18.085, PROBLEM SET 4, SOLUTION

See the webpage for the solution code, with all you need to try out the solutions below.

Question 1. (100 pts.)

a) Relevant code (no worries if you didn't put in $Minv$ or F) and picture:

```
n=15; %user, number of points in x (or number of masses)
...
U0=.5-abs(x'-.5); % user, u(x,0)
...
U(:,i)=2*U(:,i-1)-U(:,i-2)+dt^2*Minv*(-K*U(:,i-1)+F(:,i-1)); % user, leap-frog step
```

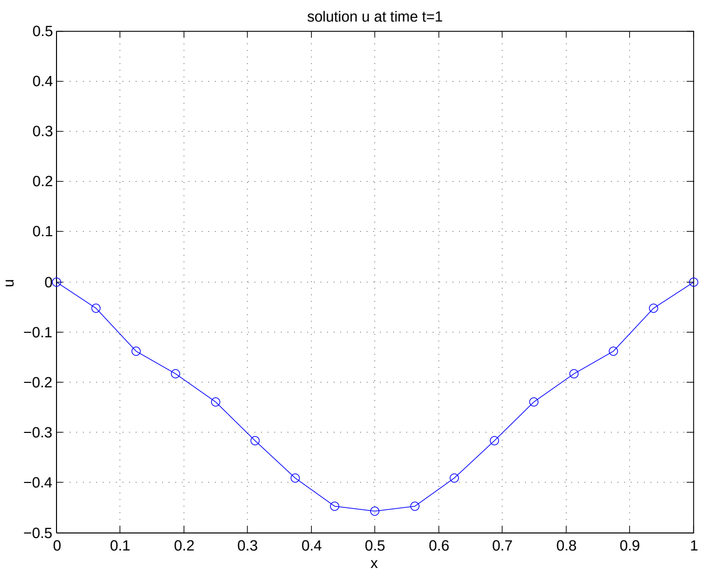

FIGURE 1. Solution for (a).

b) *Stability.* In (a), we got something that looked like how a string would behave if plucked in the middle. With $nt = (n + 1)$, we get something a little strange: the string moves as if it was made of bars with hinges, which does not look very natural, but at least after time 1 the strings ends up in the exact same position as at time 0, but with a negative sign. For $nt = (n + 1) - 1$, the solution blows up very rapidly: we must have gone over the stability limit, and the solution grows in time. (Indeed, the stability limit here is $dt \leq h$.

c) *Standing waves.* Relevant code: `U0=sin(4*2*pi*x'); % user, u(x,0)`. We notice two things. First, we obtain standing waves, that is, the same pattern in x will be repeated, and the so string comes back periodically to (approximately) the same position it was in before. Secondly, the frequency at which the pattern comes back

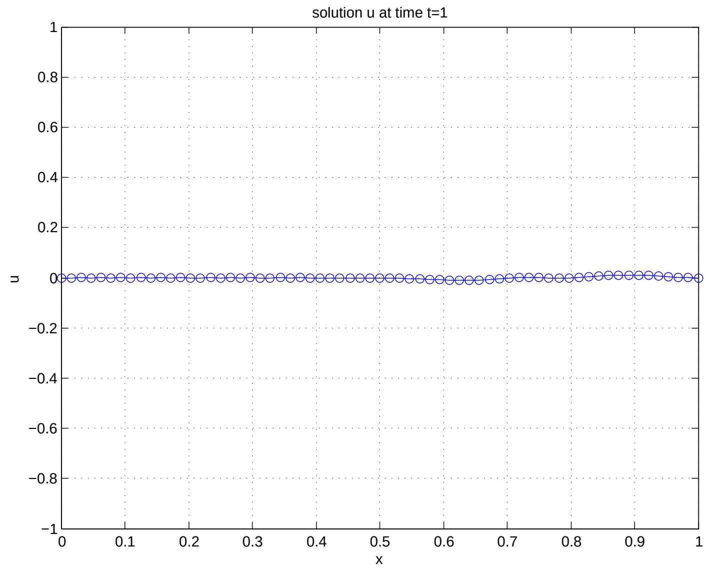

FIGURE 2. Solution for (e).

to its original position (so that's a frequency in time) is the same as its frequency in space. So if $k = 4$ as above then the pattern comes back 4 times inside a unit time interval (there is a phase error, but that's normal, no method is perfect, even leap-frog). It was ok to just say that the pattern oscillated faster in time if k was larger. We will see why in the next pset.

d) *Traveling waves.* The bump splits up in 2 halves of half the amplitude of the original bump (conservation of energy!), each half goes its own way. When they hit the walls, they reflect down (this is what a fixed end means physically for a string: reflection with a minus sign for the amplitude) and come back together again below to form the original bump again, so that again we notice conservation of energy.

e) *Free-fixed end.* I did it this way:

```
xhalf=3*h/2:h:1-h/2; %user, this is where the "springs" are
C=spdiags(xhalf'.^0,0,length(xhalf),length(xhalf)); %user, constitutive law
A=spdiags([ones(length(xhalf),1) -ones(length(xhalf),1)],[1 0],length(xhalf),n);
```

which means I got rid of the left-most spring: then vector X has one less entry, and C and A are n by n now. Some of you copied code from $bc = 0$ but put the $(1, 1)$ entry of either C or A equal to 0, which works too. We get figure 2, which makes sense because now the left-going bump reflects now with the same amplitude (no minus sign) because this is what a free end does: reflection with same amplitude. And now the two bumps (one above and one below), will cancel each other when they meet again.

f) *Free-free end.* Relevant code (removed the last spring - again, you might have done this differently):

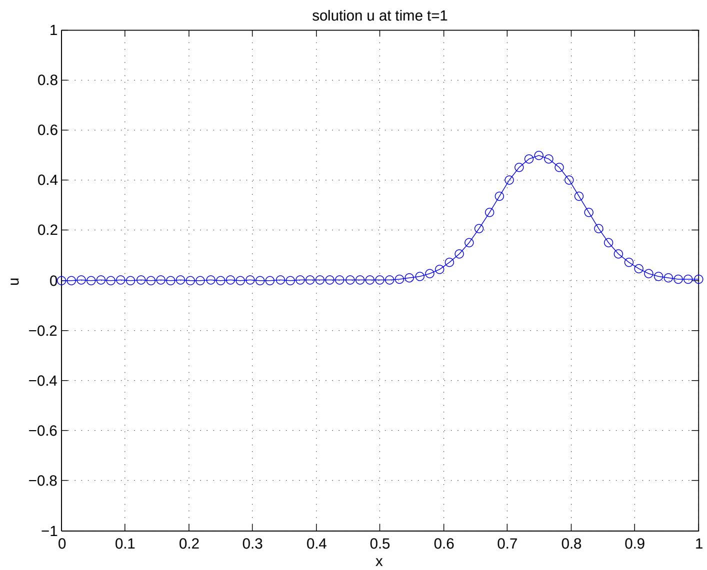

FIGURE 3. Solution for (f).

```
xhalf=3*h/2:h:1-3*h/2; %user, this is where the "springs" are
C=spdiags(xhalf'.^0,0,length(xhalf),length(xhalf)); %user, constitutive law
A=spdiags([ones(length(xhalf),1) -ones(length(xhalf),1)],[1 0],length(xhalf),n); % user,
```

And now, two free ends means the bumps reflect but neither switches sign, so that when they meet they grow back into the original bump, but towards the other end of the string: figure 3.

g) *Forcing and resonance.* Relevant code: `F(1,:)=.25*sin(2*pi*(0:dt:T)*4);`. Also, you could have done either of the following:

```
U(:,i)=2*U(:,i-1)-U(:,i-2)+dt^2*Minv*(-K*U(:,i-1)+(1/h^2)*A'*C*[F(:,i-1);0]);
% make F one row bigger since it corresponds to elongations (don't forget 1/h^2)
U(:,i)=2*U(:,i-1)-U(:,i-2)+dt^2*Minv*(-K*U(:,i-1)+[1;zeros(n-1,1)].*(K*F(:,i-1)));
% use K but make sure you take only the first row of the result
U(:,i)=2*U(:,i-1)-U(:,i-2)+dt^2*Minv*(-K*U(:,i-1)+(1/h^2)*C(1,1)*F(:,i-1));
% put the force balance in by hand
```

You might have done something else too. Either way, we see the sinusoidal forcing making its way across the string in time. If did increase T, then you would see the amplitude increase more and more. We are forcing the system at one of its natural frequencies (it has many, as we'll see when we do Fourier bases). Solution at $T = 1$: you might have used 0 initial conditions (figure 4), or the previous bump (see figure 5), that's fine.

h) *Different stiffnesses.* This is how I did it:

```
C=spdiags(.5+2*(xhalf'>1/3),0,length(xhalf),length(xhalf)); %constitutive law
```

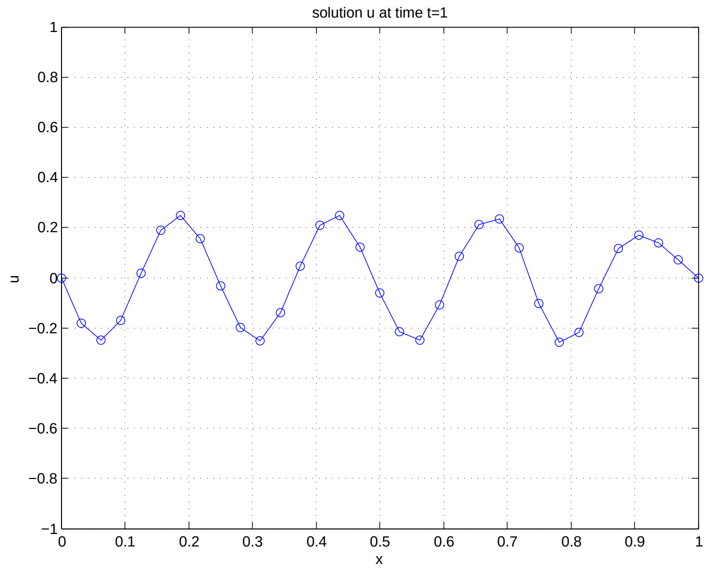

FIGURE 4. Solution for (g), 0 initial conditions.

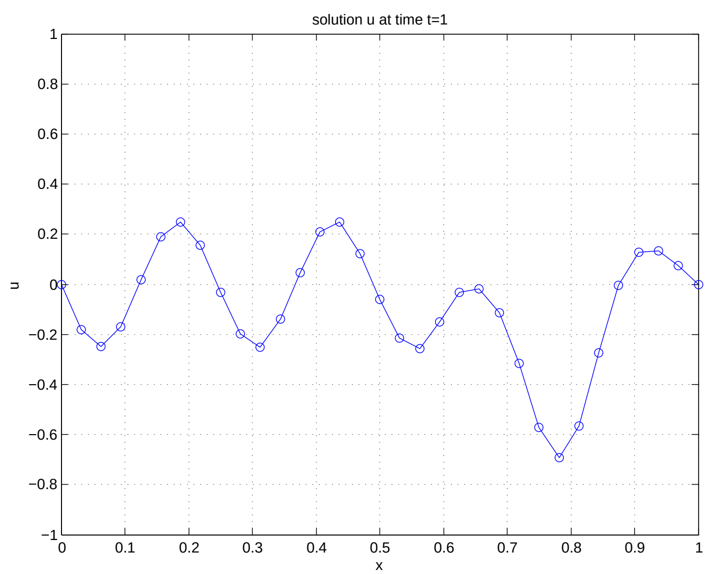

FIGURE 5. Solution for (g), bump initial conditions.

With the code from part (g), no need to change the leap-frog time-stepping, but depending on your answer maybe you had to. It was easier to see what happened with 0 initial conditions, but you can still see it with the bump too. Basically, the string on the right is stiffer, so we expect it to resist bending more, and in particular, it's frequency of oscillation in space is lower than that of the left spring. What I mean is, if you fix time and look at the string, you see about as many oscillations on the left part as on the right, but the right part is twice as long, which means its oscillations have longer period (smaller frequency).

Some of you might have been confused by the fact that we said that the frequency in *time* of a one-mass-one-spring system is $\omega = \sqrt{k/m}$, so the stiffer (higher k) the spring is, the higher the frequency. That's a frequency in *time*! And since here we have a string (similar to a system of n masses and $n+1$ springs), the true solution we found for the one-mass-one-spring system does not make sense anymore.