Sparse matrix representation for ODE Application to sound synthesis

Master 2 Acoustical Engineering
Numerical Techniques for Acoustics - Session 3

Matthieu Aussal*

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

Monday 30 September 2019 - ENSTA

Euler for 1st order linear equation

Application to 1-D string modelisation

Reminders : Taylor's expension end recursive Euler

For a function y twice derivable on $[t - \mathbf{h}, t + \mathbf{h}]$, Taylor's expension in the point t leads to :

$$y(t + \mathbf{h}) = y(t) + \mathbf{h}y'(t) + \frac{\mathbf{h}^2}{2}y''(t) + \frac{\mathbf{h}^3}{3!}y^{(3)}(t) + o(\mathbf{h}^4),$$

$$y(t - \mathbf{h}) = y(t) - \mathbf{h}y'(t) + \frac{\mathbf{h}^2}{2}y''(t) - \frac{\mathbf{h}^3}{3!}y^{(3)}(t) + o(\mathbf{h}^4).$$

Recursive Euler scheme is based on :

$$\frac{y(t) - y(t - h)}{h} = y'(t) + o(h).$$

Recursive Euler for 1st order linear ODE

Considering the first order **linear** Ordinary Differential Equation :

$$y'(t) + y(t) = 0 \quad \text{with} \quad y(0) = 1,$$

the analytical solution is given by :

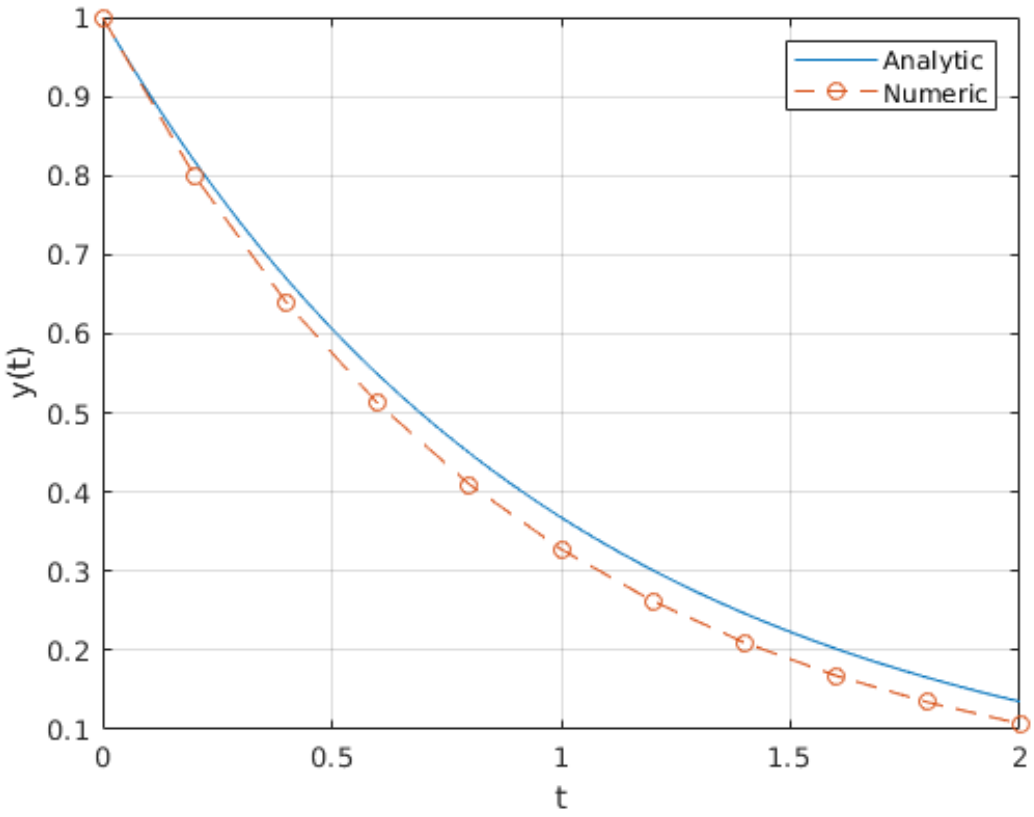
$$y(t) = y_0 e^{-t}.$$

Exercice 1 : Retrieve analytical solution. Considering a time sampling $(t_j)_{j\in[0:N]}$ of step h, compare analytical solution to a numerical resolution with a recursive Euler scheme :

$$\frac{y_j - y_{j-1}}{h} + y_{j-1} = 0 \iff y_j = y_{j-1} - h y_{j-1}.$$

Reproduce next slide figure.

Recursive Euler for 1st order linear ODE



Euler scheme with 10 steps : $\max_j (|y(t_j) - y_j)|) = 0.0402$

Sparse matrix formulation

Considering the solution vector $Y^* = (y_1, ..., y_N)^T$, the recursive Euler scheme can be rewritten using sparse matrix operators :

$$\frac{y_j - y_{j-1}}{h} + y_{j-1} = 0 \iff AY + BY = 0$$

where A, $B \in M^N(\mathbb{R})$ stand for :

$$A = \frac{1}{h}\begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ -1 & 1 & 0 & ... & 0 \\ 0 & -1 & 1 & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 & ... & 0 \\ 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 0 \end{pmatrix}$$

Exercice 2a : Find the solution of $(A + B)Y^* = 0$. What's happening ?

Sparse matrix formulation

Adding the initial condition to the linear system consist in solving matrix problem considering :

$$y_1 = y_0 - hy_0.$$

A way consist in the use of non zeros right-hand side $F = \left(\frac{(1-h)y_0}{h}, 0, ..., 0\right)^T$, such as :

$$(A + B)Y^* = F.$$

Exercice 2b : Why the first term f_1 is equal to $\frac{(1-h)y_0}{h}$? Compare solutions $(y_j)_{j\in[0:N]}$ and execution time between linear system resolution and recursive Euler scheme. Try with N increasing.

Backward Euler

Using a backward formulation from Taylor's expension :

$$\frac{y(t+h)-y(t)}{h} = y'(t) + o(h),$$

Euler scheme becomes :

$$\frac{y_j - y_{j-1}}{h} + y_j = 0 \iff AY + BY = 0,$$

modifying B matrix and F vector as :

$$B = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad F = \begin{pmatrix} \frac{y_0}{h} \\ 0 \\ \dots \\ 0 \end{pmatrix}$$

Exercice 3 : Why $f_1 = \frac{y_0}{h}$? Compare solutions $(y_j)_{j\in[0:N]}$ to forward Euler and analytical solution.

2nd order scheme : the centered scheme

A 2nd order approximation for the first derivative is build using both forward and backward formulation :

$$\frac{1}{2}\left(\frac{y(t)-y(t-h)}{h}+\frac{y(t+h)-y(t)}{h}\right)=y'(t)+o(h^2).$$

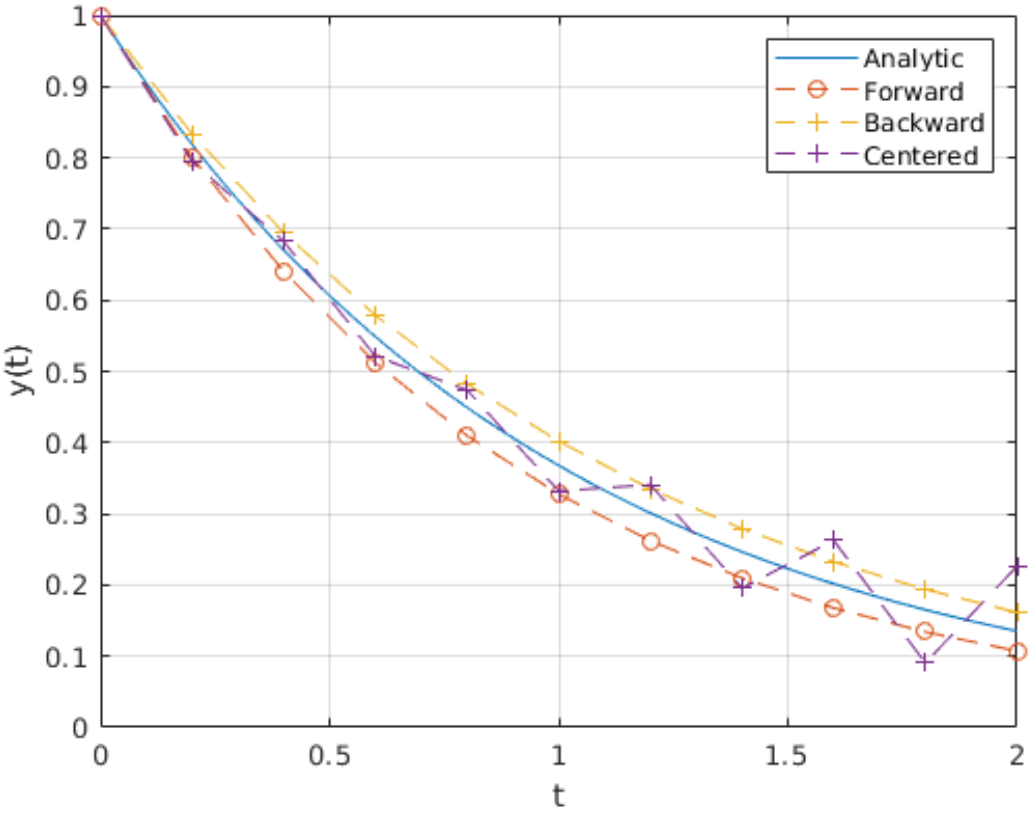
This leads to the centered scheme :

$$\frac{y_{j+1}-y_{j-1}}{2h}+y_j=0,$$

with matrix representation :

$$A=\frac{1}{2h}\begin{pmatrix}0 & 1 & 0 & \dots & 0\\ -1 & 0 & 1 & \dots & 0\\ 0 & -1 & 0 & \dots & 0\\ \dots & \dots & \dots & \dots & \dots\\ 0 & 0 & 0 & \dots & 0\end{pmatrix} \quad B=\begin{pmatrix}1 & 0 & 0 & \dots & 0\\ 0 & 1 & 0 & \dots & 0\\ 0 & 0 & 1 & \dots & 0\\ \dots & \dots & \dots & \dots & \dots\\ 0 & 0 & 0 & \dots & 1\end{pmatrix} \quad F=\begin{pmatrix}\frac{y_0}{2h}\\ 0\\ 0\\ \dots\\ 0\end{pmatrix}.$$

2nd order scheme : the centered scheme



- Forward error : 0.0402
- Backward error : 0.0340
- Centered error : 0.0911

Exercice 4 : Solve problem with Forward, Backward and Centered sheme to reproduce figure and retrieve errors. What's is going on ?

Euler for 1st order linear equation

Application to 1-D string modelisation

Time domain

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

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

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

(fr) *https ://fr.wikipedia.org/wiki/Onde_sur_une_corde_vibrante*

(en) *https ://en.wikipedia.org/wiki/String_vibration*

Fourier domain

Looking for solutions in an harmonic form (separated variables) :

$$y(x,t) = u(x)\cos(\omega t),$$

an eigen-value problem can be formalized :

$$\begin{aligned} \frac{d^2u(x)}{dx^2} &= -k^2u(x), \\ u(0) &= 0, \\ u(L) &= 0, \end{aligned}$$

where $k = \frac{\omega}{c}$ stands for the wave number. Thus, both k and $u(x)$ are unknowns.

Back to time domain

Solutions (k_n, u_n) of the eigen value equation form a basis for the time domain solution :

$$y(x,t) = \sum_{n=1}^{\infty} A_n u_n(x) \cos(\omega_n t) \quad \text{with} \quad \omega_n = c k_n$$

Considering the initial position $y(\cdot, t=0) = y_0$ of the string (loose rope case) :

$$y_0(x) = \sum_{n=1}^{\infty} A_n u_n(x),$$

wich fully determine A_n coeffiscient. Moreover, analytical solution are known as :

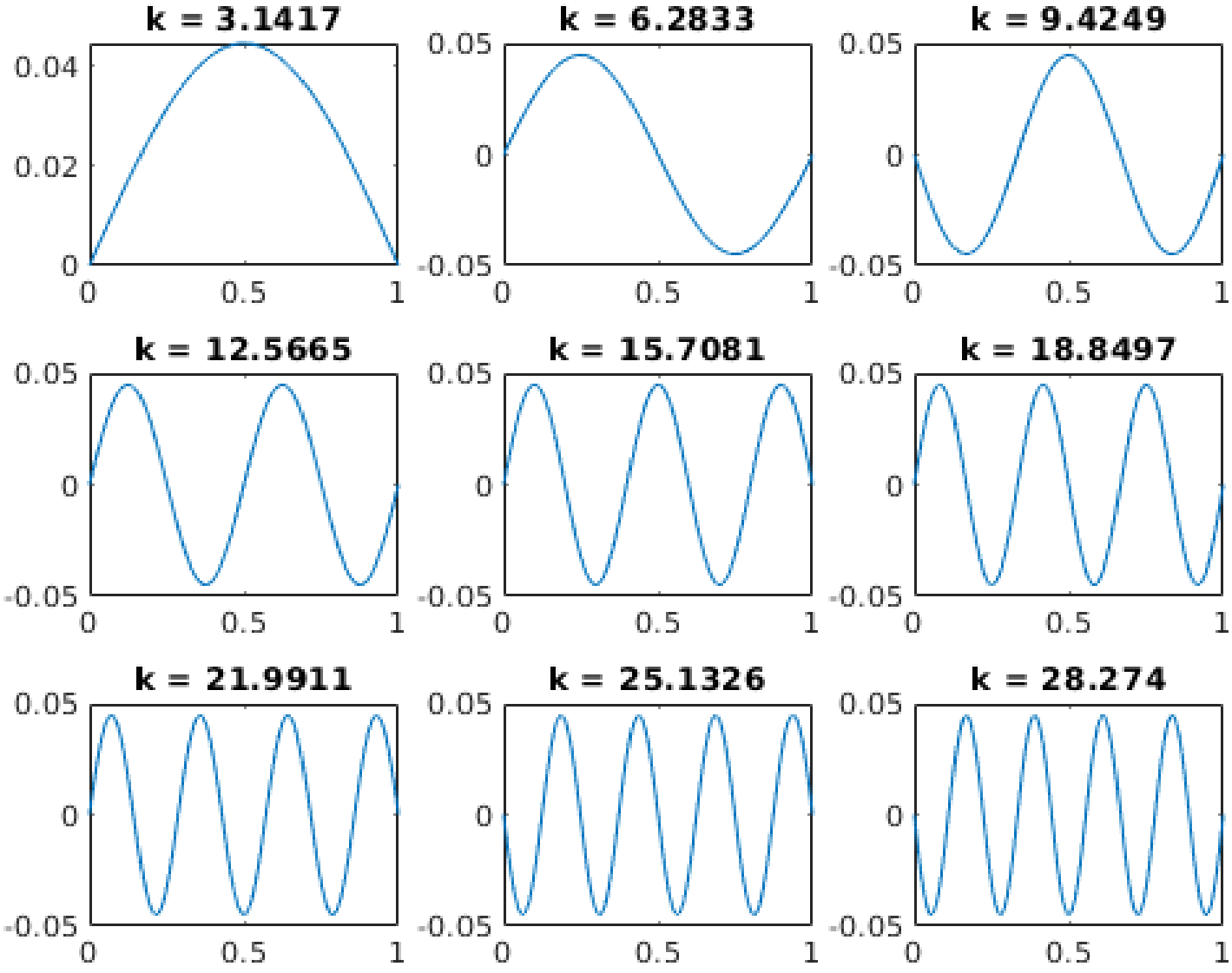
$$\begin{aligned} u_n(x) &= \sin(k_n x) \\ k_n &= \frac{n\pi}{L} \end{aligned}$$

Numerical modelisation for eigen value problem

Exercice 5 :

- Define a regular grid $X = (x_0, ..., x_N)$ to discretize the string, such as $x_0 = 0$, $x_N = L$ and $x_i = ih$, where h stand for a constant step.
- Using Taylor's expension, propose a numerical sheme to approximate $u''(x)$.
- Formulate the eigen value problem using sparse matrix.
- Add boundary conditions to the matrix formulation.
- Solve eigen value problem by forcing the smallest real eigen values (see *eigs* function).
- Plot the first nine modes (k_n, u_n) and compare to analytic values.

Numerical modelisation for eigen value problem



Numerical modelisation for time domain

Exercice 6 (facultative) :

- Define an initial position (e.g. linear hat).
- Compute A_n coefficient solving a linear system.
- Plot time domain solution using a time discretisation with 44.1 kHz sampling.
- Add wave form to the final solution (attack, release, sustain, decay) and normalize the final result.
- Compute major scale using Pythagorean dimensions ratio.
- Write wav files and listen the result.

BONUS : Define yourself Fourier series (A_n) to design your own sound, and become the next David Guetta.

Initialization of the synthesizer

```
% Gamme majeure pythagoriciene temperee
% => https://fr.wikipedia.org/wiki/Accord_pythagoricien
note  = {'DO','RE','MI','FA','SOL','LA','SI','DO'};
gamme = [ 1  9/8  81/64  4/3  3/2  27/16 243/128  2];

% Physical parameters
L  = 1/gamme(1); % string length associated to note chosen (
     here do)
N  = 1000;  % discretisation size (number of space step)
K  = 100;   % number of eigen values (eigen frequencies)
x0 = 0.2;   % mediator position (where you loose rope)
c  = 1000;  % sound celerity (m/s)
fs = 44100; % time sampling frequency (CD quality!)
```

With this configuration, verify that you find 520 Hz for the fundamental mode of fist note ($\approx DO_4$).

Useful functions

Definition of the initial position (linear hat) :

```
y0 = @(x) (x/x0).*(x<=x0) + (L-x)/(L-x0).*(x>x0);
```

Definition of a wave form :

```
waveForm = @(t) (t<=0.02).*(t/0.02) + ...  % Attack
(t>0.02) .* (t<=0.025) .* (2-t/0.02)  + ...% Decay
(t>0.025) .* (t<=0.1)  .* (0.75) + ...     % Sustain
(t>0.1)  .* (0.75*exp(-10*(t-0.1)));       % Release
```

Final solution for DO_4