

18.085, PROBLEM SET 4, DUE 7/15 IN CLASS

Final version. Please note that you should take string stiffness to be 1 everywhere, except where explicitly noted otherwise. This means that the matrix C , which gives us the constitutive law, should be the identity matrix unless noted otherwise. Wording of (h) has changed slightly. Notice line 10 of the code is useless, I just forgot to get rid of it - it certainly does no harm.

In order to save figures so you can print them, I strongly recommend you use the function `save2pdf`, available for free download here:

<http://www.mathworks.com/matlabcentral/fileexchange/16179-save2pdf>.

Save it in the same directory you use for your Matlab code. When you are ready to save a figure, click on that figure to make sure Matlab will print that one (and not another in case you have multiple figures open). Then, go to Matlab's command line and type `save2pdf('myfilename.pdf')` and a small pdf figure should have been saved to your current directory, with file name *myfilename.pdf*.

Finally, for some questions you might find it useful to copy a line of code you used for one question, comment out one of those copies (to comment out something, put a % in front of it), and then modify the other copy to answer the next question. That way, you can keep your work in case you want to go back, and you don't have to print out a different version of your code for each question for me to grade.

Question 1. (100 pts.)

In this exercise, we will apply the leap-frog method of finite differences in time to solve the wave equation for the motion of a string for $x \in [0, 1]$. That is, we solve $u'' = \frac{\partial^2 u}{\partial x^2} + F$, where $u = u(x, t)$ is the height of the string at some horizontal position x and at some time t . The function $F = F(x, t)$ is some exterior forcing, if applicable.

We know we can discretize in space using finite differences in space as we have done before, or using the K matrix: $K/h^2 = A^T C A/h^2 \approx -\frac{\partial^2}{\partial x^2}$, where h is our spatial step size. Let n be the number of points at which we evaluate u in the x direction, excluding the end-points, then $h = \frac{1}{(n+1)}$ and the column vector U contains the n approximate values of u in space ($U_j(t) \approx u(hj, t)$ for $j = 1, 2, \dots, n$), that is $U(t) = (U_1(t), U_2(t), \dots, U_n(t))^T$. So we want to solve $MU''(t) = -KU(t) + F$, where now $F = (f(h, t), f(2h, t), \dots, f(1-h, t))^T$.

For simplicity, we assume the mass of the string is equally distributed, and so we let M be the identity matrix. So we will solve $U''(t) = M^{-1}(-KU(t) + F)$, $t \in [0, T]$ or

$$(0.1) \quad U''(t) = -KU(t) + F, \quad t \in [0, T]$$

using leap-frog as we have seen in class. Please find the `pset4.m` file on the website, you will have to modify this file, but *do not* modify a line unless it has `%user` in it.

- a) Let $n = 15$, $T = 1$, we want fixed-fixed ends and a forcing $f = 0$. Modify lines 5, 8, 14 of the code accordingly (verify lines 11-12 do what they should do for f). Lines 8, 9 mean that $\delta t = h/2$, which is stable. Notice that lines 18-20 construct matrices A and C for the fixed-fixed case, and lines 49-50 put back in the 0 boundary conditions. We want initial conditions of $u(x, 0) = 1/2 - |x - 1/2|$ and $u'(x, 0) = 0$, so check lines 35, 37. Finally, change line 43 so that the leap-frog method is used to solve (0.1). Run the code, make sure the solution looks like it should (in particular,

the initial and boundary conditions) and save and print the last figure that the code shows (solution at time T). You might want to close the figure window between each run of the program.

- b) *Stability.* Try now $nt = (n + 1)$ and $nt = (n + 1) - 1$ on line 8. Describe in a few words what happens to the solution in each case (no need for plots), comparing to when $nt = (n + 1) * 2$.
- c) *Standing waves.* Let $n = 31$. Now we want to test a different initial condition, that is, $u(x, 0) = \sin(k * 2\pi x)$, where k is a positive integer. Try a few different values of $k < (n + 1)/2$ (remember you need enough points per wavelength as we briefly discussed in class, otherwise you miss out on oscillations), and describe what happens in a few words (no need for plots). There are 2 distinct things to notice!
- d) *Traveling waves.* Let $n = 31$. Now we want to test a different initial condition, that is, $u(x, 0) = .5 * e^{(-100*(x-.25)^2)}$. This is a bump! Describe in a few words what happens to the bump in time (again, no need for plots).
- e) *Free-fixed end.* Make the left end free. That is, change bc to 1 in line 14, and change lines 22-24 accordingly. Also, notice how lines 52-53 put back in the appropriate boundary conditions. Run a simulation with $nt = (n + 1) * 2$, $n = 15$ again. Make sure what happens makes sense. Then, try $n = 63$ (less error but takes more time), and now print the last figure, solution at time T for $n = 63$.
- f) *Free-free end.* Make the right end free too. That is, change bc to 2 in line 14, and change lines 26-28 accordingly. Also, notice how lines 55-56 put back in the appropriate boundary conditions. Run a simulation with $nt = (n + 1) * 2$ again. Make sure what happens makes sense. Then, try $n = 63$ (less error but takes more time), and now print the last figure, solution at time T for $n = 63$.
- g) *Forcing and resonance.* Assume someone is holding the left end of the string and making it oscillate. Then we still have fixed ends, hence bc=0, but now instead of $u(0, t) = 0$ as a boundary condition we have $u(0, t) = .25 * \sin(4 * 2\pi t)$. Show the solution at time T .

Notice how the solution seems to grow in time! This is what is called resonance. (You don't have to do this, but you can try setting say $n = 63$, $T = 5$ and $nt = 5 * (n + 1) * 2$ and check that it keeps growing.)

- h) *Different stiffnesses.* Now modify line 19 so that we have two strings put end to end, one of stiffness .5 (for $x \in [0, 1/3]$) and one of stiffness 2.5 (for $x \in [1/3, 1]$). Remember that changing the matrix C when you have a forcing means you also need to change slightly the forcing in line 43, so in the time-stepping part. Run the same simulation as in (g), and describe the difference between what happens on the left string and what happens on the right string, in a few words (no plots). Make sure you explain this difference by mentioning the different stiffnesses. You can try different stiffnesses to help you figure it out *if you feel like it*, but then, make sure you pick a dt small enough for stability (remember stability changes with C !).