

Math/AppM 4650, Spring 2008  
Chain Project

The model is  $n$  particles in the plane  $\mathbb{R}^2$ :  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ . They are joined by rigid links, each having length  $L/n$ , forming a chain of total length  $L$ . The chain is suspended from the origin  $(0, 0)$ . So there are  $n$  constraint equations describing the links:

$$\begin{aligned} x_1^2 + y_1^2 &= L^2/n^2 \\ (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 &= L^2/n^2, \quad \text{for } 2 \leq i \leq n \end{aligned} \quad (1)$$

Let's assume the links of the chain have no mass, but the joints each have mass  $M/n$  (so that the total mass of the chain is  $M$ ). We assume there is a constant force due to gravity of  $Mg/n$  acting on each joint. If the chain is hanging down, then  $g = 32$  feet-per-second-squared or  $g = 9.8$  meters-per-second-squared. If the chain is lying on a table, then  $g = 0$ . Ignoring friction and hoping there are no self-collisions, the equations of motion are (using  $x_0 = 0, y_0 = 0$ ):

$$\begin{aligned} \frac{d^2 x_i}{dt^2} &= n^2 \sigma_{i+1} (x_{i+1} - x_i) + n^2 \sigma_i (x_{i-1} - x_i), \quad \text{for } 1 \leq i \leq n-1 \\ \frac{d^2 x_n}{dt^2} &= n^2 \sigma_{n-1} (x_{n-1} - x_n), \\ \frac{d^2 y_i}{dt^2} &= n^2 \sigma_{i+1} (y_{i+1} - y_i) + n^2 \sigma_i (y_{i-1} - y_i), \quad \text{for } 1 \leq i \leq n-1 \\ \frac{d^2 y_n}{dt^2} &= -g + n^2 \sigma_{n-1} (y_{n-1} - y_n). \end{aligned} \quad (2)$$

Here the numbers  $\sigma_1, \sigma_2, \dots, \sigma_n$  are the tensions in the chain, which are different in each link but always act in the direction of each link. Mathematically, they are the Lagrange multipliers corresponding to the constraints (1).

If we write

$$(x_1, y_1) = \frac{L}{n} (\cos \theta_1, \sin \theta_1), \quad (x_i, y_i) = (x_{i-1}, y_{i-1}) + \frac{L}{n} (\cos \theta_i, \sin \theta_i), \quad \text{for } 2 \leq i \leq n, \quad (3)$$

then the evolution equations (2) simplify a bit to

$$\begin{aligned} \frac{d^2 \theta_1}{dt^2} &= -\frac{ng}{L} \cos \theta_1 + n^2 \sigma_2 \sin (\theta_2 - \theta_1) \\ \frac{d^2 \theta_i}{dt^2} &= n^2 \sigma_{i+1} \sin (\theta_{i+1} - \theta_i) - n^2 \sigma_{i-1} \sin (\theta_i - \theta_{i-1}) \quad \text{for } 2 \leq i \leq n-1 \\ \frac{d^2 \theta_n}{dt^2} &= -n^2 \sigma_{n-1} \sin (\theta_n - \theta_{n-1}). \end{aligned} \quad (4)$$

We still need to know what the tensions  $\sigma_1, \dots, \sigma_n$  are. The trick for finding them is to differentiate the constraint equations (1) twice with respect to time, and then plug in the

equations (2). This gives us the following equations:

$$\begin{aligned} \sigma_1 - \cos(\theta_2 - \theta_1)\sigma_2 &= \frac{\dot{\theta}_1^2}{n^2} - \frac{g}{nL} \sin \theta_1 \\ -\cos(\theta_{i+1} - \theta_i)\sigma_{i+1} + 2\sigma_i - \cos(\theta_i - \theta_{i-1})\sigma_{i-1} &= \frac{\dot{\theta}_i^2}{n^2}, \quad \text{for } 2 \leq i \leq n-1 \\ 2\sigma_n - \cos(\theta_n - \theta_{n-1})\sigma_{n-1} &= \frac{\dot{\theta}_n^2}{n^2}. \end{aligned} \quad (5)$$

Notice that this is a *tridiagonal* system for  $\sigma$ , so there is an efficient way to solve it given  $\theta$  and  $\dot{\theta}$ .

Defining

$$\omega_i = \frac{d\theta_i}{dt}, \quad \text{for } 1 \leq i \leq n, \quad (6)$$

the  $n$ -dimensional second-order system (4) becomes the  $2n$ -dimensional first-order system

$$\begin{aligned} \frac{d\theta_i}{dt} &= \omega_i, \quad \text{for } 1 \leq i \leq n, \\ \frac{d\omega_1}{dt} &= -\frac{ng}{L} \cos \theta_1 + n^2 \sigma_2 \sin(\theta_2 - \theta_1) \\ \frac{d\omega_i}{dt} &= n^2 \sigma_{i+1} \sin(\theta_{i+1} - \theta_i) - n^2 \sigma_{i-1} \sin(\theta_i - \theta_{i-1}) \quad \text{for } 2 \leq i \leq n-1 \\ \frac{d\omega_n}{dt} &= -n^2 \sigma_{n-1} \sin(\theta_n - \theta_{n-1}), \end{aligned} \quad (7)$$

with  $\sigma$  determined by solving the  $n \times n$  matrix

$$\begin{pmatrix} 1 & -\cos(\theta_2 - \theta_1) & 0 & \dots & 0 & 0 \\ -\cos(\theta_2 - \theta_1) & 2 & -\cos(\theta_3 - \theta_2) & \dots & 0 & 0 \\ 0 & -\cos(\theta_3 - \theta_2) & 2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -\cos(\theta_n - \theta_{n-1}) & 2 \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_n \end{pmatrix} = \begin{pmatrix} \omega_1^2/n^2 - \frac{g}{nL} \sin \theta_1 \\ \omega_2^2/n^2 \\ \vdots \\ \omega_n^2/n^2 \end{pmatrix}. \quad (8)$$

Once we solve (numerically) for  $\theta_1(t), \dots, \theta_n(t)$ , we then use the equations (3) to get  $x_1(t), \dots, x_n(t)$  and  $y_1(t), \dots, y_n(t)$ .

The main problem is to figure out what happens to this system as  $n \rightarrow \infty$ , when the “chain” approaches a “whip.” Although the equations certainly have solutions defined for all time for any particular  $n$ , because they are ordinary differential equations, in the limit they become a partial differential equation, and it is not known whether this partial differential equation actually has solutions defined for all time.

This problem is a simple analogy for the motion of an ideal, incompressible fluid: the length-preserving constraint (1) is analogous to the requirement that the fluid preserve volume, while the system (8) is analogous to the differential equation that determines the pressure of the incompressible fluid. The observation that the chain eventually “crinkles” is analogous to the development of turbulence in an incompressible fluid, and the question of whether solutions exist for all time, in fluid mechanics, is one of the most famous and long-standing problems in the field.

For the following project, you will probably want to divide up the tasks: one person good with formulas to do the first part; one person to program the system and make sure it's working; one person to actually run the simulations on a fast computer and assemble the output into pictures and movies; one person to (optionally) arrange experiments and measurements for a real chain; and one person to write it all up at the end. (This is just one suggestion; feel free to divide up the work in any way that's convenient for you.)

1. Explore the limit of the equations as  $n \rightarrow \infty$ : write  $\theta_i = \theta(\frac{Li}{n})$  and  $\sigma_i = \sigma(\frac{Li}{n})$ , with  $s = \frac{i}{n}$ . Using Taylor approximations such as

$$\sigma_{i+1} \approx \sigma(s + L/n) = \sigma(s) + \frac{L}{n}\sigma'(s) + \frac{L^2}{2n^2}\sigma''(s),$$

show that equation (4) approaches the partial differential equation

$$\frac{\partial^2 \theta}{\partial t^2}(t, s) = L^2 \left( \sigma(t, s) \frac{\partial^2 \theta}{\partial s^2}(t, s) + 2 \frac{\partial \sigma}{\partial s}(t, s) \frac{\partial \theta}{\partial s}(t, s) \right).$$

Show the boundary condition at the fixed point must be

$$\sigma(t, 0) \frac{\partial \theta}{\partial s}(t, 0) = \frac{g \cos(\theta(t, 0))}{L^2} \quad (\text{fixed!})$$

to ensure that  $\frac{\partial^2 \theta}{\partial t^2}(t, 0)$  is finite.

Similarly show that the tension given by (5) approaches the solution of the differential equation

$$\frac{\partial^2 \sigma}{\partial s^2}(t, s) - \left( \frac{\partial \theta}{\partial s}(t, s) \right)^2 \sigma(t, s) = -\frac{1}{L^2} \left( \frac{\partial \theta}{\partial t}(t, s) \right)^2,$$

and that the boundary conditions must be

$$\frac{\partial \sigma}{\partial s}(t, 0) = \frac{g}{L^2} \sin(\theta(t, 0)) \quad \text{and} \quad \sigma(t, L) = 0.$$

2. Write a fast program to solve for  $\sigma$ , given  $\theta$  and  $\omega$ , using the algorithm in the book for tridiagonal systems. Check your program for  $n = 3$  by hand to make sure it's giving correct values.
3. Write a program to use Euler's method to solve the system, with some initial conditions (see below for examples). Choose a particular  $n$  (for example,  $n = 20$ ), and figure out the smallest time step that gives a good approximation. Test it by repeating the entire calculation with half of that time step to make sure both  $(x, y)$  curves are approximately the same over some time interval.
4. Use a higher-order method to get a better approximation with a smaller time step. (Since the function is very complicated, you want to avoid Taylor methods; the derivatives are a pain to compute. Instead use Runge-Kutta style methods.) Since this is tricky to program, check your model against some reliable data from the Euler method above to make sure your program is working correctly.

5. Once you have some programs that can be run with any value of  $n$  and any time-step, start doing simulations with progressively higher values of  $n$ . (Most likely, every time you increase  $n$ , you'll have to decrease the time step to get good accuracy. Check your accuracy by running the program again with the time step halved, and make sure they both do the same thing.)

There are several things you can try simulating. (You can do one or more of these, or make up your own initial condition that does something interesting.)

- A chain extended horizontally, and then released to fall (with nonzero gravity). So the initial conditions are  $\theta_i = 0$  and  $\omega_i = 0$  for all  $i$ . The chain usually swings once to the other side smoothly, and when it comes back toward where it started, it starts to crinkle. We're especially interested in what happens as it crinkles, so make sure your approximation goes that far.
- A falling chain (again, with nonzero gravity). If  $n$  is odd, so that  $n = 2m + 1$  for some integer  $m$ , then you can simulate this with initial conditions  $\theta_i = -\frac{\pi}{2}$  for  $i \leq m$ ,  $\theta_{m+1} = 0$ , and  $\theta_i = \frac{\pi}{2}$  for  $i \geq m + 1$ . All  $\omega_i$  are zero at the initial time (because you're just letting the chain drop freely). An interesting feature of this situation is that the acceleration of the chain's free tip is *larger* than the gravitational acceleration. When the tip reaches the bottom, the chain bounces and starts to crinkle.
- Instability of the inverted chain. If you start the system with the chain held directly above the fixed point, it's an equilibrium. However, it's obviously *very* unstable, and if any one of the  $\theta_i$  is not *exactly*  $\frac{\pi}{2}$  (corresponding to being vertical), then the chain will quickly crinkle. Is the crinkling predictable in any way? (To be systematic, you could start with  $\theta_i = \frac{\pi}{2}$  for  $i < n$ , and  $\theta_n = 0$ , with all  $\omega_i = 0$  initially.)
- A stable chain? Start with some initial condition like  $\theta_i = -\frac{\pi}{4}$  for all  $i$ , where the chain would *not* be expected to crinkle. Run the simulation for a *very* long time and see if the solution does remain smooth.
- The whip. All of the above simulations involved the effects of gravity. For a harder problem, you can try making gravity zero, to simulate a chain moving on a frictionless plane. See if you can find the right initial conditions (with  $\omega$  initially nonzero, or else the chain will never move) so that you simulate a cracking whip: a small initial velocity of the entire string that eventually focuses to give a very high velocity of the tip. (A whip cracks because the free end of it breaks the sound barrier.)

For any of these simulations, one would ideally want animations on the same time-scale displaying the chains for different values of  $n$ . For example, starting with  $n = 10$ , then  $n = 20$ , then  $n = 40$ , etc., and seeing if there is any kind of convergence as  $n$  gets larger.

Other things worth investigating for any of the above initial conditions (or others that you might create):

- Does the tension  $\sigma$  always remain positive? Physically we might expect that each portion of the chain is always trying to stretch and being pulled back, but if  $\sigma$  were negative anywhere, it would mean the chain is compressing itself. In the limit of a whip, negative tension would be very surprising physically, but the equations don't rule it out.
  - What happens at the tip? The velocity  $\omega_n$  can theoretically approach infinity as  $n$  approaches infinity; do you see it doing that? Is there any relation between crinkling of the chain and high velocity at the tip?
  - What do the graphs of  $\theta$  and  $\frac{\partial\theta}{\partial t}$  look like? Do oscillations in those graphs diminish as  $n$  gets larger?
  - Of course, anything else you might want to explore would also be interesting.
6. This next bit is optional (you could certainly get an A on the project without doing it), but if you're very interested in this problem, you might try doing a real-life experiment. Take a chain (a necklace, or a swing, or a bicycle chain, or whatever you can think of), set it up in the same initial conditions as your numerical simulation above, and videotape the motion of it. See if the numerical simulation corresponds to the actual motion.

You can also try chains with a different number of links to see what happens as the number of links increases. Be careful that the motion remains planar; if the chain is allowed to twist at all in space, then all the equations change. Also be careful not to get hit by the chain: remember that the tip speed can get very high, and if you let the chain hit you, it *will* hurt!

7. As you can see, this is a very open-ended project. There are many things you can try, and there are many ways to get a good grade. What is important is that you analyze the equations correctly, that your programs work correctly, and that you have a good systematic way of doing your simulations, so that your data is trustworthy. There's no "correct" result to discover; just try to find something and demonstrate convincingly that it's true. In your final project report, describe clearly your programming methods, your simulation data, and your conclusion about your numerical experiment. What would you be willing to say with confidence about the limit as a chain approaches a whip, based on your simulations?