

Prediction models in building engineering Calculus, preliminaries

Toni Guillamon

Matemàtica Aplicada I - UPC

September 22th, 2015

An equation to start...

$$\frac{du(t)}{dt} = k(u_0 - u(t))$$



The importance of differential equations: an example

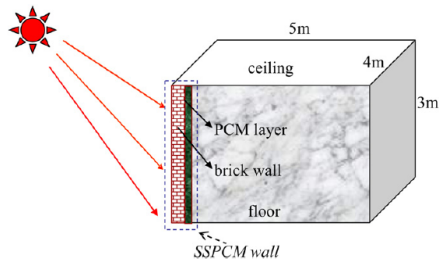


Fig. 1. Schematic of SSPCM wall and the ideal model house.

N. Zhu 1724 et al. / International Journal of Thermal Sciences 49 (2010) 1722e1731

GOAL: To provide accurate and reliable prediction of the thermodynamic performance of the building structures integrated with a shape-stabilized phase change material (SSPCM) layer.

The importance of differential equations: an example

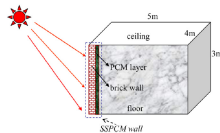


Fig. 1. Schematic of SSPCM wall and the ideal model house.

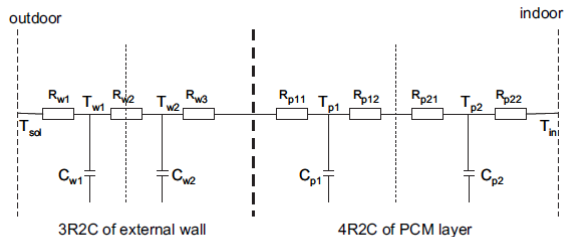


Fig. 2. Schematic of the simplified dynamic building model.

N. Zhu 1724 et al. / International Journal of Thermal Sciences 49 (2010) 1722e1731

The importance of differential equations: an example

$$C_{w1} \frac{dT_{w1}(t)}{dt} = \frac{T_{sol}(t) - T_{w1}(t)}{R_{w1}} - \frac{T_{w1}(t) - T_{w2}(t)}{R_{w2}} \quad (1)$$

$$C_{w2} \frac{dT_{w2}(t)}{dt} = \frac{T_{w1}(t) - T_{w2}(t)}{R_{w2}} - \frac{T_{w2}(t) - T_{p1}(t)}{R_{w3} + R_{p11}} \quad (2)$$

$$C_{p1} \frac{dT_{p1}(t)}{dt} = \frac{T_{w2}(t) - T_{p1}(t)}{R_{w3} + R_{p11}} - \frac{T_{p1}(t) - T_{p2}(t)}{R_{p12} + R_{p21}} \quad (3)$$

$$C_{p2} \frac{dT_{p2}(t)}{dt} = \frac{T_{p1}(t) - T_{p2}(t)}{R_{p12} + R_{p21}} - \frac{T_{p2}(t) - T_{in,a}(t)}{R_{p22}} \quad (4)$$

$$C_{air} \frac{dT_{in,a}(t)}{dt} = \frac{T_{p2}(t) - T_{in,a}(t)}{R_{p22} + R_{air}} + Q_{oe} \quad (5)$$

$$Q_{in,s} = \frac{T_{p2}(t) - T_{in,a}(t)}{R_{p22} + R_{air}} \quad (6)$$

N. Zhu 1724 et al. / International Journal of Thermal Sciences 49 (2010) 1722e1731

The importance of differential equations: an example

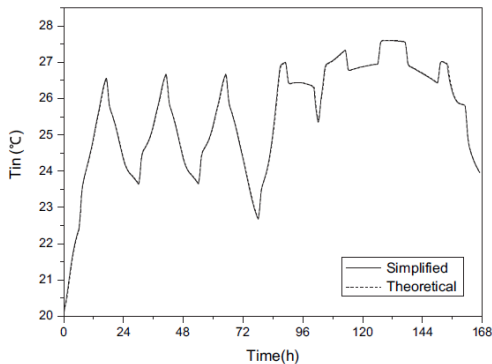


Fig. 10. T_{in} of light wall with $D_{SSPCM} = 10$ mm.

N. Zhu 1724 et al. / International Journal of Thermal Sciences 49 (2010) 1722e1731

Differential equations and dynamical systems

Trajectories and attractors

<https://www.youtube.com/watch?v=v1sbetUIHNM>

(Partial) differential equations: the wave equation

$u(x, y, t)$ Deflection of membrane from equilibrium at position (x, y) and time t .

The surface $z = u(x, y, t)$ gives the shape of the membrane at time t . Under ideal assumptions (e.g. uniform membrane density, uniform tension, no resistance to motion, small deflection, etc.) one can show that u satisfies the two dimensional wave equation:

$$u_{tt} = c^2 \nabla^2 u = c^2 (u_{xx} + u_{yy}),$$

for $0 < x < a$, $0 < y < b$.

Simulation of the wave equation

<https://www.youtube.com/watch?v=ySa6p5KGTxc>

Difference quotients

Let $f(x)$ be a continuous function defined on $D \subseteq \mathbb{R}$.

We define the **difference quotient** for $x, x_0 \in D$ as

$$\frac{f(x) - f(x_0)}{x - x_0}.$$

This correspond to the concepts of **secant**, **mean velocity** and, in general, an **average variation** of the function f along the interval $\langle x, x_0 \rangle$.

Derivatives

We define the **derivative of f at $x_0 \in D$** as

$$f'(x_0) := \frac{df(x)}{dx}(x_0) := \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}.$$

This corresponds to the concepts of **tangent**, **instantaneous velocity** and, in general, the **variation** of the function f at the point x_0 . Note that it involves the concept of limit.

Background training: You may need to go through examples of how to perform derivatives and their importance in optimization problems. Higher order derivatives have also implications in other aspects of the function (curvatures, ...).

Again the same equation...

$$\frac{du(t)}{dt} = k(u_0 - u(t))$$

k, u_0 being real constants.

- Could you give a sense to this equation now?
- Would you be able to solve it? Analytically, numerically, ...?

Maple computations

I. Derivatives and linear differential equations

Mini-project proposal on Newton's law of cooling I

We know that the Newton's law of cooling provides a formula (differential equation) for the dynamics of the temperature $T = T(t)$ of a body immersed in an environment with temperature T_A :

$$\frac{dT}{dt} = \alpha (T_A - T),$$

where α is called the *cooling rate*, can be solved as $T(t) = T_A + c \exp(-\alpha t)$, where c is a real constant.

- 1 Given T_A , explain what would you measure in order to obtain α . In particular, discuss the physical role of the constant c ?

Mini-project proposal on Newton's law of cooling II

- 2 Imagine that the environment temperature is variable, given by $T_A(t) = 20 + 4 \sin(1/12 \pi (t - 8))$, in Celsius degrees, where t units are hours. Using Maple compute and represent the temperature of a body immersed in this environment along one day ($t \in [0, 24]$) if $T(0) = 37^\circ C$. For different α values ($\alpha \in \{0.01, 0.03, 0.09\} h^{-1}$) check and interpret the role of the cooling rate. Figure 1 shows some instances.
- 3 Obtain an exact solution by means of Maple and determine the asymptotic behaviour.

Mini-project proposal on Newton's law of cooling III

Evolució de la temperatura (10 dies) per a diferents taxes de refredament

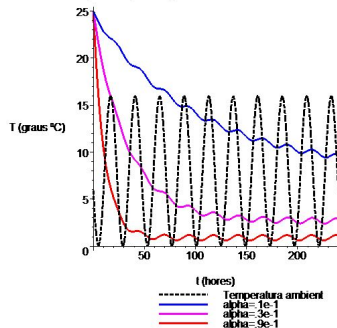


Figure : Temperature evolution of a body immersed in a variable environment. The figure shows the oscillations of the temperatures (environment and body) along 10 days. The initial temperature of the body was set at 25°C for different cooling rates ($\alpha \{0.01, 0.03, 0.09\} \text{ h}^{-1}$). The solar radiation was taken different from that of the statement.

Scalar fields

A **scalar field** is a function f from a certain domain $U \subset \mathbb{R}^n$ to a real number.

$$\begin{aligned} f : U \subset \mathbb{R}^n &\longrightarrow \mathbb{R} \\ (x_1, x_2, \dots, x_n) &\longmapsto f(x_1, x_2, \dots, x_n), \end{aligned}$$

Examples: $f(x, y) = x^2 + y^2$, $f(x, y, z) = \log(1 + z^2)(x + y), \dots$

Derivatives in higher dimensions

Let $f(x)$ be a continuous function defined on $D \subseteq \mathbb{R}^n$.

We define the **derivative of f at $x_0 \in D$** along the direction $\vec{v}$ (unitary vector) as

$$\lim_{h \rightarrow 0} \frac{f(x_0 + h \vec{v}) - f(x_0)}{h}.$$

It is called **directional derivative** and can be computed as:

$$D_{\vec{v}}f(x_0) = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right) \bigg|_{(x_0, y_0)} \cdot \vec{v}$$

Background training: Partial derivatives $\frac{\partial f}{\partial x_j}$.

Examples:

- $f(x, y) = x^2 + y^2$; $\frac{\partial f}{\partial x} = 2x$, $\frac{\partial f}{\partial y} = 2y$, $\frac{\partial^2 f}{\partial x^2} = 2$, ...
- $f(x, y, z) = \log(1 + z^2)(x + y)$; $\frac{\partial f}{\partial x} = \log(1 + z^2)$, $\frac{\partial f}{\partial y} = \log(1 + z^2)$,
 $\frac{\partial f}{\partial z} = \frac{(x+y)2z}{1+z^2}$, ...

Graphics of a scalar field: surfaces I

Consider $f : U \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$.

$\text{Graph}(f) = \{(x, y, z) \in \mathbb{R}^3 : z = f(x, y)\}$, thus obtaining a **surface** S .

Graphics of a scalar field: surfaces II

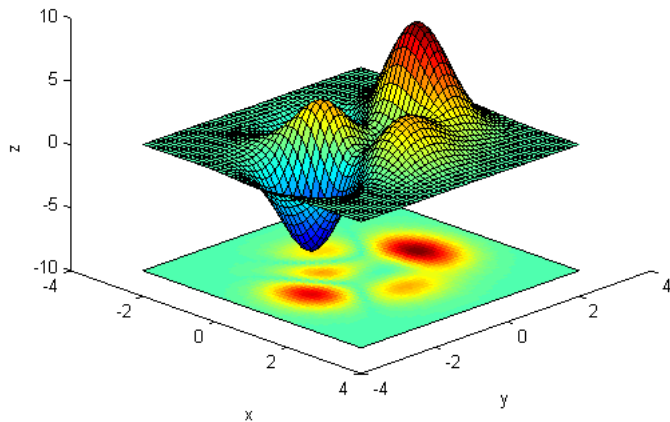


Figure : Plot of $z = f(x, y)$

Level sets I

When plotting S many difficulties may arise. The **level sets** give a good approximation of the graph. They consist of intersecting S with planes $z = k$, thus obtaining a curve given by $f(x, y) = k$.

Level sets II

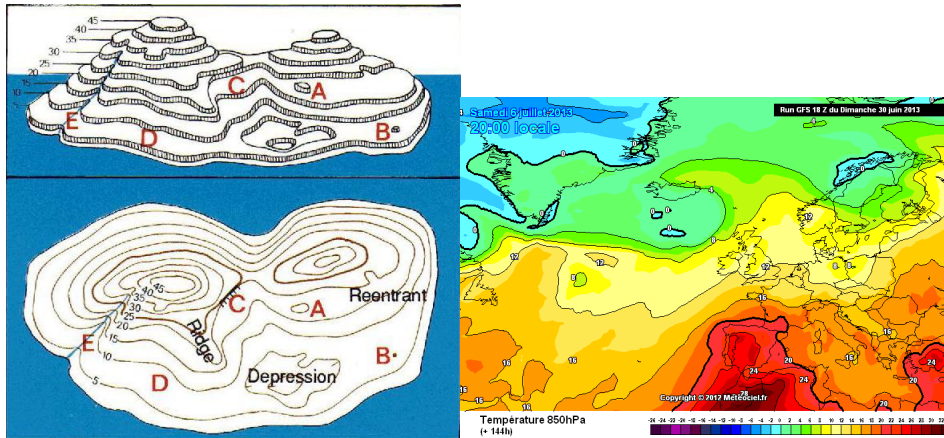


Figure : Examples of level sets. Altitude and thermal examples of level sets.

Level sets III

Question: If the levels are equispaced, what is the meaning of the level curves being less or more separated?

Maple computations

II. Derivatives in scalar fields

Mini-project proposal on level sets

We have explored the `DropDown` function in a previous example. We propose a modification of that program: to **design a path** that climbs the surface given by $z = f(x, y)$ **with a prescribed** positive **slope** $\sigma(x, y)$, where

$$\sigma(x, y) = \min\{2\%, \|\nabla f(x, y)\|\},$$

starting from the point (x_0, y_0) , where x_0 and y_0 are the second and the third coordinate of the m -th triplet of the set of points

`InitialConditionsSet` and m is index assigned to your working group.

- Compute the path until you reach a maximum point. Give a graphical representation of the solution obtained both in 3D and on the 2D plane, using the `contourplot` Maple command in the latter case.
- Compare different integration steps (3 orders of magnitude for instance). Why the solution is not the same?

Hint: It will be useful to activate the lines of the Maple code with the `#U#` notation.

Taylor expansions for scalar fields

- Let $(\mathbf{x}_0, \mathbf{y}_0) \in \mathcal{U} \subset \mathbb{R}^2$, be a point where f and its derivatives are known up to order r .
- Let $\mathcal{U}$ be an open ball centered at $(\mathbf{x}_0, \mathbf{y}_0)$.
- Let $(\mathbf{x}, \mathbf{y}) = (\mathbf{x}_0 + h_1, \mathbf{y}_0 + h_2) \in \mathcal{U} \subset \mathbb{R}^2$ be any point in a neighbourhood of $(\mathbf{x}_0, \mathbf{y}_0)$ on which f is “difficult” to compute.

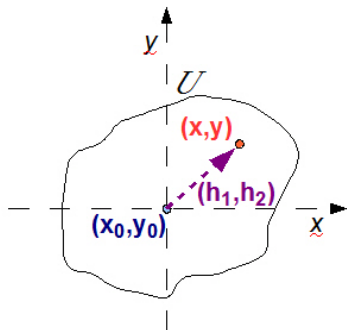


Figure : Essential elements for the Taylor expansion: domain.

Taylor expansions for scalar fields

- Let $f : \mathcal{U} \rightarrow \mathbb{R}$ be a function with continuous partial derivatives up to (at least) order 2.

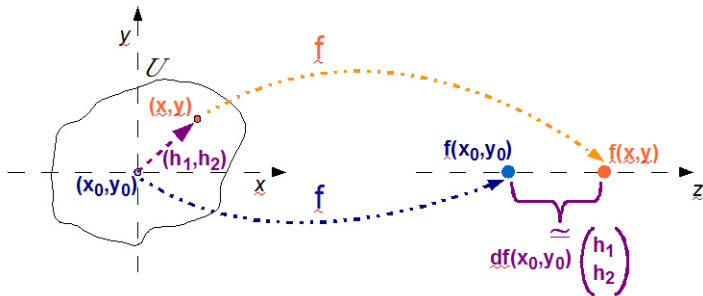


Figure : Essential elements for the Taylor expansion: function.

Taylor expansions for scalar fields

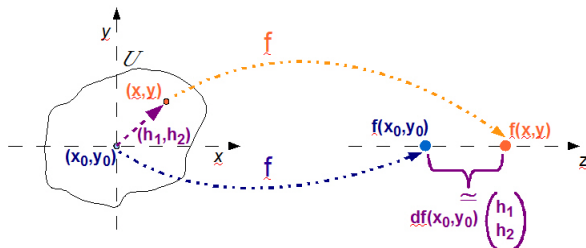


Figure : Essential elements for the Taylor expansion: function.

$$\begin{aligned}
 f(x, y) &= f(x_0, y_0) + \left[\frac{\partial f}{\partial x}(x_0, y_0)h_1 + \frac{\partial f}{\partial y}(x_0, y_0)h_2 \right] \\
 &+ \frac{1}{2!} \left[\frac{\partial^2 f}{\partial x^2}(x_0, y_0)h_1^2 + 2 \frac{\partial^2 f}{\partial x \partial y}(x_0, y_0)h_1 h_2 + \frac{\partial^2 f}{\partial y^2}(x_0, y_0)h_2^2 \right] + \dots
 \end{aligned}$$

Taylor expansions for scalar fields

$$\begin{aligned}
 f(x, y) &= f(x_0, y_0) + \left[\frac{\partial f}{\partial x}(x_0, y_0)h_1 + \frac{\partial f}{\partial y}(x_0, y_0)h_2 \right] \\
 &+ \frac{1}{2!} \left[\frac{\partial^2 f}{\partial x^2}(x_0, y_0)h_1^2 + 2\frac{\partial^2 f}{\partial x \partial y}(x_0, y_0)h_1 h_2 + \frac{\partial^2 f}{\partial y^2}(x_0, y_0)h_2^2 \right] + \dots \\
 &= f(x_0, y_0) + \left(\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right)_{(x_0, y_0)} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \\
 &+ \frac{1}{2!} (h_1 \quad h_2) \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}_{(x_0, y_0)} \begin{pmatrix} h_1 \\ h_2 \end{pmatrix}
 \end{aligned}$$

The differential and hessian matrices

Matrix of the differential:

$$df_{(x_0, y_0)} = \left(\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right)_{(x_0, y_0)}$$

The **differential operator** is a linear map that assigns the slope of the tangent plane when applied to any direction (h_1, h_2) .

The **Hessian matrix**:

$$Hf_{(x_0, y_0)} := \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}_{(x_0, y_0)}$$

The **Hessian matrix** defines a symmetric bilinear map that assigns a real number (related to the curvatures of the surface) to any direction (h_1, h_2) .

Taylor en 2D, in terms of the differential

We denote $p = (x, y)$, $p_0 = (x_0, y_0)$, $\Delta p = p - p_0 = (h_1, h_2)$ and $\Delta f = f(p) - f(p_0)$; then, the Taylor expansion up to order 1 writes as:

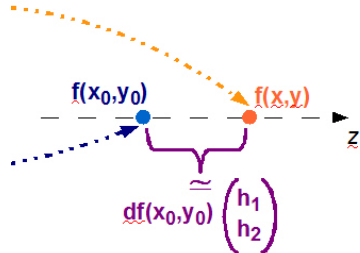


Figure : Linearization: the differential map.

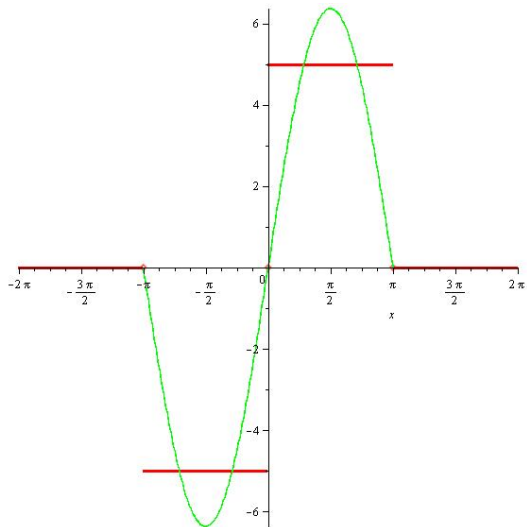
$$\Delta f = df_{p_0} \Delta p,$$

which is the compact expression of the *linearization* of the scalar field f in a neighbourhood of p_0 .

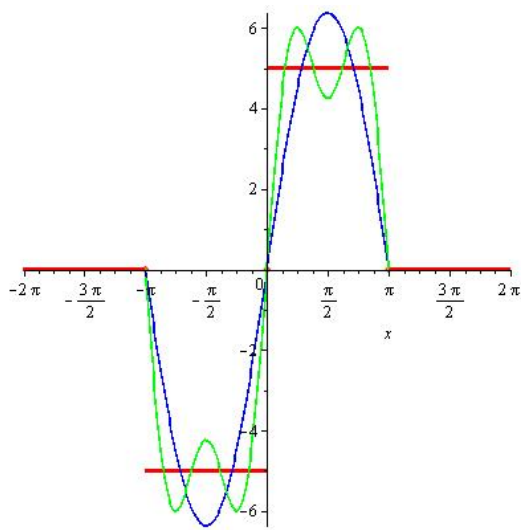
Fourier expansions I

Representation of periodic functions as a sum (possibly infinite) of simple (harmonically related) sine and cosine waves.

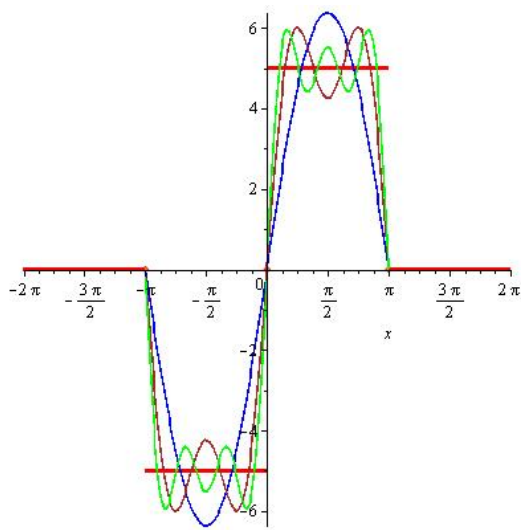
Fourier expansions II



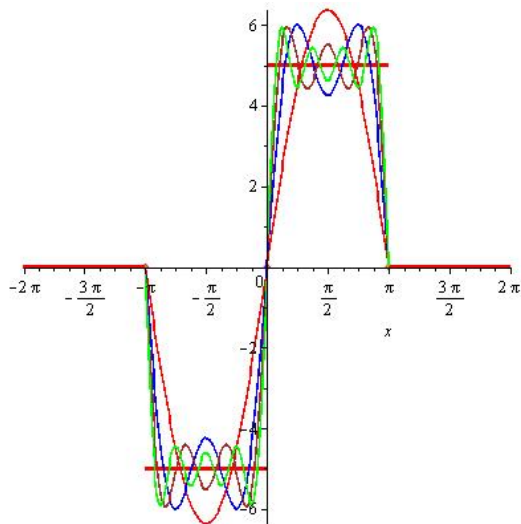
Fourier expansions III



Fourier expansions IV



Fourier expansions V



Fourier series I

In this section, $s(x)$ denotes a function of the real variable x , and s is integrable on an interval $[x_0, x_0 + T]$, for real numbers x_0 and T , being T the minimal period of s .

$$\begin{aligned} s_N(x) &= \frac{a_0}{2} + \sum_{n=1}^N a_n \cos\left(\frac{2\pi nx}{T}\right) + b_n \sin\left(\frac{2\pi nx}{T}\right) \\ &= \sum_{n=-N}^N c_n e^{i\frac{2\pi nx}{T}}, \end{aligned}$$

where

$$c_n \stackrel{\text{def}}{=} \begin{cases} \frac{1}{2}(a_n - i b_n) & \text{for } n > 0 \\ \frac{1}{2}a_0 & \text{for } n = 0 \\ c_{|n|}^* & \text{for } n < 0. \end{cases}$$

Fourier series II

The **Fourier coefficients** are computed as follows:

$$a_n = \frac{2}{T} \int_{x_0}^{x_0+T} s(x) \cdot \cos\left(\frac{2\pi n x}{T}\right) dx$$

$$b_n = \frac{2}{T} \int_{x_0}^{x_0+T} s(x) \cdot \sin\left(\frac{2\pi n x}{T}\right) dx$$

$$c_n = \frac{1}{T} \int_{x_0}^{x_0+T} s(x) \cdot e^{-i\frac{2\pi n x}{T}} dx.$$

For square-integrable functions $s(x) = \lim_{N \rightarrow \infty} s_N(x)$ except at the discontinuities.

Prediction models in building engineering

Linear Numerical Algebra

Toni Guillamon

Matemàtica Aplicada I - UPC

October 6th, 2015

Classification of linear systems

$$\mathbf{A} \mathbf{x} = \mathbf{b},$$

where $A \in M_{m \times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $m, n \in \mathbb{N}$. (m equations, n variables).

- **Compatible.** $\text{rg}A = \text{rg}(A|b)$.
 - Unique solution (**determined**). $\text{rg}A = \text{rg}(A|b) = n$.
 - Infinite solutions (**undetermined**, k degrees of freedom).
 $\text{rg}A = \text{rg}(A|b) = n - k$.
- **Incompatible.** $\text{rg}A \neq \text{rg}(A|b)$.

Classification of linear systems: examples I

- **Compatible.** $\text{rg}A = \text{rg}(A|b)$.
 - Unique solution (**determined**). $\text{rg}A = \text{rg}(A|b) = n$.

$$\begin{pmatrix} 2 & 3 \\ -1 & 3 \\ 1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ -2 \\ 3 \end{pmatrix}$$

$$m = 3, n = 2, \text{rg}A = \text{rg}(A|b) = 2 = n;$$

$$x_1 = 7/3, \quad x_2 = 1/9.$$

Classification of linear systems: examples II

- Infinite solutions (**undetermined**, k degrees of freedom).

$$\begin{pmatrix} 2 & 3 & 5 \\ -1 & 3 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

$$m = 2, n = 3, \text{rg}A = \text{rg}(A|b) = 2 = n - 1 \text{ (1 d.o.f);}$$

$$x_1 = 2/3 - x_3/3, \quad x_2 = 11/9 - 13/9 x_3, \quad x_3 = x_3.$$

Classification of linear systems: examples III

- **Incompatible.** $\text{rg}A \neq \text{rg}(A|b)$.

$$\begin{pmatrix} 2 & 3 \\ -1 & 3 \\ 1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ -2 \\ 3 \end{pmatrix}$$

$$m = 3, n = 3, \text{rg}A = 2 \neq \text{rg}(A|b) = 3.$$

Are they really negligible systems?

They appear quite naturally in applications and are usually solved using the **Least Squares Method** (LSM).

What about big matrices?

For instance, in **structural engineering** (trusses, grid structures, *estructuras reticulares*).

The **stiffness matrix** is constructed using E (Young modulus), I (inertia moment of the bar), A (transversal section area) and L (length of the bar). The result is a (6×6) symmetric local stiffness matrix.



What about big matrices?

Complex 3D
structures...

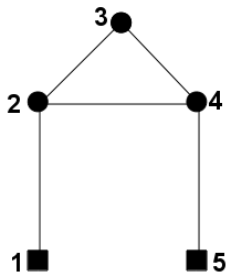


A single bar
in 2D...



$$K^L = \begin{pmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{pmatrix}$$

Example: stiffness matrices



Denoting $K^{r,s}$ the **global** (via rotation) **stiffness matrix** of the bar that joins nodes r and s , then the **stiffness matrix of the whole structure** becomes a (15×15) matrix.

Example: stiffness matrices

Denoting $K^{r,s}$ the **global** (via rotation) **stiffness matrix** of the bar that joins nodes r and s , then the **stiffness matrix of the whole structure** becomes a (15×15) matrix:

$$K_0 = \begin{pmatrix} K_{1,1}^{1,2} & K_{1,2}^{1,2} & 0 & 0 & 0 \\ K_{1,1}^{2,1} & K_{2,2}^{1,2} + K_{1,1}^{2,3} + K_{1,1}^{2,4} & K_{1,2}^{2,3} & K_{1,2}^{2,4} & 0 \\ 0 & K_{2,1}^{2,3} & K_{2,2}^{2,3} + K_{1,1}^{3,4} & K_{1,2}^{3,4} & \\ 0 & K_{2,1}^{2,4} & K_{2,1}^{3,4} & K_{2,2}^{2,4} + K_{2,2}^{3,4} + K_{1,1}^{4,5} & K_{1,2}^{4,5} \\ 0 & & & K_{2,1}^{4,5} & K_{2,2}^{4,5} \end{pmatrix}$$

Linear numerical algebra I

We would like to solve $Ax = b$, where x is the unknown and A and b are known data. We assume that A is a square matrix and writes as:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

How to solve the system $Ax = b$ for **big values of n** ?

The most useful **methods** consist of factorizing the matrix A as the product of two matrices that can have different *suitable* features:

Linear numerical algebra II

- the LU method is based on decomposing $A = L U$, being L a **lower triangular** matrix and U an **upper triangular** one. This method is based on the process of Gaussian elimination (indeed, its matrix form).

$$L = \begin{pmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{pmatrix}; \quad U = \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{pmatrix}.$$

Linear numerical algebra III

- the QR method is based on decomposing $A = Q R$, being Q an **orthogonal matrix** and R an **upper triangular** one. This method is based on the Gram-Schmidt orthogonalization.

$$Q^t Q = \text{Id}; \quad R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & r_{nn} \end{pmatrix}.$$

LU method I

The LU method is based on decomposing $A = L U$, being L a **lower triangular** matrix and U an **upper triangular** one.

Sometimes

$$P A = L U, \quad P \text{ (permutation matrix),}$$

arising from permutation of rows in A in order to use the maximum possible **pivot** in every step. Otherwise, the method can be numerically unstable. In these notes, we proceed with $P = \text{Id}$.

You can follow http://en.wikipedia.org/wiki/LU_decomposition for more information

Let us see an example, and then we will use it inside a Maple program.

Doolittle algorithm I

We create a sequence of matrices $A^{(n)}$ such that:

$$A^{(n)} = L_n A^{(n-1)}, \quad \text{with } A^{(0)} = A,$$

and:

$$L_n = \begin{pmatrix} 1 & & & & & 0 \\ & \ddots & & & & \\ & & 1 & & & \\ & & l_{n+1,n} & \ddots & & \\ & & \vdots & & \ddots & \\ 0 & & l_{N,n} & & & 1 \end{pmatrix}.$$

with $l_{i,n} := -\frac{a_{i,n}^{(n-1)}}{a_{n,n}^{(n-1)}}$ for $i = n+1, \dots, N$.

Doolittle algorithm II

After $N - 1$ steps,

$U := A^{(N-1)}$ is upper triangular and,

$L := L_1^{-1} \dots L_{N-1}^{-1}$ is lower triangular.

In particular,

$$L = \begin{pmatrix} 1 & & & & 0 \\ -l_{2,1} & \ddots & & & \\ & 1 & & & \\ \vdots & & -l_{n+1,n} & \ddots & \\ & \vdots & & 1 & \\ -l_{N,1} & & -l_{N,n} & & -l_{N,N-1} & 1 \end{pmatrix}.$$

Doolittle algorithm III

Clearly, we need to have $a_{n,n}^{(n-1)} \neq 0$ at each step. Otherwise we could do row permutations and obtain a $PA = LU$ decomposition, where P is a permutation matrix.

Example:

$$L = \begin{pmatrix} 3 & 1 & 4 \\ -2 & -4 & 1 \\ 1 & 1 & -2 \end{pmatrix}.$$

Advantages of $A = LU$ decomposition I

Assume A is an $n \times n$ square matrix.

- **Solving a linear system** $Ax = b$ reduces to solve two triangular systems; each of them can be performed (by simple substitution) in $O(n^2)$ floating point operations:
 - Step 1: Solve $Ly = b$.
 - Step 2: Solve $Ux = y$.
 - Conclusion: Since $Ax = LUx = Ly = b$, then x is the solution.
- **Computational efficiency:** Taking into account the decomposition and the resolution of the linear triangular systems, the whole process takes $2n^2/3$ floating point operations.

•

$$\det A = \det U = \prod_{i=1}^n u_{ii}.$$

Advantages of $A = L U$ decomposition II

- To compute A^{-1} , the **inverse matrix** of A , we solve the n systems $A x_i = e_i$, where e_i is the unit vector having a 1 in the i -th component. Note that these n systems share the same A matrix, so we only need 1 LU decomposition.
- If A is a **symmetric** matrix, it is possible to choose U to be the conjugate transpose of L , so that

$$A = L L^T,$$

which is called the **Cholesky decomposition**. The Cholesky decomposition always exists and is unique. For instance, stiffness matrices are symmetric.

Advantages of $A = L U$ decomposition III

- For **large sparse matrices**, special factorization algorithms have been developed. The idea is setting the sparse structure of L and U beforehand and then compute only their non-zero terms. As a rule of thumb, the cost of computation is determined by the number of nonzero entries.

Eigenvectors and eigenvalues I

Given a matrix $A \in M_{n \times n}(\mathbb{R})$, a vector $v \in \mathbb{R}^n$ is said to be an **eigenvector** (**vector propio**) of A if:

$$A v = \lambda v,$$

for some $\lambda \in \mathbb{R}$.

In this situation, λ is said to be the **eigenvalue** (**valor propio**) associated to v .

Properties:

- A matrix A can have at most n eigenvectors and n eigenvalues. They are computed from the **characteristic polynomial** of A . Indeed, they are the solutions of

$$\det(A - x \text{Id}) = 0.$$

Eigenvectors and eigenvalues II

- In case A has n real eigenvalues and diagonalizes, it admits the following decomposition:

$$A = V D V^{-1},$$

where the i -th column of V is the i -th eigenvector (for $i = 1, \dots, n$) and D is a diagonal matrix such that d_{ii} is the i -th eigenvalue of A .

- A numerical algorithm to find the eigenvalue with the maximum absolute value (say λ_1) is the **power method**:

Taking $b_{k+1} = \frac{A b_k}{\|A b_k\|}$ after an initial vector b_0 , the quotients

$$\eta_k := \frac{b_k^T A b_k}{b_k^T b_k}$$

converge to λ_1 .

References

- Golub, G.H., van Loan, C.F., “Matrix Computations”. 1992.
- Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T
“Numerical recipes: The art of scientific computing”. Cambridge Univ. Press, 1989.

Heat equation: solving the PDE on a computer

Toni Guillamon

Matemàtica Aplicada I - UPC

October 13th, 2015

Finite difference method (FDM) I

Finite differences¹

- Finite-difference methods (FDM) are numerical methods for approximating the solutions to differential equations using finite difference equations to approximate the derivatives.
- The finite differences are obtained from the Taylor series expansions:

$$f(a + h) = f(a) + f'(a)h + \frac{f''(a)}{2!}h^2 + \cdots + \frac{f^{(n)}(a)}{n!}h^n + R_n(x),$$

assuming that we are looking for functions with at least n orders of differentiability.

- In particular,

Finite difference method (FDM) II

Finite differences²

- The expressions

$$f'(a) \sim \frac{f(a+h) - f(a)}{h},$$
$$f'(a) \sim \frac{f(a+h) - f(a-h)}{2h}$$

are good approximations of the first derivative at the point a , called **lateral/forward** and **central** derivative approximations, respectively.

- The expressions

$$f''(a) \sim \frac{f(a+2h) - 2f(a+h) + f(a)}{h^2},$$
$$f''(a) \sim \frac{f(a+h) - 2f(a) + f(a-h)}{h^2}$$

are good approximations of the second derivative at the point a , called **lateral/forward** and **central** derivative approximations, respectively.

¹A review of the first chapter on Calculus preliminaries can be useful. Specially, concerning the definition of incremental quotients, partial derivatives and Taylor's expansions.

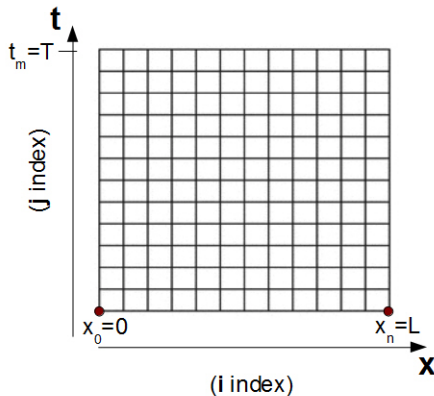
Finite difference method (FDM) I

Discretization on a grid

- We have to discretize the problem's domain, generally using a uniform grid.

Finite difference method (FDM) II

Discretization on a grid



Finite difference method (FDM) I

Example: Heat transfer in a one-dimensional rod



We have to solve the **heat equation** for $u = u(x, t)$:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2},$$

where $u(t, x)$ is the temperature at a point x at time t , and α is the thermal diffusivity, restricted to the **boundary conditions**:

$$\begin{aligned} u(x, 0) &= f(x), & 0 \leq x \leq L, \\ \begin{cases} u(0, t) &= g_1(t), \\ u(L, t) &= g_2(t), \end{cases} & 0 \leq t \leq T. \end{aligned}$$

Finite difference method (FDM) I

Example: Heat transfer in a one-dimensional rod

Fix n and m natural numbers, and define the **discretization steps** $h := L/n$, $k := T/m$.

$$x_0 = 0, x_n = L, t_0 = 0, t_m = T; \quad x_i = x_0 + i h, \quad t_j = j k.$$

We discretize the derivatives appearing in the heat equation:

$$u_t(x_i, t_j) = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{k} =: \frac{u_{i,j+1} - u_{i,j}}{k};$$

$$u_{xx}(x_i, t_j) = \frac{u(x_{i+1}, t_j) - 2 u(x_i, t_j) + u(x_{i-1}, t_j))}{h^2} =: \frac{u_{i+1,j} - 2 u_{i,j} + u_{i-1,j}}{h^2},$$

where $u_{i,j} := u(x_i, t_j)$.

Explicit Euler method I

A straightforward computation, using the **forward difference** for time, and the **central difference** for space, gives:

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \alpha \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2},$$

that is,

$$u_{i,j+1} - u_{i,j} = \frac{\alpha k}{h^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}).$$

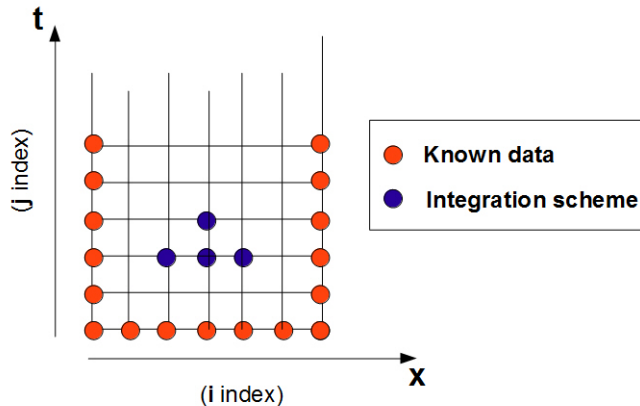
Then,

$$u_{i,j+1} = r u_{i+1,j} + (1 - 2r) u_{i,j} + r u_{i-1,j},$$

$$\text{where } r = \frac{\alpha k}{h^2} = \frac{\alpha \Delta t}{(\Delta x)^2}.$$

See the Maple file `HeatEquation.mws` for an **example**.

Explicit Euler method II



Explicit Euler method: summary

Main equation:

$$u_{i,j+1} = r u_{i+1,j} + (1 - 2r) u_{i,j} + r u_{i-1,j}, \quad r = \frac{\alpha k}{h^2} = \frac{\alpha \Delta t}{(\Delta x)^2}.$$

Boundary conditions:

$$u_{i,0} = u(x_i, 0), \quad i = 0, \dots, n$$

$$u_{0,j} = u(0, t_j), \quad j = 0, \dots, m$$

$$u_{n,j} = u(L, t_j), \quad j = 0, \dots, m$$

Sources of error in FDM and stability

The two sources of error in FDM are the **round-off error**, the loss of precision due to computer rounding of decimal quantities, and the **discretization error**, the difference between the exact solution of the finite difference equation and the exact quantity assuming no round-off.

Besides these sources of error, the Euler explicit method becomes **unstable** if $r = \frac{\alpha \Delta t}{(\Delta x)^2} \geq 0.5$.

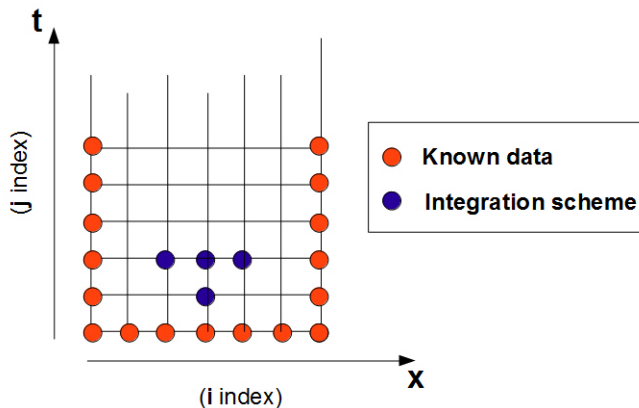
The numerical errors are proportional to the time step and the square of the space step:

$$\Delta u = O(k) + O(h^2).$$

Implicit Euler method I

To overcome the problem of **unstabilities**, one considers a different scheme, using the **backward difference** for time and keeping the **central difference** for space, which turns out to be **unconditionally stable**:

Implicit Euler method II



Implicit Euler method III

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \alpha \frac{u_{i+1,j+1} - 2 u_{i,j+1} + u_{i-1,j+1}}{h^2},$$

that is,

$$u_{i,j+1} - u_{i,j} = r (u_{i+1,j+1} - 2 u_{i,j+1} + u_{i-1,j+1}).$$

Then,

$$-r u_{i-1,j+1} + (1 + 2r) u_{i,j+1} - r u_{i+1,j+1} = u_{i,j}.$$

So, for every $j = 1, \dots, m-1$, the elements $u_{i,j+1}$ are obtained from the elements $u_{i,j}$ by solving the following linear algebra system:

$$\begin{pmatrix} 1+2r & -r & & & \\ -r & 1+2r & -r & & \\ & -r & 1+2r & -r & \\ & & \ddots & \ddots & -r \\ & & & -r & 1+2r \end{pmatrix} \begin{pmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ \vdots \\ u_{n-1,j+1} \end{pmatrix} = \begin{pmatrix} u_{1,j} + r u_{0,j+1} \\ u_{2,j} \\ u_{2,j} \\ \vdots \\ u_{n-1,j} + r u_{n,j+1} \end{pmatrix}$$

Implicit Euler method IV

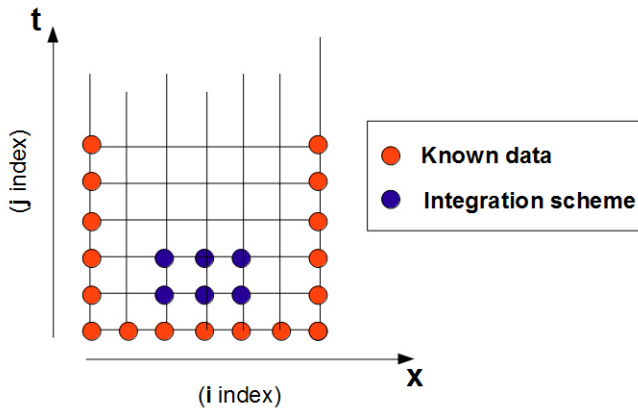
- The scheme is always **numerically stable** and convergent.
- More numerically intensive than the explicit method.
- The errors are linear over the time step and quadratic over the space step:

$$\Delta u = O(k) + O(h^2).$$

Crank-Nicolson method I

To increase the **order** of the method in time, the Crank-Nicolson method uses an average between two time levels, that can be also understood as using the **central difference** in time and a second-order **central difference** for the space derivative.

Crank-Nicolson method II



Crank-Nicolson method III

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \frac{\alpha}{2h^2}(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}) + \frac{\alpha}{2h^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}).$$

Then,

$$-\frac{r}{2} u_{i-1,j+1} + (1+r) u_{i,j+1} - \frac{r}{2} u_{i+1,j+1} = \frac{r}{2} u_{i-1,j} + (1-r) u_{i,j} + \frac{r}{2} u_{i+1,j}.$$

So, for every $j = 1, \dots, m-1$, the elements $u_{i,j+1}$ are obtained from the elements $u_{i,j}$ by solving again a linear algebra system:

$$\begin{pmatrix} 1+r & -r/2 & & & \\ -r/2 & 1+r & -r/2 & & \\ & -r/2 & 1+r & -r/2 & \\ & & \ddots & \ddots & -r \\ & & & -r/2 & 1+r \end{pmatrix} \begin{pmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ \vdots \\ u_{n-1,j+1} \end{pmatrix} = \begin{pmatrix} \frac{r}{2} u_{0,j} + (1-r) u_{1,j} + \frac{r}{2} u_{2,j} + \frac{r}{2} u_{0,j+1} \\ \frac{r}{2} u_{1,j} + (1-r) u_{2,j} + \frac{r}{2} u_{3,j} \\ \frac{r}{2} u_{2,j} + (1-r) u_{3,j} + \frac{r}{2} u_{4,j} \\ \vdots \\ \frac{r}{2} u_{n-2,j} + (1-r) u_{n-1,j} + \frac{r}{2} u_{n,j} + r u_{n,j+1} \end{pmatrix}$$

Crank-Nicolson method IV

- The errors are quadratic over both the time step and the space step:

$$\Delta u = O(k^2) + O(h^2).$$

Conclusion

- Usually the Crank-Nicolson scheme is the most accurate scheme for small time steps.
- The explicit scheme is the least accurate and can be unstable, but is also the easiest to implement and the least numerically intensive.
- The implicit scheme works the best for large time steps.

References

- Ana Lacasta. “Transferencia de calor y vapor de agua”. Slides of the *Physics* course of the Master on Building Engineering, Física Aplicada-EPSEB-UPC.
- Toni Susín. Slides of the session on the “Heat equation” of the course on *Continuous simulation*, Industrial Engineering, Matemàtica Aplicada I-ETSEIB-UPC.
- Wikipedia, *finite differences* entry.
- J. Crank. “The Mathematics of Diffusion”. 2nd Edition, Oxford, 1975, p. 143.