Studying the wave equation with Maple

Execute the worksheet and play! To manipulate a plot or an animation, click on the output and your options will appear in the menu bar.

Example 1a: Tent function for initial wave profile. Initial velocity of wave = 0, and fixed endpoints (type 1).

```
> restart:
> with(plots):  #to plot and animate
> tent:=t->(1-Heaviside(t-Pi/2))*t +Heaviside(t-Pi/2)*(Pi-t):
> plot(tent(t),t=0..Pi,color=black);
> sineseries:=proc(ff,L,b)   #ff=function, L=half period
  local m,  #dummy letter to index coefficients
        s;  #domain variable
  assume(m,integer):
  b:=m->simplify(2/L*int(ff(s)*sin(Pi/L*m*s),s=0..L)):
  end:
> sineseries(tent,Pi,B):  #to solve the fixed endpoint
       #at endpoints problem we use sine series
       #(and we'll make the speed a=1).
> B(n);
> w1a:=(x,t)->sum(B(n)*sin(n*x)*cos(n*t),n=1..10);
> plot3d(w1a(x,t),x=0..Pi,t=0..3,axes=boxed,title=`boundary temp=
  0`);
```

Make a movie!

```
> animate(w1a(x,t), x=0..Pi, t=0..2*Pi,frames=100);
    #syntax may have changed between Maple 8 and
    #later versions, check in help windows for
    #animate
```

In the first movie we used the tent function for initial position (because we used cosine functions for the functions of time, which are initially 1, with zero t−derivative). In the wave equation you also get to specify an initial velocity function!

Example 1b: Specify the initial **velocity** of the string to be the tent function of x (with zero initial displacement), and still requiring fixed endpoints, use sine series for x, and use sine functions for the t−variable. You would add a type 1a and a type 1b solution to solve the general initial boundary value problem where both the initial displacement and initial velocity of the string are non−zeron functions.

```
> w1b:=(x,t)->sum(B(n)*sin(n*x)*sin(n*t)/n,n=1..10);
      #why did we divide by n?
> animate(w1b(x,t), x=0..Pi, t=0..2*Pi,frames=100);
```

Example 2a: Free endpoint conditions (type 2), use cosine series in x. We'll continue with the tent function as initial displacement, and in this example the initial velocity is zero.

```
> cosseries:=proc(ff,L,a)   #ff=function, L=half period
  local m,  #dummy letter to index coefficients
        s;  #domain variable
  assume(m,integer):
  a:=m->simplify(2/L*int(ff(s)*cos(Pi/L*m*s),s=0..L)):
  end:
> cosseries(tent,Pi,A):  #for free endpoint condition
> A(0);
  A(n);
> w2a:=(x,t)->Pi/2+sum(A(n)*cos(n*x)*cos(n*t),n=1..10);
> plot3d(w2a(x,t),x=0..Pi,t=0..3,axes=boxed,title=‘zero flux
  boundary
  conditions‘);
```

Make a movie!

```
> animate(w2a(x,t), x=0..Pi, t=0..2*Pi,frames=100);
```

Example 2b) Or, we can use the tent function as the initial velocity:

```
> w2b:=(x,t)->Pi/4*t+sum(A(n)*cos(n*x)*sin(n*t)/n,n=1..10);
     #notice the Pi/4*t term, and also the dividing by n in the
  sum
> animate(w2b(x,t), x=0..Pi, t=0..2*Pi,frames=100);
> '?'
```

Other examples: "bump" functions:

```
> bump:=t->Heaviside(t-5*Pi/12)*Heaviside(Pi/2-t)*(t-5*Pi/12)+
  Heaviside(t-Pi/2)*Heaviside(7/12*Pi-t)*(-t+7/12*Pi);
> plot(bump(t),t=0..Pi,color=black);

> sineseries(bump,Pi,B):  #to solve the fixed endpoints
       #problem we use sine series in x.

> B(n);
> w1a:=(x,t)->sum(B(n)*sin(n*x)*cos(n*t),n=1..20);
       #fixed endpoints, zero initial velocity
  w1b:=(x,t)->sum(B(n)*sin(n*x)*sin(n*t)/n,n=1..20);
       #fixed endpoints, zero initial displacement,
       #initial bump velocity

> animate(w1a(x, t), x=-π..π, t=0..6*π, frames=300);
> animate(w1b(x, t), x=-π..π, t=0..6*π, frames=300);
> cosseries(bump, π, A) : #to solve the free endpoints
       #problem we use cosine series in x.

> A(0);
  A(n);
```

> $w2a := (x, t) \rightarrow \frac{\pi}{144} + \mathrm{sum}(A(n) * \cos(n * x) * \cos(n * t), n = 1 ..20);$

```
       #free endpoints, zero initial velocity
```

$w2b := (x, t) \rightarrow \frac{\pi}{144} \cdot t + \mathrm{sum}(A(n) * \cos(n * x) * \sin(n * t) / n, n = 1 ..20);$

```
       #free endpoints, zero initial displacement,
      #initial bump velocity


> animate(w2a(x, t), x = -Pi .. Pi, t = 0 .. 6*Pi, frames = 300);

> animate(w2b(x, t), x = -π .. π, t = 0 .. 6*π, frames = 300);
> `?`
```