Discrete Variational Symmetrical Top

Brian Beckman, May 2024

Abstract

In *Harmonic Oscillator via Discrete Lagrangian* (https://community.wolfram.com/groups/-/m/t/3161109) and in *Discrete Damped Harmonic Oscillator* (https://community.wolfram.com/groups/-/m/t/3174169), we numerically integrated harmonic oscillators via the discrete Lagrangian, comparing discrete numerical solutions favorably to analytical solutions.

This time, we tackle the classical symmetrical top via the same variational method on a discretized Lagrangian.

rested in these methods at large because they promise far better conservation of momentum and energy and scale to very large systems.

References

1. Marsden and West, 2001, *Discrete Mechanics and Variational Integrators*, http://www.cds.caltech.edu/~marsden/bib/2001/09-MaWe2001/MaWe2001.pdf

2. Matthew West, *Variational Integrators*, Caltech Thesis, May 2004, https://thesis.library.caltech.edu/2492/1/west_thesis.pdf

3. Marin Kobilarov and Jerrold E. Marsden, *Discrete Geometric Optimal Control on Lie Groups*, IEEE Transactions on Robotics, Vol. 27, No. 4, August 2011.

4. https://hepweb.ucsd.edu/ph110b/110b_notes/node36.html

5. https://bingweb.binghamton.edu/~suzuki/Math-Physics/LN-13S_Symmetric_top.pdf

6. William L. Burke, *Applied Differential Geometry*, Cambridge University Press; 1st edition (January 12, 2008)

Free Variables

- Let l be the height of the center of gravity (CG) of the top from the supporting surface when the top is fully vertical. The height of the CG when the top is inclined by an angle θ from vertical is $l \sin\left(\frac{\pi}{2} - \theta\right) = l \cos\theta$.

- Let m be the mass of the top.

- Let the radius of the top be r. For great symmetry and to reduce the number of free variables, let $r = l$.

- Let g be the acceleration of gravity at the Earth's surface.

- Approximate the moment of inertia (MOI) as that of a thin disk mounted at height l on a thin rod of negligible MOI. The MOI of the top in the body frame is:

In[1]:= `ClearAll[M]; (M = m l`^{2}` DiagonalMatrix[{1 / 4, 1 / 4, 1 / 2}]) // MatrixForm`

Out[1]//MatrixForm=

$$\begin{pmatrix} \frac{l^2 m}{4} & 0 & 0 \\ 0 & \frac{l^2 m}{4} & 0 \\ 0 & 0 & \frac{l^2 m}{2} \end{pmatrix}$$

Equations of Motion

In[2]:= `ClearAll[L, F, V, T, q, ω, ω2, ϕ, θ, ψ, ϕd, θd, ψd];`

Lagrangian

Though elementary and classical, this problem is not trivial. Let's start from first principles, but check against References 4 and 5.

Generalized Coordinates and Velocities

Our generalized coordinates are roll ϕ, pitch θ, and yaw ψ (mnemonics: ϕ has a longitudinal roll axis, θ has a lateral pitch axis, and ψ might remind one of y for *yaw*). The corresponding generalized velocities are $\dot\phi$, $\dot\theta$, and $\dot\psi$, which we'll write as ϕd, θd, ψd because $\dot\phi$, $\dot\theta$, and $\dot\psi$ are not valid names for variables. The standard notation $\partial L / \partial \dot\phi$ is a cruel pun (e.g., the dedication of Reference 6, "*To all those who, like me, have wondered how in hell you can change $\dot q$ without changing q*"). We can't take

the derivative of *L* with respect to the time derivative $\dot{\phi}$ of a function $\phi[t]$. $\partial L/\partial\dot{\phi}$ really means $\partial L/\partial(\phi d)$, where ϕd is an ordinary parameter variable of the Lagrangian function $L[\phi, \theta, \psi, \phi d, \theta d, \psi d]$.

Potential Energy

By inspection.

```
In[3]:=   ClearAll[V];
          V[state : {ϕ_, θ_, ψ_, ϕd_, θd_, ψd_}] := m l g Cos[θ];
          With[{state = {ϕ, θ, ψ, ϕd, θd, ψd}},
            V[state]]

Out[5]=   g l m Cos[θ]
```

Manually check units of measure. See dimensions of mass length^2/time^2 = energy.

Kinetic Energy

Check the following output cell visually against Reference 4.

```
In[6]:=   (* column vector *)
          ω[{ϕ_, θ_, ψ_, ϕd_, θd_, ψd_}] := {
              {ϕd Sin[θ] Sin[ψ] + θd Cos[ψ]},
              {ϕd Sin[θ] Cos[ψ] - θd Sin[ψ]},
              {ϕd Cos[θ] + ψd}};
          With[{o = ω[{ϕ, θ, ψ, ϕd, θd, ψd}]},
            With[{o12 = (o[[1, 1]]² + o[[2, 1]]²) // FullSimplify},
              T[state : {ϕ_, θ_, ψ_, ϕd_, θd_, ψd_}] :=
                1               1
                ─ M[[1, 1]] o12 + ─ M[[3, 3]] o[[3, 1]]²]];
                2               2
          With[{state = {ϕ, θ, ψ, ϕd, θd, ψd}},
            T[state] // FullSimplify]

Out[8]=   1
          ─ l² m (θd² + 2 (ψd + ϕd Cos[θ])² + ϕd² Sin[θ]²)
          8
```

Again see dimensions of mass length^2/time^2 = energy.

En passant, we've defined `state` as the sextuple $\{\phi, \theta, \psi, \phi d, \theta d, \psi d\}$.

Because $M[[1, 1]] = M[[2, 2]]$, it does not matter which we choose, here or anywhere else in this notebook. That's the defining feature of a *symmetrical* top.

Looking forward to the discrete Lagrangian, check this *T* against the simpler expression $\omega^{\intercal}.M.\omega$

In[9]:=
```
With[{state = {ϕ, θ, ψ, ϕd, θd, ψd}},

   (T[state] - 1/2 ω[state]ᵀ.M.ω[state]) // FullSimplify]
```

Out[9]=
```
{{0}}
```

Redefine *T* to be this simpler expression, and simultaneously fish out the singleton, scalar element of *T* via `〚1,1〛`, then define the Lagrangian:

In[10]:=
```
ClearAll[L, T];

T[state : {ϕ_, θ_, ψ_, ϕd_, θd_, ψd_}] := (1/2 ω[state]ᵀ.M.ω[state])〚1, 1〛;

L[state : {ϕ_, θ_, ψ_, ϕd_, θd_, ψd_}] := T[state] - V[state]
```

Checked

Take a look at it.

In[13]:=
```
With[{state = {ϕ, θ, ψ, ϕd, θd, ψd}},
   L[state]] // FullSimplify
```

Out[13]=

$$\frac{1}{16} \, l \, m \left(8 \, (-2 \, g + l \, \phi d \, \psi d) \, \mathrm{Cos}[\theta] + l \left(2 \, \theta d^2 + 3 \, \phi d^2 + 4 \, \psi d^2 + \phi d^2 \, \mathrm{Cos}[2 \, \theta]\right)\right)$$

Check our *L* against *L* hand-copied from Reference 4, remembering that *M* has dimensions of mass length^2.

In[14]:=
```
((1/2 M〚1, 1〛 (ϕd² Sin[θ]² + θd²) + 1/2 M〚3, 3〛 (ϕd Cos[θ] + ψd)² - m g l Cos[θ])

   - With[{state = {ϕ, θ, ψ, ϕd, θd, ψd}},

     L[state]]) // FullSimplify
```

Out[14]=

0

Good to go!

Euler-Lagrange Equations

Writing $\partial L / \partial v$ instead of the pun $\partial L / \partial \dot{q}$, expand out

$$\frac{d}{dt} \frac{\partial L}{\partial v} - \frac{\partial L}{\partial q} = 0 \tag{1}$$

as v ranges over {ϕd, θd, ψd} and q ranges over {ϕ, θ, ψ}. We'll get a column vector (as a flat `List`) of three terms for each of $\frac{d}{dt}\frac{\partial L}{\partial v}$ and $\frac{\partial L}{\partial q}$.

Here are the three expressions for $\frac{\partial L}{\partial q}$ as a column vector.

```
In[15]:=   ClearAll[dLdq];
           (dLdq = (With[{state = {φ, θ, ψ, φd, θd, ψd}},
                 Table[D[L[state], q], {q, {φ, θ, ψ}}]] // FullSimplify)) // MatrixForm
```

Out[16]//MatrixForm=

$$\begin{pmatrix} 0 \\ -\frac{1}{4} \, l \, m \, \left(-4 \, g + 2 \, l \, \phi d \, \psi d + l \, \phi d^2 \, Cos[\theta] \right) \, Sin[\theta] \\ 0 \end{pmatrix}$$

See dimensions of mass length^2/time^2 because the angular variables ϕ, θ, and ψ, are dimensionless.

$\frac{\partial L}{\partial \phi}$ and $\frac{\partial L}{\partial \psi}$ are zero. That means that the corresponding terms $\frac{d}{dt}\frac{\partial L}{\partial \dot\phi}$ and $\frac{d}{dt}\frac{\partial L}{\partial \dot\psi}$ are zero, or that $\frac{\partial L}{\partial \dot\phi}$ and $\frac{\partial L}{\partial \dot\psi}$ are constant. More about this below.

Here is $\frac{\partial L}{\partial v}$, or $\frac{\partial L}{\partial \dot q}$, with homage to the pun:

```
In[17]:=   ClearAll[dLdqDot];
           (dLdqDot = (With[{state = {φ, θ, ψ, φd, θd, ψd}},
                 Table[D[L[state], qd], {qd, {φd, θd, ψd}}]] // FullSimplify)) // MatrixForm
```

Out[18]//MatrixForm=

$$\begin{pmatrix} \frac{1}{8} \, l^2 \, m \, (4 \, \psi d \, Cos[\theta] + \phi d \, (3 + Cos[2\,\theta])) \\ \frac{1}{4} \, l^2 \, m \, \theta d \\ \frac{1}{2} \, l^2 \, m \, (\psi d + \phi d \, Cos[\theta]) \end{pmatrix}$$

Now see dimensions of mass length^2/time rather than mass length^2/time^2. We'll get the second time dimension back in the denominator when we compute the total time derivatives via d/dt.

Conjugate Momenta

For any Lagrangian, $\frac{\partial L}{\partial \dot q}$ are the **conjugate momenta** p_q. In our case, the first, p_ϕ, and the third, p_ψ, are **conserved momenta**, or **constants of the motion**, via Noether's theorem. In less fancy language, $\frac{\partial L}{\partial \phi}$ and $\frac{\partial L}{\partial \psi}$ are zero, so $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot\phi}\right) = \frac{dp_\phi}{dt} = 0$ and $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot\psi}\right) = \frac{dp_\psi}{dt} = 0$ by the Euler-Lagrange equations, so p_ϕ and p_ψ are constant.

Manually check these automatically calculated conjugate momenta against Reference 4:

In[19]:= `(dLdqDot〚1〛 - (((M〚1, 1〛 Sin[θ]² + M〚3, 3〛 Cos[θ]²) φd + M〚3, 3〛 ψd Cos[θ])) //`
`  FullSimplify`

Out[19]=

0

In[20]:= `(dLdqDot〚3〛 - (M〚3, 3〛 (ψd + φd Cos[θ])))`
Out[20]=

0

Define some variables:

In[21]:= `ClearAll[pφ, pψ];`
`pφ = dLdqDot〚1〛;`
`pψ = dLdqDot〚3〛;`

Check the reference's equations of motion, written in terms of the conjugate momenta:

In[24]:= $\left(\dfrac{p\phi - p\psi\ \mathrm{Cos}[\theta]}{M〚1, 1〛\ \mathrm{Sin}[\theta]^2} == \phi d\right)$ `// FullSimplify`

Out[24]=

True

In[25]:= $\left(\left(\dfrac{p\psi}{M〚3, 3〛} - \dfrac{(p\phi - p\psi\ \mathrm{Cos}[\theta])\ \mathrm{Cos}[\theta]}{M〚1, 1〛\ \mathrm{Sin}[\theta]^2}\right) == \psi d\right)$ `// FullSimplify`

Out[25]=

True

These are nice, first-order equations, alternatives to the second-order equations we'll find the hard way below. Reference 5 numerically solves these first-order equations. We'll make sure that our solutions of second-order equations matches their solutions of first-order equations.

Save these equations, minus the simplifications that reduce them to useless `True`:

In[26]:= `ClearAll[φdEqn, ψdEqn];`
$\phi d Eqn = \left(\dfrac{p\phi - p\psi\ \mathrm{Cos}[\theta]}{M〚1, 1〛\ \mathrm{Sin}[\theta]^2} == \phi d\right);$
$\psi d Eqn = \left(\left(\dfrac{p\psi}{M〚3, 3〛} - \dfrac{(p\phi - p\psi\ \mathrm{Cos}[\theta])\ \mathrm{Cos}[\theta]}{M〚1, 1〛\ \mathrm{Sin}[\theta]^2}\right) == \psi d\right);$

Equations of Motion, the Hard Way

We'll see an easier way when we get to checking Reference 5 below. For now, let's just follow the

Euler-Lagrange recipe blindly. The results will be good.

m and *l* are overt constants. Rewrite their total time derivatives to zero (it's interesting in some problems to let them vary with time, but not here)

In[29]:=
```
ClearAll[constRules];
constRules = {Dt[m, t] → 0, Dt[l, t] → 0}
```

Out[30]=

```
{Dt[m, t] → 0, Dt[l, t] → 0}
```

$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}}$, total time derivatives of the conjugate momenta, yielding second-order terms:

In[31]:=
```
ClearAll[ddLdqDotdt];
(ddLdqDotdt = (Dt[dLdqDot, t] /. constRules // FullSimplify)) // MatrixForm
```

Out[32]//MatrixForm=

$$\begin{pmatrix} \frac{1}{8} l^2 m \left((3 + \text{Cos}[2\,\theta]) \, \text{Dt}[\phi d, t] + 4 \, \text{Cos}[\theta] \, \text{Dt}[\psi d, t] - 4 \, (\psi d + \phi d \, \text{Cos}[\theta]) \, \text{Dt}[\theta, t] \, \text{Sin}[\theta] \right) \\ \frac{1}{4} l^2 m \, \text{Dt}[\theta d, t] \\ \frac{1}{2} l^2 m \left(\text{Cos}[\theta] \, \text{Dt}[\phi d, t] + \text{Dt}[\psi d, t] - \phi d \, \text{Dt}[\theta, t] \, \text{Sin}[\theta] \right) \end{pmatrix}$$

Finally, the equations of motion, $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0$:

In[33]:=
```
ClearAll[hardEqns];
(hardEqns = Table[(lhs == 0) // FullSimplify, {lhs, ddLdqDotdt - dLdq}]) //
  MatrixForm
```

Out[34]//MatrixForm=

$$\begin{pmatrix} l \, m \, ((3 + \text{Cos}[2\,\theta]) \, \text{Dt}[\phi d, t] + 4 \, \text{Cos}[\theta] \, \text{Dt}[\psi d, t] - 4 \, (\psi d + \phi d \, \text{Cos}[\theta]) \, \text{Dt}[\theta, t] \, \text{Sin}[\theta]) == (\\ l \, m \, (-4 \, g \, \text{Sin}[\theta] + l \, (\text{Dt}[\theta d, t] + \phi d \, (2 \, \psi d + \phi d \, \text{Cos}[\theta]) \, \text{Sin}[\theta])) == 0 \\ l \, m \, (\text{Cos}[\theta] \, \text{Dt}[\phi d, t] + \text{Dt}[\psi d, t] - \phi d \, \text{Dt}[\theta, t] \, \text{Sin}[\theta]) == 0 \end{pmatrix}$$

Here are some rewrites to set up the equations of motion for **NDSolve**:

In[35]:=
```
ClearAll[fTimeRules];
fTimeRules = {
    Dt[φd, t] → φ''[t], Dt[θd, t] → θ''[t], Dt[ψd, t] → ψ''[t],
    φd → φ'[t], θd → θ'[t], ψd → ψ'[t],
    φ → φ[t], θ → θ[t], ψ → ψ[t]};
```

It's not easy to interpret, or even believe, these equations straight-up, but the numerical solution delivers the goods.

In[37]:= `(hardEqns /. fTimeRules) // MatrixForm`

Out[37]//MatrixForm=

$$\left(\begin{array}{c} l\, m\, (-4\, \text{Sin}[\theta[t]]\, \theta'[t]\, (\text{Cos}[\theta[t]]\, \phi'[t] + \psi'[t]) + (3 + \text{Cos}[2\, \theta[t]])\, \phi''[t] + 4\, \text{Cos}[\theta[t]]\, \psi \\ l\, m\, (-4\, g\, \text{Sin}[\theta[t]] + l\, (\text{Sin}[\theta[t]]\, \phi'[t]\, (\text{Cos}[\theta[t]]\, \phi'[t] + 2\, \psi'[t]) + \theta''[t])) == (\\ l\, m\, (-\text{Sin}[\theta[t]]\, \theta'[t]\, \phi'[t] + \text{Cos}[\theta[t]]\, \phi''[t] + \psi''[t]) == 0 \end{array} \right.$$

Numerical Solution

Pick some numerical values for the free variables. The "4" denotes some rules to help checking Reference 4.

In[38]:=
```
ClearAll[numRules4];
numRules4 = {l → 1, m → 1, g → 9.81};
(hardEqns /. fTimeRules /. numRules4) // MatrixForm
```

Out[40]//MatrixForm=

$$\left(\begin{array}{c} -4\, \text{Sin}[\theta[t]]\, \theta'[t]\, (\text{Cos}[\theta[t]]\, \phi'[t] + \psi'[t]) + (3 + \text{Cos}[2\, \theta[t]])\, \phi''[t] + 4\, \text{Cos}[\theta[t]]\, \psi''[t] \\ -39.24\, \text{Sin}[\theta[t]] + \text{Sin}[\theta[t]]\, \phi'[t]\, (\text{Cos}[\theta[t]]\, \phi'[t] + 2\, \psi'[t]) + \theta''[t] == 0 \\ -\text{Sin}[\theta[t]]\, \theta'[t]\, \phi'[t] + \text{Cos}[\theta[t]]\, \phi''[t] + \psi''[t] == 0 \end{array} \right.$$

Change tN if you want a longer integration.

In[41]:=
```
ClearAll[tN];
tN = 4;
```

Mathematica has no trouble, numerically, with these equations as they stand, algebraic wrestling not needed.

```
In[43]:=  ClearAll[nsoln];
          nsoln = NDSolve[(hardEqns /. fTimeRules /. numRules4) ~ Join ~ {
              ϕ[0] == 0.1, ϕ'[0] == 10.0,
              θ[0] == 0.4, θ'[0] == 0.1,
              ψ[0] == 0.1, ψ'[0] == 0.1},
            {ϕ[t], θ[t], ψ[t]},
            {t, 0, tN}]
```

Out[44]=

$$\Big\{\Big\{ \phi[t] \to \text{InterpolatingFunction}\Big[\quad \begin{array}{l} \text{Domain:} \quad \{\{0., 4. \}\} \\ \text{Output:} \quad \text{scalar} \end{array} \Big][t],$$

$$\theta[t] \to \text{InterpolatingFunction}\Big[\quad \begin{array}{l} \text{Domain:} \quad \{\{0., 4. \}\} \\ \text{Output:} \quad \text{scalar} \end{array} \Big][t],$$

$$\psi[t] \to \text{InterpolatingFunction}\Big[\quad \begin{array}{l} \text{Domain:} \quad \{\{0., 4. \}\} \\ \text{Output:} \quad \text{scalar} \end{array} \Big][t]\Big\}\Big\}$$

Conservation of Energy

Again, the "4" denotes that we're checking against Reference 4.

```
In[45]:=  ClearAll[E4];
          E4[state : {ϕ_, θ_, ψ_, ϕd_, θd_, ψd_}] := T[state] + V[state];
```

```
In[47]:=  With[{state = {ϕ[t], θ[t], ψ[t], ϕ'[t], θ'[t], ψ'[t]}},
              E4[state]] /. numRules4 // FullSimplify
```

Out[47]=

$$9.81 \cos[\theta[t]] + 0.125\, \theta'[t]^2 +$$
$$(0.1875 + 0.0625 \cos[2\,\theta[t]])\ \phi'[t]^2 + 0.5 \cos[\theta[t]]\, \phi'[t]\, \psi'[t] + 0.25\, \psi'[t]^2$$

Some Funny Business

It's not immediately straightforward to compute derivatives of the `InterpolatingFunctions` produced by `NDSolve`. The reason is that the solutions are not functions, but expressions depending on the hard-coded parameter t, but Mathematica's `Derivative` function requires functions, and not expression. We must convert the expressions to functions and, furthermore, to avoid any use of the `Symbol` t, even if apparently safely encapsulated in `Modules` or as dummy variables in `Plot` commands. To wit, the following does not work:

In[48]:=
```
Module[{fθ},
  fθ[u_] := ((θ[t] /. nsoln) /. {t → u});
  Plot[{fθ[t], fθ'[t]}, {t, 0, tN}]]
```
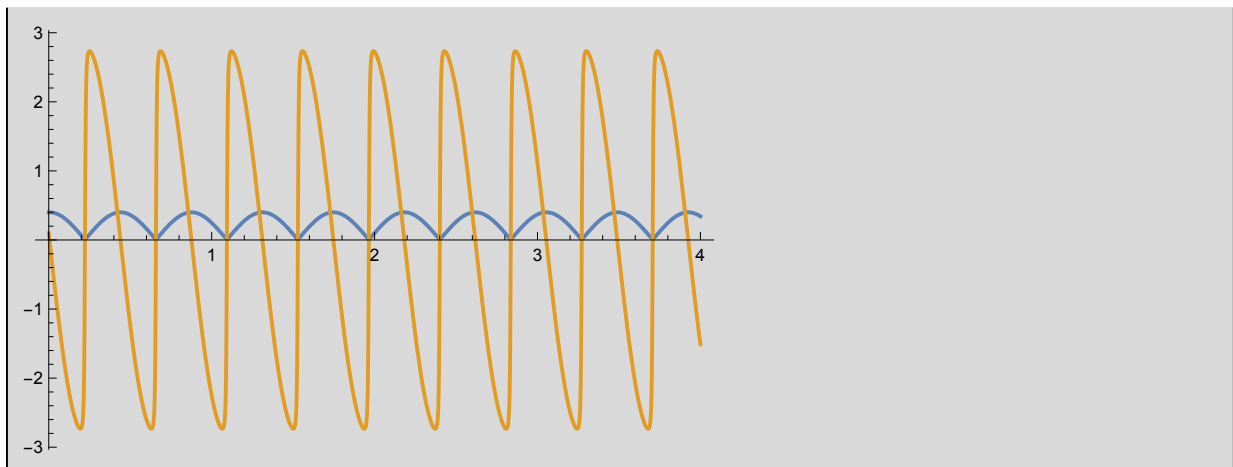
Out[48]=
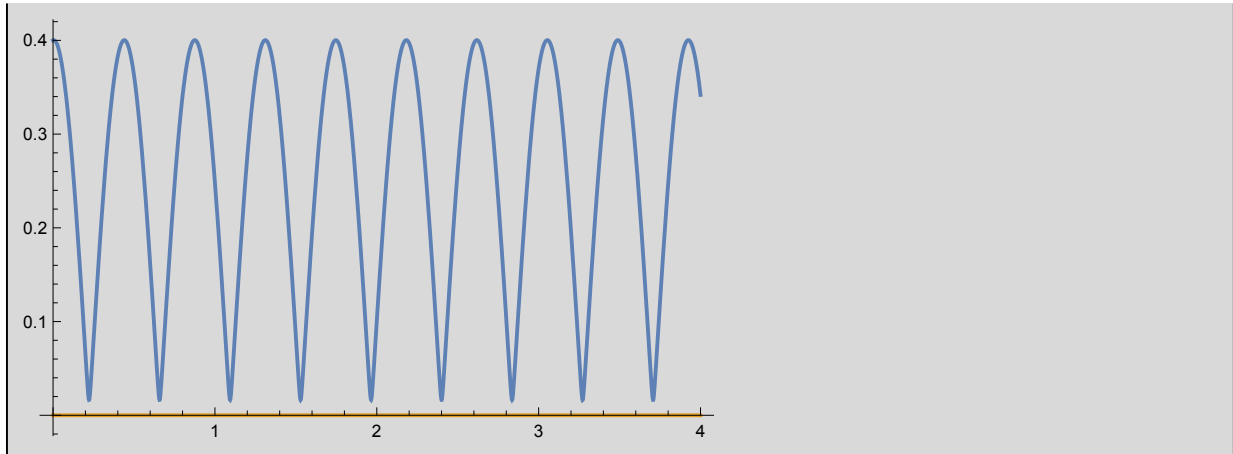
The following DOES work, so long as the dummy variable in the `Plot` command is anything *except* `t`:

In[49]:=
```
Module[{fθ},
  fθ[u_] := ((θ[t] /. nsoln[[1]]) /. {t → u});
  Plot[{fθ[u], fθ'[u]}, {u, 0, tN}]]
```

Out[49]=

We discovered these limitations by trial-and-error, but we're now equipped to compute energy from our solution `nsoln`.

Checked

Energy is conserved to six figures, at least. It is, however, trending downwards, certainly due to numeri-cal effects, so we might lose a significant amount for a longer integration. Let's see below whether the discrete method does better.

```
In[50]:=  Module[{fϕ, fθ, fψ, meanE4},
            fϕ[u_] := ((ϕ[t] /. nsoln〚1〛) /. {t → u});
            fθ[u_] := ((θ[t] /. nsoln〚1〛) /. {t → u});
            fψ[u_] := ((ψ[t] /. nsoln〚1〛) /. {t → u});
            meanE4 =
             Mean[Table[(E4[{fϕ[u], fθ[u], fψ[u], fϕ'[u], fθ'[u], fψ'[u]}] /. numRules4),
                {u, 0, tN, (tN - 0) / 100}]];
            Plot[(E4[{fϕ[u], fθ[u], fψ[u], fϕ'[u], fθ'[u], fψ'[u]}] /. numRules4) - meanE4,
             {u, 0, tN}, PlotLabel → "Residual Energy Versus Time",
             Frame → Automatic, GridLines → Automatic]]
```

Out[50]=
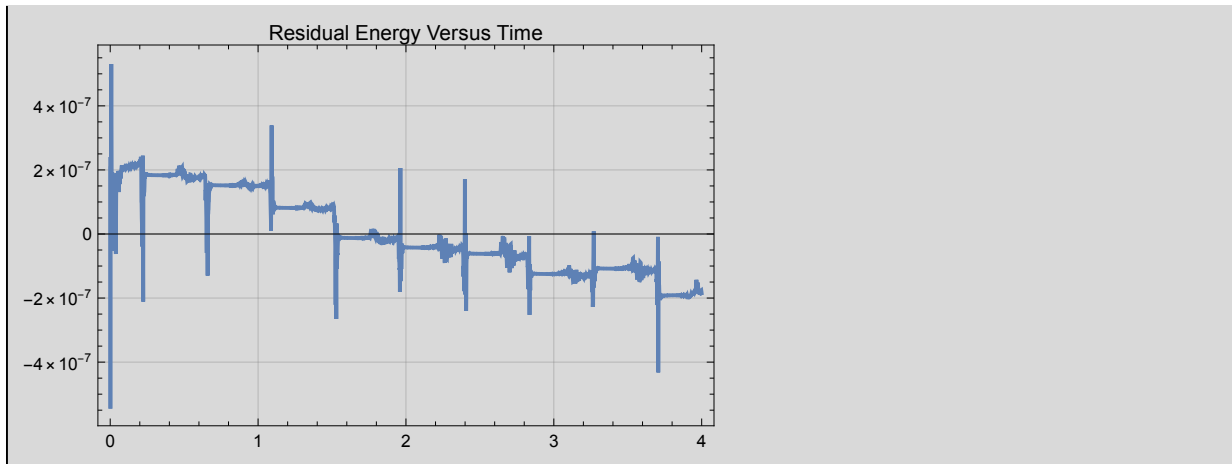
Conservation of Momenta

Likewise, we should check conservation of our momenta, azimuthal $pϕ$ and nodal $pψ$.

```
In[51]:=  {pϕ, pψ} /. numRules4
```

Out[51]=

$$\left\{ \frac{1}{8} \left(4\, \psi d\, Cos[\theta] + \phi d\, (3 + Cos[2\,\theta]) \right), \; \frac{1}{2} \left(\psi d + \phi d\, Cos[\theta] \right) \right\}$$

We see conservation within eight figures with downward trends.

In[52]:=
```
Module[{fϕ, fθ, fψ, meanPϕ, meanPψ},
  fϕ[u_] := ((ϕ[t] /. nsoln〚1〛) /. {t → u});
  fθ[u_] := ((θ[t] /. nsoln〚1〛) /. {t → u});
  fψ[u_] := ((ψ[t] /. nsoln〚1〛) /. {t → u});
  meanPϕ = Mean[Table[
      pϕ /. numRules4 /. {θ → fθ[u], ψd → fψ'[u], ϕd → fϕ'[u]}, {u, 0, tN, 100}]];
  meanPψ = Mean[Table[
      pψ /. numRules4 /. {θ → fθ[u], ψd → fψ'[u], ϕd → fϕ'[u]}, {u, 0, tN, 100}]];
  Column[{
    Plot[(pϕ /. numRules4 /. {θ → fθ[u], ψd → fψ'[u], ϕd → fϕ'[u]}) - meanPϕ,
     {u, 0, tN}, ImageSize → Medium,
     PlotLabel → "Residual Azimuthal Angular Momentum Versus Time",
     Frame → Automatic, GridLines → Automatic],
    Plot[(pψ /. numRules4 /. {θ → fθ[u], ψd → fψ'[u], ϕd → fϕ'[u]}) - meanPψ,
     {u, 0, tN}, ImageSize → Medium,
     PlotLabel → "Residual Nodal Angular Momentum Versus Time",
     Frame → Automatic, GridLines → Automatic]}]]
```

Out[52]=
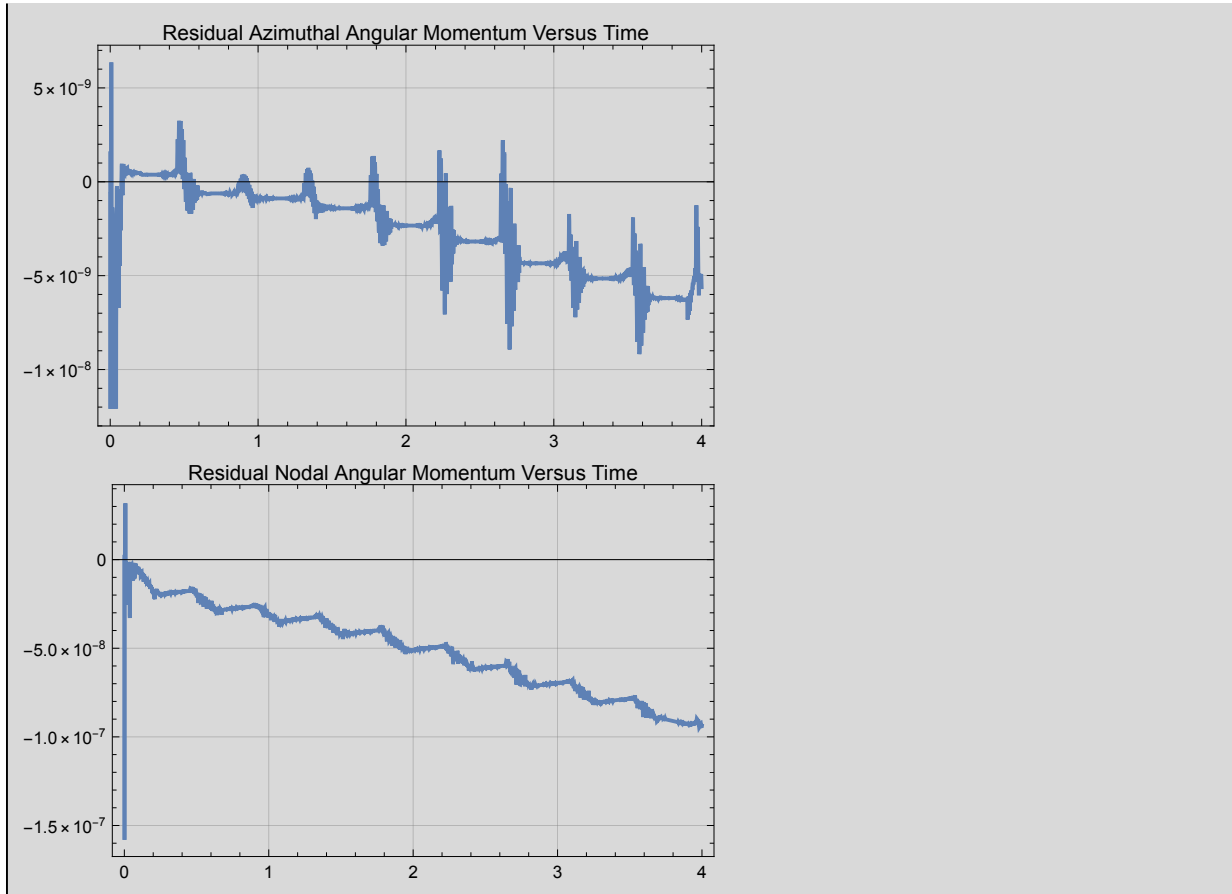
Interactive Exploration

`NDSolve` is so fast that we can put it in a `Manipulate` to play around with initial conditions and integration time.

Some graphics helpers:

```
In[53]:=   ClearAll[jack, eulerRotate];
           With[{o = {0, 0, 0},
             e1 = {1, 0, 0}, e2 = {0, 1, 0}, e3 = {0, 0, 1},
             tip = {0, 0, 1}, pr = 1.25},
            eulerRotate[obj_, ϕ_, θ_, ψ_] :=
             With[{r3 = RotationMatrix[ϕ, e3]},
              With[{e1Prime = r3.e1},
               With[{e3DoublePrime =
                   RotationMatrix[θ, e1Prime].r3.e3},
                Rotate[
                 Rotate[
                  Rotate[
                   obj,
                   ϕ, e3],
                  θ, e1Prime],
                 ψ, e3DoublePrime]]]];
            jack[opacity_ : 0.1, diameter_ : 0.01] := {
              Opacity[opacity],
              {Red, Arrow[Tube[{o, e1}, diameter]]},
              {Darker[Green], Arrow[Tube[{o, e2}, diameter]]},
              {Blue, Arrow[Tube[{o, e3}, diameter]]},
              {Lighter[Cyan], Arrow[Tube[{o, -e1}, diameter]]},
              {Magenta, Arrow[Tube[{o, -e2}, diameter]]},
              {Yellow, Arrow[Tube[{o, -e3}, diameter]]}}]
```

A rendering function:

In[55]:=
```
ClearAll[renderSoln];
renderSoln[nsoln_, tn_] :=
  With[{pr = 1.25, e1 = {1, 0, 0}, e2 = {0, 1, 0}, e3 = {0, 0, 1}, squish = 0.0001},
   With[{prl = {{-pr, pr}, {-pr, pr}, {-pr, pr}}},
    Show[
     Graphics3D[{
       jack[.25, 0.0125],
       Opacity[0.1], Sphere[],
       Opacity[0.05],
       Blue, Cylinder[squish {-e3, e3}],
       Red, Cylinder[squish {-e1, e1}],
       Green, Cylinder[squish {-e2, e2}]}],
     ParametricPlot3D[{
        Sin[θ[t]] Cos[ϕ[t]],
        Sin[θ[t]] Sin[ϕ[t]],
        Cos[θ[t]]} /. nsoln,
       {t, 0, tn}, PlotRange → prl, AspectRatio → {1, 1, 1}]]]];
```

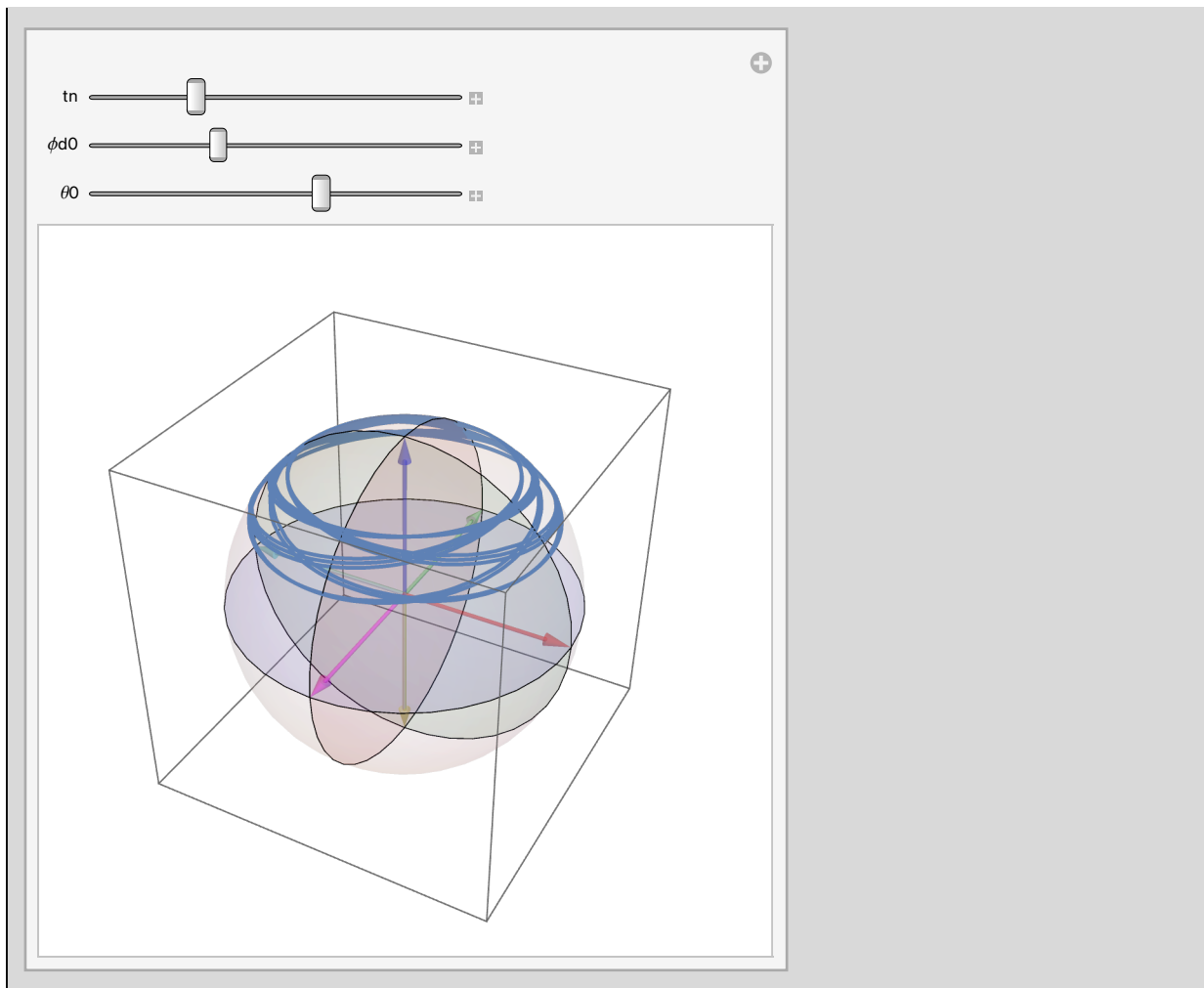A playground:

In[57]:=

```
Manipulate[
 Module[{nsoln =
    NDSolve[(hardEqns /. fTimeRules /. {l → 1, m → 1, g → 9.81}) ~ Join ~ {
        ϕ[0] == 0, ϕ'[0] == ϕd0,
        θ[0] == θ0, θ'[0] == 0,
        ψ[0] == 0, ψ'[0] == 0},
      {ϕ[t], θ[t], ψ[t]},
      {t, 0, tn}]},
  renderSoln[nsoln, tn]],
 {{tn, 8}, 0.001, 30}, {{ϕd0, 10}, 0.001, 30}, {{θ0, 1}, 0.001, π / 2 - 0.001}]
```

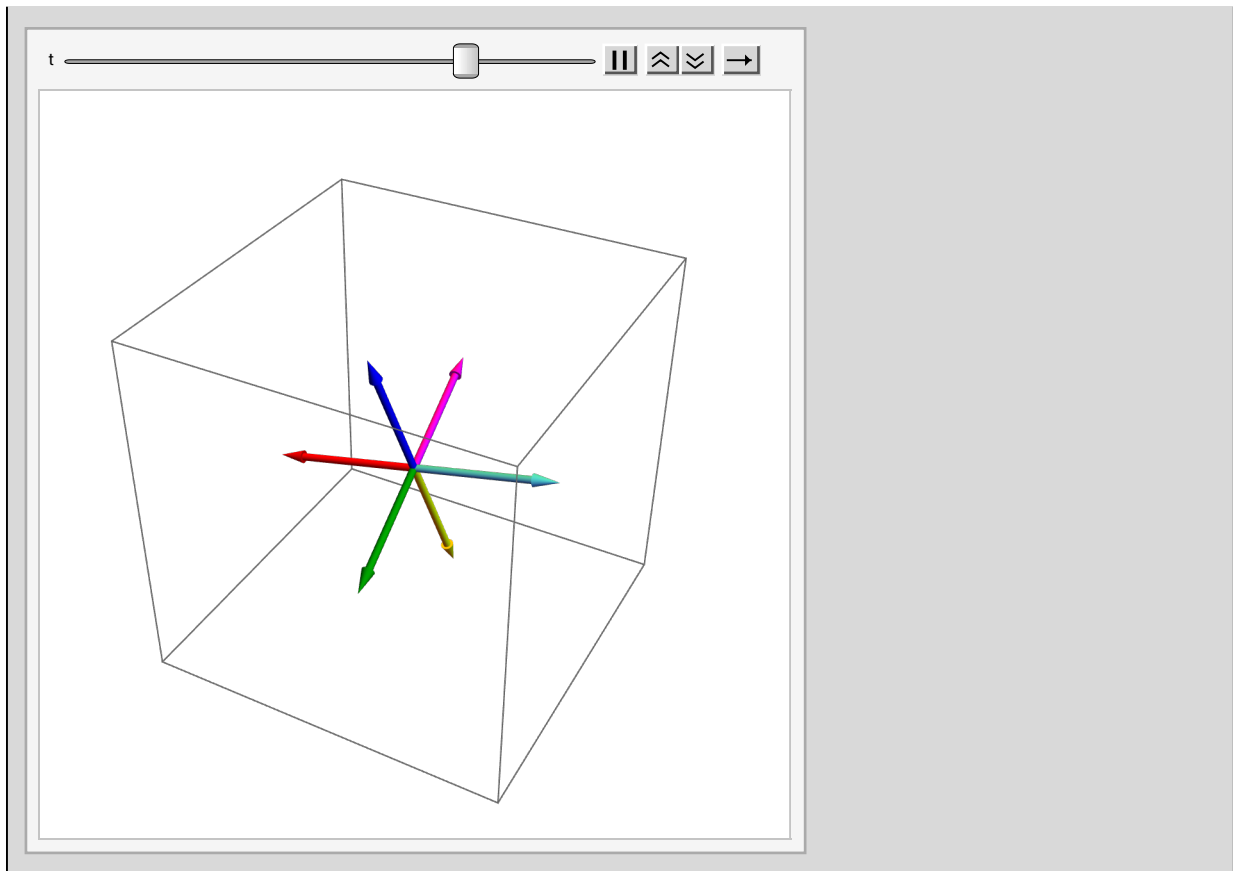Out[57]=

Another playground:

In[58]:=
```
With[{pr = 1.25},
 With[{prl = {{-pr, pr}, {-pr, pr}, {-pr, pr}}},
  With[{ϕ = nsoln〚1, 1, 2〛, θ = nsoln〚1, 2, 2〛, ψ = nsoln〚1, 3, 2〛},
   Animate[
    Evaluate@
     Graphics3D[{
       eulerRotate[
        jack[1.0, 0.025], ϕ, θ, ψ]}, PlotRange → prl],
     {t, 0, 4}]]]]
```

Out[58]=

Again via Reference 5

After writing a Lagrangian equal to ours, Reference 5 takes a different tack to the equations of motion. Their tack is optimized for calculation by hand. It's manifestly shorter, simpler, and easier to read than ours. We'll compare our results, already checked against Reference 4, to theirs.

Lagrangian

Reference 5 makes no assumptions other than lateral symmetry about the moment of inertia. Write a Lagrangian in their terms.

```
In[59]:=  ClearAll[T5, L5];
          T5[state : {φ_, θ_, ψ_, φd_, θd_, ψd_}] :=
             ⎛ 1          ⎛ I1  0   0 ⎞          ⎞
             ⎜ ─ ω[state]ᵀ.⎜ 0   I1  0 ⎟.ω[state]⎟ [[1, 1]];
             ⎝ 2          ⎝ 0   0   I3 ⎠          ⎠

          L5[state : {φ_, θ_, ψ_, φd_, θd_, ψd_}] := T5[state] - V[state]

          With[{state = {φ, θ, ψ, φd, θd, ψd}},
             L5[state]] // Expand // FullSimplify
```

Out[62]=

$$\frac{1}{4}\left(2\,I1\,\theta d^2 + I1\,\phi d^2 + I3\,\phi d^2 + 2\,I3\,\psi d^2 + \right.$$
$$\left.(-4\,g\,l\,m + 4\,I3\,\phi d\,\psi d)\,Cos[\theta] + (-I1 + I3)\,\phi d^2\,Cos[2\,\theta]\right)$$

Compare to a handwritten transcription of their 13S3:

```
In[63]:=  ⎛⎛ 1                               1                            ⎞
          ⎜⎜ ─ I1 (θd² + φd² Sin[θ]²) + ─ I3 (ψd + φd Cos[θ])² - m g l Cos[θ]⎟ -
          ⎝⎝ 2                               2                            ⎠

             With[{state = {φ, θ, ψ, φd, θd, ψd}}, L5[state]]⎞ // FullSimplify
```

Out[63]=

0

Temporary Equations of Motion

Build up the equations of motion, as before. Reference 5 doesn't ultimately use them, so this is just an exercise.

Here is $\frac{\partial L}{\partial q}$:

```
In[64]:=  ClearAll[dLdq5];
          (dLdq5 = (With[{state = {φ, θ, ψ, φd, θd, ψd}},
                Table[D[L5[state], q], {q, {φ, θ, ψ}}]] // FullSimplify)) // MatrixForm
```

Out[65]//MatrixForm=

$$\begin{pmatrix} 0 \\ \left(g\,l\,m - I3\,\phi d\,\psi d + (I1 - I3)\,\phi d^2\,Cos[\theta]\right)\,Sin[\theta] \\ 0 \end{pmatrix}$$

Here is $\frac{\partial L}{\partial \dot{q}}$:

In[66]:=
```
ClearAll[dLdqDot5];
(dLdqDot5 = (With[{state = {ϕ, θ, ψ, ϕd, θd, ψd}},
      Table[D[L5[state], qd], {qd, {ϕd, θd, ψd}}]])) // FullSimplify // MatrixForm
```

Out[67]//MatrixForm=

$$\begin{pmatrix} \text{I3 } \psi d \, \text{Cos}[\theta] + \frac{1}{2} \, \phi d \, (\text{I1} + \text{I3} + (-\text{I1} + \text{I3}) \, \text{Cos}[2\,\theta]) \\ \text{I1 } \theta d \\ \text{I3 } (\psi d + \phi d \, \text{Cos}[\theta]) \end{pmatrix}$$

Here is $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}}$:

In[68]:=
```
ClearAll[ddLdqDotdt5, constRules5];
constRules5 = constRules ~ Join ~ {Dt[I1, t] → 0, Dt[I3, t] → 0};
(ddLdqDotdt5 = (Dt[dLdqDot5, t] /. constRules5 /. fTimeRules // FullSimplify)) //
  MatrixForm
```

Out[70]//MatrixForm=

$$\begin{pmatrix} \text{Sin}[\theta[t]] \, \theta'[t] \, (2 \, (\text{I1} - \text{I3}) \, \text{Cos}[\theta[t]] \, \phi'[t] - \text{I3} \, \psi'[t]) + \frac{1}{2} \, (\text{I1} + \text{I3} + (-\text{I1} + \text{I3}) \, \text{Cos}[2\,\theta[t] \\ \text{I1 } \theta''[t] \\ \text{I3 } (-\text{Sin}[\theta[t]] \, \theta'[t] \, \phi'[t] + \text{Cos}[\theta[t]] \, \phi''[t] + \psi''[t]) \end{pmatrix}$$

Here are the equations of motion, $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = 0$:

In[71]:=
```
ClearAll[eqns5];
(eqns5 = Table[(lhs == 0), {lhs, ddLdqDotdt5 - dLdq5}]) // MatrixForm
```

Out[72]//MatrixForm=

$$\begin{pmatrix} \text{Sin}[\theta[t]] \, \theta'[t] \, (2 \, (\text{I1} - \text{I3}) \, \text{Cos}[\theta[t]] \, \phi'[t] - \text{I3} \, \psi'[t]) + \frac{1}{2} \, (\text{I1} + \text{I3} + (-\text{I1} + \text{I3}) \, \text{Cos}[2\,\theta[t] \\ - \left(\left(g \, l \, m - \text{I3} \, \phi d \, \psi d + (\text{I1} - \text{I3}) \, \phi d^2 \, \text{Cos}[\theta] \right) \, \text{Sin}[\theta] \right) + \text{I1} \, \theta''[t] \\ \text{I3 } (-\text{Sin}[\theta[t]] \, \theta'[t] \, \phi'[t] + \text{Cos}[\theta[t]] \, \phi''[t] + \psi''[t]) \, == \end{pmatrix}$$

These equations are still second-order, but Reference 5 reduces two of them to first order. Let's go!

Conjugate Momenta

pϕ5: Azimuthal Momentum

The first (azimuthal) and third (nodal) conjugate momenta should be conserved. Mathematica expresses the first one, *pϕ,* via the school half-angle formula:

In[73]:=
```
(2 Cos[θ]^2 - 1 == Cos[2 θ]) // FullSimplify
```

Out[73]=

```
True
```

Reference 5 writes a prettier, more compact version:

In[74]:=
```
ClearAll[pϕ5];
pϕ5 = I3 ψd Cos[θ] + (I1 Sin[θ]² + I3 Cos[θ]²) ϕd;
```

Mathematica easily proves the two equal:

In[76]:=
```
(dLdqDot5〚1〛 == pϕ5) // FullSimplify
```

Out[76]=

```
True
```

Reference 5 assumes $p\phi$ is constant and writes a new constant, b:

In[77]:=
```
ClearAll[b];
b = pϕ5 / I1
```

Out[78]=

$$\frac{\text{I3 } \psi\text{d Cos}[\theta] + \phi\text{d } \left(\text{I3 Cos}[\theta]^2 + \text{I1 Sin}[\theta]^2\right)}{\text{I1}}$$

pψ5: Nodal Momentum

Mathematica and Reference 5 agree completely on the expression of $p\psi$

In[79]:=
```
ClearAll[pψ5];
pψ5 = dLdqDot5〚3〛
```

Out[80]=

```
I3 (ψd + ϕd Cos[θ])
```

Reference 5 assumes $p\psi$ is constant and writes a new constant, a:

In[81]:=
```
ClearAll[a];
a = pψ5 / I1
```

Out[82]=

$$\frac{\text{I3 } (\psi\text{d} + \phi\text{d Cos}[\theta])}{\text{I1}}$$

Two First-Order Equations of Motion

Reference 5 claims that $\dot{\phi} = \frac{b - a \cos[\theta]}{\sin[\theta]^2}$. Let's check it:

In[83]:=
```
b - a Cos[θ]
──────────── // FullSimplify
   Sin[θ]²
```

Out[83]=

ϕd

Reference 5 claims that $\dot{\psi} = a\frac{I1}{I3} - \frac{(b - a\,Cos[\theta])\,Cos[\theta]}{Sin[\theta]^2}$. Let's check it:

In[84]:=
```
⎛    I1   (b - a Cos[θ]) Cos[θ]⎞
⎜a ──── - ─────────────────────⎟ // FullSimplify
⎝    I3          Sin[θ]²        ⎠
```

Out[84]=

ψd

Package these facts in two equations, getting ready for a numerical solution:

In[85]:=
```
ClearAll[foEqns, a, b];
          ⎧      b - a Cos[θ]         I1   (b - a Cos[θ]) Cos[θ]⎫
foEqns = ⎨ϕd == ──────────── , ψd == a ── - ─────────────────────⎬
          ⎩        Sin[θ]²             I3          Sin[θ]²        ⎭
```

Out[86]=

$$\left\{\phi d == (b - a\,Cos[\theta])\,Csc[\theta]^2,\ \psi d == \frac{a\,I1}{I3} - (b - a\,Cos[\theta])\,Cot[\theta]\,Csc[\theta]\right\}$$

One Second-Order Equation for Nutation

In[87]:=
```
ClearAll[soEqn];
       ⎛                        ⎛ Cos[θ] ⎞          ⎛3 + Cos[2 θ]⎞                  ⎞
soEqn = ⎜I1 Dt[θd, t] == I1 (a² + b²) ⎜────────⎟ - I1 a b ⎜────────────⎟ + m g l Sin[θ]⎟
       ⎝                        ⎝Sin[θ]³ ⎠          ⎝ 2 Sin[θ]³ ⎠                  ⎠
```

Out[88]=

$$I1\,Dt[\theta d, t] == (a^2 + b^2)\,I1\,Cot[\theta]\,Csc[\theta]^2 - \frac{1}{2}\,a\,b\,I1\,(3 + Cos[2\,\theta])\,Csc[\theta]^3 + g\,l\,m\,Sin[\theta]$$

New Constants

α and β are obviously constant. We'll numerically check the claim that E_1 is also constant.

In[89]:=
```
ClearAll[E1, α, β, newConstRules];
               ⎧                     1    ⎛       ⎛b - a Cos[ϕ]⎞²⎞
newConstRules = ⎨E1 → m g l Cos[ϕ] + ── I1 ⎜θd² + ⎜────────────⎟ ⎟,
               ⎩                     2    ⎝       ⎝  Sin[ϕ]    ⎠ ⎠

      2 E1      2 m g l⎫
   α → ────, β → ──────⎬;
       I1          I1  ⎭
```

New Nutation Equation

In[91]:=
```
ClearAll[soEqn2];
soEqn2 = (Dt[θd, t] == (a² + b²) (Cos[θ]/Sin[θ]³) - a b ((3 + Cos[2 θ])/(2 Sin[θ]³)) + β/2 Sin[θ])
```

Out[92]=
$$Dt[θd, t] == (a^2 + b^2) \, Cot[θ] \, Csc[θ]^2 - \frac{1}{2} \, a \, b \, (3 + Cos[2θ]) \, Csc[θ]^3 + \frac{1}{2} \, β \, Sin[θ]$$

It's easy to prove the two equations for $\ddot{θ}$ are equivalent.

In[93]:=
```
(soEqn[[2]] - I1 soEqn2[[2]]) /. newConstRules // FullSimplify
```

Out[93]=

0

Numerical Solution

In[94]:=
```
ClearAll[α, β, a, b];
numRules5 = {α → 1.6, β → 2.0, a → 2.5, b → 1.7, I1 → 2, I3 → 1};
```

In[96]:=
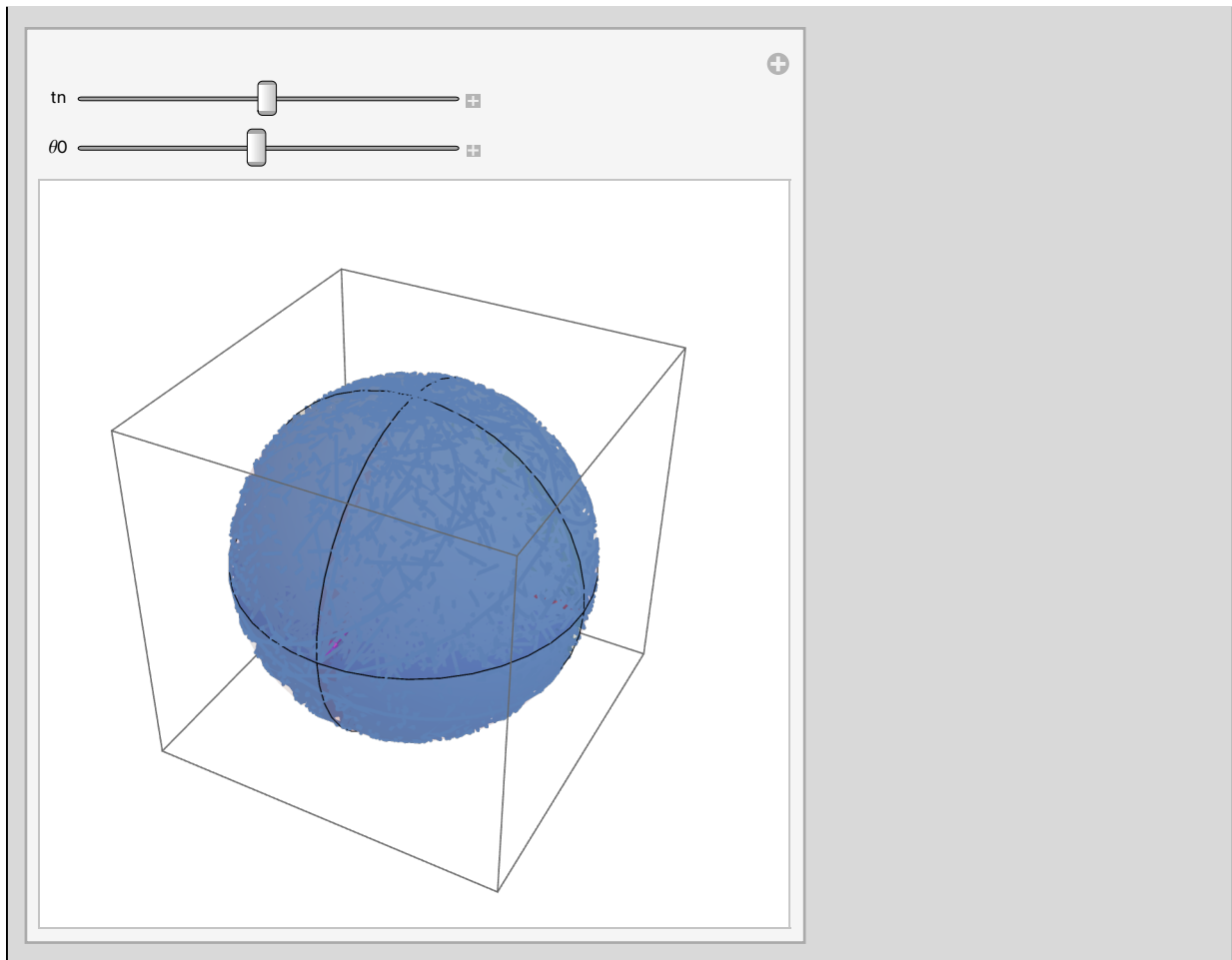```
(foEqns ~ Join ~ {soEqn2}) /. numRules5
```

Out[96]=
$$\{φd == (1.7 - 2.5 \, Cos[θ]) \, Csc[θ]^2, \, ψd == 5. - (1.7 - 2.5 \, Cos[θ]) \, Cot[θ] \, Csc[θ],$$
$$Dt[θd, t] == 9.14 \, Cot[θ] \, Csc[θ]^2 - 2.125 \, (3 + Cos[2θ]) \, Csc[θ]^3 + 1. \, Sin[θ]\}$$

In[97]:=
```
ClearAll[nsoln5];
Manipulate[
 Module[{nsoln5 =
    Quiet@NDSolve[((foEqns~Join~{soEqn2}) /. numRules5 /. fTimeRules)~Join~
      {φ[0] == 0, θ[0] == θ0*°, θ'[0] == 0, ψ[0] == 0},
     {φ[t], θ[t], ψ[t]},
     {t, 0, tN}]},
  renderSoln[nsoln5, tn]],
 {{tn, 70}, 0.001, 140}, {{θ0, 42}, 0, 90}]
```

Out[98]=



First, we reproduce the beautiful numerical solutions in Reference 5 verbatim, then we show them equivalent to our reductions over their developments above.

```
In[99]:= ClearAll[Firsteq, Secondeq, Thirdeq, def1, Initial, eq1];
          Firsteq = θ''[t] ==
                (a² + b²) Cot[θ[t]] Csc[θ[t]]² - 1/2 a b (3 + Cos[2 θ[t]]) Csc[θ[t]]³ + β/2 Sin[θ[t]];
          Secondeq = ϕ'[t] == Csc[θ[t]] (-a Cot[θ[t]] + b Csc[θ[t]]);
          Thirdeq = ψ'[t] == a I1/I3 + a Cot[θ[t]]² - b Cot[θ[t]] Csc[θ[t]];
          Initial = {θ'[0] == 0, ϕ[0] == 0 °, ψ[0] == 0 °};
          def1 = {Firsteq, Secondeq, Thirdeq} /. numRules5;
          Manipulate[
           Module[{
             eq1 = NDSolve[def1 ~ Join ~ Initial ~ Join ~ {θ[0] == θ0 * °},
               {ϕ[t], θ[t], ψ[t]}, {t, 0, tn}]},
            renderSoln[eq1, tn]],
           {{θ0, 43}, 0, 90}, {{tn, 70}, 0.0001, 1470}]
```

Out[105]=
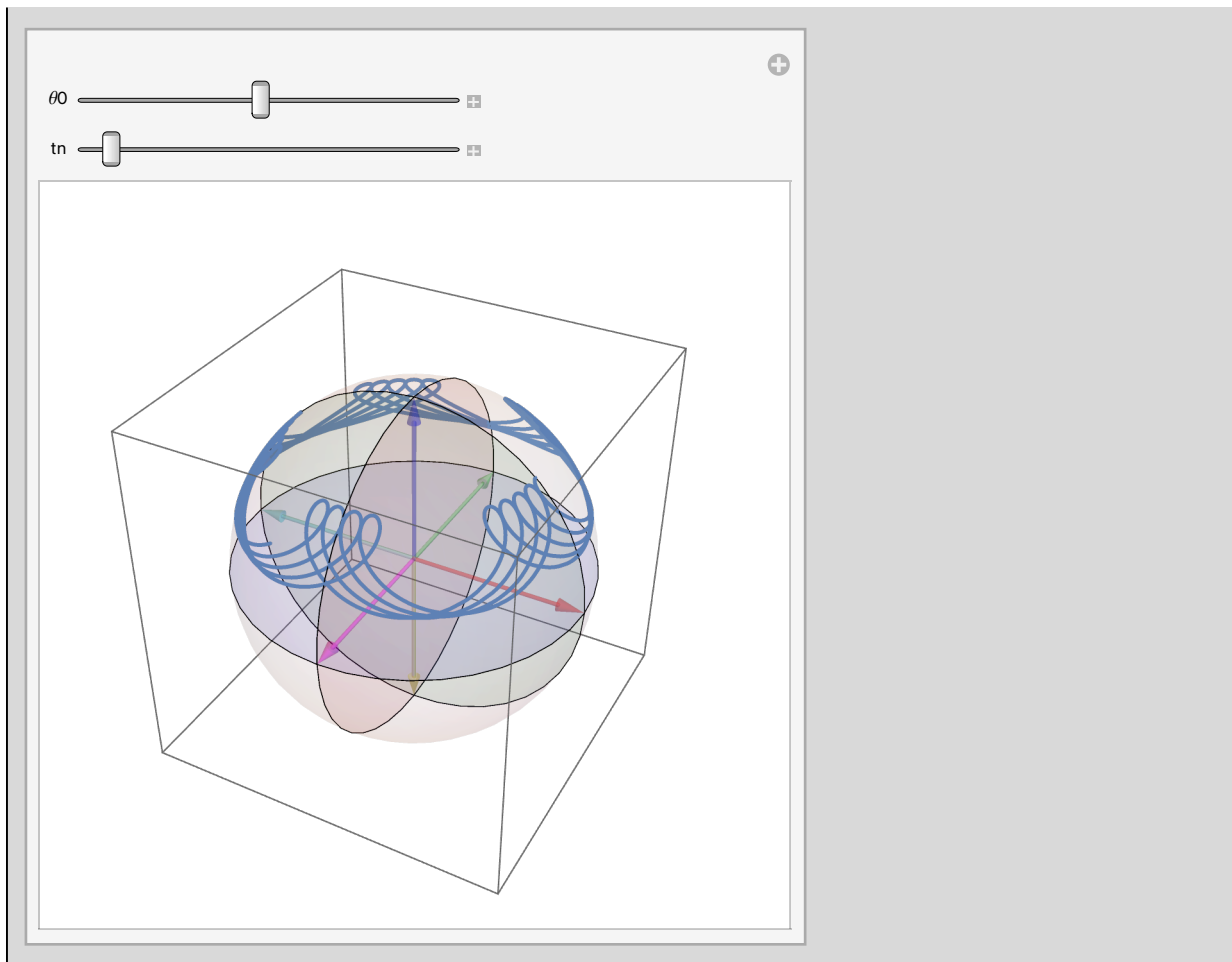
In[106]:=

```
(soEqn2 /. numRules5 /. fTimeRules) - (Firsteq /. numRules5)
```

Out[106]=

0

In[107]:=

```
(foEqns[[1]] /. fTimeRules) -
 Secondeq
```

Out[107]=

$\left(\phi'[t] == (b - a \, Cos[\theta[t]]) \, Csc[\theta[t]]^2\right) -$
$(\phi'[t] == Csc[\theta[t]] \, (-a \, Cot[\theta[t]] + b \, Csc[\theta[t]]))$

In[108]:=

In[109]:=

```
(foEqns[[1]] /. numRules5 /. fTimeRules) - (Secondeq /. numRules5) // FullSimplify
```

Out[109]=

$\left(\phi'[t] == (1.7 - 2.5 \, Cos[\theta[t]]) \, Csc[\theta[t]]^2\right) -$
$(\phi'[t] == Csc[\theta[t]] \, (-2.5 \, Cot[\theta[t]] + 1.7 \, Csc[\theta[t]]))$

In[110]:=

```
ClearAll[def2];
def2 = (foEqns ~ Join ~ {soEqn2}) /. numRules5 /. fTimeRules // RotateRight
```

Out[111]=

$\{\theta''[t] == 9.14 \, Cot[\theta[t]] \, Csc[\theta[t]]^2 - 2.125 \, (3 + Cos[2 \, \theta[t]]) \, Csc[\theta[t]]^3 + 1. \, Sin[\theta[t]],$
$\phi'[t] == (1.7 - 2.5 \, Cos[\theta[t]]) \, Csc[\theta[t]]^2,$
$\psi'[t] == 5. - (1.7 - 2.5 \, Cos[\theta[t]]) \, Cot[\theta[t]] \, Csc[\theta[t]]\}$

In[112]:=

```
(#[[2]] & /@ def1 - #[[2]] & /@ def2) // FullSimplify
```

Out[112]=

$\{\{0., Cot[\theta[t]] \, (-2.5 - 9.14 \, Csc[\theta[t]]) \, Csc[\theta[t]] +$
$\quad Csc[\theta[t]]^2 \, (1.7 + (6.375 + 2.125 \, Cos[2 \, \theta[t]]) \, Csc[\theta[t]]) - 1. \, Sin[\theta[t]],$
$\quad 5. + 2.5 \, Cot[\theta[t]]^2 + Cot[\theta[t]] \, (-1.7 - 9.14 \, Csc[\theta[t]]) \, Csc[\theta[t]] +$
$\quad (6.375 + 2.125 \, Cos[2 \, \theta[t]]) \, Csc[\theta[t]]^3 - 1. \, Sin[\theta[t]]\},$
$\{Cot[\theta[t]] \, Csc[\theta[t]] \, (2.5 + 9.14 \, Csc[\theta[t]]) +$
$\quad Csc[\theta[t]]^2 \, (-1.7 + (-6.375 - 2.125 \, Cos[2 \, \theta[t]]) \, Csc[\theta[t]]) + 1. \, Sin[\theta[t]],$
$\quad 0., 5. + 2.5 \, Cot[\theta[t]]^2 + 0.8 \, Cot[\theta[t]] \, Csc[\theta[t]] - 1.7 \, Csc[\theta[t]]^2\},$
$\{-5. - 2.5 \, Cot[\theta[t]]^2 + (-6.375 - 2.125 \, Cos[2 \, \theta[t]]) \, Csc[\theta[t]]^3 +$
$\quad Cot[\theta[t]] \, Csc[\theta[t]] \, (1.7 + 9.14 \, Csc[\theta[t]]) + 1. \, Sin[\theta[t]],$
$\quad -5. - 2.5 \, Cot[\theta[t]]^2 - 0.8 \, Cot[\theta[t]] \, Csc[\theta[t]] + 1.7 \, Csc[\theta[t]]^2, 0.\}\}$

Energy E5

E is a reserved symbol in Mathematica, so we called our our old energy E4, with homage to Reference 4, and the new energy E5, with homage to Reference 5.

The nutation equation, for $\theta[t]$, is the only second-order equation we'll need to deal with after incorporating the conserved momenta later. Check ours against a hand-written copy from Reference 5, Section 13S3.

Mathematica has a harmless minus sign

In[113]:=

```
-eqns5〚2, 1〛 // Expand // FullSimplify
```

Out[113]=

$$\left(g\, l\, m - I3\, \phi d\, \psi d + (I1 - I3)\, \phi d^2\, Cos[\theta] \right) Sin[\theta] - I1\, \theta''[t]$$

Here is a handwritten copy from Reference 5; by inspection, it perfectly matches ours:

In[114]:=

```
Sin[θ[t]] (m g l + (I1 - I3) ϕ'[t]² Cos[θ[t]]) -
    I3 ϕ'[t] × ψ'[t] Sin[θ[t]] - I1 θ''[t] // Expand // FullSimplify
```

Out[114]=

$$g\, l\, m\, Sin[\theta[t]] + (I1 - I3)\, Cos[\theta[t]]\, Sin[\theta[t]]\, \phi'[t]^2 -$$
$$I3\, Sin[\theta[t]]\, \phi'[t]\, \psi'[t] - I1\, \theta''[t]$$

Mathematica is not able to cancel out the terms, but it is of no consequence. The following has a leading factor of $\sin\theta - \sin\theta$, obviously zero:

In[115]:=

```
(-eqns5〚2, 1〛 - (Sin[θ[t]] (m g l + (I1 - I3) ϕ'[t]² Cos[θ[t]]) -
    I3 ϕ'[t] × ψ'[t] Sin[θ[t]] - I1 θ''[t])) // FullSimplify
```

Out[115]=

$$\left(g\, l\, m - I3\, \phi d\, \psi d + (I1 - I3)\, \phi d^2\, Cos[\theta] \right) Sin[\theta] -$$
$$Sin[\theta[t]] \left(g\, l\, m + (I1 - I3)\, Cos[\theta[t]]\, \phi'[t]^2 \right) + I3\, Sin[\theta[t]]\, \phi'[t]\, \psi'[t]$$

Junkyard

Ignore stuff in here.

Damped Solution

Mathematica easily solves the equation and we pick the only pertinent part of the expression returned by `DSolve`:

In[116]:=

```
dsoln = (DSolve[deqn, x, t])〚1, 1, 2〛
```

⋯ DSolve : Equation or list of equations expected instead of deqn in the first argument deqn.   ⓘ

⋯ Part : Part specification DSolve [deqn, x, t]〚1, 1, 2 〛 is longer than depth of object.   ⓘ

Out[116]=

```
DSolve[deqn, x, t]〚1, 1, 2〛
```

For plotting, we must be a bit careful, because if b^2 is smaller than $4\,m\,\kappa$, we get complex arguments to the exponentials, yielding oscillation, but that's the interesting part! Only the real parts of the solution are physical.

Convince yourself that as b^2 varies from 0 to $4\,m\,\kappa$, the system transitions from undamped, to under-damped (showing some oscillation), to critically damped, to overdamped (showing no oscillation). See this page on damping.

In[117]:=

```
Manipulate[Plot[Re[dsoln[t] /. {κ → 1, m → 1, b → bManip, C[2] → 1, C[1] → 0}],
   {t, 0, 25}, GridLines → Automatic],
  {bManip, 0, 4, Appearance → "Open"}]
```

Out[117]=
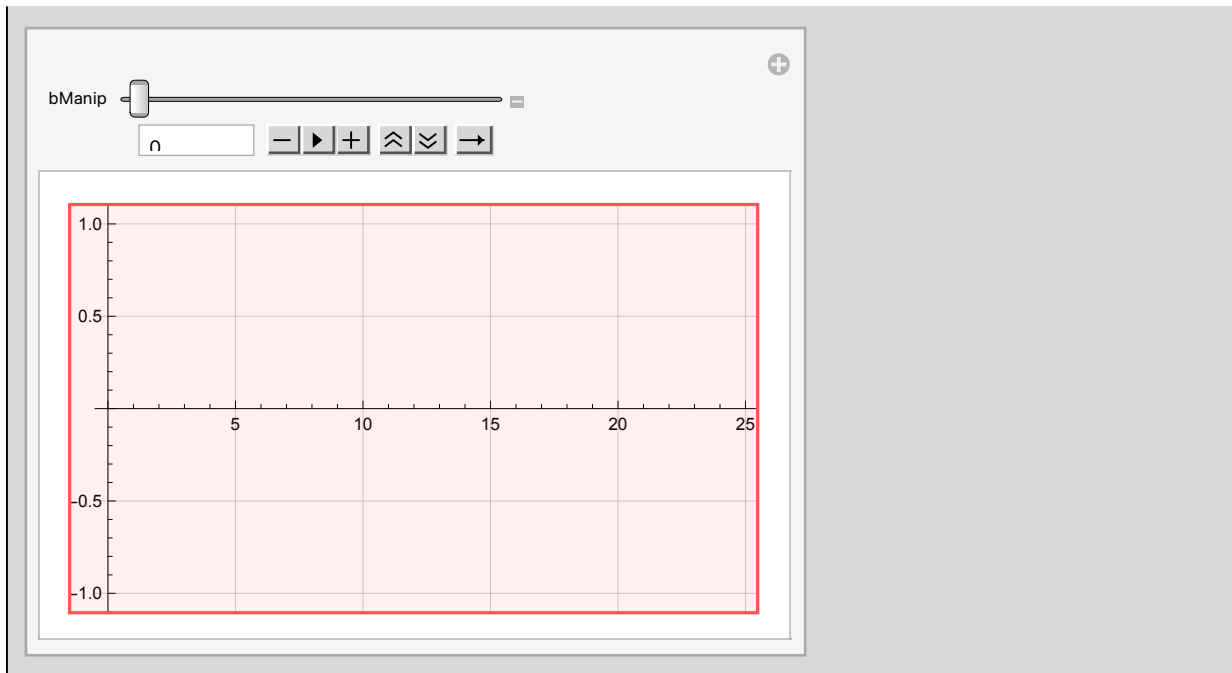
Discrete Solution

Reference 2 gives us the **forced discrete Euler-Lagrange Equations**:

where L_d is the discrete Lagrangian and F_d^- and F_d^+ are the left and right discrete forces. These should approximate the continuous forcing so that

$$F_d^-(q_k, q_{k+1}) \cdot \delta q_k + F_d^+(q_k, q_{k+1}) \cdot \delta q_{k+1} \approx \int_{t_k}^{t_{k+1}} F(q(t), \dot{q}(t)) \cdot \delta q \ dt.$$

The equation (1.27) defines an integrator $(q_k, q_{k+1}) \mapsto (q_{k+1}, q_{k+2})$ given implicitly by the **forced discrete Euler-Lagrange equations**:

$$D_1 L_d(q_{k+1}, q_{k+2}) + D_2 L_d(q_k, q_{k+1}) + F_d^-(q_{k+1}, q_{k+2}) + F_d^+(q_k, q_{k+1}) = 0. \qquad (1.28)$$

The simplest example of discrete forces is to take

$$F_d^-(q_k, q_{k+1}) = F(q_k)$$
$$F_d^+(q_k, q_{k+1}) = 0,$$

which, together with the discrete Lagrangian (1.3), gives the forced Euler-Lagrange equations

$$M \left(\frac{q_{k+1} - 2q_k + q_{k-1}}{h^2} \right) = -\nabla V(q_k) + F(q_k).$$

The position-momentum form of a variational integrator with forcing is useful for implementation purposes. This is given by

$$p_k = -D_1 L_d(q_k, q_{k+1}) - F_d^-(q_k, q_{k+1})$$
$$p_{k+1} = D_2 L_d(q_k, q_{k+1}) + F_d^+(q_k, q_{k+1}).$$

References 2 and 3 are very difficult and long. The derivations dive very deeply into differential geometry, topology, and more. No shallow reading can give us a recipe for the force terms F_d^- and F_d^+. However, from our success with the undamped simple harmonic oscillator, we can hope for an easy extension.

Start with quadrature of the unforced discrete Lagrangian according to the third, unnumbered equation on page 643 of Reference 3, and introducing another free variable, the time step h:

> *Simple Example:* Consider a continuous, typical Lagrangian of the form $L(q, \dot{q}) = \frac{1}{2}\dot{q}^T M \dot{q} - V(q)$ (V being a potential function) and define the discrete Lagrangian $L_d(q_k, q_{k+1}) = hL(q_{k+\frac{1}{2}}, (q_{k+1} - q_k)/h)$ by the use of the notation $q_{k+\frac{1}{2}} := (q_k + q_{k+1})/2$. The resulting update equation is
>
> $$M\frac{q_{k+1} - 2q_k + q_{k-1}}{h^2} = -\frac{1}{2}(\nabla V(q_{k-\frac{1}{2}}) + \nabla V(q_{k+\frac{1}{2}}))$$

Guess that

In[118]:=

```
ClearAll[FMinus, FPlus];

FMinus[q_] := D[F[q, v, t], v] /. {v → (q[k + 1] - q[k])/h};

FPlus[q_] := D[F[q, v, t], v] /. {v → (q[k] - q[k - 1])/h}
```

and write

In[121]:=

```
discreteEqn =
  m ((q[k + 1] - 2 q[k] + q[k - 1])/h^2) == -1/2 ((D[V[q], q] /. {q → (q[k - 1] + q[k])/2}) +
    (D[V[q], q] /. {q → (q[k] + q[k + 1])/2}) + FMinus[q] + FPlus[q])
```

Out[121]=

$$\frac{m (q[-1 + k] - 2 q[k] + q[1 + k])}{h^2} == \frac{1}{2} \left(-V'\left[\frac{1}{2} (q[-1 + k] + q[k])\right] - V'\left[\frac{1}{2} (q[k] + q[1 + k])\right] - \right.$$
$$\left. F^{(0,1,0)}\left[q, \frac{-q[-1 + k] + q[k]}{h}, t\right] - F^{(0,1,0)}\left[q, \frac{-q[k] + q[1 + k]}{h}, t\right]\right)$$

Algebraic solution of the discrete equation for time step $k + 1$ given solutions for time steps $k - 1$ and k. Do not confuse time step k with spring constant κ. As with most quadratics, we get two solutions.

In[122]:=

```
discreteSolns = (Solve[discreteEqn, q[1 + k]] // FullSimplify)
```

⦿ Solve : This system cannot be solved with the methods available to Solve. Try Reduce or FindInstance instead.

Out[122]=

$$\text{Solve}\Big[\frac{2\,m\,(q[-1+k] - 2\,q[k] + q[1+k])}{h} + h\,\Big(V'\Big[\frac{1}{2}\,(q[-1+k] + q[k])\Big] + V'\Big[\frac{1}{2}\,(q[k] + q[1+k])\Big] + F^{(0,1,0)}\Big[q, \frac{-q[-1+k] + q[k]}{h}, t\Big] + F^{(0,1,0)}\Big[q, \frac{-q[k] + q[1+k]}{h}, t\Big]\Big) == 0, q[1+k]\Big]$$

Pick the first one to start:

In[123]:=

```
discreteSoln1 = discreteSolns〚1, 1, 2〛 /. {κ → 1, m → 1}
```

Out[123]=

$$h\,\Big(V'\Big[\frac{1}{2}\,(q[-1+k] + q[k])\Big] + V'\Big[\frac{1}{2}\,(q[k] + q[1+k])\Big] + F^{(0,1,0)}\Big[q, \frac{-q[-1+k] + q[k]}{h}, t\Big] + F^{(0,1,0)}\Big[q, \frac{-q[k] + q[1+k]}{h}, t\Big]\Big)$$

Compare the plot of the analytical solution versus the first discrete solution for various choices of the time step, represented by its negative logarithm base 10, and for the damping parameter *b*.

In[124]:=

```
ClearAll[experiment];
experiment[bManip_, timeSteps_, h_] :=
  Module[{k, q, x},
    q[k_] := x[[1 + k]]; (* Let us start at k == 0 *)
    x = ConstantArray[0, timeSteps + 1];
    (* bootstrap discrete solution from analytical solution *)
    x[[1]] = Re[dsoln[0] /. {b → bManip, κ → 1, m → 1, C[2] → 1, C[1] → 0}];
    x[[2]] = Re[dsoln[h] /. {b → bManip, κ → 1, m → 1, C[2] → 1, C[1] → 0}];
    For[k = 1, k < timeSteps, k++,
      (* Must manually paste the solution -- reason unknown *)
      x[[k + 2]] = ((-4 + 2 b h - h^2) q[-1 + k] + 2 (4 - h^2) q[k]) / (4 + 2 b h + h^2) /. b → bManip];
    GraphicsRow[{
      ListLinePlot[x, Frame → True,
        FrameLabel → {{"Amplitude", ""}, {"Time Step -- units of h",
          "Discrete Solution"}}, GridLines → Automatic],
      Plot[Re[dsoln[τ] /. {b → bManip, κ → 1, m → 1, C[2] → 1, C[1] → 0}],
        {τ, 0, timeSteps h}, Frame → True,
        GridLines → Automatic, FrameLabel → {{"Amplitude", ""},
          {"Time Step -- units of 1/h", "Analytical Solution"}}}]}]];
Manipulate[experiment[bManip, 25 / 10^minuslog10h, 10.^minuslog10h],
  {bManip, 0, 4, Appearance → "Open"},
  {minuslog10h, {0, -1, -2, -3, -4}}]
```

Out[126]=

Total success!