

We want to find out what a guitar string with variable mass density would sound like. Our model is based on (1): a 1D wave equation with variable speed of sound:

$$\boxed{\frac{1}{c(x)^2} \frac{\partial^2 \Phi}{\partial t^2} = \frac{\partial^2 \Phi}{\partial x^2}} \quad (1)$$

From the mathematical theory of partial differential equations (EDPs), we know that we need both boundary and initial conditions to get a unique solution. For a guitar string, we impose:

$$\left\{ \begin{array}{l} \frac{1}{c(x)^2} \frac{\partial^2 \Phi}{\partial t^2} = \frac{\partial^2 \Phi}{\partial x^2}, \quad 0 < x < L, \quad t > 0 \\ \Phi(t, 0) = \Phi(t, L) = 0 \\ \Phi(0, x) = f(x) \\ \frac{\partial \Phi}{\partial t}(0, x) = g(x) \end{array} \right. , \quad (2)$$

where L is the length of the string and $f(x)$ and $g(x)$ are functions representing the shape and the velocity of the string, respectively, upon being plucked.

For our purposes, it is more convenient to work with first order equations in time. Define $\Pi(t, x) \equiv \frac{\partial \Phi}{\partial t}$. The problem (2) is then translated into:

$$\left\{ \begin{array}{l} \frac{\partial \Phi}{\partial t} = \Pi \quad 0 < x < L, \quad t > 0 \\ \frac{\partial \Pi}{\partial t} = c(x)^2 \frac{\partial^2 \Phi}{\partial x^2} \\ \Phi(t, 0) = \Phi(t, L) = 0 \\ \Phi(0, x) = f(x) \\ \Pi(0, x) = g(x) \end{array} \right. . \quad (3)$$

1 Finite Differences Method

In the finite differences approach, we need to discretize the domain into a grid of points. For simplicity, we choose a regular grid, as shown in figure 1.

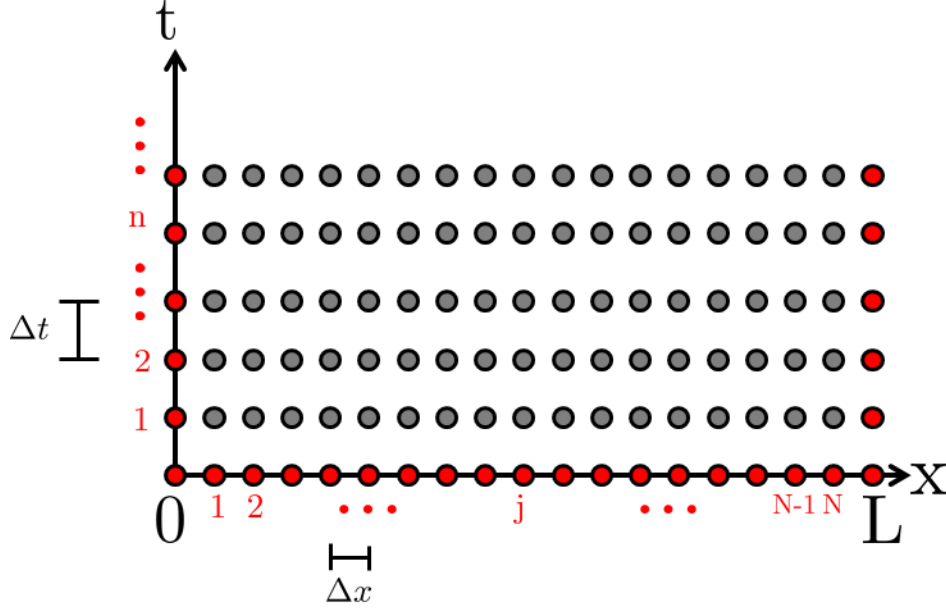


Figure 1: Spacetime discretization grid to be used in the numerical solution of the problem. We show in red the boundary of the domain where the problem is defined. The labeling is chosen so that there are N spatial points *not on the boundary* for a given time step.

As can be seen in figure 1, each point in the grid (t_n, x_j) is labeled by a spatial index $j = 0, 1, \dots, N+1$ and a temporal index $n = 0, 1, \dots$. The points are separated from each other by both a spatial step Δx and a time step Δt . Our unknown functions $\Phi(t, x)$ and $\Pi(t, x)$ can then be evaluated on a specific point (t_n, x_j) ; we define $\Phi(t_n, x_j) \equiv \Phi_j^n$ and $\Pi(t_n, x_j) \equiv \Pi_j^n$.

Now, we know that, using Taylor expansions, we can approximate derivatives on a grid. For example, the following is true *if the spatial grid is regular*:

$$\left. \frac{\partial^2 \Phi}{\partial x^2} \right|_{(t_n, x_j)} \approx \frac{\Phi_{j+1}^n - 2\Phi_j^n + \Phi_{j-1}^n}{\Delta x^2}, \quad (4)$$

as a second-order approximation for the laplacian operator. Let us then choose a general time step t_n , $n \neq 0$, and apply the equations of motion for every point x_j *not on the boundary*:

$$\left\{ \begin{array}{l} \left(\frac{\partial \Phi}{\partial t} \right)_j^n = \Pi_j^n \\ \left(\frac{\partial \Pi}{\partial t} \right)_j^n = (c_j^n)^2 \left(\frac{\Phi_{j+1}^n - 2\Phi_j^n + \Phi_{j-1}^n}{\Delta x^2} \right) \end{array} \right. \quad j = 1, 2, \dots, N, \quad n = 1, 2, \dots \quad (5)$$

The reason we apply the functions only on points not on the boundary is very simple: strictly speaking, differential equations are not defined on it. The boundary conditions, however, fix the values of the unknowns beforehand.

For a given time step t_n , $n \neq 0$, we can interpret (5) as a system of equations for the components of the vector:

$$\mathbf{Y}^n \equiv \begin{bmatrix} \Phi_1^n \\ \Phi_2^n \\ \vdots \\ \Phi_{N-1}^n \\ \Phi_N^n \\ \Pi_1^n \\ \Pi_2^n \\ \vdots \\ \Pi_{N-1}^n \\ \Pi_N^n \end{bmatrix}. \quad (6)$$

Indeed, because the wave equation is linear, it is convenient to condense the unknowns into a big vector. We can then write the right-hand side of (5) as $\mathbf{M}^n \cdot \mathbf{Y}^n$ for a suitable matrix \mathbf{M}^n that depends only on the parameters of the problem (in the present case, the speed of sound function $c(x)$).

However, one must be careful in constructing the matrix \mathbf{M}^n . Remember that the boundary values Φ_0^n , Φ_{N+1}^n , Π_0^n and Π_{N+1}^n do not enter \mathbf{Y}^n as they are not unknowns. In order to see this clearly, take for example $j = 1$ and $j = 2$ in the second set of equations in (5):

$$\begin{cases} \left(\frac{\partial \Pi}{\partial t} \right)_1^n = -2 \frac{(c_1^n)^2}{\Delta x^2} \Phi_1^n + \frac{(c_1^n)^2}{\Delta x^2} \Phi_2^n + \frac{(c_1^n)^2}{\Delta x^2} \Phi_0^n \\ \left(\frac{\partial \Pi}{\partial t} \right)_2^n = \frac{(c_2^n)^2}{\Delta x^2} \Phi_1^n - 2 \frac{(c_2^n)^2}{\Delta x^2} \Phi_2^n + \frac{(c_2^n)^2}{\Delta x^2} \Phi_3^n \end{cases}. \quad (7)$$

In (7), we use the same color scheme as in figure 1: gray indicates points inside the open domain and red indicates points on its boundary. Something similar happens for $j = N - 1$ and $j = N$:

$$\begin{cases} \left(\frac{\partial \Pi}{\partial t} \right)_{N-1}^n = \frac{(c_{N-1}^n)^2}{\Delta x^2} \Phi_{N-2}^n - 2 \frac{(c_{N-1}^n)^2}{\Delta x^2} \Phi_{N-1}^n + \frac{(c_{N-1}^n)^2}{\Delta x^2} \Phi_N^n \\ \left(\frac{\partial \Pi}{\partial t} \right)_N^n = \frac{(c_N^n)^2}{\Delta x^2} \Phi_{N-1}^n - 2 \frac{(c_N^n)^2}{\Delta x^2} \Phi_N^n + \frac{(c_N^n)^2}{\Delta x^2} \Phi_{N+1}^n \end{cases}. \quad (8)$$

Therefore, if we want to write the right-hand side of the second group of equations in (5) as a linear system on the variables $(\Phi_1^n, \Phi_2^n, \dots, \Phi_{N-1}^n, \Phi_N^n)$, the red terms in (7) and (8) have to be written as a non-homogeneous contribution. In other words, we will have $A \cdot (\Phi_1^n, \Phi_2^n, \dots, \Phi_{N-1}^n, \Phi_N^n) + \mathbf{b}$ for given A and \mathbf{b} . In particular, it is clear that A is a tridiagonal matrix and the only non-zero entries of \mathbf{b} are its first and last components.

We still have to consider the first set of equations in (5). It poses no problem though, since only the same points j appear. Therefore, equations (5) can finally be written:

$$\begin{bmatrix} \left(\frac{\partial \Phi}{\partial t}\right)_1^n \\ \vdots \\ \left(\frac{\partial \Phi}{\partial t}\right)_N^n \\ \hline \left(\frac{\partial \Pi}{\partial t}\right)_1^n \\ \vdots \\ \left(\frac{\partial \Pi}{\partial t}\right)_N^n \end{bmatrix} = \begin{bmatrix} & & & & & \\ & \mathbb{O} & & & \mathbb{I} & \\ & & & & & \\ \hline & & & & & \\ & \ddots & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & \ddots & \\ & & & & & \mathbb{O} \end{bmatrix} \begin{bmatrix} \Phi_1^n \\ \vdots \\ \Phi_N^n \\ \hline \Pi_1^n \\ \vdots \\ \Pi_N^n \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \hline \frac{(c_1^n)^2}{\Delta x^2} \Phi_0^n \\ 0 \\ \vdots \\ 0 \\ \hline \frac{(c_N^n)^2}{\Delta x^2} \Phi_{N+1}^n \end{bmatrix}, \quad (9)$$

where \mathbb{O} and \mathbb{I} represent the null matrix and the identity matrix, respectively. The tridiagonal block is the matrix A discussed before. Defining \mathbf{B} as the boundary condition vector, we can then write:

$$\begin{bmatrix} \left(\frac{\partial \Phi}{\partial t}\right)_1^n \\ \vdots \\ \left(\frac{\partial \Phi}{\partial t}\right)_N^n \\ \hline \left(\frac{\partial \Pi}{\partial t}\right)_1^n \\ \vdots \\ \left(\frac{\partial \Pi}{\partial t}\right)_N^n \end{bmatrix} = M^n \cdot \mathbf{Y}^n + \mathbf{B}^n \equiv \mathbf{F}^n(\mathbf{Y}^n) \quad (10)$$

Only one time step t_n , $n \neq 0$ has been dealt with so far. Now, we want to evolve our initial conditions in time using (10).

1.1 Explicit Euler Method

We can again make use of Taylor expansions. Now we do it in the time coordinate instead. The following is true:

$$\mathbf{Y}^n \equiv \mathbf{Y}|_{t_n} \approx \mathbf{Y}|_{t_{n-1}} + \left. \frac{d\mathbf{Y}}{dt} \right|_{t_{n-1}} \Delta t. \quad (11)$$

But $\frac{d\mathbf{Y}}{dt}$ is given by (10). Therefore:

$$\boxed{\mathbf{Y}^n \approx \mathbf{Y}^{n-1} + \mathbf{F}^{n-1}(\mathbf{Y}^{n-1})\Delta t, \quad n = 1, 2, \dots} \quad (12)$$

This is the *explicit Euler method*: we can find the values of the unknowns on a chosen time step using only their values on the previous time step. Notice that, from (10), the function \mathbf{F}^n *already incorporates the boundary conditions* and, thus, the evolution takes them into account. Furthermore, \mathbf{F}^n depends on the time step itself, so that in general we have to build a new matrix for each step. Nevertheless, our problem is a little bit simpler: because neither the speed of sound nor the boundary conditions are functions of time, the evolution matrix coming from \mathbf{F} is the same for all iterations. We then only need to build one matrix and keep applying it to the each time step vector.

1.2 Implicit Euler Method

Explicit methods have a caveat: they usually amplify numerical errors at each iteration, leading to an unphysical growing solution. This can be dealt with by introducing numerical dissipation terms into the wave equation itself. Alternatively, one can try using implicit methods instead.

Implicit methods are also based on Taylor expansions. In (11), we used a Taylor series to approximate the value of the solution vector at $t_n = t_{n-1} + \Delta t$ from its value at t_{n-1} . The opposite also works. Given $\mathbf{Y}^n = \mathbf{Y}|_{t_n}$, we can approximate $\mathbf{Y}^{n-1} = \mathbf{Y}|_{t_{n-1}} = \mathbf{Y}|_{t_n - \Delta t}$ by:

$$\mathbf{Y}^{n-1} \approx \mathbf{Y}|_{t_n} - \left. \frac{d\mathbf{Y}}{dt} \right|_{t_n} \Delta t. \quad (13)$$

Again, from (10), we can write:

$$\boxed{\mathbf{Y}^n - \mathbf{F}^n(\mathbf{Y}^n)\Delta t \approx \mathbf{Y}^{n-1}, \quad n = 1, 2, \dots} \quad (14)$$

However, we do not know \mathbf{Y}^n ; this is exactly what we want to find from a given \mathbf{Y}^{n-1} . Equation (14), then, involves calculating the inverse of the matrix $\mathbb{I} - \mathbf{M}^n$ for each iteration. Once again, our guitar string setting is simpler: we only need to calculate this inverse matrix once, since neither the speed of sound function nor the boundary conditions depend on time.

1.3 Explicit Runge-Kutta Method

Involves calculating \mathbf{F} in “virtual” points between t_{n-1} e t_n .

1.4 Implicit Runge-Kutta Method

Same thing but implicit.

1.5 Crank-Nicolson Method

Average between explicit and implicit Euler.