Finite Difference Time Domain
Application to room acoustics

Master 2 Acoustical Engineering
Numerical Techniques for Acoustics - Session 4

Matthieu Aussal*

*Centre de Mathématiques Appliquées de l'École Polytechnique
Route de Saclay - 91128 Palaiseau CEDEX France

Monday 07 October 2019 - ENSTA

Wave equation in time domain

For a wave on a vibrating string, magnitude of the vibration $y(x,t)$ is governed by a Partial Derivative Equation (PDE) :

$$\begin{aligned} \frac{1}{c^2}\frac{\partial^2 y}{\partial t^2} - \frac{\partial^2 y}{\partial x^2} &= 0, \\ y(x=0,\cdot) &= 0, \\ y(x=L,\cdot) &= 0, \\ y(\cdot,t=0) &= y_0, \\ \frac{\partial y}{\partial t}(\cdot,t=0) &= v_0, \end{aligned}$$

with c the sound celerity, t the time, x the position on the string, L the length of the string and y_0, v_0 the initial position and speed.

Note : 4 derivatives $\rightarrow$ 4 initial conditions.

Classical scheme for **space** derivative

As usual, we consider Taylor's expansion for all functions $y \in C^4([0, L])$:

$$y''(x) = \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} + O(h^4).$$

Considering a string discretization by a regular grid $X = (x_0, ..., x_N)$, such as :

- $x_0 = 0$,
- $x_N = L$,
- $x_n = n\delta_x$ with $\delta_x = \frac{L}{N+1}$ and $n \in [0, N]$,

a **centered scheme** can be used :

$$\begin{aligned} y''(x) &= \frac{y(x+\delta_x) - 2y(x) + y(x-\delta_x)}{\delta_x^2} + O(\delta_x^4) \\ y''_n &\approx \frac{y_{n+1} - 2y_n + y_{n-1}}{\delta_x^2} \end{aligned}$$

Classical scheme for **time** derivative

Fixing a final time t_f for the wave propagation, a time discretisation $T = (t_0, ..., t_P)$ can be defined such as :

- $t_0 = 0$,
- $t_P = t_f$,
- $t_p = p\delta_t$ with $\delta_t = \frac{t_f}{P+1}$ and $p \in [0, P]$.

In the same way as for space, second order derivative in time leads to another centered sheme :

$$\begin{aligned} y''(t) &= \frac{y(t+\delta_t) - 2y(t) + y(t-\delta_t)}{\delta_t^2} + O(\delta_t^4) \\ y''_p &\approx \frac{y_{p+1} - 2y_p + y_{p-1}}{\delta_t^2} \end{aligned}$$

Note : Not so far from the space scheme...

Classical scheme for **wave equation**

Combine each partial derivative approximation leads to the leap-frog scheme :

$$\frac{1}{c^2}\frac{\partial^2 y}{\partial t^2} \quad - \quad \frac{\partial^2 y}{\partial x^2} = 0$$

$$\frac{1}{c^2}\frac{y_n^{p+1} - 2y_n^p + y_n^{p-1}}{\delta_t^2} \quad - \quad \frac{y_{n+1}^p - 2y_n^p + y_{n-1}^p}{\delta_x^2} = 0$$

- Time indices up, space indices down,
- Implicit scheme in space,
- Explicit scheme in time (two step scheme),
- Stable $\iff$ $\frac{\delta_x}{\delta_t} \geq c$ (CFL condition).

Note : CFL from Richard Courant, Kurt Friedrichs and Hans Lewy : the "speed" of the scheme has to be greater than the speed of the equation.

Algorithm

- Use sparse matrix for the discrete derivative of the space part :

$$M_x = \frac{1}{\delta_x^2}\begin{pmatrix} -2 & 1 & 0 & \dots & 0 & 0 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -2 & 1 \\ 0 & 0 & 0 & \dots & 1 & -2 \end{pmatrix},$$

- Add initial condition to the linear system :

$$M_x = \frac{1}{\delta_x^2}\begin{pmatrix} \delta_x^2 & 0 & 0 & \dots & 0 & 0 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & \delta_x^2 \end{pmatrix}$$

Algorithm

- Initialize 2 vectors of unknowns in space :

$$Y^0 = \begin{pmatrix} y_0^0 \\ y_1^0 \\ \dots \\ y_N^0 \end{pmatrix} = \begin{pmatrix} y(x_0, 0) \\ y(x_1, 0) \\ \dots \\ y(x_N, 0) \end{pmatrix}$$

$$Y^1 = \begin{pmatrix} y_0^1 \\ y_1^1 \\ \dots \\ y_N^1 \end{pmatrix} = Y^0 + \delta_t \begin{pmatrix} \partial_t y(x_0, 0) \\ \partial_t y(x_1, 0) \\ \dots \\ \partial_t y(x_N, 0) \end{pmatrix}$$

- Compute Y^2, using the leap-frog scheme :

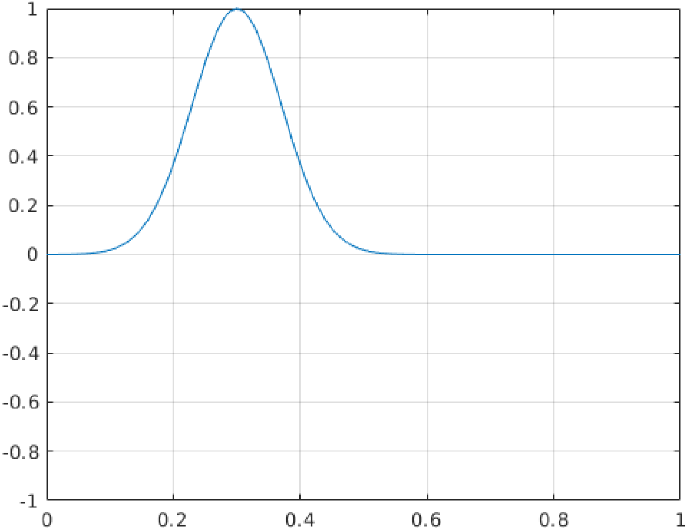
$$Y^2 = 2Y^1 - Y^0 + (c\delta_t)^2 M x Y^1$$

- Make a recursion until the final time t_f.

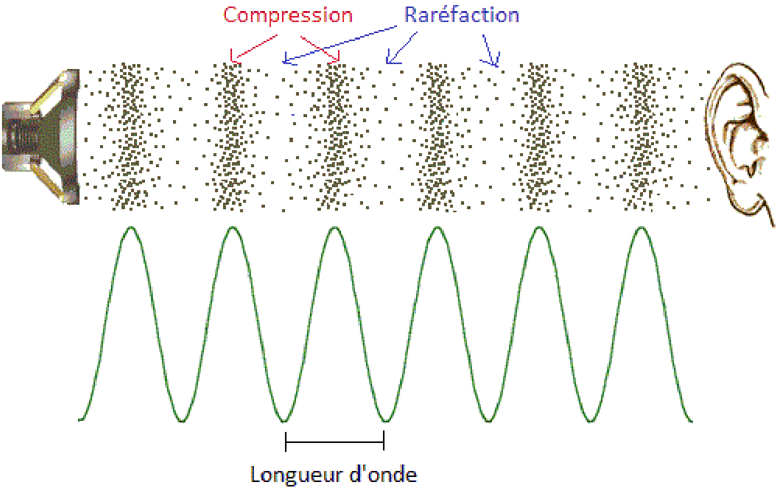
Starting code

```
% Clean up
clear all
close all
clc

% Physical parameters
L  = 1;          % String size
x0 = 0.3;        % Initial position
tf = 1;          % Final time
c  = 1;          % Sound celerity

% Initial condition for magnitude and speed
u0 = @(x) exp(-(x-x0).^2/1e-2); % gaussian
v0 = @(x) zeros(size(x));       % null

% Numerical discretization
dx = 0.01;              % Space step
dt = 0.5*dx/c;          % Time step (following CFL)
```

Numerical result



1-D air propagation



1-D room acoustic

For a wave propagating in a 1-D room, relative magnitude of the wave $u(x,t)$ is governed by a PDE :

$$
\begin{aligned}
\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} &= 0, \\
\frac{\partial u}{\partial x}(x=0,\cdot) &= 0, \\
\frac{\partial u}{\partial x}(x=L,\cdot) &= 0, \\
u(\cdot, t=0) &= u_0, \\
\frac{\partial u}{\partial t}(\cdot, t=0) &= v_0,
\end{aligned}
$$

with c the sound celerity, t the time, x the position in the room, L the length of the room and u_0, v_0 the initial position and speed.

Note : Physics is completely different, but equation is almost the same. **Only initial conditions in space have changed.**

Algorithm modifications

As only initial conditions in space has changed, only sparse matrix has to be modified :

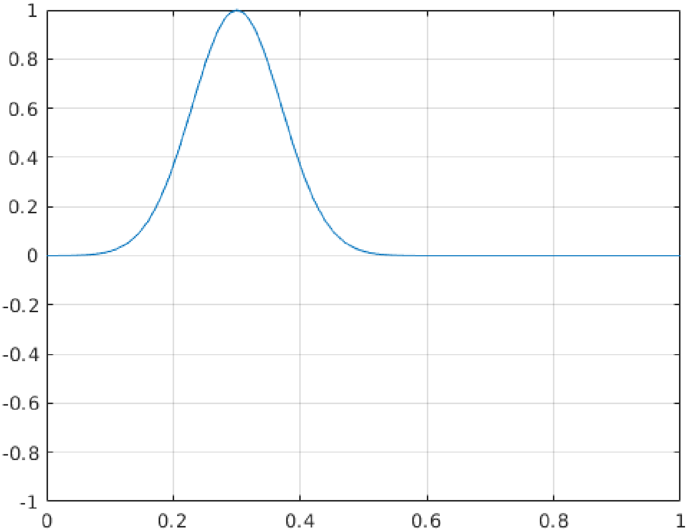
$$\begin{aligned} \frac{\partial u}{\partial x}(x=0,\cdot) &= 0 \iff \frac{u_0 - u_1}{\delta_x} = 0, \\ \frac{\partial u}{\partial x}(x=L,\cdot) &= 0 \iff \frac{u_{N-1} - u_N}{\delta_x} = 0. \end{aligned}$$

which implies :

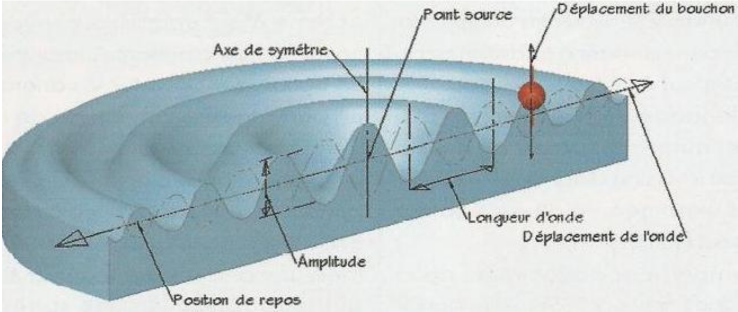
$$M_x = \frac{1}{\delta_x^2} \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 1 & -2 & 1 & \dots & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -2 & 1 \\ 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix}$$

Note : $u(x=0,\cdot) = 0$ is a **Dirichlet** condition, $\frac{\partial u}{\partial x}(x=0,\cdot) = 0$ is a **Neumann** condition.

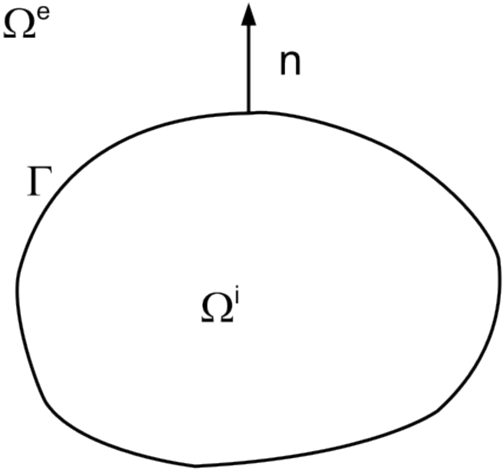
Numerical result



2-D air propagation



2-D bounded domain



2-D room acoustic

For a wave propagating in a 2-D room Ω^i, relative magnitude of the wave $u(\mathbf{x}, t) = u(x, y, t)$ is governed by a PDE :

$$\begin{aligned}
\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} - \Delta_{\mathbf{x}} u &= 0, \\
\frac{\partial u}{\partial n}(\mathbf{x} \in \Gamma, \cdot) &= 0, \\
u(\cdot, t = 0) &= u_0, \\
\frac{\partial u}{\partial t}(\cdot, t = 0) &= v_0,
\end{aligned}$$

with c the sound celerity, t the time, $\mathbf{x} = (x, y)$ the position in the room and u_0, v_0 the initial position and speed.

Reminder :

$$\Delta_{\mathbf{x}} u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

Note : Physics is completely different, but equation is almost the same. **Only initial conditions in space have changed !**

Still the same approach...

Considering a tensor product with two regular grids :

- $x_m = m\delta_{\mathbf{x}}$ for all $m \in [0, M]$,
- $y_n = n\delta_{\mathbf{x}}$ for all $n \in [0, N]$,

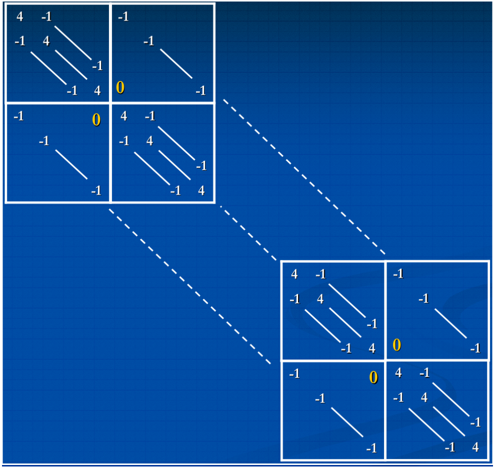
Taloyr expansion can be applied simultaneously :

$$\begin{aligned}
\Delta_{\mathbf{x}} u(x,y) &= \frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y), \\
\Delta_{\mathbf{x}} u(x,y) &= \frac{u(x+h,y) - 2u(x,y) + u(x-h,y)}{h^2} + \\
& \quad \frac{u(x,y+h) - 2u(x,y) + u(x,y-h)}{h^2} + O(h^4), \\
\Delta_{\mathbf{x}} u_{m,n} &\approx \frac{u_{m+1,n} + u_{m,n+1} - 4u_{m,n} + u_{m-1,n} + u_{m,n-1}}{\delta_{\mathbf{x}}^2}.
\end{aligned}$$

Note : It's a five points sheme (e.g. blackboard).

Algorithm

- Use sparse matrix for the discrete derivative of the Laplacian,
- Add initial condition (Dirichlet or Neumann)



Example with $-\delta_{\mathbf{x}}^2 \Delta_{\mathbf{x}}$ matrix with **Dirichlet** condition.

Algorithm

- Initialize 2 vectors of unknowns in space :

$$U^0 = \begin{pmatrix} u^0_{0,0} \\ u^0_{1,0} \\ \dots \\ u^0_{M,0} \\ u^0_{0,1} \\ u^0_{1,1} \\ \dots \\ u^0_{M,N} \end{pmatrix} = \begin{pmatrix} u(x_0, y_0, 0) \\ u(x_1, y_0, 0) \\ \dots \\ y(x_M, y_0, 0) \\ u(x_0, y_1, 0) \\ u(x_1, y_1, 0) \\ \dots \\ y(x_M, y_N, 0) \end{pmatrix}$$

  and $U^1 = U^0 + \delta_t v_0$.
- Compute Y^2, using the leap-frog scheme :

$$Y^2 = 2Y^1 - Y^0 + (c\delta_t)^2 M_{xy} Y^1$$

- Make a recursion until the final time t_f.

Starting code

```
% Clean up
clear all
close all
clc

% Physical parameters
L  = [3 2];          % Room size
X0 = [2.2 1.2];      % Initial position
tf = 5;              % Final time
c  = 1;              % Sound celerity

% Initial condition for magnitude and speed
ui = @(x,y) exp(-(x-X0(1)).^2/1e-2) .* ...
            exp(-(y-X0(2)).^2/1e-2);      % gaussian
vi = @(x,y) zeros(size(x));               % null

% Numerical discretization
dx = 0.01;            % Space step (both for x and y)
dt = 0.5*dx/c;        % Time step (CFL)
```

Numerical result