

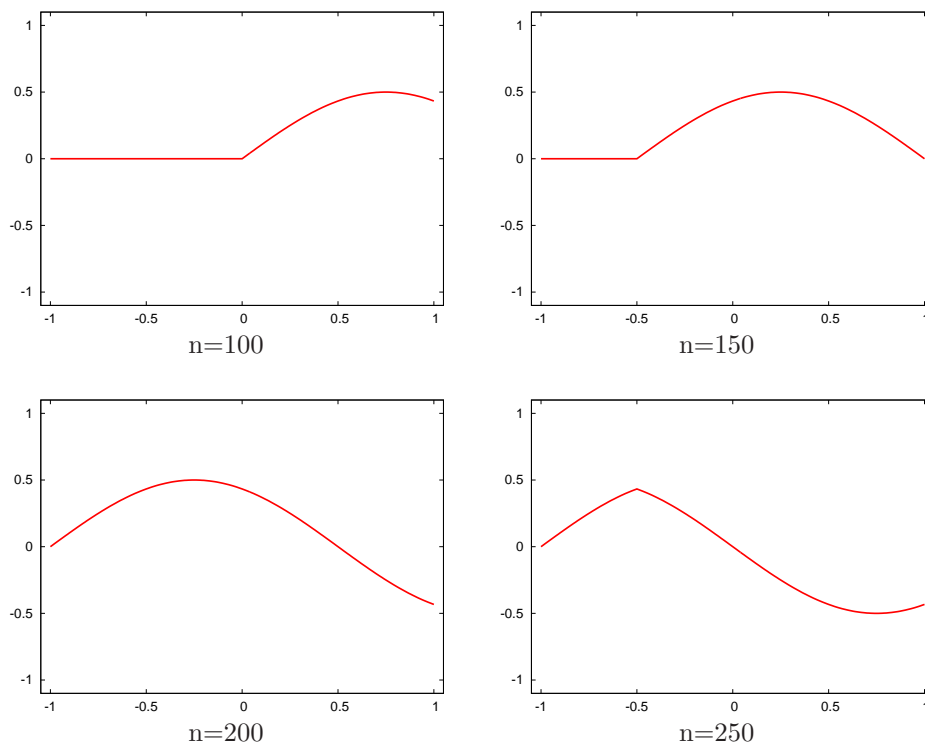
Homework IV : Partial differential equations and traveling waves on strings

Guidelines for Homework IV : Please read carefully!

1. Homework IV is due Thursday, 22/11 by 22:00.
2. Homework will be submitted via email to ustunel@metu.edu.tr.
3. **VERY IMPORTANT!** The subject line of your email MUST read Phys343 Homework – nothing more, nothing less.
4. **VERY IMPORTANT!** Homework turned in between 22:00 on 22/11 and 8:00 on 23/11 will only receive 50% of the full credit. Homework turned in later than that will NOT be accepted.
5. I'm expecting four .m files from you. PLEASE DO NOT SEND WORD DOCUMENTS.

Goal of the assignment : In this assignment, you will modify the script we wrote in class to explore different boundary conditions.

Question 1 (50 points): In this question, you will study the effect of driving the string periodically at one end keeping the other end fixed. The time evolution of the profile might look something like this :



Download the files `propagate.m` and `string_fixed.m` from the web site. Copy `propagate.m` to a new file called `propagate_driven.m` and copy `string_fixed.m` to a new file called `string_driven.m`. Modify `propagate_driven.m` and `string_driven.m` in the following way :

1. In `string_driven.m` instead of the Gaussian, now assign the initial profile to zeros such that at all times before the simulation the string is at rest with all of its elements at zero.
2. `propagate_driven.m` should keep one end of the string fixed while it drives the other end with a sinusoidal driving force such that it oscillates as follows $y = A \sin(\omega t)$. Modify `propagate_driven.m` such that it now takes the angular velocity of the periodical driving force ω , amplitude of oscillation A , time index n and the time step dt . In short, your new function declaration should be

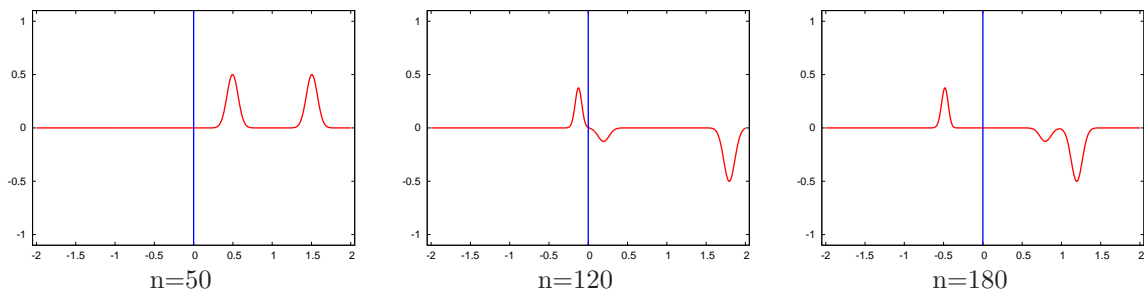
```
function ynext=propagate_driven(ynow,yprev,r,omega,A,n,dt)
.
.
.
endfunction
```

3. In `string_driven.m`, instead of `propagate`, call the new function `propagate_driven` with the following constants.

Property	Symbol	Value
Rate	<code>rate</code>	100 cycles/sec
Angular frequency	<code>omega</code>	To be obtained from <code>rate</code>
Amplitude	<code>A</code>	0.5 m

Send me both `string_driven.m` and `propagate_driven.m`.

Question 2 (50 points): In this question, you are going to see the reflection and transmission of waves going from strings with different speeds of sound or equivalently different mass densities. The time evolution should look something like this



where the left-hand side is the heavier part and the right-hand side is the lighter part. Download the codes `initial_profile.m`, `propagate.m` and `string_fixed.m`. We are now going back to the Gaussian initial profile.

Copy `propagate.m` into a new file called `propagate_two_parts.m` and `string_fixed.m` into a new file called `string_two_parts.m`. Assume that the string is made up of two parts where the first part has a sound velocity $c_1 = 300$ m/sec and the second part has a sound velocity $c_2 = 500$ m/sec. In other words, the second part of the string is *lighter* than the first part. The two parts have equal length, which means that the point where they meet is in the middle. Make the following modifications :

1. Instead of a single `r`, `propagate_two_parts.m` should now take as input two `rs`, `r1` and `r2`, each corresponding to one of the two parts of the string.
2. In the constants section in `string_two_parts`, you now have to choose a single `dt` for two different velocities. There are two available choices, `dt1=dx/c1` or `dt2=dx/c2`. Make the appropriate choice. Be careful that the choice is related to the maximum of c_1 and c_2 . You may use the `max` function in Octave to help you make this choice. Because there are two different c s there are also two different `rs`. Define `r1` and `r2`.
3. Divide the `for` loop inside `propagate_two_parts.m` into two : one before the midpoint and one after the midpoint. For those points before the midpoint use `r1` to obtain the profile in the next step and for those after the midpoint use `r2`.
4. Change the center point of the initial Gaussian. First create it in the lighter part and then the heavier part. You may increase the length of the string if it's too short to completely fit your Gaussian. Run your codes and see if you get the expected result.

Note that after you run your code for a while you see some trailing ripples before and after the wavefront. Why does this happen?

Send me both your `propagate_two_parts.m` and `string_two_parts.m` files.