Z_{ij} u^j$$

We can obtain the $Z$s from the usual momentum expression

$$p_i = M_{ij} \dot{q}^j = M_{ik} \dot{q}^k = M_{ik} S^k_j u^j \Rightarrow Z_{ij} = M_{ik} S^k_j$$

or

$$Z_{ij} = \frac{\dot{p}_i}{\dot{u}^j}$$

The latter is often easier
It is clear from the expression

$$Z_{ij} = M_{ik} S_{jk}$$

that $Z$ does not depend on $u$
We already have the equations for the evolution of $\mathbf{q}$: $\dot{\mathbf{q}} = S^i_j \mathbf{u}^j$

Hamilton’s equations become

$$
\dot{p}_i = Z_{ij} \dot{u}^j + \dot{Z}_{ij} \mathbf{u}^j = \frac{\partial L}{\partial \dot{q}^i} + Q_i
$$

where any explicit

$$
\dot{q}^i = S^i_j \mathbf{u}^j
$$

We need to replace

$$
\dot{Z}_{ij} = \frac{\partial Z_{ij}}{\partial \dot{q}^k} \dot{q}^k + \frac{\partial Z_{ij}}{\partial \dot{q}^k} S^k_m \mathbf{u}^m
$$
So that we have Hamilton’s equations in the form

\[ Z_{ij} \dot{u}^j + \frac{\partial Z_{ij}}{\partial q^k} S_m^k u^m u^j = \frac{\partial L}{\partial q^i} + Q_i \]

Remember that these are not actually correct because I have not written the Lagrange multipliers

\[ Z_{ij} \dot{u}^j + \frac{\partial Z_{ij}}{\partial q^k} S_m^k u^m u^j = \frac{\partial L}{\partial q^i} + Q_i + \lambda_k C_i^k \]

is the correct form
We have learned to multiply by $S$ to obtain the correct form
the reduced Hamilton’s equations

$$Z_{ij} u^i S_n^i + \frac{\partial Z_{ij}}{\partial q^k} S_m^k u^m u^j S_n^i = \frac{\partial L}{\partial q^i} S_n^i + Q_i S_n^i$$

We need to solve these simultaneously with the $q$ equations

$$\dot{q}^i = S_j^i u^j$$
\[ \frac{\partial L}{\partial q^i} \] is, formally, a terribly complicated expression

\[ \frac{\partial L}{\partial q^i} = \frac{1}{2} \dot{q}_p \frac{\partial M_{pq}}{\partial q^i} \dot{q}_q - \frac{\partial V}{\partial q^i} \]

\[ \frac{\partial L}{\partial q^i} = \frac{1}{2} S_m^p u^m \frac{\partial M_{pq}}{\partial q^i} S_n^q u^n - \frac{\partial V}{\partial q^i} \]

However, if we relegate the connectivity constraints to the pseudononholonomic world most of the derivatives of \( \mathbf{M} \) are zero, and it is not at all bad. We will see this in context.
The full formal expression for the reduced Hamilton’s equations

\[ Z_{ij} \dot{u}^j S^i_n + \frac{\partial Z_{ij}}{\partial q^k} S^k_m u^m u^j S^i_n = \left( \frac{1}{2} S^p_m u^m \frac{\partial M_{pq}}{\partial q^i} S^q_n u^n - \frac{\partial V}{\partial q^i} \right) S^i_n + Q_i S^i_n \]

This is a set of nonlinear first order ordinary differential equations in \( u \)
the coefficients of which are functions of \( q \), often quite complicated functions

This would be a snap to integrate numerically were the coefficients to be constant

The method of Zs pretends that they are
I have adapted the method of Zs from Kane and Levinson’s 1983 paper, cited in the text.

Their work used Kane’s method, which I explore in the text, but not in class.

My method of Zs is not as complete as it might be, because I allow some of the coefficients to remain as explicit functions of q.

This relies on \( \frac{\partial L}{\partial q^i} \) and \( S^j_i \) actually being pretty simple functions of q.

(and also the generalized forces)
Write Hamilton’s equations as follows

\[ Z_{ij} S_n^i u^j + \frac{\partial Z_{ij}}{\partial q^k} S_m^k S_n^i u^m u^j = \frac{\partial L}{\partial q^i} S_n^i + Q_i S_n^i \]

We replace the \( Z \) variables by constants

\[ Z_{ij} \rightarrow T_{ij}, \quad \frac{\partial Z_{ij}}{\partial q^k} S_m^k \rightarrow T_{ijm} \]

\[ T_{ij} S_n^i u^j + T_{ijm} S_n^i u^m u^j = \frac{\partial L}{\partial q^i} S_n^i + Q_i S_n^i \]

This choice of substitutions aligns with Hamilton’s equations, not the reduced equations
Of course, they are not constants, so we have to add algebraic equations to our system to allow us to calculate them as we go forward

\begin{align*}
T_{ij} &= Z_{ij}(q^k), \quad T_{ijm} = \frac{\partial Z_{ij}(q^p)}{\partial q^k} S_m^k(q^q)
\end{align*}

That has been a lot to digest, and we’ll go through an example shortly

Let me try to summarize this first
Start from square one

the **simple** holonomic constraints

generalized coordinates

Lagrangian

nonholonomic and pseudononholonomic constraints

generalized forces

\[ \dot{q}^i = S^i_j u^j \]

Hamilton’s equations

introduce the Zs

reduced Hamilton’s equations with Zs
Consider the little sphere on the big sphere — both free to roll
How do we set this up?

Set the $\mathbf{K}$ axis of the big sphere horizontal as before for rolling.

I can choose the inertial axis such that the small sphere is on the $\mathbf{i}$ axis without loss of generality.

If all I care about is the ball rolling down starting from rest then the simplest thing to do is but its $\mathbf{K}$ axis horizontal as well.
It might be fun to start the ball rolling on a line of latitude

The easiest way to do that is to put the $K$ axis on a line of longitude which we can do by the proper choice of $\phi$ and $\theta$.

We want $K$ to be

$$K_2 = \begin{cases} 
-\cos \xi \cos \eta \\
-\cos \xi \sin \eta \\
\sin \xi
\end{cases}$$

Pick $\phi_2 = \eta + \frac{\pi}{2}$, $\theta_2 = \xi - \frac{\pi}{2}$
Go to Mathematica to look at this