Other information:

Center for Applied Math
3rd year PhD student
Tel. no.: 847-644-2031
Email: iml32@cornell.edu

Contents

1 Four-Bar Linkage
  1.1 Specification of coordinates and initial conditions ................. 2
  1.2 Degrees of freedom and system dimensionality .......................... 3
  1.3 Equations of motion ..................................................... 4
  1.4 Numerical integration of DAEs ........................................... 8
  1.5 Checking the integration: constraints .................................... 9
  1.6 Numerical results .......................................................... 10
  1.7 A note on animation ....................................................... 17

2 The Triple Pendulum
  2.1 Specification of coordinates and initial conditions ..................... 18
  2.2 Derivation 1: the Lagrangian ............................................. 19
  2.3 Derivation 2: the Newtonian approach .................................... 20
  2.4 Numerical results .......................................................... 21

3 Appendix
  3.1 Code for the four-bar linkage ............................................. 28
  3.2 Code for the triple pendulum ............................................. 35
1 Four-Bar Linkage

1.1 Specification of coordinates and initial conditions

We set up the four-bar linkage as follows:

By convention:
1. Set bar AB to be the fixed horizontal bar.

2. Specify initial configuration via \((l_0, p_{1x}, p_{1y}, p_{2x}, p_{2y})\). Lengths and angles \(l_i, \theta_i\) are calculated therefrom.

3. Set initial \(\dot{\theta}_i = 0\) for \(i = 1, 2, 3\). So we specify the initial structure and let it drop under its weight.

4. Initial parameters:
   (a) masses \(m_i\)
   (b) moments of inertia \(I_i^G\)
   (c) gravitational constant \(g\)

1.2 Degrees of freedom and system dimensionality

The four-bar linkage may be specified by the dynamics of three free-floating bars each with coordinates \((x_i, y_i, \theta_i)\) (having nine degrees of freedom altogether), together with a set of eight constraints specified as follows:

1. Point \(A\) fixed (2)
2. Point \(B\) fixed (2)
3. Acceleration of bar AD at point \(D\) is equal to acceleration of bar CD and point \(D\) (2)
4. Acceleration of bar CD at point \(C\) is equal to acceleration of bar BC and point \(C\) (2)

Therefore the constrained system has one degree of freedom, while integration of the numerical system requires solving a set of \(9 + 8 = 17\) DAEs.
1.3 Equations of motion

Before writing down the equations of motion, we fix notation for the tension forces at the end of each bar.

1. Bar AD: Tension at point $A$ is $(A_x, A_y)$, tension at point $B$ is $(B_x, B_y)$

2. Bar BC: Tension at point $B$ is $(B_x, B_y)$, tension at point $C$ is $(C_x, C_y)$

3. Bar CD: By Newton’s third law, tension at point $C$ is $-(C_x, C_y)$, tension at point $D$ is $-(D_x, D_y)$.

Applying linear momentum balance (LMB) to each of the three moving bars and then dotting by the basis vectors $\hat{i}$ and $\hat{j}$, we will obtain 6 equations.

Applying angular momentum balance (AMB) about the centers of gravity of each of the moving bars, we obtain 3 equations.

Specifying the constraint equations as discussed in Sec. 1.2, we obtain 8 equations.

Thus, we find $6 + 3 + 8 = 17$ equations of motion. These equations are linear in the following seventeen variables (stored in the vector $\xi$):

$$\xi = (\ddot{x}_1, \ddot{y}_1, \ddot{x}_2, \ddot{y}_2, \ddot{x}_3, \ddot{y}_3, \ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3, A_x, A_y, B_x, B_y, C_x, C_y, D_x, D_y).$$

The free-body diagrams of each bar are shown:
We now write down the equations explicitly.

**Linear momentum balance:**

Bar AD: we have

\[ m_1\ddot{x}_1 = A_x + D_x \quad (1\text{-LMB1}) \]

and

\[ m_1\ddot{y}_1 = A_y + D_y - m_1g. \quad (2\text{-LMB2}) \]

Bar BC: we have

\[ m_2\ddot{x}_2 = B_x + C_x \quad (3\text{-LMB3}) \]
and

\[ m_2\ddot{y}_2 = B_y + C_y - m_2g. \]  

(4-LMB4)

Bar CD: we have

\[ m_3\ddot{x}_3 = -C_x - D_x \]  

(5-LMB5)

and

\[ m_3\ddot{y}_3 = -C_y - D_y - m_3g. \]  

(6-LMB6)

**Angular momentum balance:**

We consider \( \dot{\mathbf{H}}/Q = \mathbf{M}/Q \) for \( Q \) the center of mass in each case (suppressing indices). In each case, \( \mathbf{r}_{G/Q} = 0 \), so the only term surviving on the lhs is \( I^G_i \ddot{\theta}_i \).

Bar AD: We have \( \mathbf{r}_{D/Q} = d_1(\sin \theta_1 \hat{i} - \cos \theta_1 \hat{j}) \) and \( \mathbf{r}_{A/Q} = -\mathbf{r}_{D/Q} \). The \( \hat{k} \)-th component of the AMB equation becomes

\[ I^G_1 \ddot{\theta}_1 = d_1(-\sin \theta_1 A_x - \cos \theta_1 A_y + \sin \theta_1 D_x + \cos \theta_1 D_y). \]  

(7-AMB1)

Bar BC: We have \( \mathbf{r}_{C/Q} = d_2(\sin \theta_2 \hat{i} - \cos \theta_2 \hat{j}) \) and \( \mathbf{r}_{B/Q} = -\mathbf{r}_{C/Q} \). The \( \hat{k} \)-th component of the AMB equation becomes

\[ I^G_2 \ddot{\theta}_2 = d_2(-\sin \theta_2 B_x - \cos \theta_2 B_y + \sin \theta_2 C_x + \cos \theta_2 C_y). \]  

(8-AMB2)

Bar CD: We have \( \mathbf{r}_{C/Q} = d_3(\cos \theta_3 \hat{i} + \sin \theta_3 \hat{j}) \) and \( \mathbf{r}_{D/Q} = -\mathbf{r}_{C/Q} \). The \( \hat{k} \)-th component of the AMB equation becomes

\[ I^G_3 \ddot{\theta}_3 = d_3(\sin \theta_3 C_x - \cos \theta_3 C_y - \sin \theta_3 D_x + \cos \theta_3 D_y). \]  

(9-AMB3)

**Constraint equations:**

Point \( A \) fixed:

Point \( A = (x_A, y_A) \) may be written in terms of \((x_1, y_1)\) as follows:
\[
\begin{pmatrix}
  x_A \\
y_A
\end{pmatrix} = \begin{pmatrix} x_1 - d_1 \sin \theta_1 \\
y_1 + d_1 \cos \theta_1 \end{pmatrix}.
\]

Differentiate this expression twice and take \((\ddot{x}_A, \ddot{y}_A) = (0, 0)\) to obtain

\[
\ddot{x}_1 - d_1 \ddot{\theta}_1 \cos \theta_1 = -d_1 \dot{\theta}_1^2 \sin \theta_1 \tag{10-CON1}
\]

and

\[
\ddot{y}_1 - d_1 \ddot{\theta}_1 \sin \theta_1 = d_1 \dot{\theta}_1^2 \cos \theta_1. \tag{11-CON2}
\]

Point B fixed:
This is nearly identical to the case above. Just replace labels:
Point \(B = (x_B, y_B)\) may be written in terms of \((x_2, y_2)\) as follows:

\[
\begin{pmatrix}
x_B \\
y_B
\end{pmatrix} = \begin{pmatrix} x_2 - d_2 \sin \theta_2 \\
y_2 + d_2 \cos \theta_2 \end{pmatrix}.
\]

Differentiate this expression twice and take \((\ddot{x}_B, \ddot{y}_B) = (0, 0)\) to obtain

\[
\ddot{x}_2 - d_2 \ddot{\theta}_2 \cos \theta_2 = -d_2 \dot{\theta}_2^2 \sin \theta_2 \tag{12-CON3}
\]

and

\[
\ddot{y}_2 - d_2 \ddot{\theta}_2 \sin \theta_2 = d_2 \dot{\theta}_2^2 \cos \theta_2. \tag{13-CON4}
\]

Point C fixed:
Point \(C\) may be written in two different ways, since the constraint is precisely that one end of the bars \(BC\) and \(CD\) coincide:

\[
\begin{pmatrix}
x_2 + d_2 \sin \theta_2 \\
y_2 - d_2 \cos \theta_2
\end{pmatrix} = \begin{pmatrix} x_3 + d_3 \cos \theta_3 \\
y_3 + d_3 \sin \theta_3 \end{pmatrix}.
\]
Differentiate this vector equality twice to obtain

$$\ddot{x}_3 - \ddot{x}_2 - d_3 \ddot{\theta}_3 \sin \theta_3 - d_2 \ddot{\theta}_2 \cos \theta_2 = d_3 \dot{\theta}_3^2 \cos \theta_3 - d_2 \dot{\theta}_2^2 \sin \theta_2$$  \hspace{1cm} (14-CON5)

and

$$\ddot{y}_3 - \ddot{y}_2 + d_2 \ddot{\theta}_2 \sin \theta_2 - d_3 \ddot{\theta}_3 \cos \theta_3 = -d_3 \dot{\theta}_3^2 \sin \theta_3 - d_2 \dot{\theta}_2^2 \cos \theta_2.$$  \hspace{1cm} (15-CON6)

Point $D$ fixed:

Point $D$ may be written in two different ways, since the constraint is precisely that one end of the bars $CD$ and $AD$ coincide:

$$\begin{pmatrix} x_1 + d_1 \sin \theta_1 \\ y_1 - d_1 \cos \theta_1 \end{pmatrix} = \begin{pmatrix} x_3 - d_3 \cos \theta_3 \\ y_3 - d_3 \sin \theta_3 \end{pmatrix}.$$  \hspace{1cm} (16-CON7)

Differentiate this vector equality twice to obtain

$$\ddot{x}_1 - \ddot{x}_3 - d_3 \ddot{\theta}_3 \sin \theta_3 + d_1 \ddot{\theta}_1 \cos \theta_1 = d_3 \dot{\theta}_3^2 \cos \theta_3 + d_1 \dot{\theta}_1^2 \sin \theta_1$$  \hspace{1cm} (16-CON7)

and

$$\ddot{y}_1 - \ddot{y}_3 - d_1 \ddot{\theta}_1 \sin \theta_1 + d_3 \ddot{\theta}_3 \cos \theta_3 = d_3 \dot{\theta}_3^2 \sin \theta_3 + d_1 \dot{\theta}_1^2 \cos \theta_1.$$  \hspace{1cm} (17-CON8)

This concludes the list of 17 DAEs. We now proceed to show how they may be numerically integrated.

1.4 Numerical integration of DAEs

Inspection of the equations derived above shows that each is indeed a linear combination of the components of $\xi$. Therefore we may write the equations compactly as a matrix-vector equation:

$$M \xi = \mathbf{v},$$
where \( \mathbf{v} \) is a vector of functions of ‘known’ quantities— the positions, velocities, and parameters of the problem.

Recall that our system has one degree of freedom. Indeed, all we care about is \( \xi_7 = \ddot{\theta}_1 \) given initial conditions \( \theta_{1,0}, \dot{\theta}_{1,0} \). The other angles and angular velocities are in principle derivable from this information.

However, in our numerical integration, we decide instead to record all three angles and velocities (i.e. we consider the dynamical system \( (\xi_7, \xi_8, \xi_9) = F(\theta_i; \text{parameters}) \) instead).

Therefore, our numerical integration is performed for the six variables \( (\theta_1, \theta_2, \theta_3, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3) \).

Our numerical integration involves setting up the matrix \( M \), the vector \( \mathbf{v} \), and then backsolving \( \xi = M\backslash \mathbf{v} \). The matrix \( M \) has the following structure:

\[
M = \begin{bmatrix} \mathbf{D}_M & \mathbf{J}_1 \\ \mathbf{J}_2 & 0 \end{bmatrix},
\]

where \( \mathbf{D}_M \) is a 9 \times 9 diagonal matrix consisting of the masses \( m_i \) in pairs as well as the moments of inertia, and \( \mathbf{J}_1 \) (resp. \( \mathbf{J}_2 \)) is a 9 \times 8 (resp. 8 \times 9) matrix of coefficients. \( \mathbf{J}_1 \) and \( \mathbf{J}_2 \) are nearly antisymmetric with respect to each other, but differences in sign due to the definition of the signs of the tensions and the choice of angular measurements prevent them from being a completely antisymmetric matrix pair.

The code used to integrate the EOM is relegated to the appendix. Before continuing to the numerical results, we briefly discuss algebraic constraints we can use to check our integration.

### 1.5 Checking the integration: constraints

We can check the following constraints:

1. **Two constraints**: The location of hinge \( C \) must agree as measured in \((x_3, y_3)\) coordinates and \((x_2, y_2)\) coordinates:

   \[
   C_{\text{bar} \ 3} = C_{\text{bar} \ 2} \\
   l_1(\sin \theta_1, -\cos \theta_1) + l_3(\cos \theta_3, \sin \theta_3) = (l_0, 0) + l_2(\sin \theta_2, -\cos \theta_2)
   \]

2. **Two constraints**: Equivalent to above: the location of hinge \( D \) must agree as measured in \((x_3, y_3)\) coordinates and \((x_1, y_1)\) coordinates:

   \[
   D_{\text{bar} \ 3} = D_{\text{bar} \ 1} \\
   (l_0, 0) + l_2(\sin \theta_2, -\cos \theta_2) - l_3(\cos \theta_3, \sin \theta_3) = l_1(\sin \theta_1, -\cos \theta_1)
   \]
3. **One constraint:** The total energy of the four-bar linkage must be conserved.

**On conservation of bar lengths:**
The lengths of bars AD and BC are trivially conserved since:

(a) \( D = l_1(\sin \theta_1, -\cos \theta_1) \Rightarrow \|AD\| = l_1 \) and

(b) \( B = (l_0, 0), C = B + l_2(\sin \theta_2, -\cos \theta_2) \Rightarrow \|BC\| = l_2 \) for all \( t \).

If the hinges measured with respect to two adjacent bars agree (as in the constraint conditions above), **the length of bar CD is conserved:**

\[
\|CD\| = \|C_{\text{bar }3} - D_{\text{bar }1}\| \\
= \|l_1(\sin \theta_1, -\cos \theta_1) + l_3(\cos \theta_1, \sin \theta_1) - l_1(\sin \theta_1, -\cos \theta_1)\| \\
= \|l_3(\cos \theta_3, \sin \theta_3)\| = l_3.
\]

### 1.6 Numerical results

1. **Structure released from near steady state**

We initialize with the following i.c.s and parameters:

```plaintext
1 tspan=[0,40];
2 l0=1;
3 p1=[0.5;-4];
4 p2=[3;-3];
5 m1=1;
6 m2=1;
7 m3=1;
8 IG1=2;
9 IG2=2;
10 IG3=2;
11 g=10;
```

The initial configuration is given as follows:
Figure 1: Configuration for parameters given above. Black bar is fixed. Black circles represent massless hinges. Rods are rigid.

The angle evolution over 40 time units is given as follows (color coding matches across all graphs):

Figure 2: Dynamical evolution of angular variables $\theta_1, \theta_2, \theta_3$. 
We show the drift in the hinges $C$ and $D$ as measured from adjacent bars (the error term is measured like $|C_{\text{bar } 2,x} - C_{\text{bar } 3,x}|$, etc. for the other components):

Finally, we plot energy as a function of time for this initialization:

Figure 3: Hinge drift.
We notice a linear energy drift as we numerically integrate. In this case, the error function $E_{\text{end}}/E_{\text{start}} - 1$ was shown to evaluate to $-8.8486 \times 10^{-5}$.

2. Structure released with high angle

We now initialize with the following i.c.s and parameters:

```matlab
1 tspan=[0,40];
2 l0=1;
3 p1=[1;4];
4 p2=[2;3];
5 m1=1;
6 m2=1;
7 m3=1;
8 IG1=2;
9 IG2=2;
10 IG3=2;
11 g=10;
```

The initial configuration now has a large angle from the rest position, as shown:
The resulting (one-dimensional) behavior of the structure remains periodic but with a different amplitude structure:

The order of the two sets of sharp peaks in this figure is interesting: in the animation,
it is shown that the green bar flips at the apex of the first swing, and then flips again at the apex of the second swing, before returning to its original motion (run the file DEMOtwobarsup.m to see this behavior!)

This behavior can also be studied in the context of the following phase space projection of the dynamics onto the \((\theta_1, \theta_3)\) plane:

![Phase space projection of dynamics onto \((\theta_1, \theta_3)\) plane.](image)

Figure 7: Phase space projection of dynamics onto \((\theta_1, \theta_3)\) plane.

We note that the dynamics lies on a 1-dimensional level set of the entire 2-dimensional phase space \((\theta_1, \dot{\theta}_1)\). This level set is a level set of the energy function \(H(\theta_1, \dot{\theta}_1) = c\) (in other words, energy conservation reduces the dimension of the dynamics by one).

A manifestation of this one-dimensional behavior can be seen in the plot above: the trajectory goes up and down the curve shown in a periodic motion. For an interval of \(\theta_1\), we have \(\theta_3\) (for the green bar) either above \(-\pi\) or below \(-\pi\), indicating there are portions of the trajectory where the green bar has been flipped.

We plot hinge and energy drift:
Figure 8: Energy conservation over 40 time units.

Figure 9: Hinge drift.
1.7 A note on animation

We define the hinges $C$ and $D$ by measuring them from the fixed hinges $B$ and $A$, respectively:

\[
C_{animate} = C_{bar\ 2} \\
D_{animate} = D_{bar\ 1}
\]

Bar CD (i.e., bar 3) is then drawn by connecting the hinges $C_{animate}$ and $D_{animate}$.

If the hinges drift apart between the moving bars (i.e. if the constraint equations in the problem are not satisfied), the lengths of the bars AD and BC trivially remain fixed, but the length of bar CD will not be conserved. This can be seen, for instance, by inputting (nonzero) values of $\dot{\theta}_i$ which do not satisfy the kinematic constraints of the problem.

This concludes the study of the four-bar linkage.
2 The Triple Pendulum

2.1 Specification of coordinates and initial conditions

The triple pendulum is shown as follows:

We note that the angle variables $\theta_i$ are measured from vertically downward. We limit ourselves to the case where the center of gravity of each bar is in the middle of the bar: $d_i = L_i/2$.

By definition of our basis vectors, we have

$$\mathbf{v}_i = \dot{\mathbf{r}}_i = \omega_i \times \mathbf{r}_i = r_i \omega_i \hat{\theta}_i,$$

where $i$ here means:

1. Measured from $O$ if $i = 1$,
2. Measured from $E_1$ if $i = 2$,
3. Measured from $E_2$ if $i = 3$. 

We use these formulas without further comment in the following derivations using the Lagrangian and the Newtonian formalisms.

2.2 Derivation 1: the Lagrangian

We first derive the Lagrangian \( L = T - V \), where \( T = T_1 + T_2 + T_3 \) and similarly \( V = V_1 + V_2 + V_3 \). The kinetic energy of the first bar \( T_1 \) is then given by

\[
T_1 = \frac{1}{2} m_1 \left\| \mathbf{v}_{G1/O} \right\|^2 + \frac{1}{2} I^G_1 \dot{\theta}_1^2
= \frac{1}{2} m_1 d_1^2 \dot{\theta}_1^2 + \frac{1}{2} I^G_1 \dot{\theta}_1^2.
\]

The kinetic energy of the second bar \( T_2 \) is given by

\[
T_2 = \frac{1}{2} m_2 \left\| \mathbf{v}_{E1/O} + \mathbf{v}_{G2/E1} \right\|^2 + \frac{1}{2} I^G_2 \dot{\theta}_2^2
= \frac{1}{2} m_2 (L_1^2 \dot{\theta}_1^2 + d_2^2 \dot{\theta}_2^2 + 2L_1 d_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)) + \frac{1}{2} I^G_2 \dot{\theta}_2^2.
\]

Finally, the kinetic energy of the third bar \( T_3 \) is given by

\[
T_3 = \frac{1}{2} m_3 \left\| \mathbf{v}_{E1/O} + \mathbf{v}_{E2/E1} + \mathbf{v}_{G3/E2} \right\|^2 + \frac{1}{2} I^G_3 \dot{\theta}_3^2
\]

Define the function \( L_2'(m_2, d_2) = \frac{1}{2} m_2 (L_1^2 \dot{\theta}_1^2 + d_2^2 \dot{\theta}_2^2 + 2L_1 d_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)) \) (the first part of \( T_2 \), except with \((m_2, d_2)\) variable). Then we can write \( T_3 \) as follows:

\[
T_3 = L_2'(m_3, L_2) + \frac{1}{2} m_3 d_3^2 \dot{\theta}_3^2 + m_3 d_3 \dot{\theta}_3 (L_1 \dot{\theta}_1 \cos(\theta_1 - \theta_3) + L_2 \dot{\theta}_2 \cos(\theta_2 - \theta_3)) + \frac{1}{2} I^G_3 \dot{\theta}_3^2.
\]

The potential energies are more straightforward— we simply compute \( m_i g h_i \) (we measure from the common basepoint \( h = 0 \) for ease of computation):

\[
V_1 = m_1 g d_1 (1 - \cos \theta_1)
V_2 = m_2 g (L_1 (1 - \cos \theta_1) + d_2 (1 - \cos \theta_2))
V_3 = m_3 g (L_1 (1 - \cos \theta_1) + L_2 (1 - \cos \theta_2) + d_3 (1 - \cos \theta_3))
\]

The EOM are then computed by solving the Euler-Lagrange equations:
\[ \frac{\partial \mathcal{L}}{\partial \dot{\theta}_i} = \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{\theta}}. \]

Because these equations are a mess, we don’t write them here. See instead the code in the appendix, which derives the equations of motion.

### 2.3 Derivation 2: the Newtonian approach

We compute angular momentum balance (AMB) for three subsystems. As before, the notation \( G_i \) indicates the center of gravity of bar \( i \) (for eg. \( r_{G_i/O} \) is the position of the center of gravity of bar \( i \) about \( O \)).

The three AMB equations are as follows:

1. About \( O \) for the entire system with three bars:
   
   We have
   
   \[ \mathbf{M}_{/O} = \mathbf{H}_{/O} \]
   
   \[ \sum_{i=1}^{3} \mathbf{r}_{G_i/O} \times (-m_i g \hat{j}) = \sum_{i=1}^{3} (m_i \mathbf{r}_{G_i/O} \times \mathbf{a}_i + I_{G_i}^{\ddot{\theta}_i} \hat{k}), \]

   where \( \mathbf{a}_i = -\omega_i^2 \mathbf{r}_i + \ddot{\omega}_i \hat{k} \times \mathbf{r}_i \) is the acceleration of the \( i \)-th center of gravity in the inertial reference frame (note: \( \omega_i \) here is not necessarily \( \dot{\theta}_i \) for \( i = 2, 3 \) since this would be the angle with respect to the origin. To bypass this issue, simply add the accelerations componentwise from the topmost mass).

2. About \( E_1 \) for the subsystem consisting of bars 2 and 3:
   
   We have
   
   \[ \mathbf{M}_{/E_1} = \mathbf{H}_{/E_1} \]
   
   \[ \sum_{i=2}^{3} \mathbf{r}_{G_i/E_1} \times (-m_i g \hat{j}) = \sum_{i=2}^{3} (m_i \mathbf{r}_{G_i/E_1} \times \mathbf{a}_i + I_{G_i}^{\ddot{\theta}_i} \hat{k}). \]

3. About \( E_2 \) for the subsystem consisting of bar 3:
   
   We have
   
   \[ \mathbf{M}_{/E_2} = \mathbf{H}_{/E_2} \]
   
   \[ \mathbf{r}_{G_3/E_2} \times (-m_3 g \hat{j}) = m_3 \mathbf{r}_{G_3/E_2} \times \mathbf{a}_3 + I_{G_3}^{\ddot{\theta}_3} \hat{k}. \]
Equality of the derivations:

In order to check that the solution set \((\ddot{\theta}_i)^{\text{Newtonian}}\) is exactly equal to the solution set \((\ddot{\theta}_i)^{\text{Lagrangian}}\), we carried out these symbolic manipulations in Mathematica. The resulting two vectors of equations were subtracted from one another. It was found that

\[\ddot{\theta}_i^{\text{Newtonian}} - \ddot{\theta}_i^{\text{Lagrangian}} = 0\]

for \(i = 1, 2, 3\), implying that the derivations were symbolically identical. This calculation is found in the notebook file triplependderivecheck.nb.

2.4 Numerical results

Satisfied that either derivation procedure gives identical results, we integrate the equations of motion given by the Lagrangian formulation. The matlab code used for this derivation is given in derivetriple.m and may be found in the appendix.

We consider three cases of interest: low energy oscillations about the stable point, medium energy oscillations resulting in chaotic behavior, and high energy whirling motion that is ‘above’ the region of chaotic trajectories in the 6-dimensional phase space.

1. Low energy case: small oscillations:

The following output is provided by the code DEMOsmall.m.

We initialize with the following parameter values:

```plaintext
1 ttot=30;
2 m1 = 1.0;
3 m2 = 1.0;
4 m3 = 1;
5 L1 = 1.0;
6 L2 = 1.0;
7 L3 = 1.0;
8 g = 10;
9 th1zero = 0;
10 th2zero = -pi/30;
11 th3zero = pi/15;
12 th1dotzero = 0;
13 th2dotzero = 0;
14 th3dotzero = 0;
```

The configuration is given as follows:
We find superpositions of small modal oscillations in this regime:
Consider $\theta_3$. We find small discrepancies as we compare ‘beats’ of the oscillations. The low energy regime apparently does not prevent the dynamical variables from traversing the accessible portion of the phase space in a quasiperiodic fashion.

We measure the energy drift for this set of parameters:
The energy drift in this case (measured identically to the energy drift function in the four-bar linkage) was measured to be $-1.4395 \times 10^{-11}$.

2. **Medium energy case: complex and chaotic oscillations**

We now consider the case where the triple pendulum is initialized pointing to the right:

```matlab
1 th1zero = pi/2;
2 th2zero = pi/2;
3 th3zero = pi/2;
```

The angles evolve in a highly nonlinear fashion. The following figure depicts a snapshot of the evolution after a few swings:
Figure 13: Triple pendulum dynamics after a few swings of bar 1.

The following plot of the angular variables also depicts the nonlinear motion. In particular, the bottom-most bar eventually experiences epochs of whirling, as shown in the plot for times in the approximate interval of $(13, 17)$:

Figure 14: Evolution of angular variables
The energy in this case suffers from very little drift: $-3.6116 \times 10^{-13}$.

![Figure 15: Energy conservation](image)

3. High energy case: whirling behavior

Finally, we briefly show an instance for which high enough energy 'dampens out' the chaotic motion. The following graphs are generated via the matlab file DEMOwhirligig.m. Our initial conditions are given as follows (all other parameters are as in the other cases):

```matlab
1 th1zero = 4*pi/5;
2 th2zero = 4*pi/5;
3 th3zero = 4*pi/5;
4 th1dotzero = 10;
5 th2dotzero = 0;
6 th3dotzero = 0;
```

We begin with the triple pendulum close to vertically up and give the innermost bar a large angular velocity. The resulting behavior is shown in the angular evolution plot:
Figure 16: Evolution of angular variables in the very high energy case

In this case, the angular velocity appears to be linear plus some small perturbations.
3 Appendix

3.1 Code for the four-bar linkage

The four-bar linkage program consists of the following four pieces:

1. an input file, of which any of the DEMO*.m files serve as a template
2. fourbarsdriver.m, the main program which integrates and then plots things
3. fourbarsode2.m, the ODE files used by the driver
4. fourbarsanimate.m, the code for animation also used by the driver

Since the files are all well-commented, we simply list them here:

1. Input file:

```matlab
1 %Input parameters and initial conditions
2 tspan=[0,80]; %Integration time
3 l0=1; %Length of fixed bar
4 p1=[1;4]; %Point D
5 p2=[2;3]; %Point C
6 m1=1; %Masses
7 m2=1;
8 m3=1;
9 IG1=2; %Moments of inertia
10 IG2=2;
11 IG3=2;

12 close all;
13 %Integration:
14 [tout thout] = fourbardriver(tspan, l0, p1, p2, m1, m2, m3, IG1, IG2, IG3);
```

2. fourbarsdriver.m:

```matlab
function [tout thout] = ...
    fourbardriver(tspan, l0, p1, p2, m1, m2, m3, IG1, IG2, IG3)

%NOTE: the best way to use this function is to modify one of the demos
%included in the folder. For example twobarsdown.m.

%This is the main function to integrate the four-bar system. It has
%three main parts:
% 1. Specifying initial conditions
```
% 2. Numerical integration
% 3. Figure plotting

% Let's explain the three main parts in turn:

% 1. Converts our input (length of fixed rod, two other points p1 ... and p2 in
% the plane) into initial conditions for the three angles theta_1, ...
% theta_2, % and theta_3. It also finds the lengths (l1, l2, l3), as well as
% midpoints (d1,d2,d3), of the bars, which we expect to remain fixed.

% WE ASSUME THE CENTERS OF MASS ARE IN THE MIDDLE: d_i = l_i/2. ...
% This eases
% the algebraic computation in the ODE files.

% Remark about 1: *by convention* we set the following:
% length AB = l_0 is always fixed horizontally (so A = (0,0), B = ...
% (l_0,0))
% C = p2
% D = p1.
% Just keep this in mind. So if p1 is to the left of p2, you get ...
% the bars
% crossing each other (eg. demo downflipped.m).

% 2. Numerically integrates the initial conditions for the system of
% variables (theta_1, theta_2, theta_3) and their first ...
% derivatives. ode23
% is used.

% 3. Plots four different things:
% - a plot of the drift in hinges as measured from adjacent bars
% - a plot of the evolution of the three angles th1, th2, and th3
% - a plot of the total energy of the system vs time
% - an animation of the four-bar system

% %0. MAIN PROGRAM
% %1. SPECIFY INITIAL CONDITIONS
% A=[0;0];
% B=[l0;0];
% C=p2;
% D=p1;
% l1=norm(D);
% 12=norm(C-B);
l3=norm(D−C);

d1=l1/2;
d2=l2/2;
d3=l3/2;

th1init=atan2(p1(1),−p1(2));
th2init=atan2(p2(1)−10,−p2(2));
th3init=atan2(p2(2)−p1(2),p2(1)−p1(1));

fprintf(['l1 = ',num2str(l1), ', l2 = ',num2str(l2), ', l3 = ... 
',num2str(l3),']
')
fprintf(['d1 = ',num2str(d1), ', d2 = ',num2str(d2), ', d3 = ... 
',num2str(d3),']
')
fprintf(['th1 = ',num2str(th1init), ', th2 = ',num2str(th2init), ... 
', 'th3 = ', num2str(th3init),']
')

%%2. NUMERICAL INTEGRATION

%Initial conditions:
dd=[d1;d2;d3];
ths=[th1init;th2init;th3init];
% thdots=[0;0;0];
% thdots=[0;0;0];
m = [m1;m2;m3];
IGs=[IG1;IG2;IG3];
g=1;
x0=[ths;thdots];
options = odeset('RelTol', 1e−16);
[tout,thout] = ode23(@(t,x) ... 
 fourbarodes2(t,x,dd,m,g,IGs),tspan,x0,options);

%%3. PLOTTING FIGURES

% Figure 1: Length constraint check
% Figure 2: Angular evolution
% Figure 3: Energy conservation check
% Figure 4: Animation
%%HINGE CONSTRAINT CHECK

```matlab
figure(1); hold on;

% Measure hinge C from bar 2 (BC)
Cx2 = B(1) + 12*sin(thout(:,2));
Cy2 = B(2) - 12*cos(thout(:,2));

% Measure hinge C from bar 3 (CD)
Cx3 = l1*sin(thout(:,1)) + l3*cos(thout(:,3));
Cy3 = -l1*cos(thout(:,1)) + l3*sin(thout(:,3));

% Measure hinge D from bar 1 (AD)
Dx1 = l1*sin(thout(:,1));
Dy1 = -l1*cos(thout(:,1));

% Measure hinge D from bar 3 (CD)
Dx3 = 10 + l2*sin(thout(:,2)) - l3*cos(thout(:,3));
Dy3 = -l2*cos(thout(:,2)) - l3*sin(thout(:,3));

plot(tout,abs(Cx2 - Cx3),'b','LineWidth',4)
plot(tout,abs(Cy2 - Cy3),'r','LineWidth',4)
plot(tout,abs(Dx1 - Dx3),'g','LineWidth',2)
plot(tout,abs(Dy1 - Dy3),'y','LineWidth',2)
legend('Cx Error', 'Cy Error', 'Dx Error', 'Dy Error');
set(gca,'FontSize',20);
xlabel('time','FontSize',20);
ylabel('hinge error','FontSize',20);
```

%%ANGLE EVOLUTION

```matlab
figure(2); hold on;
plot(tout,thout(:,1),'b','LineWidth',2)
plot(tout,thout(:,2),'r','LineWidth',2)
plot(tout,thout(:,3),'g','LineWidth',2)
legend('Bar 1', 'Bar 2', 'Bar 3');
title('Angle check','FontSize',20);
set(gca,'FontSize',20);
xlabel('time','FontSize',20);
ylabel('\theta_1','FontSize',20);
hold off
```

%%ENERGY CONSERVATION CHECK

```matlab
Pots = g*(+(m1+2*m3)*d1*(1 - cos(thout(:,1))) + m2*d2*(1 - ...
```
cos(thout(:,2)) + m3*(d3*sin(thout(:,3)));

Kins=(m1/2)*d1^2*thout(:,4).^2 + (m2/2)*d2^2*thout(:,5).^2 + ...
(m3/2)*d3^2*thout(:,6).^2 + 11^2*thout(:,4).^2 + ...
2*d3*l1*thout(:,4).*thout(:,6).*sin(thout(:,1)-thout(:,3)));

Kins = Kins + (1/2)*(IG1*thout(:,4).^2 + IG2*thout(:,5).^2 + ...
IG3*thout(:,6).^2);

figure(3); hold on;
plot(tout,Pots+Kins,'k','LineWidth',2);

fprintf(['Energy drift = ...
    ',num2str((Pots(end)+Kins(end))/(Pots(1)+Kins(1))−1),'

title('Energy conservation check','FontSize',20)

set(gca,'FontSize',20);
xlabel('time','FontSize',20);
ylabel('E_{total}','FontSize',20);
hold off

%%%ANIMATION

figure(4);

fourbarsanimate

3. fourbarod2.m:

function yout = fourbarodes2(t,y,d,m,g,IGs)

yout=zeros(6,1);

Big = zeros(17,17);
rhsvec=zeros(17,1);
rhsvec2=zeros(17,1);

% y = [th1;th2;th3;th1dot;th2dot;th3dot]

%Method: Solve Big * rhsvec2 = rhsvec for rhsvec2.

%Meaning of these matrices and vectors: solving an extended system of 
% of ODEs + constraint equations. More information in project doc.

%%%RHS Equation
%% Auxiliary things
\theta_d \dot{\theta}_d^2
\theta_d = y(4:6)^2;

% Pieces that show up
\theta_1 v = d(1) \times [-\sin(y(1)); \cos(y(1))];
\theta_2 v = d(2) \times [-\sin(y(2)); \cos(y(2))];
\theta_3 v = d(3) \times [\cos(y(3)); \sin(y(3))];

% Write down 17-dimensional rhs vector
rhsvec(2:2:6) = -g*m;
rhsvec(10:13) = [\theta_d(1)\times \theta_1 v; \theta_d(2)\times \theta_2 v];
rhsvec(14:17) = \theta_d(3) \times [\theta_3 v; \theta_3 v] + [\theta_d(2)\times \theta_2 v; -\theta_d(1)\times \theta_1 v];

%% Now to write down the big matrix

% Auxiliary things:
\theta_1 v = [\cos(y(1)); \sin(y(1))];
\theta_2 v = [\cos(y(2)); \sin(y(2))];
\theta_3 v = [\sin(y(3)); \cos(y(3))];

% Top left: Mass and moment of inertia part
Big(1:9,1:9) = diag([m(1) m(1) m(2) m(2) m(3) m(3) IGs(1) IGs(2) ... IGs(3)]);

% Top right
% Subsection: LMB tension part
Big(1:6,10:15) = diag([-1 -1 -1 -1 1 1]);
Big(3:6,14:17) = Big(3:6,14:17) + diag([-1 -1 1 1]);
Big(1:2,16:17) = diag([-1,-1]);

% Subsection: AMB tension part
Big(7,[10,11,16,17]) = d(1) \times [\theta_1 v', -\theta_1 v'];
Big(8,12:15) = d(2) \times [\theta_2 v', -\theta_2 v'];
Big(9,14:17) = d(3) \times [-\theta_3 v(1), \theta_3 v(2), \theta_3 v(1), -\theta_3 v(2)];

% Bottom left: Constraints
Big(10:15,1:6) = eye(6);
Big(14:17,3:6) = Big(14:17,3:6) - eye(4);
Big(16:17,1:2) = eye(2);
Big([10,11,16,17],7) = d(1) \times [ -\theta_1 v; \theta_1 v];
%Bottom right: zeros

%Compute entire extended system and record theta double dots:
rhsvec2 = Big \( \{ \text{rhsvec} \};
\)

yout(1:3) = y(4:6);
yout(4:6) = rhsvec2(7:9);

4. fourbarsanimate.m:

moviescaling = 1;
n = length(tout); dur = tout(end);
delay = floor(moviescaling * 1000 * dur / n); % delay per frame in .001 secs

axis('equal')

maxright = max([l0; abs(Cx); abs(Dx)]);
mindown = min([Cy; Dy]);
maxdown = max([Cy; Dy; 1]);

axis([-maxright - 1 maxright + 1 mindown maxdown])

bar0pic = line('xdata', [0; l0], 'ydata', [0; 0], ...
    'linewidth', 3, 'erase', 'xor', 'color', 'black');
bar1pic = line('xdata', [0; p1(1)], 'ydata', [0; p1(2)], ...
    'linewidth', 3, 'erase', 'xor', 'color', 'blue');
bar2pic = line('xdata', [l0; p2(1)], 'ydata', [0; p2(2)], ...
    'linewidth', 3, 'erase', 'xor', 'color', 'red');
bar3pic = line('xdata', [p1(1); p2(1)], 'ydata', [p1(2); p2(2)], ...
    'linewidth', 3, 'erase', 'xor', 'color', 'green');
hingeA = line('xdata', 0, 'ydata', 0, 'marker', '.', 'markersize', [30], ...
    'color', 'black', 'erase', 'xor');
hingeB = line('xdata', 10, 'ydata', 0, 'marker', '.', 'markersize', [30], ...
    'color', 'black', 'erase', 'xor');
hingeC = line('xdata', p2(1), 'ydata', p2(2), ...
    'marker', '.', 'markersize', [30], ...
    'color', 'black', 'erase', 'xor');
hingeD = line('xdata', p1(1), 'ydata', p1(2), ...
    'marker', '.', 'markersize', [30], ...
3.2 Code for the triple pendulum

The triple pendulum program in matlab consists of the following five pieces:

1. an input file, of which any of the DEMO*.m files serve as a template
2. triplependdriver.m, the main program that integrates and then plots things
3. derivetriple.m, a symbolic program that generates the EOM from the Lagrangian and inserts it into
4. triplependodes.m, the ODE file used by the driver
5. triplependanimate.m, the code for animation also used by the driver

In addition, we also include a mathematica notebook file called triplependderivecheck.nb containing a proof of the equality of the two derivations of the EOM.

We list each of these commented programs in turn:

1. Input file:

```matlab
clear functions
close all
clear all

%%%Parameters
%ttot=30;
ml = 1.0;
```
m2 = 1.0;
m3 = 1;
L1 = 1.0;
L2 = 1.0;
L3 = 1.0;
g = 10;

th1zero = 0;
th2zero = -pi/30;
th3zero = pi/15;

th1dotzero = 0;
th2dotzero = 0;
th3dotzero = 0;

[t z] = triplependdriver(ttot,m1,m2,m3,L1,L2,L3,g,th1zero,...
    th2zero, th3zero, th1dotzero, ...
    th2dotzero,th3dotzero);

2. derivetriples.m:

  syms th1 th2 th3 L1 L2 L3 th1dot th1ddot th2dot th2ddot real
  syms th3dot th3ddot real
  syms m1 m2 m3 I1 I2 I3 C1g C2g C3g g ... real
  syms T1 T2 T3 V1 V2 V3 LL ...
  syms eqs1 eqs2 eqs3 th1eqs th2eqs the3eqs ... real

% Moments of inertia
I1=(1/12)*m1*L1^2;
I2=(1/12)*m2*L2^2;
I3=(1/12)*m3*L3^2;

% Centers of mass
C1g=L1/2;
C2g=L2/2;
C3g=L3/2;

% Kinetic energies
T1=(1/2)*m1*C1g^2*th1dot^2 + (1/2)*I1*th1dot^2;
T2=(1/2)*m2*(L1^2*th1dot^2 + ... C2g^2*th2dot^2+L1*L2*th1dot*th2dot*cos(th1-th2)) + (1/2)*I2*th2dot^2;
T3=(1/2)*m3*(L1^2*th1dot^2 + ... L2^2*th2dot^2+2*L1*L2*th1dot*th2dot*cos(th1-th2));
T3=T3+ (1/2)*I3*th3dot^2+ (m3/2)* C3g^2*th3dot^2 + ...
    m3*C3g*th3dot*(L1*th1dot*cos(th1 - th3) + L2*th2dot*cos(th2 - ... th3));
%Potential energies
V1=m1*g*(1-cos(th1));
V2=m2*g*(L1*(1-cos(th1))+C2g*(1-cos(th2)));
V3=m3*g*(L1*(1-cos(th1))+L2*(1-cos(th2)) + C3g*(1-cos(th3)));

Ttotal=simple(T1+T2+T3);
Vtotal=simple(V1+V2+V3);

%Lagrangian
LL=Ttotal-Vtotal;

%Computation of d(del L/ del thetadot)/dt
y=[th1dot,th1,th2dot,th2,th3dot,th3];
xi=[th1ddot,th1dot,th2ddot,th2dot,th3ddot,th3dot];
M1=diff(LL,th1dot);
M2=diff(LL,th2dot);
M3=diff(LL,th3dot);
dLthdot1dt=xi*(jacobian(M1,y)');
dLthdot2dt=xi*(jacobian(M2,y)');
dLthdot3dt=xi*(jacobian(M3,y)');

% Computation of dL/dt − d(del L/ del thetadot)/dt
eqs1=diff(LL,th1)-dLthdot1dt;
eqs2=diff(LL,th2)-dLthdot2dt;
eqs3=diff(LL,th3)-dLthdot3dt;

%Solve for theta double dot
[th1eqs th2eqs th3eqs]=solve(eqs1,eqs2,eqs3,th1ddot,th2ddot,th3ddot);

th1char = char(simple(th1eqs));
th2char = char(simple(th2eqs));
th3char = char(simple(th3eqs));

% Create ODE files
fid=fopen( 'triplependodes.m','w' ... );
fprintf(fid, 'function zdot=triplependodes(t,z,flag,m1, m2, m3, g, ... 
L1, L2, L3) ; \n');
fprintf(fid, 'th1 = z(1) ; ... 
\n');
fprintf(fid, 'th1dot = z(2) ; ... 
\n');
fprintf(fid, 'th2 = z(3) ; ... 
\n');
fprintf(fid, 'th2dot = z(4) ; ... 
\n');
fprintf(fid, 'th3 = z(5) ; ... 
\n');
fprintf(fid, 'th3dot = z(6) ; ... 

n');
fprintf(fid,['alpha1 = ...\n' th1char '; ... 

n']);
fprintf(fid,['alpha2 = ...\n' th2char '; ... 

n']);
fprintf(fid,['alpha3 = ...\n' th3char '; ... 

n']);
fprintf(fid, 'zdot = [th1dot alpha1 th2dot alpha2 th3dot alpha3]'' ... 

); fclose(fid);

3. triplependodes:

function zdot=triplependodes(t,z,flag,m1, m2, m3, g, L1, L2, L3) ;
th1 = z(1) ;
th1dot = z(2) ;
th2 = z(3) ;
th2dot = z(4) ;
th3 = z(5) ;
th3dot = z(6) ;
alpha1 = ... 

−(60*g*m2^2*sin(th1) + 36*g*m3^2*sin(th1) + 36*g*m2^2*sin(th1 − ... 

2*th2) + 36*g*m3^2*sin(th1 − 2*th2) + ... 

48*L2*m2^2*th2dot^2*sin(th1 − th2) + ... 

72*L2*m3^2*th2dot^2*sin(th1 − th2) + ... 

18*L3*m3^2*th2dot^2*sin(th1 − th3) + 48*g*m1*m2*sin(th1) + ... 

90*g*m1*m3*sin(th1) + 150*g*m2*m3*sin(th1) − 27*g*m1*m3*sin(th1 ... 

− 2*th2 + 2*th3) − 27*g*m1*m3*sin(th1 + 2*th2 − 2*th3) − ... 

27*g*m2*m3*sin(th1 − 2*th2 + 2*th3) − 27*g*m2*m3*sin(th1 + ... 

2*th2 − 2*th3) + 90*g*m2*m3*sin(th1 − 2*th2) − ... 

18*g*m2*m3*sin(th1 − 2*th3) + 36*L1*m2^2*th1dot^2*sin(2*th1 − ... 

2*th2) + 36*L1*m3^2*th1dot^2*sin(2*th1 − 2*th2) + ... 

18*L3*m3^2*th3dot^2*sin(th1 − 2*th2 + th3) + ... 

150*L2*m2*m3*th2dot^2*sin(th1 − th2) + ... 

12*L3*m2*m3*th3dot^2*sin(th1 − th3) + ... 

90*L1*m2*m3*th1dot^2*sin(2*th1 − 2*th2) − ... 

18*L1*m2*m3*th1dot^2*sin(2*th1 − 2*th3) − ... 

18*L2*m2*m3*th2dot^2*sin(th1 + th2 − 2*th3) + ... 

36*L3*m2*m3*th3dot^2*sin(th1 − 2*th2 + th3)/(2*L1*(16*m1*m2 + ... 

30*m1*m3 + 75*m2*m3 − 18*m2^2*cos(2*th1 − 2*th2) − ... 

18*m^2*cos(2*th1 − 2*th2) + 90*m2^2 + 18*m^2) − ... 

18*m1*m3*cos(2*th2 + 2*th3) − 45*m2*m3*cos(2*th1 − 2*th2) + ... 

9*m2*m3*cos(2*th1 − 2*th3) − 27*m2*m3*cos(2*th2 − 2*th3));

alpha2 = ... 

−(72*g*m2^2*sin(th2) − 36*g*m3^2*sin(2*th1 − th2) − ... 

72*g*m2^2*sin(2*th1 − th2) + 36*g*m3^2*sin(th2) − ... 

36*g*m1*m2*sin(2*th1 − th2) − 45*g*m1*m3*sin(2*th1 − th2) − ...
\[ \begin{align*}
135g \cdot m2 \cdot m3 \cdot \sin(2 \cdot \theta 1 - \theta 2) - 144L1 \cdot m2^2 \cdot \dot{\theta}1 \cdot \dot{\theta}2 \cdot \sin(\theta 1 - \ldots \\
\theta 2) - 72L1 \cdot m3^2 \cdot \dot{\theta}3 \cdot \sin(\theta 1 - \theta 2) + \ldots \\
18L3 \cdot m3^2 \cdot \dot{\theta}2 \cdot \sin(\theta 2 - \ldots \\
15g \cdot m1 \cdot m3 \cdot \sin(\theta 2) + 135g \cdot m2 \cdot m3 \cdot \sin(\theta 2) + \ldots \\
27g \cdot m1 \cdot m3 \cdot \sin(\theta 1 + \ldots \\
\theta 2 - \ldots \\
9g \cdot m1 \cdot m3 \cdot \sin(\theta 2 - \ldots \\
36L2 \cdot m2^2 \cdot \dot{\theta}2 \cdot \sin(\theta 1 - 2 \cdot \theta 2) - \ldots \\
36L2 \cdot m3^2 \cdot \dot{\theta}2 \cdot \sin(2 \cdot \theta 1 - 2 \cdot \theta 2) - \ldots \\
48L1 \cdot m1 \cdot m2 \cdot \dot{\theta}1 \cdot \dot{\theta}2 - \ldots \\
60L1 \cdot m1 \cdot m3 \cdot \dot{\theta}1 \cdot \dot{\theta}2 - \ldots \\
270L1 \cdot m2 \cdot m3 \cdot \dot{\theta}1 \cdot \dot{\theta}2 + \ldots \\
48L3 \cdot m1 \cdot m3 \cdot \dot{\theta}1 \cdot \dot{\theta}2 + \ldots \\
\alpha 3 = \ldots \\
\end{align*} \]

\[ \begin{align*}
14 \cdot (27g \cdot m2^2 \cdot \sin(2 \cdot \theta 2 - \ldots \\
9g \cdot m1 \cdot m3 \cdot \sin(\theta 3) - 27g \cdot m2^2 \cdot \sin(2 \cdot \theta 1 - 2 \cdot \theta 2 + \theta 3) + \ldots \\
9g \cdot m1 \cdot m2 \cdot \sin(2 \cdot \theta 1 - \ldots \\
54g \cdot m1 \cdot m3 \cdot \sin(2 \cdot \theta 1 - 3 \cdot \theta 2) + 18g \cdot m1 \cdot m3 \cdot \sin(2 \cdot \theta 2 - \ldots \\
18g \cdot m2 \cdot m3 \cdot \sin(2 \cdot \theta 1 - \ldots \\
18 \cdot L1 \cdot m2 \cdot \dot{\theta}1 \cdot \dot{\theta}2 \cdot \sin(\theta 1 - \ldots \\
54L2 \cdot m2^2 \cdot \dot{\theta}2 \cdot \sin(\theta 2 - 3 \cdot \theta 2 + \ldots \\
18L2 \cdot m2^2 \cdot \dot{\theta}2 \cdot \sin(2 \cdot \theta 1 - 2 \cdot \theta 2 + \theta 3) - 54g \cdot m1 \cdot m3 \cdot \sin(2 \cdot \theta 1 - \ldots \\
2 \cdot \theta 2 + \theta 3) - 54g \cdot m2 \cdot m3 \cdot \sin(2 \cdot \theta 1 - 2 \cdot \theta 2 + \theta 3) - \ldots \\
18L2 \cdot m2^2 \cdot \dot{\theta}2 \cdot \sin(2 \cdot \theta 1 - 2 \cdot \theta 2 - \ldots \\
54L1 \cdot m2 \cdot \dot{\theta}1 \cdot \dot{\theta}2 \cdot \sin(\theta 1 - 2 \cdot \theta 2 + \ldots \\
12L1 \cdot m1 \cdot m2 \cdot \dot{\theta}1 \cdot \dot{\theta}2 \cdot \sin(\theta 1 - \ldots \\
72L1 \cdot m1 \cdot m3 \cdot \dot{\theta}1 \cdot \dot{\theta}2 \cdot \sin(\theta 1 - \ldots \\
48L2 \cdot m1 \cdot m2 \cdot \dot{\theta}2 \cdot \sin(\theta 2 - \ldots \\
36L1 \cdot m2 \cdot m3 \cdot \dot{\theta}1 \cdot \dot{\theta}2 - \ldots \\
144L2 \cdot m1 \cdot m3 \cdot \dot{\theta}2 \cdot \sin(2 \cdot \theta 2 - \ldots \\
180L2 \cdot m2 \cdot m3 \cdot \dot{\theta}2 \cdot \sin(\theta 2 - \ldots \\
36L2 \cdot m2 \cdot m3 \cdot \dot{\theta}2 \cdot \sin(2 \cdot \theta 1 - \ldots \\
18L3 \cdot m2 \cdot m3 \cdot \dot{\theta}2 \cdot \sin(2 \cdot \theta 1 - 2 \cdot \theta 2 + \ldots \\
54L3 \cdot m2 \cdot m3 \cdot \dot{\theta}2 \cdot \sin(2 \cdot \theta 2 - \ldots \\
36L1 \cdot m1 \cdot m2 \cdot \dot{\theta}1 \cdot \dot{\theta}2 \cdot \sin(\theta 1 - 2 \cdot \theta 2 + \ldots \\
72L1 \cdot m1 \cdot m3 \cdot \dot{\theta}1 \cdot \dot{\theta}2 \cdot \sin(\theta 1 - \ldots \\
108L1 \cdot m2 \cdot m3 \cdot \dot{\theta}1 \cdot \dot{\theta}2 \cdot \sin(\theta 1 - 2 \cdot \theta 2 + \ldots \\
30m1 \cdot m3 + 75m2 \cdot m3 - 18m2^2 \cdot \cos(2 \cdot \theta 1 - 2 \cdot \theta 2) - \ldots \\
18m3^2 \cdot \cos(2 \cdot \theta 1 - 2 \cdot \theta 2) + 30m2^2 + 18m3^2 - \ldots \\
18m1 \cdot m3 \cdot \cos(2 \cdot \theta 2 - 2 \cdot \theta 3) - 45m2 \cdot m3 \cdot \cos(2 \cdot \theta 1 - 2 \cdot \theta 2) + \ldots \\
9m2 \cdot m3 \cdot \cos(2 \cdot \theta 1 - 2 \cdot \theta 3) - 27m2 \cdot m3 \cdot \cos(2 \cdot \theta 2 - 2 \cdot \theta 3)) ;
\end{align*} \]
\[
9 \cdot m2 \cdot m3 \cdot \cos(2 \cdot \theta1 - 2 \cdot \theta3) - 27 \cdot m2 \cdot m3 \cdot \cos(2 \cdot \theta2 - 2 \cdot \theta3))
\]

4. \texttt{triplependanimate.m}:

```matlab
clf;
n=length(t); dur=t(end);

th1=z(1,1);
th2=z(1,3);
th3=z(1,5);

R1=[cos(th1) -sin(th1); sin(th1) cos(th1)];
R2=[cos(th2) -sin(th2); sin(th2) cos(th2)];
R3=[cos(th3) -sin(th3); sin(th3) cos(th3)];

bar1=R1*[0;−L1];
bar2=R2*[0;−L2]+bar1;
bar3=R3*[0;−L3]+bar2;

hinge=plot(0,0, 'marker','.','markersize',50, ...'
    'color','black');hold on;
axis('equal')
Ltotal=(L1+L2+L3)*(5/4);
axis([-Ltotal Ltotal -Ltotal Ltotal]);

bar1pic=plot([0;bar1(1)],[0;bar1(2)], ...
    'linewidth', 3, 'color','blue');
bar2pic=plot([bar1(1);bar2(1)],[bar1(2);bar2(2)], ...
    'linewidth', 3, 'color','red');
bar3pic=plot([bar2(1);bar3(1)],[bar2(2);bar3(2)], ...
    'linewidth', 3, 'color','green');
barsizes=[7 4 2];
 [~,masslist]=sort([m1 m2 m3], 'descend');
set(bar1pic,'linewidth',barsizes(masslist(1))
set(bar2pic,'linewidth',barsizes(masslist(2))
set(bar3pic,'linewidth',barsizes(masslist(3))

pause(0.5)% Hold still for a second before starting
for i=1:n
    pause(.03)
```
% The purpose of this code is to ensure that two different methods of computing thetadoubledots give us the same answer.
% 1. Compute the Lagrangian and take Euler–Lagrange equations to find %VSol1 (list of three thetadoubledot equations)
% 2. Use AMB for three subsystems and solve to find VSol2 (list ... of %three thetadoubledot equations)
% 3. Simplify VSol1 – VSol2 and check that the result is (0,0,0)

% 1. LAGRANGIAN METHOD
functransform = {Th1 -> Th1[t], Th2 -> Th2[t], Th3 -> Th3[t],
                w1 -> Th1'[t], w2 -> Th2'[t], w3 -> Th3'[t]};

d1 = l1/2;
d2 = l2/2;
d3 = l3/2;
IG1 = (1/12)*m1*l1^2;
IG2 = (1/12)*m2*l2^2;
IG3 = (1/12)*m3*l3^2;

T1 = (1/2) m1 d1^2 w1^2 + (1/2) IG1 w1^2;
T2 = (1/2) m2 (l1^2 w1^2 + d2^2 w2^2 +
       2 d1 d2 w1 w2 Cos[Th1 - Th2]) + (1/2) IG2 w2^2;
T3 = (1/2) m3 (l1^2 w1^2 + l2^2 w2^2 +
       2 l1 l2 w1 w2 Cos[Th1 - Th2]);
\[ T_3 = T_3 + \frac{1}{2} m_3 d_3^2 w_3^2 + m_3 d_3 w_3 (11 w_1 \cos[\text{Th}_1 - \text{Th}_3] + 12 w_2 \cos[\text{Th}_2 - \text{Th}_3]) + \frac{1}{2} IG_3 w_3^2; \]

\[ T = \text{FullSimplify}[T_1 + T_2 + T_3]; \]

\[ V_1 = m_1 g d_1 (1 - \cos[\text{Th}_1]); \]

\[ V_2 = m_2 g (11 (1 - \cos[\text{Th}_1]) + d_2 (1 - \cos[\text{Th}_2])); \]

\[ V_3 = m_3 g (11 (1 - \cos[\text{Th}_1]) + 12 (1 - \cos[\text{Th}_2]) + d_3 (1 - \cos[\text{Th}_3])); \]

\[ V = \text{FullSimplify}[V_1 + V_2 + V_3]; \]

\[ LL = T - V; \]

\[ LTh = D[LL, \{\text{Th}_1, \text{Th}_2, \text{Th}_3\}] /. \text{functransform}; \]

\[ dLdw = D[LL, \{w_1, w_2, w_3\}] /. \text{functransform}; \]

\[ LThd1 = D[dLdw[[1]], t]; \]

\[ LThd2 = D[dLdw[[2]], t]; \]

\[ LThd3 = D[dLdw[[3]], t]; \]

\[ r1 = \text{FullSimplify}[LTh[[1]] - LThd1]; \]

\[ r2 = \text{FullSimplify}[LTh[[2]] - LThd2]; \]

\[ r3 = \text{FullSimplify}[LTh[[3]] - LThd3]; \]

\[ VSol1 = \text{Solve}\{r1 == 0, r2 == 0, r3 == 0, \{\text{Th}_1^{\prime}[t], \text{Th}_2^{\prime}[t], \text{Th}_3^{\prime}[t]\}\}; \]

% 2. NEWTONIAN METHOD

\[ d_1 = \frac{11}{2}; \]

\[ d_2 = \frac{12}{2}; \]

\[ d_3 = \frac{13}{2}; \]

\[ IG_1 = \frac{1}{12} m_1 l_1^2; \]

\[ IG_2 = \frac{1}{12} m_2 l_2^2; \]

\[ IG_3 = \frac{1}{12} m_3 l_3^2; \]

\[ ii = \{1, 0, 0\}; jj = \{0, 1, 0\}; kk = \{0, 0, 1\}; \]

\[ \lambda_1 = \{\sin[\text{Th}_1], -\cos[\text{Th}_1], 0\}; \]

\[ \lambda_2 = \{\sin[\text{Th}_2], -\cos[\text{Th}_2], 0\}; \]

\[ \lambda_3 = \{\sin[\text{Th}_3], -\cos[\text{Th}_3], 0\}; \]

\[ r_{01} = \lambda_1; \]

\[ r_{01d1} = \lambda_1 \cdot d_1; \]

\[ r_{12} = \lambda_2; \]

\[ r_{12d2} = \lambda_2 \cdot d_2; \]

\[ r_{23} = \lambda_3; \]

\[ r_{23d3} = \lambda_3 \cdot d_3; \]

\[ r_{02} = r_{01} + r_{12}; \]

\[ r_{03} = r_{02} + r_{23}; \]
\[ r_{1to3} = r_{0to3} - r_{0to1}; \]
\[ a_1 = -w_1^2 \cdot r_{0tod1} + w_1 \cdot \text{Cross}[kk, r_{0tod1}]; \]
\[ a_2 = -w_1^2 \cdot r_{0to1} + w_1 \cdot \text{Cross}[kk, r_{0to1}] + (-w_2^2 \cdot r_{1tod2} + w_2 \cdot \text{Cross}[kk, r_{1tod2}]); \]
\[ a_3 = -w_1^2 \cdot r_{0to1} + w_1 \cdot \text{Cross}[kk, r_{0to1}] + (-w_2^2 \cdot r_{1to2} + w_2 \cdot \text{Cross}[kk, r_{1to2}]) + (-w_3^2 \cdot r_{2tod3} + w_3 \cdot \text{Cross}[kk, r_{2tod3}]); \]
\[ GF_1 = -m_1 \cdot g \cdot jj; \]
\[ GF_2 = -m_2 \cdot g \cdot jj; \]
\[ GF_3 = -m_3 \cdot g \cdot jj; \]
\[ M_0 = \text{Cross}[r_{0tod1}, GF_1] + \text{Cross}[r_{0to1} + r_{1tod2}, GF_2] + \text{Cross}[r_{0to2} + r_{2tod3}, GF_3]; \]
\[ M_1 = \text{Cross}[r_{1tod2}, GF_2] + \text{Cross}[r_{1to2} + r_{2tod3}, GF_3]; \]
\[ M_2 = \text{Cross}[r_{2tod3}, GF_3]; \]
\[ Hd_0 = m_1 \cdot \text{Cross}[r_{0tod1}, a_1] + m_2 \cdot \text{Cross}[r_{0to1} + r_{1tod2}, a_2] + m_3 \cdot \text{Cross}[r_{0to2} + r_{2tod3}, a_3] + \text{IG}_1 \cdot w_1 \cdot d \cdot kk + \text{IG}_2 \cdot w_2 \cdot d \cdot kk + \text{IG}_3 \cdot w_3 \cdot d \cdot kk; \]
\[ Hd_1 = m_2 \cdot \text{Cross}[r_{1tod2}, a_2] + m_3 \cdot \text{Cross}[r_{1to2} + r_{2tod3}, a_3] + \text{IG}_2 \cdot w_2 \cdot d \cdot kk + \text{IG}_3 \cdot w_3 \cdot d \cdot kk; \]
\[ Hd_2 = m_3 \cdot \text{Cross}[r_{2tod3}, a_3] + \text{IG}_3 \cdot w_3 \cdot d \cdot kk; \]
\[ r_{1b} = \text{FullSimplify}[M_0[[3]] - Hd_0[[3]]] /\text{functransform}; \]
\[ r_{2b} = \text{FullSimplify}[M_1[[3]] - Hd_1[[3]]] /\text{functransform}; \]
\[ r_{3b} = \text{FullSimplify}[M_2[[3]] - Hd_2[[3]]] /\text{functransform}; \]
\[ VSol_2 = \text{Solve}\{r_{1b} == 0, r_{2b} == 0, r_{3b} == 0\}, \{w_1d, w_2d, w_3d\}; \]

\section*{3. COMPARISON OF SOLUTIONS}
\begin{verbatim}
FullSimplify[(Th1"[t] /. \{VSol1[[1, 1]]\}) - (w_1d /. \{VSol2[[1, 1]]\})]
FullSimplify[(Th2"[t] /. \{VSol1[[1, 2]]\}) - (w_2d /. \{VSol2[[1, 2]]\})]
FullSimplify[(Th3"[t] /. \{VSol1[[1, 3]]\}) - (w_3d /. \{VSol2[[1, 3]]\})]
\end{verbatim}

0
0
0
0