

FYS2130 Prosjekt

Kadnr.: 15070

6. mai 2017

1 Exercise 1)

We are here going to find the expression for a mass point y_i of a string from its neighbours and previous position. From the exercise text we are given the force on the a mass point as

$$F_i = F_{i,l} + F_{i,r} = -(k_{i-1} + k_i)y_i + k_{i-1}y_{i-1} + k_iy_{i+1} \quad (1)$$

Where $F_{i,l}$ is the force acting on the mass point from the left, and $F_{i,r}$ is the force acting on it from the right. k_i is the spring constant of the i 'th spring; y_{i-1} and y_{i+1} are the left and right neighbour, respectively; and k_{i-1} the $i - 1$ 'th spring.

From Newton's second law we get that

$$F_i = m_i \ddot{y}_i \quad (2)$$

And from the exercise text we lastly get the numerical approximation of the second derivative:

$$\ddot{y}_i = \frac{d^2 y_i}{dt^2} \approx \frac{y_i^+ - 2y_i^0 + y_i^-}{(\Delta t)^2} \quad (3)$$

Where y_i^+ is what are interested in finding, namely the position of the mass point y_i in the next time step. y_i^0 is the current position of the mass point, and y_i^- the previous position. We are only going to look at the y -position of each mass point. In reality the mass point would be able to move in the x -direction as well (and z -direction in a 3D case), but we are simplifying the system by only looking at the y -position.

We can now use the information above to find a expression for the position of a mass point in the next time step y_i^+ . We begin by inserting equation (1) and (3) into (2):

$$-(k_{i-1} + k_i)y_i + k_{i-1}y_{i-1} + k_iy_{i+1} = m_i \frac{y_i^+ - 2y_i^0 + y_i^-}{(\Delta t)^2} \quad (4)$$

We can now simply solve for y_i^+ . We then get the expression for the next y -position/amplitude of the i 'th mass point:

$$\boxed{y_i^+ = \frac{\Delta t^2}{m_i} (k_{i-1}y_{i-1} - (k_{i-1} + k_i)y_i + k_iy_{i+1}) + 2y_i^0 - y_i^-} \quad (5)$$

Notice that this equation is dependent on the current and previous position of the mass point, meaning that we need an initial position *and* a pre-initial position. Finding the pre-initial conditions will depend on the behaviour we want the system to have, and will be discussed below for some of the later cases.

We also have to be careful of the neighbouring mass points. In the middle of the string this poses no problem, but at the ends we have to impose some special rules on the mass points. We are going to look at two methods of dealing with the endpoints:

Open ends:

For the open ends we want the mass points to only feel the force of one of its neighbours – the right neighbour for the first point, and the left of the last point –. This means that we can write Newton's second law for the first and last mass point as:

$$m_0 \ddot{y}_0 = F_{0,r} \quad m_{N-1} \ddot{y}_{N-1} = F_{N-1,l} \quad (6)$$

We can thus rewrite expression (5) explicitly for the endpoints:

$$y_0^+ = \frac{\Delta t^2}{m_0} (-k_0 y_0 + k_0 y_1) + 2y_0^0 - y_0^- \quad (7)$$

$$y_{N-1}^+ = \frac{\Delta t^2}{m_{N-1}} (k_{N-2} y_{N-2} - k_{N-2} y_{N-1}) + 2y_{N-1}^0 - y_{N-1}^- \quad (8)$$

Reflective ends:

The second type of ends are the reflective ends. These are ends that don't move from their initial position. This means that every wave hitting this point will be reflected. There are two ways of implementing this. The first and simplest method is to just force the position of these mass points to be the initial condition:

$$y_{end}^+ = y_{end}^0 = y_{end}^- \quad (9)$$

The second method is more like reality. A total reflection will happen when a wave hits a domain with a very large impedance, given as

$$z_i = \sqrt{k_i m_i} \quad (10)$$

or a wave on string. We can there for give the endpoints very large impedance. In our case we can give the point a mass $M_i \gg m_i$. This will ensure a reflective end. But again we have to find a way to treat the neighbours at the endpoint. The way I did this in my simulation was to simply have open ends with mass M_i , this means that we don't have to think about the neighbours and we get reflective ends. So for the reflective ends y_{end}^+ is given as:

$$y_0^+ = \frac{\Delta t^2}{M_0} (-k_0 y_0 + k_0 y_1) + 2y_0^0 - y_0^- \quad (11)$$

$$y_{N-1}^+ = \frac{\Delta t^2}{M_{N-1}} (k_{N-2} y_{N-2} - k_{N-2} y_{N-1}) + 2y_{N-1}^0 - y_{N-1}^- \quad (12)$$

2 Exercise 2)

We are here going to show that the above way of describing a wave on a string reduces to the 1D wave equation.

We are going to start by introducing a mass density $\mu = m/\Delta x$ and the constant spring stiffness $\kappa = k\Delta x$, here Δx is the distance between the mass points. We are going to assume the spring constants k_i and masses m_i to be constant along the whole spring ¹. This means that we can use that

$$k = \frac{\kappa}{\Delta x}, \quad m = \mu\Delta x \quad (13)$$

We can now insert these into equation (1) and (2)

$$m_i\ddot{y}_i = \mu\Delta x\ddot{y}_i = F_i = \frac{\kappa}{\Delta x}(y_{i-1} - 2y_i + y_{i+1}) \quad (14)$$

$$\Rightarrow \mu\Delta x\ddot{y}_i = \frac{\kappa}{\Delta x}(y_{i-1} - 2y_i + y_{i+1}) \quad (15)$$

In the expression for F_i I pulled out $\kappa/\Delta x$ and rearranged the y -terms. We then move $\mu\Delta x$ to the right side, and get

$$\ddot{y}_i = \frac{\kappa}{\mu} \cdot \frac{y_{i-1} - 2y_i + y_{i+1}}{(\Delta x)^2} \quad (16)$$

If we remember back to how discretized the second derivative of y with respect to time in (3) we see that the last term on the right side look eerily similar. This is infact the discretized of the second derivative of y with respect to x

$$\frac{d^2y_i}{dx^2} \approx \frac{y_{i-1} - 2y_i + y_{i+1}}{(\Delta x)^2} \quad (17)$$

We are going to use that this go to a equivalence as Δx goes to zero. So if we insert this into (16) we get

$$\ddot{y}_i = \frac{\kappa}{\mu} \frac{d^2y_i}{dx^2} \quad (18)$$

This means that a wave on a string obeys the differential equation:

$$\boxed{\frac{d^2y_i}{dt^2} = v_B^2 \frac{d^2y_i}{dx^2}} \quad (19)$$

This is the 1D wave equation. Comparing this with (18), we get that a wave on a string propagate with the velocity v_B given as

$$\boxed{v_B^2 = \frac{\kappa}{\mu}} \quad (20)$$

¹This is not necessary to for the end points, and especially for reflective ends. But for a solution of the wave equation, the imposed boundary conditions will ensure the right behaviour of the ends. We can therefore ignore the ends for the derivation of the wave equation.

3 Exercise 3)

When we are simulating the wave, we need to ensure that the program is numerically stable. The requirement for this to be true is that

$$\frac{\Delta x}{\Delta t} \geq v_B \quad (21)$$

This comes from the Courant–Friedrichs–Lewy condition for convergence of a numerical solution for partial differential equations like the wave equation. Broadly this condition says that the time step has to be smaller than the time it takes the wave to move the next mass point, which is equivalent to the above (21).

To see what kind of limitations this means for our simulation, we are going to rewrite v_B with k and m :

$$v_B = \sqrt{\frac{\kappa}{\mu}} = \sqrt{\frac{k\Delta x}{m/\Delta x}} = \Delta x \sqrt{\frac{k}{m}} \quad (22)$$

We insert this into (21)

$$\frac{\Delta x}{\Delta t} \geq \Delta x \sqrt{\frac{k}{m}} \quad (23)$$

Doing the algebra we get the convergence condition:

$$\boxed{\Delta t \leq \sqrt{\frac{m}{k}}} \quad (24)$$

As we can see, the convergence condition no longer involves Δx . If we look at the expression for y_i^+ (5) we see that there is no mention of Δx , meaning that both the stability and precision of the simulation is wholly independent of Δx .

The conclusion is thus that while the value of Δt is highly important for the accuracy of the simulation, Δx is just a scaling factor which we are free to set as 1 for our simulation.

4 Exercise 4)