

pde.wave The 1D wave equation

The one-dimensional wave equation is the linear PDE

$$\partial_t^2 u(t, x) = c^2 \partial_x^2 u(t, x), \quad (1)$$

with real constant c . This equation models such phenomena as strings, fluids, sound, and light. It is subject to initial and boundary conditions and can be extended to multiple spatial dimensions. For 2D and 3D examples in rectangular and polar coordinates, see Kreyszig⁸ and Haberman.⁹

8. Kreyszig, *Advanced Engineering Mathematics*, § 12.9, 12.10.
9. Haberman, *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*, Chapter § 13, 7.3.
REDUCING STRING PDE'S TO ODE'S BY separation of variables

Example pde.wave-1

Consider the one-dimensional wave equation PDE

$$\partial_t^2 u(t, x) = c^2 \partial_x^2 u(t, x) \quad (2)$$

with real constant c and with Dirichlet boundary conditions on interval $x \in [0, L]$

$$u(t, 0) = 0 \quad \text{and} \quad u(t, L) = 0, \quad (3a)$$

and with initial conditions (we need two because of the second time-derivative)

$$u(0, x) = f(x) \quad \text{and} \quad \partial_t u(0, x) = g(x), \quad (4)$$

where f and g are some piecewise continuous functions on $[0, L]$.

Assume a product solution

First, we assume a product solution of the form $u_p(t, x) = T(t)X(x)$ where T and X are unknown functions on $t > 0$ and $x \in [0, L]$.

Separate PDE

Second, we substitute the product solution into Eq. 2 and separate variables:

$$T''X = c^2 TX'' \Rightarrow \quad (5)$$

$$\frac{T''}{c^2 T} = \frac{X''}{X}. \quad (6)$$

So it is separable! Note that we chose to group c with T , which was arbitrary but conventional.

Set equal to a constant

Since these two sides depend on different independent variables (t and x), they must equal the same constant we call $-\lambda$, so we have two ODEs:

$$\frac{T''}{c^2 T} = -\lambda \Rightarrow T'' + \lambda c^2 T = 0 \quad (7)$$

$$\frac{X''}{X} = -\lambda \Rightarrow X'' + \lambda X = 0. \quad (8)$$

Solve the boundary value problem

The latter of these equations with the boundary conditions (3) is precisely the same Sturm-Liouville boundary value problem from Example pde.sturm-1, which had eigenfunctions

$$X_n(x) = \sin\left(\sqrt{\lambda_n} x\right) \quad (9a)$$

$$= \sin\left(\frac{n\pi}{L} x\right) \quad (9b)$$

with corresponding (positive) eigenvalues

$$\lambda_n = \left(\frac{n\pi}{L}\right)^2. \quad (10)$$

Solve the time variable ODE

The time variable ODE is homogeneous and, with λ restricted by the reals by the boundary value problem, has the familiar general solution

$$T(t) = k_1 \cos(c\sqrt{\lambda}t) + k_2 \sin(c\sqrt{\lambda}t) \quad (11)$$

with real constants k_1 and k_2 . However, the boundary value problem restricted values of λ to λ_n , so

$$T_n(t) = k_1 \cos\left(\frac{cn\pi}{L}t\right) + k_2 \sin\left(\frac{cn\pi}{L}t\right). \quad (12)$$

Construct the product solution

The product solution is

$$u_p(t, x) = T_n(t)X_n(x) \\ = k_1 \sin\left(\frac{n\pi}{L}x\right) \cos\left(\frac{cn\pi}{L}t\right) + k_2 \sin\left(\frac{n\pi}{L}x\right) \sin\left(\frac{cn\pi}{L}t\right).$$

This is a family of solutions that each satisfy only exotically specific initial conditions.

Apply the initial conditions

Recall that superposition tells us that any linear combination of the product solution is also a solution. Therefore,

$$u(t, x) = \sum_{n=1}^{\infty} a_n \sin\left(\frac{n\pi}{L}x\right) \cos\left(\frac{cn\pi}{L}t\right) + b_n \sin\left(\frac{n\pi}{L}x\right) \sin\left(\frac{cn\pi}{L}t\right) \quad (13)$$

is a solution. If a_n and b_n are properly selected to satisfy the initial conditions, Eq. 13 will be the solution to the entire problem. Substituting $t = 0$ into our potential solution gives

$$u(0, x) = \sum_{n=1}^{\infty} a_n \sin\left(\frac{n\pi}{L}x\right) \quad (14a)$$

$$\partial_t u(t, x)|_{t=0} = \sum_{n=1}^{\infty} b_n \frac{cn\pi}{L} \sin\left(\frac{n\pi}{L}x\right). \quad (14b)$$

Let us extend f and g to be periodic and odd over x ; we call the extensions f^* and g^* . From Eq. 14, the initial conditions are satisfied if

$$f^*(x) = \sum_{n=1}^{\infty} a_n \sin\left(\frac{n\pi}{L}x\right) \quad (15a)$$

$$g^*(x) = \sum_{n=1}^{\infty} b_n \frac{cn\pi}{L} \sin\left(\frac{n\pi}{L}x\right). \quad (15b)$$

We identify these as two odd Fourier syntheses.

The corresponding Fourier analyses are

$$a_n = \frac{2}{L} \int_0^L f^*(x) \sin\left(\frac{n\pi}{L}x\right) dx \quad (16a)$$

$$b_n \frac{cn\pi}{L} = \frac{2}{L} \int_0^L g^*(x) \sin\left(\frac{n\pi}{L}x\right) dx \quad (16b)$$

So the complete solution is Eq. 13 with components given by Eq. 16. Notice this satisfies the PDE, the boundary conditions, and the initial condition!

Discussion

It can be shown that this series solution is equivalent to two traveling waves that are interfering (see Haberman⁹ and Kreyszig⁸). This is convenient because computing the series solution exactly requires an infinite summation. We show in the following section that the approximation by partial summation is still quite good.

Choosing specific initial conditions

If we want to plot solutions, we need to specify initial conditions over $[0, L]$. Let's model a string being suddenly struck from rest as

$$f(x) = 0 \quad (17)$$

$$g(x) = \delta(x - \Delta L) \quad (18)$$

where δ is the Dirac delta distribution and $\Delta \in [0, L]$ is a fraction of L representing the location of the string being struck. The odd periodic extension is an odd pulse train. The integrals of 16 give

$$a_n = 0 \quad (19a)$$

$$b_n = \frac{2}{cn\pi} \int_0^L \delta(x - \Delta L) \sin\left(\frac{n\pi}{L}x\right) dx \\ = \frac{2}{cn\pi} \sin(n\pi\Delta). \quad (\text{sifting property})$$

Now we can write the solution as

$$u(t, x) = \sum_{n=1}^{\infty} \frac{2}{cn\pi} \sin(n\pi\Delta) \sin\left(\frac{n\pi}{L}x\right) \sin\left(\frac{cn\pi}{L}t\right). \quad (20)$$

Plotting in Python

First, load some Python packages.

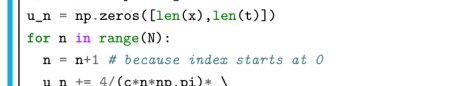
```
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, Markdown, LaTeX
```

Set $c = L = 1$ and sum values for the first 8 terms of the solution for some striking location Δ .

```
delta = 0.1 # 0 <= delta <= L
L = 1
c = 1
N = 100
t = np.linspace(0, (L/c)*pi**2, 100)
x = np.linspace(0, L, 100)
u, u_t = np.meshgrid(t, x)
u_n = np.zeros([len(t), len(x)])
for n in range(8):
    u_n += (-1)**n * np.sin(c*(n*pi)*t) * \
        np.sin(c*(n*pi)*x) * \
        np.sin(c*(n*pi)*delta) * \
        np.sin(c*(n*pi)*x)
u_t_n = np.zeros([len(t), len(x)])
for n in range(8):
    u_t_n += (-1)**n * c * \
        np.cos(c*(n*pi)*t) * \
        np.cos(c*(n*pi)*x) * \
        np.cos(c*(n*pi)*delta) * \
        np.sin(c*(n*pi)*x)
```

Let's first plot some early snapshots of the response.

```
import seaborn as sns
u_nsnap = 7
sns.set_palette('magma')
sns.set_style('whitegrid')
fig, ax = plt.subplots(2, 1)
ax[0].plot(x, u_nsnap)
ax[1].plot(x, u_t_nsnap)
ax[0].set_xlabel('x')
ax[1].set_xlabel('t')
```



Now we plot the entire response.

```
fig, ax = plt.subplots(2, 1)
p = ax.contourf(t, x, u)
c = fig.colorbar(p, ax=ax)
ax.set_xlabel('x')
ax.set_ylabel('t')
plt.xlabel('x')
plt.ylabel('t')
```



We see a wave develop and travel, reflecting and inverting off each boundary.

change to neumann bc
 $\frac{\partial}{\partial x} u(t, x) \Big|_{x=0, L} = 0$

$$X_n(x) = \cos\left(\frac{n\pi}{L} x\right)$$

$$u_p(t, x) = T_n(t) X_n(x) \\ = \left(k_1 \cos\left(\frac{cn\pi}{L} t\right) + k_2 \sin\left(\frac{cn\pi}{L} t\right) \right) \cos\left(\frac{n\pi}{L} x\right)$$

$$u(t, x) = \sum_{n=1}^{\infty} k_1 \cos\left(\frac{cn\pi}{L} t\right) \cos\left(\frac{n\pi}{L} x\right) + k_2 \sin\left(\frac{cn\pi}{L} t\right) \cos\left(\frac{n\pi}{L} x\right)$$

$$k_1 = a_n \quad k_2 = b_n$$

$$u(0, x) = \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi}{L} x\right)$$

$$\frac{\partial}{\partial t} u(t, x) \Big|_{t=0} = \sum_{n=1}^{\infty} \frac{L}{cn\pi} b_n \cos\left(\frac{n\pi}{L} x\right)$$

$$a_n = \frac{2}{L} \int_0^L f(x) \cos\left(\frac{n\pi}{L} x\right) dx$$

$$b_n = \frac{2}{L} \frac{L}{cn\pi} \int_0^L g(x) \cos\left(\frac{n\pi}{L} x\right) dx$$

$$f(x) = 0 \Rightarrow a_n = 0$$

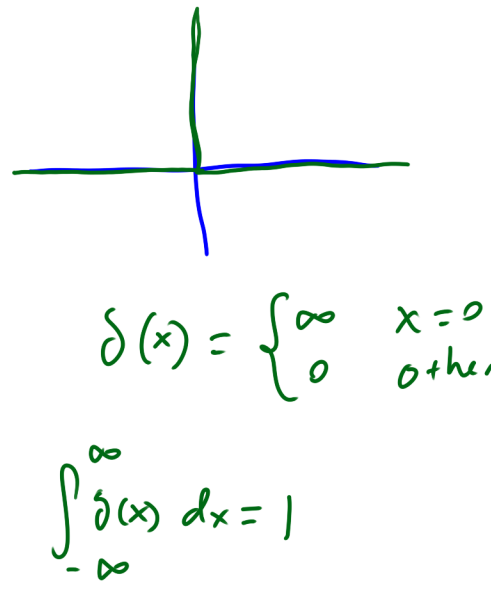
$$g(x) = \delta(x - \Delta L)$$

$$b_n = \frac{2}{cn\pi} \int_0^L \delta(x - \Delta L) \cos\left(\frac{n\pi}{L} x\right) dx$$

$$= \frac{2}{cn\pi} \cos\left(\frac{n\pi}{L} \Delta L\right)$$

$$= \frac{2}{cn\pi} \cos(n\pi\Delta)$$

$$u(t, x) = \sum_{n=1}^{\infty} \frac{2}{cn\pi} \cos(n\pi\Delta) \sin\left(\frac{cn\pi}{L} t\right) \cos\left(\frac{n\pi}{L} x\right)$$



⁸ Haberman, *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems* (Classic Version), § 4.4.
⁹ Kreyszig, *Advanced Engineering Mathematics*, § 12.2.
¹⁰ Python code in this section was generated from a Jupyter notebook named `pde_separation_example_02.ipynb` with a `py2html` kernel.