EE2 2005
University of Iceland

Hannes Jónsson
Kristján Alexandersson

A: Vibrating string

v2

A1. Traveling waves

In this lab session you will use Matlab to study traveling waves. The goal is to prepare you to better understand time dependent quantum mechanics, the motion of quantum mechanical particles. This will lead to a clearer connection between quantum mechanics and Newton mechanics which gives approximate, but typically accurate enough results for large particles. Before getting to the time dependent quantum mechanics, it is useful to understand a simpler problem where our every day intuition can guide us, the motion of a vibrating string. You will start out by calculating and visualizing waves in a string which is held fixed at both ends (such as a guitar string). This has a direct anology with a quantum mechanical particle confined to a region in space, the so called 'particle-in-a-box'. You will also explore important wave phenomena, in particular constructive and destructive interference of waves.

Waves appear in many different contexts. For example: ocean waves, a vibrating guitar string, sound waves, earthquake waves, oscillating electric and magnetic field in electromagnetic radiation, etc. All these different types of waves are described by very similar mathematics. From basic equation of motion and restoring forces a differential equation can be derived, the so called 'wave equation'. The solution to a wave equation is a function describing the wave in time and space. For example, consider an ocean wave. If u(x,t) is the displacement of the surface of the water at location x and time t, then the wave equation can be written as

$$\frac{\partial^2 u(x,t)}{\partial x^2} = \frac{1}{V^2}\frac{\partial^2 u(x,t)}{\partial t^2}$$

The left hand side of this equation represents the second derivative of the displacement function with respect to location (the symbol ∂ is used when a function of two or more variables is differentiated with respect to just one of the variables while the others are held fixed). The factor V in the right hand side is the speed of the wave, and the right most factor is the second derivative with respect to time. A solution to the wave equation can be written as

$$u(x,t) = A \sin(kx-\omega t+\phi)$$

where k, ω and ϕ are constants. The maximum displacement, A, is referred to as the amplitude. Here, the amplitude will be set to unity, A=1.

Q1: Verify that this is indeed a solution to the wave equation by evaluating the derivatives (by hand) on the left and right hand side. Give an expression for V in terms of k and ω.

The constant k is called the wavevector and is related to the wavelength, λ, by the equation $k=2\pi/\lambda$. The constant ϕ (the Greek letter 'phi') changes the 'phase' of the wave.The factor ω (the Greek letter 'omega') is called the angular frequency and is related to the frequency ν (the Greek letter 'nu') by $\omega=2\pi\nu$. The frequency is the inverse of the period, i.e. the time interval between full oscillations. These concepts can all be illustrated by plotting the wave. Choose the values k=10 and ω=3 as an example, and to begin with set t=0 and ϕ=0.

First create the a file named v1.m in your working directory and type in the following two lines

```
-------------------------------------------
function y=v1(x,t,k,omega,phi)

y=sin(k*x-omega*t+phi);
-------------------------------------------
```

Then, make a graph showing the wave

```
>> x=[0:0.01:1];
>> plot(x,v1(x,0,10,3,0));
```

Q2: What is the wavelength of this wave (read it from the graph)? How does the number you obtain agree with the specified value of the wavenumber (recall the relationship between the two from above).

Q3: Change the phase by π (setting now phi=π) and plot the wave again. How does shifting the phase by π affect the wave?

When two waves with the same wavelength are such that a peak in one occurs at the same location as a trough in the other, then the two waves are said to be 'out-of-phase'. When two waves meet, for example when there are two sources of waves and the waves meet in between the two sources, the wave displacements add up. For example, if another wave, v2 is defined in a way similar to v1

```
-------------------------------------------
function y=v2(x,t,k,omega,phi)

y=sin(k*x-omega*t+phi);
-------------------------------------------
```

then the superposition of the two waves will result in a wave given by vsum=v1+v2

Q4: Plot the total wave, vsum, when the phase difference between the two waves, v1 and v2, is: 0, $\pi/4$, $\pi/2$, π, and 2π. You can show all five plots simultaneously by giving a list of x,y pairs to the plot command

```
>>plot(x,y1,x,y2,x,y3,x,y4.........)
```

When the waves add up in-phase we say that there is 'constructive interference' between the waves. The amplitude is then the sum of the amplitudes of the v1 and v2 waves. When the waves add up out-of-phase, we say that there is 'destructive interference' and the amplitude is then the difference of the amplitudes of the v1 and v2 waves. Label the plots with 'constructive intereference' and 'destructive interference' where appropriate (only some phase differences lead to these two extreme cases) using the legend command.

Now, consider the time dependence (so far we only considered the initial time, t=0). In order to see the effect of both variables x and t simultaneously, an animation of the motion of the wave is needed. This can be done easily in Matlab using the movie function.

Octave doesn't offer this option but you can see how the wave moves, frame by frame, by amending the function below by removing the getframe and movie commands. Also it is a good idea when using Octave to quadruple tmax to get a better feel for the movement of the wave, as the animation will not loop as in Matlab. If the animation is too rapid, add the pause command into the the for loop.

The function for creating an animation of the v1 wave is

```
--------------------------------------------------------------------------------
function travwave(xmin,xmax,k,omega,phi)

tmax=2*pi/omega;
% The total amount of time of animation

tincrement = 0.05;
% The increment in time between frames

nframes = round(tmax/tincrement);
% The number of frames

x=[xmin:0.01:xmax];

for i=0:nframes
   plot(x,v1(x,i*tincrement,k,omega,phi));
   axis([xmin xmax -1.2 1.2])
   M(i+1) = getframe; % remove this command if using Octave
end

movie(M,20)  % remove this command if using Octave
--------------------------------------------------------------------------------
```

then evaluate using (for example)

```
>> travwave(0,1,10,3,0)
```

Note that the animation is carried out over a time interval from zero to $2\pi/\omega$.

Q5: Why is this a reasonable choice for the length of the time interval?

The wave completes one oscillation at a given point in space over the time interval $1/\nu = 2\pi/\omega$. This is the period. During a period the wave travels one wavelength. The speed is distance traveled divided by time interval.

Q7: Derive from this an expression for the speed, V, of the wave in terms of λ and ν.

Q8: What determines the direction that the wave is traveling in? To answer this, define another wave, v3, where the sign of k*x has been changed and animate. Which direction is the v3 wave traveling in?

A2. Standing waves

When a wave that is traveling to the left is added to a wave which has the same amplitude, frequency and wavelength but is traveling to the right, the resulting wave is stationary, i.e. it is neither going to the right nor to the left. First define the sum of v1 and v3

```
------------------------------------------------------------
function y=vsum(x,t,k,omega,phi)

y=v1(x,t,k,omega,phi)+v3(x,t,k,omega,phi);
------------------------------------------------------------
```

and then animate using using the function defined by standwave.m

```
-----------------------------------------------------------------
function standwave(xmin,xmax,k,omega,phi)

tmax=2*pi/omega;

tincrement = 0.05;

nframes = round(tmax/tincrement);

x=[xmin:0.01:xmax];

for i=0:nframes
   plot(x,vsum(x,i*tincrement,k,omega,phi));
   axis([xmin xmax -2.2 2.2])
   M(i+1) = getframe;  % remove if using Octave
end

movie(M,20)  % remove if using Octave
-----------------------------------------------------------------

>> standwave(0,1,10,3,0)
```

Note that there is no displacement at some points. These points are called 'nodes' in the 'standing wave'.

When a string is held fixed at both ends, the displacement of a wave must necessarily be zero at the ends. The allowed solutions to the wave equation are then restricted to the standing waves that have nodes at the location of the two endpoints of the string. That is, if the string stretches from x=0 to x=1, then the fact that the string is held fixed at the ends means that

$$u(0,t) = u(1,t) = 0$$

These two conditions are referred to as 'boundary conditions'. They greatly restrict the acceptable solutions to the wave eqation.

This could for example be a string in a guitar. The tension in the string and the length of the string determine the frequency of the oscillations and thereby the tone given by the string. The allowed solutions to the wave equation are standing waves. They are sometimes referred to as the 'fundamental vibrational modes'. These kinds of waves can be written as (this is a simplification of the sum of two traveling waves, using trigonometry)

```
-------------------------------------------------------
function y=u(x, t, n, omega, phase)

y=cos(n*omega*t + phase)*sin(n*pi*x);
-------------------------------------------------------
```

Here n is an integer related to the number of nodes between the endpoints. There are n-1 nodes between the end points.

For example, to animate the n=2 mode use

```
--------------------------------------------------------------------
function trigstandwave(xmin,xmax,n,omega,phase)

tmax=2*pi/omega;

tincrement = 0.05;

nframes = round(tmax/tincrement);

x=[xmin:0.01:xmax];

for i=0:nframes
   plot(x,u(x,i*tincrement,n,omega,phase));
   axis([xmin xmax -2.2 2.2])
   M(i+1) = getframe;
end

movie(M,20)
--------------------------------------------------------------------

>> trigstandwave(0,1,2,3,0)
```

Q9: Make animations of the n=2, 3, and 4 vibrational modes. If you use the same number of frames and animate the same time interval, then you should note that the oscillations become faster as n increases. For a smooth animation, more frames are needed when the oscillations are faster. In order to get the smoothest animation, choose a time interval which covers an integral number of periods (then there is no discontinuity as the animation repeats).

Q10: Define a new wave which is a sum of the n=2 vibrational mode with phase ϕ=0 and the n=2 vibrational mode with phase ϕ=2π and animate the result. The phase difference of the two waves that are added together is 2π. What is this phenomenon called (what type of interference is it)?

Q11: Repeat the question above for the n=2 vibrational mode with ϕ=0 and the n=2 vibrational mode with ϕ=π. The phase difference of the two waves that are added together is π. What is this phenomenon called?

Q12: Define a new wave which is the sum of the n=3 vibrational mode and the n=2 vibrational mode (take ϕ=0 in both cases). This new wave is still a valid solution of the wave equation for the string because it satisfies the differential equation and boundary conditions (the end points remain fixed, u(0,t)=u(1,t)=0). But, the sum of the two standing waves is not a standing wave in this case. What is the time period of this new wave?

B. Two dimensional waves:

The above analysis of the vibrating string can be generalized in a rather straightforward way to a vibrating rectangular membrane (a rectangular drum head). The result is a set of vibrational modes which now are indexed with two integers, n and m (one for each spatial degree of freedom)

```
--------------------------------------------------------------------------
function z=u2d(x, y, t, n, m, omega, phase)

z=cos(n*omega*t + phase).*sin(n*pi*x).*sin(m*pi*y);
--------------------------------------------------------------------------
```

The animation is analogous to the one-dimensional case, except that the plots now need to be 3D plots so the plotting function is mesh rather than plot. Here the n=1, m=2 mode is animated

```
------------------------------------------------------------------------------------------------
function a=trigstandwave3d(xmin,xmax,ymin,ymax,n,m,omega,phase)

tmax=2*pi/omega;

tincrement = 0.05;

nframes = round(tmax/tincrement);

x=[xmin:0.1:xmax];
y=[ymin:0.1:ymax];

[x,y] = meshgrid(x,y);

for i=0:nframes
   mesh(x,y,u2d(x,y,i*tincrement,n,m,omega,phase));
   axis([xmin xmax ymin ymax -1.2 1.2])
   M(i+1) = getframe;
end

movie(M,20)
------------------------------------------------------------------------------------------------
```

```
>> trigstandwave3d(0,1,0,1,1,2,3,0)
```

Q13: Which axis is the x-axis and which is the y-axis (left or right)?

Q14: Animate the n = 2, m = 2 mode.

Q15: Animate the n = 2, m = 3 mode.