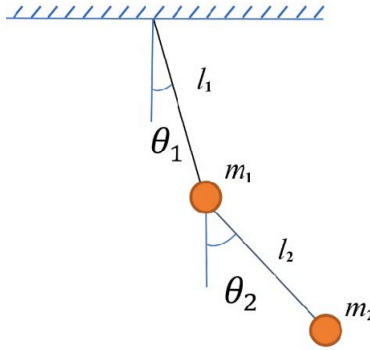


# The Double Pendulum

## MATH1090: Directed Study in Differential Equations

[http://people.sc.fsu.edu/~jburkardt/classes/math1090.2020/double\\_pendulum/double\\_pendulum.pdf](http://people.sc.fsu.edu/~jburkardt/classes/math1090.2020/double_pendulum/double_pendulum.pdf)



We track the angles  $\theta_1$  and  $\theta_2$ ; masses and lengths are parameters

### The double pendulum

The double pendulum system can exhibit sensitivity and chaotic solutions.

## 1 The double pendulum model

In the double pendulum problem, a rod of length  $l_1$  is fixed at one end  $(0,0)$ , and forms an angle  $\theta_1$  with the downward vertical, so that its endpoint is at  $(x_1, y_1) = (l_1 \cos(\theta_1), l_1 \sin(\theta_1))$ . A weight of mass  $m_1$  is attached to this end of the first rod.

A second rod, of length  $l_2$  is also attached to this end of the first rod. It forms an angle  $\theta_2$  with the downward vertical. A weight of mass  $m_2$  is attached to the free end of the second rod. The position of this weight is  $(x_2, y_2) = (x_1, y_1) + (l_2 \cos(\theta_2), l_2 \sin(\theta_2))$ .

Gravity has a force coefficient of  $g$ .

The unknowns are the angles and their time derivatives. We store the unknowns in a vector  $z$ :

$$z = \begin{bmatrix} \theta_1 \\ \frac{d\theta_1}{dt} \\ \theta_2 \\ \frac{d\theta_2}{dt} \end{bmatrix}$$

The differential equations become:

$$\begin{aligned} z'_1 &= z_2 \\ z'_2 &= -\frac{((g(2m_1 + m_2) \sin(z_1) + m_2(g \sin(z_1 - 2z_3) + 2(l_2 z_4^2 + l_1 z_2^2 \cos(z_1 - z_3)) \sin(z_1 - z_3))))}{2l_1(m_1 + m_2 - m_2 \cos(z_1 - z_3)^2)} \\ z'_3 &= z_4 \\ z'_4 &= \frac{(((m_1 + m_2)(l_1 z_2^2 + g \cos(z_1)) + l_2 m_2 z_4^2 \cos(z_1 - z_3)) \sin(z_1 - z_3))}{l_2(m_1 + m_2 - m_2 \cos(z_1 - z_3)^2)} \end{aligned}$$

These derivatives can be evaluated by calling the following function, which is available on the web page.

```
1 dzdt = double_pendulum_deriv ( t, z );
```

Listing 1: double\_pendulum\_data.m returns parameters.

The system has 5 parameters. To start with, we will assign them the following default values:

- $\mathbf{g} = 9.81 \frac{\text{meters}}{\text{second}^2}$
- $\mathbf{m}_1 = 1$  kilogram;
- $\mathbf{m}_2 = 1$  kilogram;
- $\mathbf{l}_1 = 2$  meters;
- $\mathbf{l}_2 = 1$  meter;

The values of the parameters can be reported by calling the following function, available on the web page.

```
1 [ g, m1, m2, l1, l2 ] = double_pendulum_data ( );
```

Listing 2: double\_pendulum\_data.m returns parameters.

Time  $t$  will be measured in seconds, and angles  $\theta$  in radians.

## 2 The system energy

We are already familiar with the fact that when we model a physical system, it can be important to evaluate the energy. If no energy is being added or subtracted from the system, and there's no friction, then a perfect model would report the same energy at all times.

We know that the midpoint method is one example of an ODE solver that can exactly conserve energy, if that energy can be expressed as a quadratic expression in terms of the variables. As it turns out, the energy of the double pendulum system is not expressible this way, and so we can't hope for exact conservation. Nonetheless, we will be interested in seeing how well our various solvers do in at least approximately conserving this quantity.

Here is the somewhat awful expression for the energy  $E$  of the double pendulum, for a given set of variables  $z$ :

$$\begin{aligned}T_1 &= 0.5 m_1 l_1^2 \dot{z}_2^2 \\T_2 &= 0.5 m_2 l_1^2 \dot{z}_2^2 + 0.5 m_2 l_2^2 \dot{z}_4^2 + m_2 l_1 l_2 \dot{z}_2 \dot{z}_4 \cos(z_1 - z_3) \\V_1 &= m_1 g l_1 \cos(z_1) \\V_2 &= m_2 g l_1 \cos(z_1) + m_2 g l_2 \cos(z_3) \\E &= T_1 + T_2 + V_1 + V_2\end{aligned}$$

The value of the energy can be reported by calling the following function, which is available on the web page.

```
1 e = double_pendulum_energy ( z );
```

Listing 3: double\_pendulum\_energy.m returns the energy

## 3 Simulate the small perturbation problem with Forward Euler

We will consider a “small perturbation” problem, for which the double pendulum starts at time  $t = 0$  with the initial condition  $z(t = 0) = [0.25, 0, 0, 0]$ . Thus the first pendulum has been deflected to an angle of

about  $15^\circ$ . We are interested in studying the system over the time period  $0 \leq t \leq 50$ . We start with our forward Euler code, available on the web page as *euler.m*.

The data will be numerous, and complicated. In order to have a chance of comprehending what is going on, we will start by converting the data from angles  $z_1 = \theta_1$ ,  $z_3 = \theta_2$ , to Cartesian coordinates  $(x_1, y_1), (x_2, y_2)$ . Because we are measuring angles starting from the vertical downward direction, these formulas are not quite what you would have expected:

```

1  x1 = l1 * sin(z1);
2  y1 = - l1 * cos(z1);
3  x2 = x1 + l2 * sin(z3);
4  y2 = y1 - l2 * cos(z3);

```

Then we will be interested in the following plots:

1. plot  $(t, x_1(t))$  and  $(t, y_1(t))$ , the coordinates of the end of pendulum 1;
2. plot  $(t, x_2(t))$  and  $(t, y_2(t))$ , the coordinates of the end of pendulum 2;
3. plot  $(x_1(t), y_1(t))$ , the positions of the end of pendulum 1;
4. plot  $(x_2(t), y_2(t))$ , the positions of the end of pendulum 2;
5. plot  $(t, e(t))$ , the energy at each time; include a second reference line displaying  $e = 0$ ;

For small perturbations, plots #1 and #2 should seem roughly periodic and roughly regular. For larger perturbations, where one or both pendulums can actually swing through the vertical upright position, we may expect much more complicated behavior. Plot #3 should be a circle, or a portion of a circle for small problems. Plot #4 should be made by circular paths looping along the circle of plot #1. Plot #5 would be a perfectly straight line if our ODE solver was exactly conservative. Instead, we can only hope that the line stays close to its initial value at  $t = 0$ . In order to judge whether the variations are significant, it's helpful to draw a second line for  $e = 0$ , so that the plot has an implicit scale.

## 4 A code to call `euler()` for the double pendulum

The Forward Euler code uses a fixed stepsize. To get reasonable results over our time interval, it is necessary to use a very large number of steps. Here, we didn't get believable pictures until trying  $n = 100,000$  steps. And this is for the small perturbation problem!

```

1  f = @ double_pendulum_deriv;
2  tspan = [ 0.0, 50.0 ];
3  z0 = [ 0.25; 0.0; 0.0; 0.0 ];
4  n = 100000;
5
6  [ t, z ] = euler ( f, tspan, z0, n );
7  %
8  % Compute Cartesian coordinates.
9  %
10 [ g, m1, m2, l1, l2 ] = double_pendulum_data ( );
11
12 x1 = l1 * sin ( z(1,:) );
13 y1 = - l1 * cos ( z(1,:) );
14 x2 = x1 + l2 * sin ( z(3,:) );
15 y2 = y1 - l2 * cos ( z(3,:) );
16 %
17 % Plot #1.
18 %
19 figure ( 1 );
20 plot ( t, x1, 'g', ...
21       t, y1, 'r', ...
22       'LineWidth', 2 );
23 grid ( 'on' );

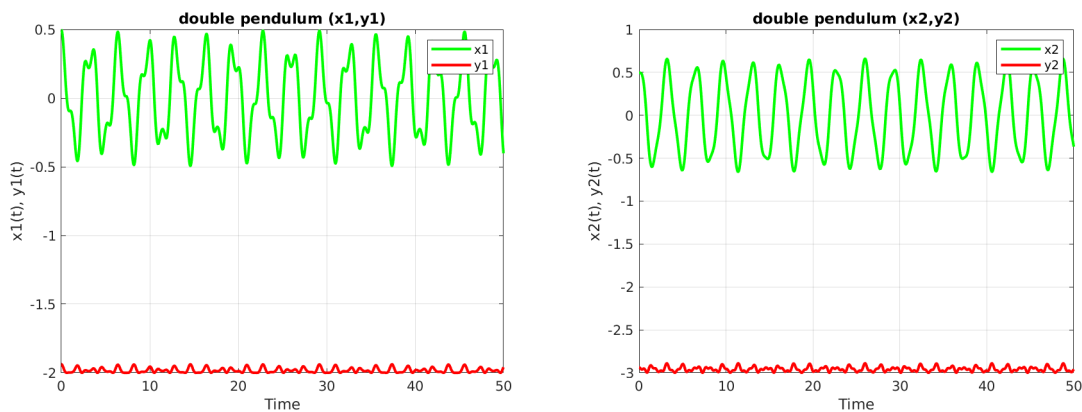
```

```

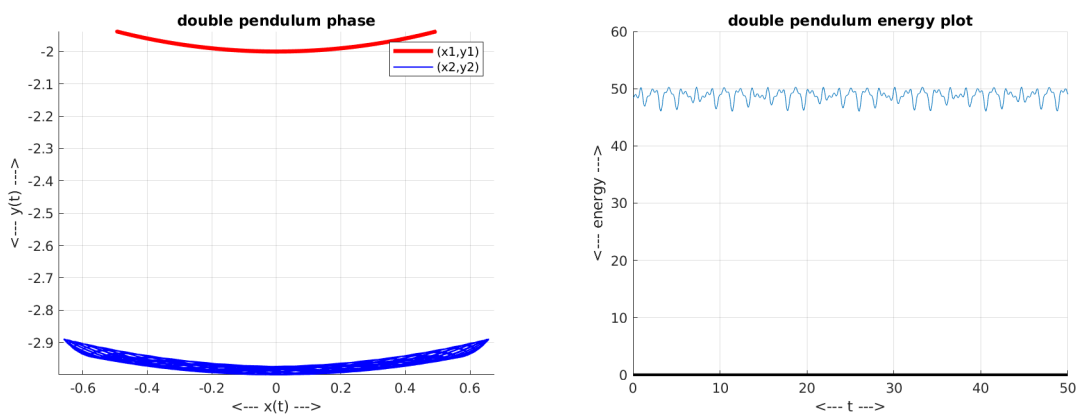
24 xlabel ( 'Time' );
25 ylabel ( 'x1(t), y1(t)' );
26 legend ( 'x1', 'y1' );
27 title ( 'double pendulum (x1,y1)' )
28 filename = 'double_pendulum_x1y1.png';
29 print ( '-dpng', filename );
30 fprintf ( 1, ' Graphics saved as "%s"\n', filename );
31 %
32 % More plots ...
33 %

```

## 5 Graphic results for the small perturbation problem



*Time plots of  $(t, x_1)$ ,  $(t, y_1)$  and  $(t, x_2)$ ,  $(t, y_2)$  show regularity and periodicity*



*The phase plot shows that pendulum #1 stays on a circular arc, and #2 gently loops around it. below it; the energy plot shows approximate conservation.*

## 6 The medium and large perturbation problems

The graphic results from the small perturbation problem suggest that the solution can exhibit regularity and periodicity for both  $(x_1, y_1)$  and  $(x_2, y_2)$ , the positions of the two pendulum masses. We see that the phase portrait is pretty calm, with the first pendulum tracing a thin line, while the thicker line for the second

pendulum mass indicates that its motion is somewhat more complicated. Finally, the energy plot shows that the solver can only attain a “reasonable” amount of energy conservation, but the energy clearly varies, so that our computational model is only approximately tracking the mathematical ideal of perfect energy conservation.

Now let us consider three sets of initial conditions:

1. *small*:  $z = [0.25, 0, 0, 0]$ ;
2. *medium*:  $z = [\pi/3, 0, 0, 2]$ ;
3. *large*:  $z = [\pi, 0, \pi, 4]$ ;

We have already considered the small perturbation case, so it is up to you to run the medium and large perturbations, make the plots, and try to decide what they are telling you.

You may want to check whether the results would change significantly if you increase the number of steps (for a fixed step method) or tighten the error tolerance (for an adaptive method). If the energy conservation improves, for instance, then you might conclude that the first computation was simply not accurate enough. Otherwise, you might wonder whether this ODE system can be too wild to approximate accurately.

## 7 An animation option

There is a MATLAB program, written by Alexander Erlich, which can solve the double pendulum problem, display the pendulum system, and make a movie of the results. Watching the animation for various perturbation sizes, it is easy to see how the system can be regular and rhythmic for small perturbation, becoming chaotic when more energy is available, so that the pendulums can both swing full circle.

On the course website, there is a copy of the program, set up to solve the small perturbation problem, in the file *double\_pendulum\_movie\_small.m*, as well as the resulting animation *double\_pendulum\_movie\_small.avi*. To change the code to do the medium or large perturbation problems only requires changing the data in the initial segment of the code.

In that initial code, you can also easily see how to choose different values for the gravitational, mass, and length parameters, or to experiment with other initial conditions.

## 8 REPORT

We have presented all the theoretical material and examples that we wanted. In the remaining time, we are asking you to do some independent investigation and prepare a final *LaTeX* report. You don't have to repeat everything we have talked about, but you should be able to make an intelligent discussion of the issues we have looked into for solving differential equations.

Unless you find another problem that is more interesting, I would suggest that you concentrate on this double pendulum problem, and try out some of the techniques we have discussed. You can take a look at the Wikipedia page about the double pendulum problem to get some more background and references. The equations are pretty complicated, but we can still approximate solutions using the methods we have discussed.

A suitable report might consider some (but not all!) of the following items:

1. Describe the problem with picture and equations;
2. Solve the small perturbation problem with the forward Euler method, and explain why you need so many steps to get a reasonable answer.

3. Discuss the fact that the solution is easy to approximate for low energy, but becomes very sensitive and irregular for higher energy.
4. Define a “good” approximation as one for which the energy is never more than 10% above or below the initial energy;
5. Look at how, when using a fixed step size, a higher order method gives a “good” approximation using much fewer steps.
6. Try an adaptive stepsize method like *rk12\_adapt.m*, and describe what you have to do to get a “good” approximation.
7. Consider the “medium” or “large” perturbation problems. Try to get a “good” approximation. (This may not be possible.) Use plots to show that the solution behaves much differently than it did for the small perturbation.
8. Compare the midpoint method and a comparable second order fixed-step method like *rk2*, and plot the energy over time in both cases. Does the midpoint method do a better job of energy conservation?

Please work on this report over the next three weeks, and check in with me or Professor Trenchea each week. Your reports will be due by April 17, 2020.