Università degli Studi di Napoli Federico II

Finite element approach for a heat diffusion problem

Simulations and control

Course of “Mathematical methods for Fields”
2019/2020

Prof. Guglielmo Rubinacci

Luigi D’Amico
Alessandro Melone

Physical model

From the physics to the mathematical model

Luigi D’Amico
Alessandro Melone

Physical model

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Heat Transmission

Thermodynamically, **Heat** is a form of energy that is transferred between two systems by virtue of a temperature difference.

Heat can be transferred in three different ways:

- **Conduction**: heat is transmitted by the direct contact between particles of a body without any motion of the material as a whole
- **Convection**: transport of heat as a volume of liquid or gas moves from a region of one temperature to that of another temperature
- **Radiation:** heat propagation by means of electromagnetic waves produced by virtue of the temperature of the body

Conduction occurs in all solids, liquids and gases when a temperature gradient exists. In the fluids, the molecules have freedom of motion and energy is also transferred by movement of the fluid. However, if the fluid is at rest, the heat is transferred by conduction.

Physical problem

Heat Diffusion equation, (1/3)

The diffusion equation is the equation of a scalar field. It can describe the heat diffusion by conduction, in this case the scalar field at each point gives the temperature of the media.

To give an example of how is obtained, consider a rigid, homogeneus and isotropic body with constant mass density.

Is obtained considering the conservation of energy principle:

considering a control volume[1] $\Delta\Omega$ in the time interval $(\mathrm{t}, \mathrm{t} + \mathrm{dt})$ we impose that

$$dU_{\Delta\Omega} = -dQ_{\Delta\Omega} + dW_{\Delta\Omega}$$

- $dU_{\Delta\Omega} \ = internal\ energy\ variation$
- $dQ_{\Delta\Omega} \ = heat\ quantity\ that\ goes\ out\ \Delta\Omega\ passing\ through\ \partial(\Delta\Omega)$
- $dW_{\Delta\Omega} \ = energy\ generated\ by\ the\ source$

[1] control volume: a box shaped region so small that the field quantities can be assumed to be uniform in it

Heat Diffusion equation, (2/3)

To express the equation in terms of field quantities:

$$dU_{\Delta\Omega} = -dQ_{\Delta\Omega} + dW_{\Delta\Omega}$$

- $dU_{\Delta\Omega} = \rho c_v \Theta \, \Delta\Omega dt$
- $dQ_{\Delta\Omega} = \left(\int_{\partial(\Delta\Omega)} q(M,t) \cdot \hat{n} \, dS_M \right) dt$ [1]
- $dW_{\Delta\Omega} = p_s(P,t) \, \Delta\Omega dt$

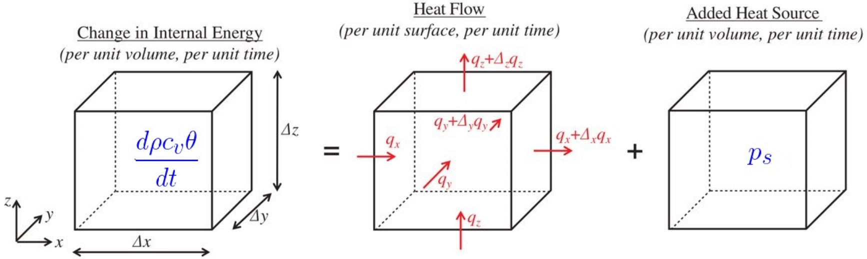
With:

- $\Theta = temperature \, [K]$
- $c_v = specific\ heat\ capacity \left[\frac{J}{K \cdot kg}\right];$
- $q = heat\ flux \left[\frac{J}{s \cdot m^2}\right];$
- $p_s = power\ density\ of\ the\ external\ source \left[\frac{J}{m^3 \cdot s}\right];$

[1] M *is the baricenter of the elementary surface* S_M

Università degli Studi di Napoli Federico II

Heat Diffusion equation, (3/3)



Divide for $\Delta\Omega dt$ the previous equation and let $\Delta\Omega \to 0$, so we obtain:

$$\rho c_v \frac{\partial \Theta}{\partial t} = -\nabla \cdot q + p_s$$

Let be $k = thermal\ conductivity \left[\frac{J}{K \cdot s \cdot m}\right]$, so for the Fourier's law:$\Theta = -k\nabla q$

then finally:

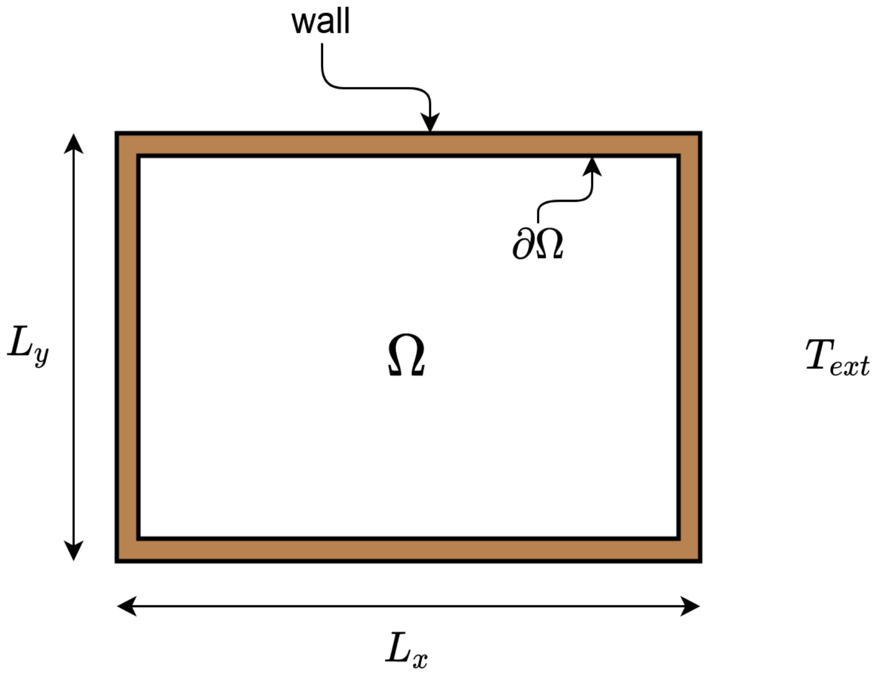
$$\frac{\partial \Theta}{\partial t} = D\Delta\Theta + s(x, y, z)$$

with $\mathrm{D} = \frac{\mathrm{k}}{\rho c_v}$ $diffusion\ coefficient$ and $\mathrm{s(x, y, z)} = \frac{\mathrm{p_s}}{\rho c_v}$

Physical problem

Case study

- The aim of this project is to implement a finite element model that allow to simulate the temperature evolution of an empty room, that exchange heat with the external world assumed to have uniform and constant temperature T_{ext}.
- Thanks to the simple geometric of the problem, is considered the evolution only respect the (x, y) coordinates, i.e. consider the 2-D problem of compute the evolution of $\theta(x, y)$ for a given T_{ext} and heat source $s(x, y)$
- The boundary $\partial\Omega$ correspond to the internal face of the walls



Physical problem

Boundary conditions

In this project are considered two different boundary conditions:

Neumann boundary condition on $\Gamma_N \subset \partial\Omega$

This condition impose a given heat flow $\bar{q}$ on $\partial\Omega$, i.e.

$$\vec{q} \cdot \vec{n} = \bar{q} \; on \; \Gamma_N$$

Robin boundary condition on $\Gamma_R \subset \partial\Omega$

This condition has the aim to model the heat flow due to the difference between the room temperature and the external temperature.

$$\vec{q} \cdot \vec{n} = \frac{\Theta - T_{ext}}{R_{tot}} \; on \; \Gamma_R$$

With $\mathrm{R}_{tot} = thermal\ resistance \left[\frac{K \cdot m^2}{W}\right]$

Luigi D’Amico
Alessandro Melone

Computing R_{tot}

The points of Γ_R belong to the internal face of the wall, in order to compute the insulating effect of the wall is necessary to compute R_{tot}, the dynamics is neglected through the use of equilibrium equations for the *heat transfer* Q $[W]$.

Let be k_w the *thermal conductivity of the wall*, A the surface of the wall and considering the notation in the image on the right:

$$Q_w = kA\frac{\Theta - T_w}{L} = \frac{\Theta - T_w}{R_{cond}} \quad with \;\; R_{cond} = \frac{L}{k_w A}$$

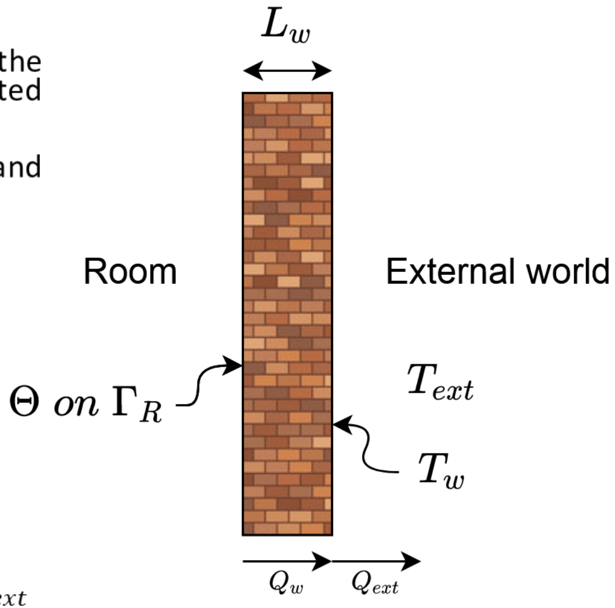
$$Q_{ext} = hA(\mathrm{T_w} - T_{ext}) = \frac{\mathrm{T_w} - T_{ext}}{R_{conv}} \quad with \;\; R_{conv} = \frac{1}{hA}$$

Since at the equilibrium $Q_w = Q_{ext} = Q$:

$$Q = \frac{\Theta - T_{ext}}{R_{tot}} \quad with \;\; R_{tot} = R_{cond} + R_{conv} \; [1]$$

Considering *heat flux* q instead of Q is imposed $A = 1$ that gives: $\vec{q} \cdot \hat{n} = \frac{\Theta - T_{ext}}{R_{tot}}$

Finally: $\Theta = -k\nabla q$ implies $\nabla\Theta \cdot \hat{n} = \frac{T_{ext} - \Theta}{kR_{tot}}$



[1] *There is an electrical analogy with conduction heat transfer: the analog of Q is current, and the analog of the temperature difference is voltage difference.*

Physical problem

Simplifying assumption

- Numerical simulations for temperature distribution are often modeled based on Navier-Stokes Equations (NSE) for air flow coupled with an equation for heat transfer. While this approach is highly accurate and has been used successfully, heavy computational effort is required due to the nonlinearity of NSE. Thus, it is often not suitable for real-time simulations.
- The approach used in this work cannot capture small-scale details but it may however capture global behaviour.
- Assumed there are no windows in the room, the velocity of the air on the boundary is zero, so we will assume that the convective term (term that consider the heat exchange due to the movement of the air) is null.

Physical model

Strong form equations

$$\frac{\partial \Theta}{\partial t} = D\Delta\Theta + s(x,y)\,\mu(t)$$

$$\vec{q}\,\vec{n} = -k\vec{\nabla}\Theta\,\vec{n} = \bar{q} \qquad \text{on } \Gamma_q$$

$$\vec{q}\,\vec{n} = -k\vec{\nabla}\Theta\,\vec{n} = -\frac{(T-\Theta)}{R_{tot}} \qquad \text{on } \Gamma_R$$

With: $\vec{q} = \vec{\nabla}\Theta$

$\Theta = \Theta(x,y,t)$

$T = T_{ext}$

We assume that the source term can be factored in the form: $s(x,y)\,\mu(t)$

This assumption allow to compute the integral to obtain the fem matrix once and not at each step, infact those integral do not depend on time (it will be shown after few slides)

From the strong to the weak form

Mathematical derivation and equivalence

From the strong to the weak form

Derivation

Let be $w(x,y) \in H^1(\Omega)$ the *weight function* with $H^1(\Omega)$ the Sobolev space, that is the space of the function square-integrable with also the first derivative square-integrable.

Multiply each side of the strong form equation for $w(x,y)$ and integrate:

$$\iint_\Omega w\dot{\Theta}\,dS = D\iint_\Omega w\nabla^2\Theta\,dS + \mu(t)\iint_\Omega ws\,dS$$

UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Derivation

Let be $w(x,y) \in H^1(\Omega)$ the *weight function* with $H^1(\Omega)$ the Sobolev space, that is the space of the function square-integrable with also the first derivative square-integrable.

Multiply each side of the strong form equation for $w(x,y)$ and integrate:

$$\iint_\Omega w\dot{\Theta}\, dS = D\iint_\Omega w\nabla^2\Theta\, dS + \mu(t)\iint_\Omega ws\, dS$$

Now expand the first term of the second member, and apply Gauss-Green:

$$D\iint_\Omega w\nabla\cdot\nabla\Theta\, dS = -D\iint_\Omega \vec{\nabla}w\,\vec{\nabla}\Theta\, dS + D\int_\Gamma w\vec{\nabla}\Theta\cdot\vec{n}\, d\Gamma$$

Finally use the additive property of the integrals:

$$D\int_\Gamma w\vec{\nabla}\Theta\cdot\vec{n}\, d\Gamma = D\int_{\Gamma_N} -w\frac{\bar{q}}{k}\, d\Gamma_N + D\int_{\Gamma_R} w\frac{T-\Theta}{kR_{tot}}\, d\Gamma_R$$

From the strong to the weak form

Derivation

Let be $w(x,y) \in H^1(\Omega)$ the *weight function* with $H^1(\Omega)$ the Sobolev space, that is the space of the function square-integrable with also the first derivative square-integrable.

Multiply each side of the strong form equation for $w(x,y)$ and integrate:

$$\iint_\Omega w\dot{\Theta}\, dS = D \iint_\Omega w\nabla^2\Theta\, dS + \mu(t) \iint_\Omega ws\, dS$$

Now expand the first term of the second member, and apply Gauss-Green:

$$D \iint_\Omega w\nabla \cdot \nabla\Theta\, dS = -D \iint_\Omega \vec{\nabla} w\, \vec{\nabla}\Theta\, dS + D \int_\Gamma w\vec{\nabla}\Theta \cdot \vec{n}\, d\Gamma$$

Finally use the additive property of the integrals:

$$D \int_\Gamma w\vec{\nabla}\Theta \cdot \vec{n}\, d\Gamma = D \int_{\Gamma_N} -w\frac{\bar{q}}{k}\, d\Gamma_N + D \int_{\Gamma_R} w\frac{T-\Theta}{kR_{tot}}\, d\Gamma_R$$

In conclusion:

$$\iint_\Omega w\dot{\Theta}\, dS + D \iint_\Omega \vec{\nabla} w\, \vec{\nabla}\Theta\, dS = \mu(t) \iint_\Omega ws\, dS - \frac{D}{k}\int_{\Gamma_N} w\bar{q}\, dS + \frac{D}{kR_{tot}}\int_{\Gamma_R} wT\, dS - \frac{D}{kR_{tot}}\int_{\Gamma_R} w\Theta\, dS$$

Luigi D'Amico
Alessandro Melone

From the strong to the weak form

Consideration

In conclusion: $\forall w(x,y) \in H^1(\Omega)$

$$\iint_\Omega w\dot{\Theta}\,dS + D\iint_\Omega \vec{\nabla} w\,\vec{\nabla}\Theta\,dS = \mu(t)\iint_\Omega ws\,dS - \frac{D}{k}\int_{\Gamma_N} w\bar{q}\,dS + \frac{D}{kR_{tot}}\int_{\Gamma_R} wT\,dS - \frac{D}{kR_{tot}}\int_{\Gamma_R} w\Theta\,dS$$

- The equation above is the *weak form* and can be proved that is equivalent to the *strong form*
- Is named *weak* form because the regularity condition imposed on the temperature field function are less demanding respect that of the strong form: is not necessary that exist the Laplacian of the solution, but just the first derivative.
- The f.e.m. approach used in this project wants to find a piecewise approximation of the solution, so the second derivative exist only in the sense of distributions. For this reason it was necessary to introduce the weak form.

Finite Element Method

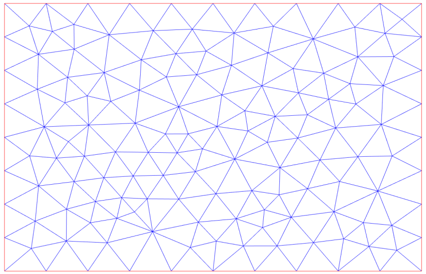
Implementation of the 2D FEM algorithm with triangle elements

Luigi D'Amico
Alessandro Melone

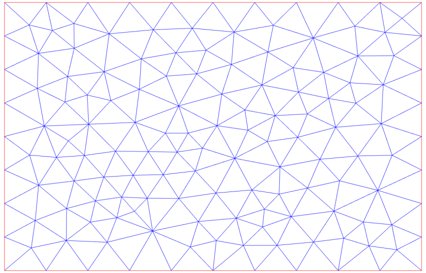
Introduction

- The f.e.m. approach wants to approximate the functional space in which to find the solution, the approximate solution and the test function will belong to the functional space of the piecewise function.

- The two-dimensional domain Ω is *discretized* into N_e triangular subdomains (elements). Each element consists of *3* nodes.



Introduction



- The f.e.m. approach wants to approximate the functional space in which to find the solution, the approximate solution and the test function will belong to the functional space of the piecewise function.
- The two-dimensional domain Ω is *discretized* into N_e triangular subdomains (elements). Each element consists of *3* nodes.
- We will rely on a piecewise approximation, therefore we will stipulate the approximation of the field for each element separately.
- The approximate temperature field in element *e* will be given by a polynomial expression involving shape functions and the nodal temperature vector of the element.

$$\Theta^{(e)}(x,y) = [N^{(e)}]\{\Theta^{(e)}\}$$

Shape function array of the element

$$[N^{(e)}] = [N_1^{(e)}(x,y)\ \ N_2^{(e)}(x,y)\ \ N_1^{(e)}(x,y)]$$

Nodal temperature vector of the element

$$\{\Theta^{(e)}\} = [\Theta_1^{(e)}\ \ \Theta_2^{(e)}\ \ \Theta_3^{(e)}]^T$$

Node temperature and weighting functions

We can also define the gradient of the nodal temperature

$$\{\nabla\Theta^{(e)}\} = \{B^{(e)}\}\,[\Theta^{(e)}]$$

$$\{B^{(e)}\} = \begin{bmatrix} \frac{\partial N_1^{(e)}}{\partial x} & \frac{\partial N_2^{(e)}}{\partial x} & \frac{\partial N_3^{(e)}}{\partial x} \\ \frac{\partial N_1^{(e)}}{\partial y} & \frac{\partial N_2^{(e)}}{\partial y} & \frac{\partial N_3^{(e)}}{\partial y} \end{bmatrix}$$

The nodal values of each element *e*, are given from the global node vector by a gather operation

$$\{\Theta^{(e)}\} = [L^{(e)}]\{\Theta\}$$

$\{\Theta\}$: temperature values of all the nodal points in the finite element mesh

$[L^{(e)}]$: gather array of element e

Node temperature and weighting functions

We can also define the gradient of the nodal temperature

$$\{\nabla\Theta^{(e)}\} = \{B^{(e)}\}\,[\Theta^{(e)}]$$

$$\{B^{(e)}\} = \begin{bmatrix} \frac{\partial N_1^{(e)}}{\partial x} & \frac{\partial N_2^{(e)}}{\partial x} & \frac{\partial N_3^{(e)}}{\partial x} \\ \frac{\partial N_1^{(e)}}{\partial y} & \frac{\partial N_2^{(e)}}{\partial y} & \frac{\partial N_3^{(e)}}{\partial y} \end{bmatrix}$$

The nodal values of each element *e*, are given from the global node vector by a gather operation

$$\{\Theta^{(e)}\} = [L^{(e)}]\{\Theta\}$$

$\{\Theta\}$: temperature values of all the nodal points in the finite element mesh

$[L^{(e)}]$: gather array of element e

The piecewise approximation for the actual field is also used for the weighting functions field, *w(x,y)*

$$w^{(e)}(x,y) = [N^{(e)}]\{W^{(e)}\} = \{W^{(e)}\}^T[N^{(e)}]^T$$

$$\{\nabla w^{(e)}\} = \{B^{(e)}\}\,[W^{(e)}]$$

$$\{W^{(e)}\} = [L^{(e)}]\{W\}$$

$\{W^{(e)}\}$: Nodal values of weighting functions for element e

Finite Element Method

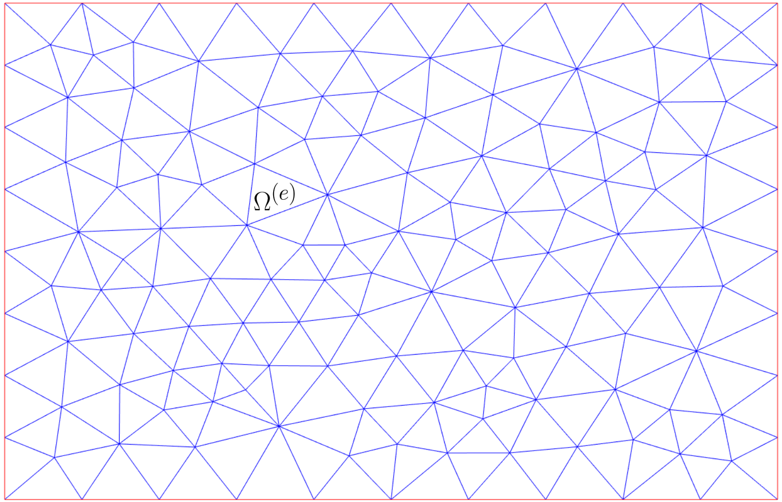
How to solve the weak form

$$\iint_\Omega w\dot{\Theta}\, dS + D\iint_\Omega \vec{\nabla} w\, \vec{\nabla}\Theta\, dS = \mu(t)\iint_\Omega ws\, dS - \frac{D}{k}\int_{\Gamma_N} w\bar{q}\, dS + \frac{D}{kR_{tot}}\int_{\Gamma_R} wT\, dS - \frac{D}{kR_{tot}}\int_{\Gamma_R} w\Theta\, dS$$

How to solve the weak form

$$\iint_{\Omega} w\dot{\Theta}\, dS + D\iint_{\Omega} \vec{\nabla} w\, \vec{\nabla}\Theta\, dS = \mu(t)\iint_{\Omega} ws\, dS - \frac{D}{k}\int_{\Gamma_N} w\bar{q}\, dS + \frac{D}{kR_{tot}}\int_{\Gamma_R} wT\, dS - \frac{D}{kR_{tot}}\int_{\Gamma_R} w\Theta\, dS$$

We now separate the domain and boundary integrals into the contributions from the various element subdomain $\Omega^{(e)}$, and boundary segment $\Gamma_N^{(e)}\Gamma_R^{(e)}$



Finite Element Method

Left side, 2nd term

$$\iint_{\Omega} w\dot{\Theta}\, dS + D\iint_{\Omega} \vec{\nabla} w\, \vec{\nabla}\Theta\, dS = \mu(t)\iint_{\Omega} ws\, dS - \frac{D}{k}\int_{\Gamma_N} w\bar{q}\, dS + \frac{D}{kR_{tot}}\int_{\Gamma_R} wT\, dS - \frac{D}{kR_{tot}}\int_{\Gamma_R} w\Theta\, dS$$

$$D\iint_{\Omega}\{\nabla w\}^T\{\nabla\Theta\}\, dV = \sum_{e=1}^{N_e}\left(D\iint_{\Omega^{(e)}}\{\nabla w\}^T\{\nabla\Theta\}\, dV\right)$$

$$D\iint_{\Omega^{(e)}}\{\nabla w\}^T\{\nabla\Theta\}\, dV = \{W^{(e)}\}^T[K^{(e)}]\{\Theta^{(e)}\} = \{W\}^T[L^{(e)}]^T[K^{(e)}][L^{(e)}]\{\Theta\}$$

$$\boxed{D\iint_{\Omega}\{\nabla w\}^T\{\nabla\Theta\}\, dV = \{W\}^T[K]\{\Theta\}}$$

$$[K^{(e)}] = D\iint_{\Omega^{(e)}}[B^{(e)}][B^{(e)}]dV$$

$$[K] = \sum_{e=1}^{N_e}\left([L^{(e)}]^T[K^{(e)}][L^{(e)}]\right)$$

Right side, 4th term

$$\iint_\Omega w\dot{\Theta}\, dS + D\iint_\Omega \vec{\nabla} w\, \vec{\nabla}\Theta\, dS = \mu(t)\iint_\Omega w s\, dS - \frac{D}{k}\int_{\Gamma_N} w\bar{q}\, dS + \frac{D}{kR_{tot}}\int_{\Gamma_R} wT\, dS - \frac{D}{kR_{tot}}\int_{\Gamma_R} w\Theta\, dS$$

$$\frac{D}{k\,R_{tot}}\int_{\Gamma_R} w\,\Theta\,dS = \sum_{e=1}^{N_e}\left(\frac{D}{k\,R_{tot}}\int_{\Gamma_R^{(e)}} w\,\Theta\,dS\right)$$

$$\frac{D}{k\,R_{tot}}\iint_{\Gamma_R^{(e)}} w\,\Theta\,dS = \{W^{(e)}\}^T[C^{(e)}]\{\Theta^{(e)}\} = \{W\}^T[L^{(e)}]^T[C^{(e)}][L^{(e)}]\{\Theta\}$$

$$\boxed{\frac{D}{k\,R_{tot}}\int_{\Gamma_R} w\,\Theta\,dS = \{W\}^T[C]\{\Theta\}}$$

$$[C^{(e)}] = \frac{D}{k\,R_{tot}}\int_{\Gamma_R^{(e)}}[N^{(e)}][N^{(e)}]dS$$

$$[C] = \sum_{e=1}^{N_e}\left([L^{(e)}]^T[C^{(e)}][L^{(e)}]\right)$$

Right side, 1st 2nd 3rd term

$$\iint_\Omega w\dot{\Theta}\, dS + D \iint_\Omega \vec{\nabla} w\, \vec{\nabla}\Theta\, dS = \mu(t) \iint_\Omega w s\, dS - \frac{D}{k}\int_{\Gamma_N} w\bar{q}\, dS + \frac{D}{kR_{tot}}\int_{\Gamma_R} wT\, dS - \frac{D}{kR_{tot}}\int_{\Gamma_R} w\Theta\, dS$$

$$\boxed{-\frac{D}{k}\int_{\Gamma_q} w\,\bar{q}\, dS + \frac{D}{k\,R_{tot}}\int_{\Gamma_R} w\,T\, dS + \iint_\Omega w\, s\, \mu\, dV = \{W\}^T\{f\}}$$

$$\{f\} = \begin{bmatrix}\{f_\Omega\} & \{f_{\Gamma_N}\} & \{f_{\Gamma_{R_1}}\}\end{bmatrix}\begin{bmatrix}\mu \\ \bar{q} \\ T\end{bmatrix}$$

$$\{f_\Omega^{(e)}\} = \iint_{\Omega^{(e)}} [N^{(e)}]^T\, s^{(e)} dV$$

$$\{f_{\Gamma_q}^{(e)}\} = -\frac{D}{k}\int_{\Gamma_q^{(e)}} [N^{(e)}]^T dS$$

$$\{f_{\Gamma_{R_1}}^{(e)}\} = \frac{D}{k\,R_{tot}}\int_{\Gamma_R^{(e)}} [N^{(e)}]^T dS$$

Left side, 1st term

$$\iint_\Omega w\dot{\Theta}\,dS + D\iint_\Omega \vec{\nabla} w\,\vec{\nabla}\Theta\,dS = \mu(t)\iint_\Omega ws\,dS - \frac{D}{k}\int_{\Gamma_N} w\bar{q}\,dS + \frac{D}{kR_{tot}}\int_{\Gamma_R} wT\,dS - \frac{D}{kR_{tot}}\int_{\Gamma_R} w\Theta\,dS$$

$$\boxed{\iint_\Omega w\,\dot{\Theta}\,dV = \{W\}^T[M]\{\dot{\Theta}\}}$$

$$[M^{(e)}] = \iint_{\Omega^{(e)}} [N^{(e)}][N^{(e)}]dV$$

$$[M] = \sum_{e=1}^{N_e}\left([L^{(e)}]^T[M^{(e)}][L^{(e)}]\right)$$

Matrix formula for the weak form

$$\boxed{\iint_{\Omega} w\dot{\Theta}\, dS + D \iint_{\Omega} \vec{\nabla} w\, \vec{\nabla}\Theta\, dS = \mu(t) \iint_{\Omega} ws\, dS - \frac{D}{k} \int_{\Gamma_N} w\bar{q}\, dS + \frac{D}{kR_{tot}} \int_{\Gamma_R} wT\, dS - \frac{D}{kR_{tot}} \int_{\Gamma_R} w\Theta\, dS}$$

Putting all terms together...

$$\{W\}^T[M]\{\dot{\Theta}\} + \{W\}^T[K]\{\Theta\} = -\{W\}^T[C]\{\Theta\} + \{W\}^T\{f\}$$

Università degli Studi di Napoli Federico II

Matrix formula for the weak form

$$\iint_\Omega w\dot{\Theta}\,dS + D\iint_\Omega \vec{\nabla}w\,\vec{\nabla}\Theta\,dS = \mu(t)\iint_\Omega ws\,dS - \frac{D}{k}\int_{\Gamma_N} w\bar{q}\,dS + \frac{D}{kR_{tot}}\int_{\Gamma_R} wT\,dS - \frac{D}{kR_{tot}}\int_{\Gamma_R} w\Theta\,dS$$

Putting all terms together...

$$\{W\}^T[M]\{\dot{\Theta}\} + \{W\}^T[K]\{\Theta\} = -\{W\}^T[C]\{\Theta\} + \{W\}^T\{f\}$$

Once the FEM algorithm has been runned, the quantities [*M*] [*K*] [*C*] {*f*} is obtained.

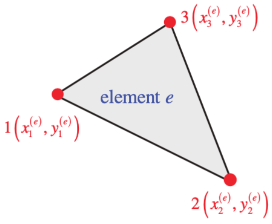
$$\boxed{[M]\{\dot{\Theta}\} + [L]\{\Theta\} = \{f\}}$$

A linear dynamic system in the $\boldsymbol{\Theta}$ variable has to be solved in order to obtain the nodal temperature vector.

$$[L] = [K] + [C]$$

Shape functions, 1

We have three nodes and we want to introduce a polynomial interpolation for the temperature of the element *e*



We impose that the temperature at each nodal points is equal to the corresponding nodal temperature

$$\Theta^{(e)}(x,y) = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{Bmatrix} a_0^{(e)} \\ a_1^{(e)} \\ a_2^{(e)} \end{Bmatrix} = [p(x,y)]\{a\}$$

$$\begin{bmatrix} 1 & x_1^{(e)} & y_1(e) \\ 1 & x_2^{(e)} & y_2(e) \\ 1 & x_3^{(e)} & y_3(e) \end{bmatrix} \begin{bmatrix} a_0^{(e)} \\ a_1^{(e)} \\ a_2^{(e)} \end{bmatrix} = \begin{bmatrix} \Theta_0^{(e)} \\ \Theta_1^{(e)} \\ \Theta_2^{(e)} \end{bmatrix}$$

$$[M^{(e)}]\{a^{(e)}\} = \{\Theta^{(e)}\}$$

Finite Element Method

Shape functions, 2

We obtain an expression for the shape function

$$\Theta^{(e)}(x,y) = [p(x,y)][M^{(e)}]^{-1}\{\Theta^{(e)}\}$$

$$[N^{(e)}] = [p(x,y)][M^{(e)}]^{-1}$$

If one does the math, the following expression are obtained for the three shape functions

$$N_1^{(e)}(x,y) = \frac{1}{2A^{(e)}}\left[x_2^{(e)}y_3^{(e)} - x_3^{(e)}y_2^{(e)} + \left(y_2^{(e)} - y_3^{(e)}\right)x + \left(x_3^{(e)} - x_2^{(e)}\right)y\right]$$

$$N_2^{(e)}(x,y) = \frac{1}{2A^{(e)}}\left[x_3^{(e)}y_1^{(e)} - x_1^{(e)}y_3^{(e)} + \left(y_3^{(e)} - y_1^{(e)}\right)x + \left(x_1^{(e)} - x_3^{(e)}\right)y\right]$$

$$N_3^{(e)}(x,y) = \frac{1}{2A^{(e)}}\left[x_1^{(e)}y_2^{(e)} - x_2^{(e)}y_1^{(e)} + \left(y_1^{(e)} - y_2^{(e)}\right)x + \left(x_2^{(e)} - x_1^{(e)}\right)y\right]$$

where $A^{(e)}$ is the area of the element, obtained from the nodal coordinates as follows

$$A^{(e)} = \frac{1}{2}\det\left(\left[M^{(e)}\right]\right) = \frac{1}{2}\left[\left(x_2^{(e)}y_3^{(e)} - x_3^{(e)}y_2^{(e)}\right) - \left(x_1^{(e)}y_3^{(e)} - x_3^{(e)}y_1^{(e)}\right) + \left(x_1^{(e)}y_2^{(e)} - x_2^{(e)}y_1^{(e)}\right)\right]$$

Stiffness matrix [K]

Through the matrix [$B^{(e)}$] we can also obtain the shape functions partial derivatives with respect to x and y.

[$B^{(e)}$] for the three-node triangular element is *constant* (it does not depend on x and y).

$$\left[B^{(e)}\right] = \frac{1}{2A^{(e)}} \begin{bmatrix} y_2^{(e)} - y_3^{(e)} & y_3^{(e)} - y_1^{(e)} & y_1^{(e)} - y_2^{(e)} \\ x_3^{(e)} - x_2^{(e)} & x_1^{(e)} - x_3^{(e)} & x_2^{(e)} - x_1^{(e)} \end{bmatrix}$$

It's now possible to compute the [$K^{(e)}$] matrix.

Note that numerical error is not present here because [$B^{(e)}$] can be extracted from the integral

$$\boxed{[K^{(e)}] = D \iint_{\Omega^{(e)}} [B^{(e)}][B^{(e)}] dV = D\,[B^{(e)}]\,[B^{(e)}]\,A^{(e)}}$$

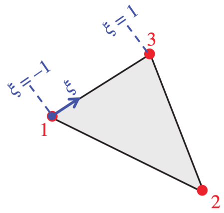
Vector {f}

For the boundary condition, we *parametrize* the boundary segment.

Since the natural boundary of the element is a line segment, we can describe the position of a point along this segment by means of a single parameter ξ

$$x(\xi)=\frac{x_1^{(e)}+x_3^{(e)}}{2}+\frac{x_3^{(e)}-x_1^{(e)}}{2}\xi,\ -1\le\xi\le 1$$

$$y(\xi)=\frac{y_1^{(e)}+y_3^{(e)}}{2}+\frac{y_3^{(e)}-y_1^{(e)}}{2}\xi,\ -1\le\xi\le 1$$



It’s now possible compute the integral term in function of ξ variable.

$$\{f_{\Gamma_N}^{(e)}\}=-\frac{D}{k}\int_{\Gamma_N^{(e)}}[N^{(e)}]^T dS=-\frac{D}{k}\int_{-1}^{+1}\begin{Bmatrix}N_1^{(e)}(x(\xi),y(\xi))\\ N_2^{(e)}(x(\xi),y(\xi))\\ N_3^{(e)}(x(\xi),y(\xi))\end{Bmatrix}\frac{\ell_{31}}{2}d\xi$$

$$\{f_{\Gamma_{R_1}}^{(e)}\}=\frac{Dh}{k}\int_{\Gamma_R^{(e)}}[N^{(e)}]^T dS=\frac{Dh}{k}\int_{-1}^{+1}\begin{Bmatrix}N_1^{(e)}(x(\xi),y(\xi))\\ N_2^{(e)}(x(\xi),y(\xi))\\ N_3^{(e)}(x(\xi),y(\xi))\end{Bmatrix}\frac{\ell_{31}}{2}d\xi$$

The integrals are evaluated using the 1D gaussian quadrature

$$\ell_{31}=\sqrt{\left(x_3^{(e)}-x_1^{(e)}\right)^2+\left(y_3^{(e)}-y_1^{(e)}\right)^2}$$

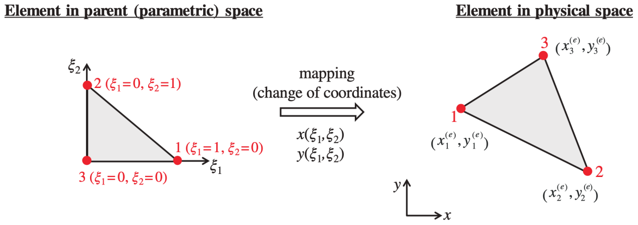
$$dS=\sqrt{\left(\frac{\partial x}{\partial\xi}\right)^2+\left(\frac{\partial y}{\partial\xi}\right)^2}d\xi=\frac{\ell_{31}}{2}d\xi$$

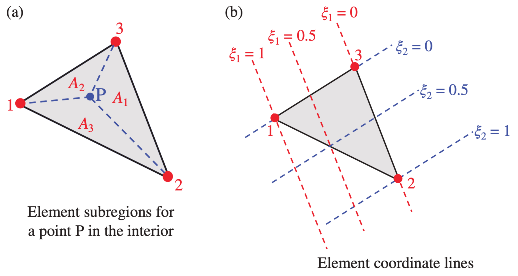
Matrix [*M*] and $\{f_\Omega\}$

The isoparametric concept is used to integrate over the triangular domain.
Natural coordinates approach: given a node *P* in the interior of the element, the three areas of the three sub-triangles are defined.

ξ_1 and ξ_2 can be used to establish the isoparametric element

We can now establish the associated mapping





Element subregions for a point P in the interior

Element coordinate lines

$$\xi_1 = \frac{A_1}{A^{(e)}}$$

$$\xi_2 = \frac{A_2}{A^{(e)}}$$

$$\xi_3 = \frac{A_3}{A^{(e)}} = 1 - \xi_1 - \xi_2$$

$$0 \leq \xi_1 \leq 1$$

$$0 \leq \xi_2 \leq 1$$

$$x(\xi_1, \xi_2) = \sum_{i=1}^{3} N_i^{(3T)}(\xi_1, xi_2)\, x_i^{(e)}$$

$$y(\xi_1, \xi_2) = \sum_{i=1}^{3} N_i^{(3T)}(\xi_1, xi_2)\, y_i^{(e)}$$

Matrix $[M]$ and $\{f_\Omega\}$

We now establish the shape functions in the parametric (natural) coordinate space

$$N_1^{(3T)}(\xi_1,\xi_2) = \xi_1$$

$$N_2^{(3T)}(\xi_1,\xi_2) = \xi_2$$

$$N_3^{(3T)}(\xi_1,\xi_2) = 1-\xi_1-\xi_2$$

Concerning the integration, for any function we can use the change-of-coordinates formula to obtain

$$\iint_{\Omega^{(e)}} f(x,y)dV = \int_0^1 \int_0^{1-\xi_1} f(x(\xi_1,\xi_2),y(\xi_1,\xi_2)) \cdot J d\xi_1 d\xi_2$$

This formula is used to compute the surface integral for $[M^{(e)}]$ and $\{f_\Omega^{(e)}\}$

$$J = \det([J]), \quad [J] = \begin{bmatrix} \partial x/\partial \xi_1 & \partial x/\partial \xi_2 \\ \partial y/\partial \xi_1 & \partial y/\partial \xi_2 \end{bmatrix}$$

Numerical quadrature can then be used to evaluate domain integrals

Finite Element Method

Steady-state case

Output of the FEM algorithm: [*L*] {*f*}

$$\boxed{[L]\{\Theta\} = \{f\}}$$

A linear algebraic system has to be solved to obtain the nodal temperature vector {**Θ**}.
The command “\” in *Matlab* has been used.

$$\{f\} = \left[\{f_\Omega\}\ \{f_{\Gamma_N}\}\ \{f_{\Gamma_{R_1}}\}\right]\begin{bmatrix}\mu \\ \bar{q} \\ T\end{bmatrix}$$

Steady-state case

Output of the FEM algorithm: [*L*] {*f*}

$$\boxed{[L]\{\Theta\} = \{f\}}$$

A linear algebraic system has to be solved to obtain the nodal temperature vector {**Θ**}.

The command “\” in *Matlab* has been used.

$$\{f\} = \left[\{f_\Omega\}\ \{f_{\Gamma_N}\}\ \{f_{\Gamma_{R_1}}\}\right]\begin{bmatrix}\mu\\ \bar{q}\\ T\end{bmatrix}$$

Unsteady case

Output of the FEM algorithm: [*M*] [*L*] $\{f_\Omega\}$ $\{f_N\}$ $\{f_{R1}\}$

$$\boxed{[M]\{\dot{\Theta}\} + [L]\{\Theta\} = \{f\}}$$

A linear dynamic system in the {**Θ**} variable and with time-variant input has to be solved in order to obtain the nodal temperature vector.

$$\{f\} = \left[\{f_\Omega\}\quad \{f_{\Gamma_N}\}\quad \{f_{\Gamma_{R_1}}\}\right]\begin{bmatrix}\mu(t)\\ \bar{q}(t)\\ T(t)\end{bmatrix}$$

FEM Code Verification

execution of the order-of-accuracy test through the MMS method

Luigi D’Amico
Alessandro Melone

Università degli Studi di Napoli Federico II

Verification and Validation

Verification and validation provide a means for assessing the credibility and accuracy of mathematical models and their subsequent simulations.

- **Verification:** assessing the numerical accuracy of a simulation relative to the true solution to the mathematical model. Is composed by two parts:
  - **code verification**
  - **solution verification**
- **Validation:** assessment of the accuracy of the mathematical model relative to the observations of nature that come in the form of experimental measurements. In this case the relationship of the strong form model with the natural phenomena is proved. In other words is checked if is solved the right equation.

In this project is not considered the validation problem, because there is no physical experiment to observe.

FEM code verification

Verification problem

•**Code Verification**: the goal is to ensure that the code is a faithful representation of the underlying mathematical model. Thus assess both the correctness of the chosen numerical algorithm and the correctness of the instantiation of that algorithm into written source code (i.e. there are no coding mistakes).

In other words (considering the Roache's definition), it is a procedure to demonstrate that the discretization error decreases to zero as the mesh dimensions decreases to zero through the verification of the order-of-accuracy.

•**Solution Verification:** the goal is the estimation the magnitude (and not the order like the code verification) of the numerical errors that occur when the governing equations are discretized and solved numerically.

FEM code verification

Order-of-accuracy test (1/2)

The **Order-of-accuracy test** assess if the numerical solutions converge to the exact solution to the governing equation at the expected rate (i.e. the formal order of accuracy).

Observed order of accuracy is the actual order of accuracy obtained on a series of systematically-refined meshes.

Let be $\tilde{u}$ the *exact solution to the mathematical model* u_h the *exact solution to the discrete equation* and h the *mesh spacing* ($\Delta x, \Delta y$ *and* Δt denoted here collectively by the parameter h).

Consider a series expansion of the solution to the discrete equations u_h in terms of the *mesh spacing* h in the limit as $h \to 0$:

$$\mathrm{u}_h = u_{h=0} + \left.\frac{\partial u}{\partial h}\right|_{h=0} h + \left.\frac{\partial^2 u}{\partial h^2}\right|_{h=0} \frac{h^2}{2} + O(h^3) \qquad with \quad \lim_{h\to 0}\frac{O(h^p)}{h^p} = l \neq 0$$

If a *convergent* numerical scheme is employed $u_{h=0} = \tilde{u}$, moreover if is pth-order accurate terms of order h^{p-1} do not appear in the truncation error ε_h :

$$\varepsilon_h = u_h - \tilde{u} = g_p h^p + O(h^{p+1})$$

with $g_p = g_p(x, y, t)$ independent from h

In the limit as $h \to 0$, the higher-order terms in will become small relative to the leading term and can be neglected, this is approximately true if h is in the *asymptotic region*.

Order-of-accuracy test (2/2)

Introducing the *grid refinement factor* r which is defined as $\mathrm{r} \equiv \frac{h_{coarse}}{h_{fine}}$ we can consider the discretization error expansion for the two mesh levels h and rh:

$$\varepsilon_h = g_p h^p \qquad and \qquad \varepsilon_{rh} = g_p (rh)^p$$

Dividing the first equation by the second one, then taking the natural log, we can finally obtain the *general expression observed order of accuracy*:

$$\hat{\rho} = \frac{\ln\left(\frac{\varepsilon_{rh}}{\varepsilon_h}\right)}{\ln(r)}$$

The observed order of accuracy can be evaluated using the norms of the discretization error:

$$\hat{\rho} = \frac{\ln\left(\frac{\|\varepsilon_{rh}\|}{\|\varepsilon_h\|}\right)}{\ln(r)}$$

Trade-off choice mesh refinement:

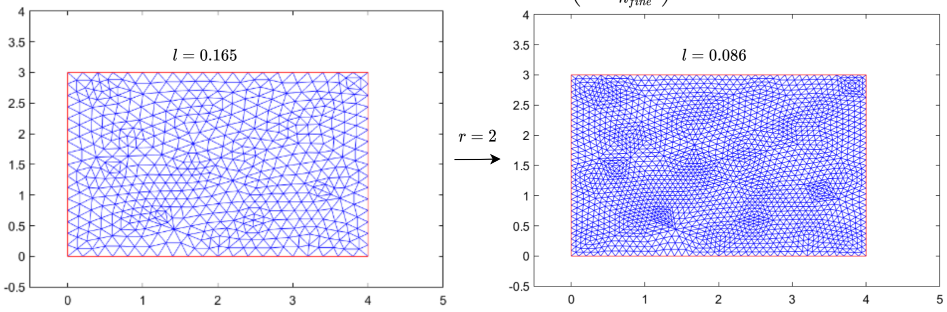
- The numerical solution of the discrete equations differs from the exact solution u_h because the presence of both round-off and iterative error. In order to use the numerical solution as a surrogate for u_h when computing ε_h we assume that these errors are at least 100 times smaller than the discretization error on the finest mesh employed (i.e., $\leq 0{,}01 \cdot \varepsilon_h$)
- The observed order of accuracy $\hat{\rho}$ will only match the formal order of accuracy when the higher-order terms are in fact small, i.e., for $h \to 0$. But for highly refined spatial meshes and/or small time steps, the discretization error can be small and thus round-off error can adversely impact the order-of-accuracy test.

FEM Code verification

Mesh refinement

- *pdetool* of *Partial Differential Equation toolbox* by *MatLab* is used to generate the triangulation of the domain.
- The uniform mesh refinement is obtained by dividing each triangle side in two parts, therefore from one triangle 4 triangles are obtained.
- The discretization parameter is $h = \Delta l / \Delta l_{ref}$ with Δl be the average triangle side length and Δl_{ref} is computed on the more coarse mesh.

h: average lenght of mesh segment   r: refinement factor $\left(r = \frac{h_{coarse}}{h_{fine}}\right)$



Note that the mesh refinement is uniform despite the mesh are not uniform.

Are considered 4 different meshes obtained by successive refinement

h	Δl	num. triangles	num. nodes	num. segments
1	0.3288	262	150	71
0.5	0.16528	1048	561	249
0.25	0.086124	4192	2169	981
0.125	0.042625	16768	8529	3734

FEM code verification

Finding the exact solution

In order to achieve the exact solution the main approaches are two:

- **Method of Exact Solutions** (MES): find the exact solution by solving the governing equations, for the given boundary and initial conditions. The exact solution for complex problem often depend on significant simplifications in dimensionality, geometry, physics, etc…
- **Method of Manufactured Solutions** (MMS): choose a solution a priori that respect the boundary and initial conditions, then plug this solution in the governing equations and find the corresponding source term.
    - For code verification purposes, is not required that the manufactured solution be related to a physically realistic problem (e.g. the heat source could be meaningless in our case).
    - The manufactured solutions should be analytic functions with smooth derivatives. Trigonometric and exponential functions are recommended.

FEM Code verification

Construct manufactured solution (steady state case) (1/2)

Assume $T = 0$ the BC become: $\vec{\nabla}\Theta \cdot \hat{n} = \left[\, \Theta_x(x,y) \;\; \Theta_y(x,y) \,\right] \left[\begin{smallmatrix} n_1(x,y) \\ n_2(x,y) \end{smallmatrix} \right] = -\dfrac{\Theta(x,y)}{kR_{tot}} \;\forall (x,y) \in \Gamma_R$

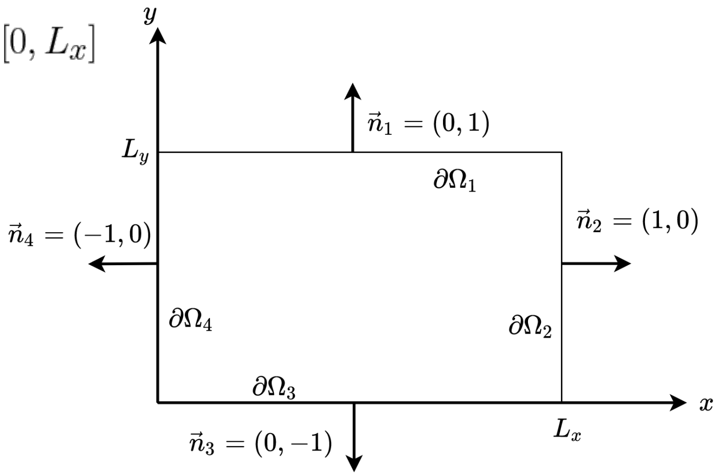
Given that: $\Gamma_R = \{\partial\Omega_1, \partial\Omega_2, \partial\Omega_3, \partial\Omega_4\}$:

BC on $\partial\Omega_1$ where $\hat{n} = \left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix} \right] \Rightarrow \vec{\nabla}\Theta \cdot \hat{n} = \dfrac{\partial \Theta}{\partial y}(x, L_y) = -\dfrac{\Theta(x, L_y)}{kR_{tot}}, \forall x \in [0, L_x]$

BC on $\partial\Omega_2$ where $\hat{n} = \left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix} \right] \Rightarrow \dfrac{\partial \Theta}{\partial x}(L_x, y) = -\dfrac{\Theta(L_x, y)}{kR_{tot}}, \forall y \in [0; L_y]$

BC on $\partial\Omega_3$ where $\hat{n} = \left[\begin{smallmatrix} 0 \\ -1 \end{smallmatrix} \right] \Rightarrow \dfrac{\partial \Theta}{\partial y}(x, 0) = -\dfrac{\Theta(x, 0)}{kR_{tot}}, \forall x \in [0; L_x]$

BC on $\partial\Omega_4$ where $\hat{n} = \left[\begin{smallmatrix} -1 \\ 0 \end{smallmatrix} \right] \Rightarrow \dfrac{\partial \Theta}{\partial x}(0, y) = -\dfrac{\Theta(0, y)}{kR_{tot}}, \forall y \in [0; L_y]$



UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

Construct manufactured solution (steady state case) (2/2)

Choice the solution with this form: $\Theta(x,y) = e^{p(x)}\, e^{g(y)}$

Now compute $p(x)$ and $g(y)$ imposing the BC:

$$\Theta_x(x,y) = p_x(x)\, e^{p(x)} e^{g(y)} \Rightarrow \begin{array}{l} \text{on } \partial\Omega_2 \;\; p_x(L_x) = -\dfrac{1}{kR_{tot}} \\ \text{on } \partial\Omega_4 \;\;\; p_x(0) = +\dfrac{1}{kR_{tot}} \end{array} \Rightarrow p_x(x) = (1 - \frac{2}{L_x}x)\frac{1}{kR_{tot}} \Rightarrow p(x) = x(1 - \frac{1}{L_x}x)\frac{1}{kR_{tot}} + C_x$$

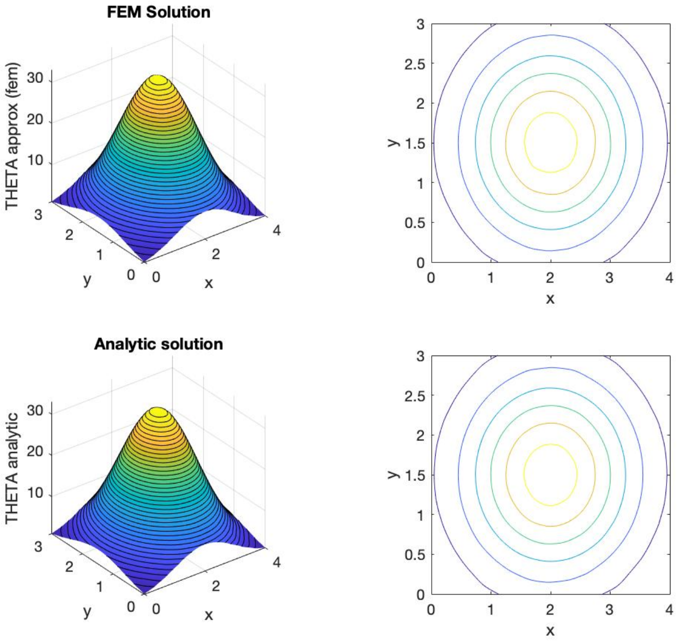
$$\Theta_y(x,y) = g_y(y)\, e^{p(x)} e^{g(y)} \Rightarrow \begin{array}{l} \text{on } \partial\Omega_1 \;\; g_y(L_y) = -\dfrac{1}{kR_{tot}} \\ \text{on } \partial\Omega_3 \;\;\; g_y(0) = +\dfrac{1}{kR_{tot}} \end{array} \Rightarrow g_y(y) = (1 - \frac{2}{L_y}y)\frac{1}{kR_{tot}} \Rightarrow g(y) = y(1 - \frac{1}{L_y}y)\frac{1}{kR_{tot}} + C_y$$

Thus:
$$\begin{array}{l} p(x) = x(1 - \dfrac{1}{L_x}x)\dfrac{1}{kR_{tot}} \\ g(y) = y(1 - \dfrac{1}{L_y}y)\dfrac{1}{kR_{tot}} \end{array}$$

Finally: $s(x,y) = -D\nabla^2\Theta = -D(p_{xx} + p_x^2 + p_{yy} + p_y^2)e^{p(x)}e^{g(x)}$

FEM code verification

Test results (steady state case)



First, compute the error norms considering the true solution and the numerical solutions obtained with different mesh refinements.

$$||\varepsilon||_2 = \left[A_\Omega \int_\Omega (u_h - \tilde{u})^2 dS \right]^{1/2}$$

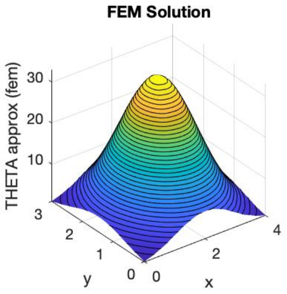
$$||\varepsilon||_{inf} = \max_\Omega (u_h - \tilde{u})$$

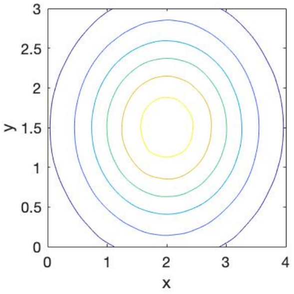
- *L2* norm is computed using gaussian quadrature
- H*inf* norm is computed considering the nodal values

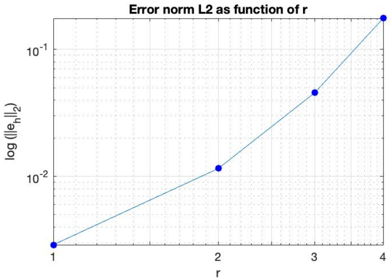
h	1	0.5	0.25	0.125
$\|\|\varepsilon\|\|_2$	1.75×10^{-1}	4.56×10^{-2}	1.15×10^{-2}	2.89×10^{-3}
$\|\|\varepsilon\|\|_{inf}$	1.79×10^{-1}	7.28×10^{-2}	2.52×10^{-2}	8.03×10^{-3}

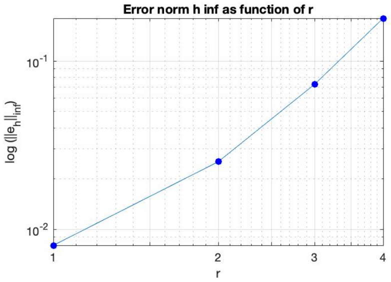
FEM code verification

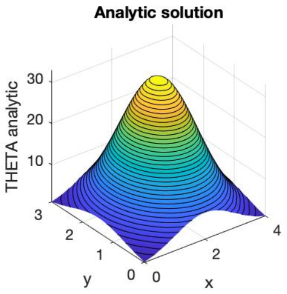
Test results (steady state case)

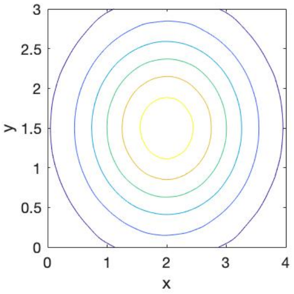










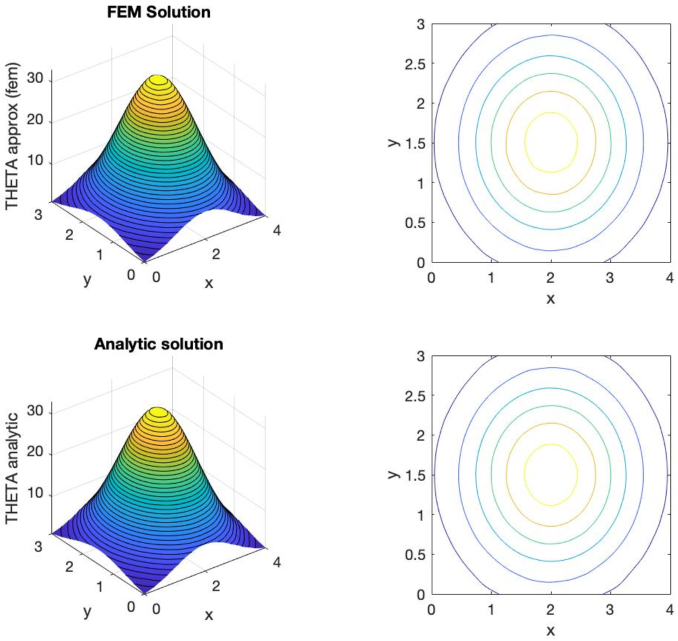


The slope of the line passing through two points can be computed using:

$$\hat{\rho}_k = \frac{ln(\|\frac{\varepsilon_{2kh}}{\varepsilon_{kh}}\|)}{ln(\|2\|)} \text{ for } k = 1, 2, 4$$

FEM code verification

Test results (steady state case)



The observed order of accuracy of the 3 tests are

ρ_1	ρ_2	ρ_3
1.9466	1.9811	1.9941

Shows that they tends to the value of *2* as expected.

Increasing the mesh refinement the value is closer to 2 showing that the numerical solution is in the *asymptotic convergence range*, thus, the lowest-order in the truncation an discretization error expansions dominate the higher-order terms in the power series expansion for the numerical solution.

Verification of unsteady case

execution of the order-of-accuracy test through the MMS method

Luigi D'Amico
Alessandro Melone

Construct manufactured solution (unsteady state case)

$$\boxed{[M]\{\dot{\Theta}\} + [L]\{\Theta\} = \{f\}}$$

$$\{f\} = \begin{bmatrix} \{f_\Omega\} & \{f_{\Gamma_q}\} & \{f_{\Gamma_{R_1}}\} \end{bmatrix} \begin{bmatrix} \mu(t) \\ \bar{q}(t) \\ T(t) \end{bmatrix}$$

The integration of the LTI system has been experimented with different technique

- Explicit Euler
- Crank-Nicolson
- Runge-Kutta 4th order
- Simulink solver (variable step)

Code verification

As already did for the steady-state case, we first build a manufactured solution. Then the obtained solution is compared to the numerical solution in space and in time. The difference is used to determine the test.

Construct manufactured solution (unsteady state case)

The following form is chosen for the solution:

$$\Theta(x, y, t) = e^{p(x)}\, e^{g(y)} e^{q(t)}$$

Imposing the same boundary Robin condition as we did previously, we can obtain the same functions $p(x)$ and $g(y)$.

$$p(x) = x(1 - \frac{1}{L_x}x)\frac{1}{kR_{tot}}$$
$$g(y) = y(1 - \frac{1}{L_y}y)\frac{1}{kR_{tot}}$$

Plugging the theta function into the model:

$$s(x, y)\mu(t) = \frac{\partial \Theta}{\partial t} - D\nabla^2\Theta$$
$$s(x, y)\mu(t) = (q_t - p_{xx} - p_x^2 - g_{yy} - g_y^2)e^{p(x)}e^{g(y)}e^{q(t)}$$

Imposing the initial condition:

$$\Theta(x, y, 0) = \Theta_0(x, y) = e^{p(x)}e^{g(y)}e^{q(0)}$$

Choosing: $q(t) = -ct \rightarrow e^{q(0)} = 1$ with $c \in \mathbb{R}$

thus the initial condition become: $\Theta_0(x, y) = e^{p(x)}\, e^{g(y)}$

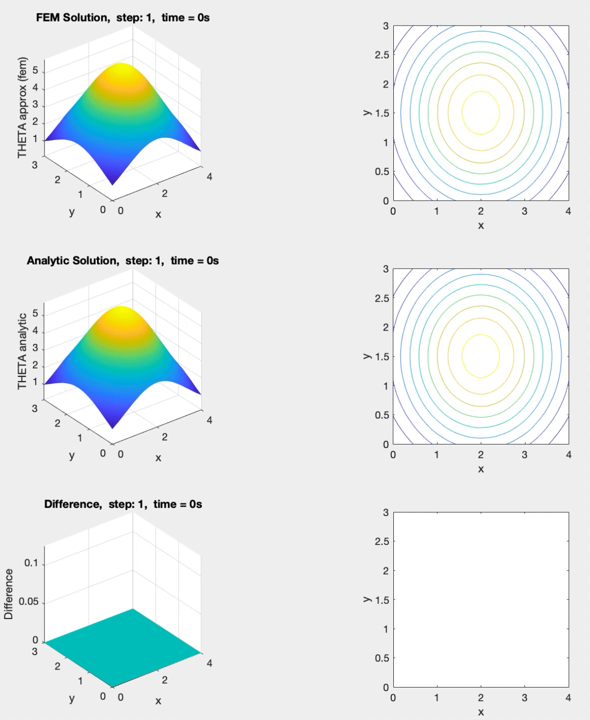
This choice of $q(t)$ allows the factorization:

$$s(x, y) = (c - p_{xx} - p_x^2 - g_{yy} - g_y^2)e^{p(x)}e^{g(y)}$$
$$\mu(t) = e^{-ct}$$

In the following is chosen c=1/20 in order to slow down the convergence to zero of the manufactured solution.

FEM Code verification

Construct manufactured solution (unsteady state case)



The video shows that the numerical solution follows the analytic manufactured solution.

Luigi D’Amico
Alessandro Melone

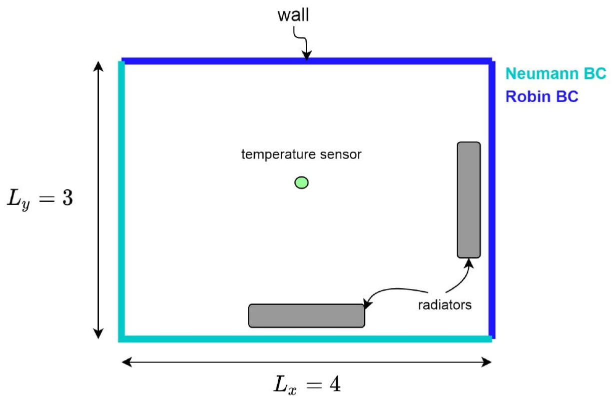
Control and simulations

Implementation of a Relay controller

Luigi D’Amico
Alessandro Melone

Introduction to the control problem

- The goal is to control the nodal temperature located in a specific node on which we suppose that a sensor is installed.
- The sensor value is used as feedback to the control system.
- The control system is a simple relay with hysteresis to avoid excessive switches once the nodal temperature has reached the set-point.



Luigi D'Amico
Alessandro Melone

Introduction to the control problem

- The goal is to control the nodal temperature located in a specific node on which we suppose that a sensor is installed.
- The sensor value is used as feedback to the control system.
- The control system is a simple relay with hysteresis to avoid excessive switches once the nodal temperature has reached the set-point.

The heat source position (i.e. the radiators) is specified by:

$$s(x,y) = \begin{cases} 0 & \text{source is present} \\ 1 & \text{source is not present} \end{cases}$$

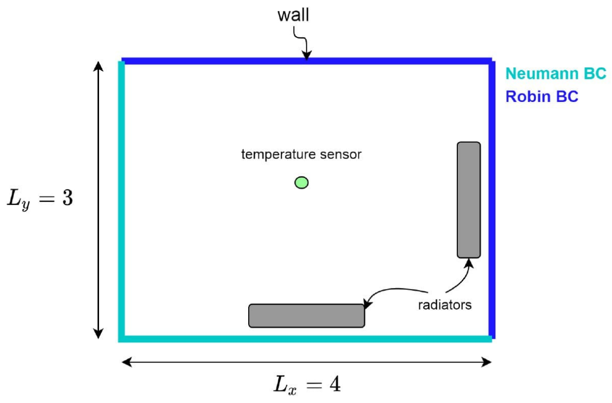
The time-variant input is:

$$\mu(t) = G(s)\bar{\mu}(t)$$

Where:

$G(s)$ is a LPF with gain to model the radiators slow-dynamic and its power

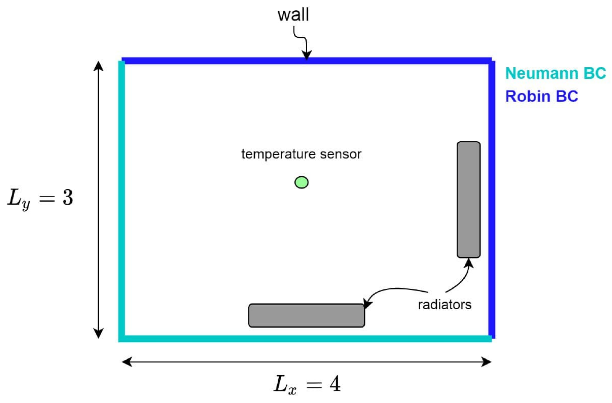
$\bar{\mu}(t)$ is the relay output (1 or 0)



Control and simulations

Case study

- The lengths of the room's wall are set to $L_x = 4$, $L_y = 3$
- Two walls are supposed to exchange heat with external environment (*Robin conditions*) and two wall are supposed to be isolated (*Neumann conditions, q=0*), as shown in the figure.
- The sensor is placed at the center of the room $(x,y) = (2, 1.5)$

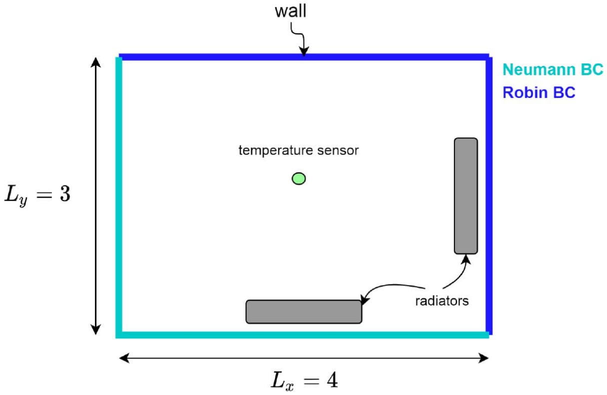


Control and simulations

Choosing parameters to match with physics

- Note that we have neglected the natural convection effects. This assumption would lead to a non satisfactory behaviour compared with the natural phenomena.

- In order to cope with this phenomena and achieve reasonable physical behaviour of the system, some parameters has been modified as the *diffusion coefficient D* and the *wall resistance* R_{tot}.



Control and simulations

Evolution of temperature with no heat source

First of all, it is shown an experiment with no input and with a fixed external temperature $T = 20$.

The experiment shows that all the nodal temperatures go to the external starting from the nodes closest to the Robin condition's walls

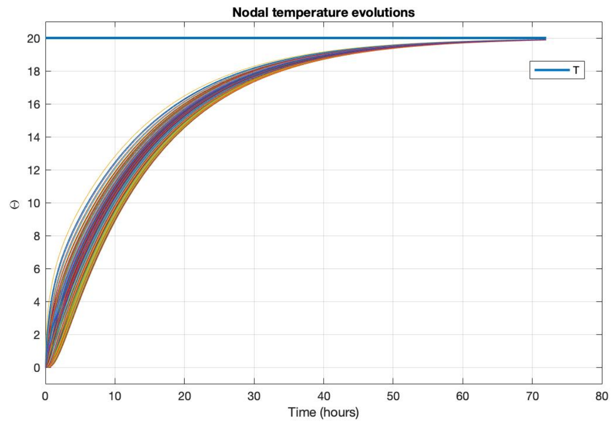
Luigi D'Amico
Alessandro Melone

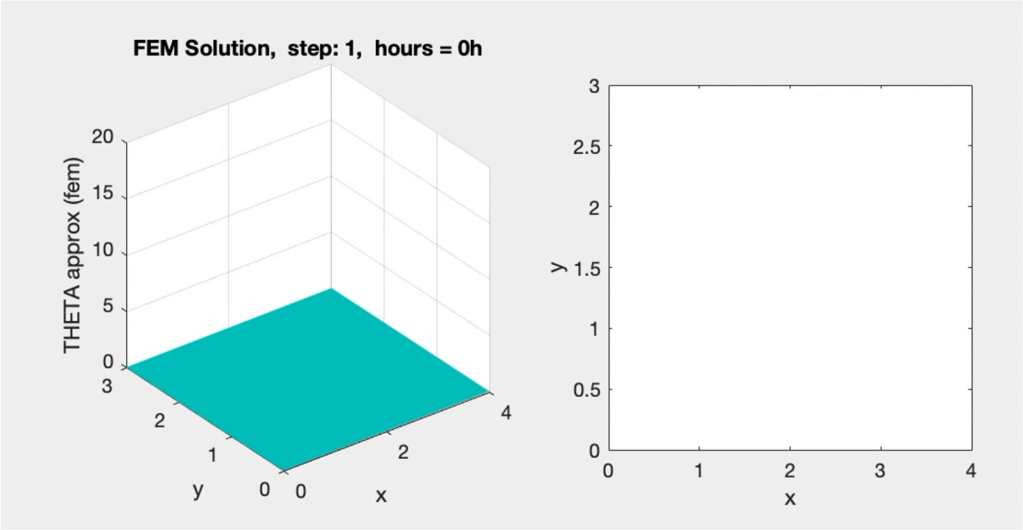
Control and simulations

Evolution of temperature with no heat source

First of all, it is shown an experiment with no input and with a fixed external temperature *T = 20*.

The experiment shows that all the nodal temperatures go to the external starting from the nodes closest to the Robin condition's walls

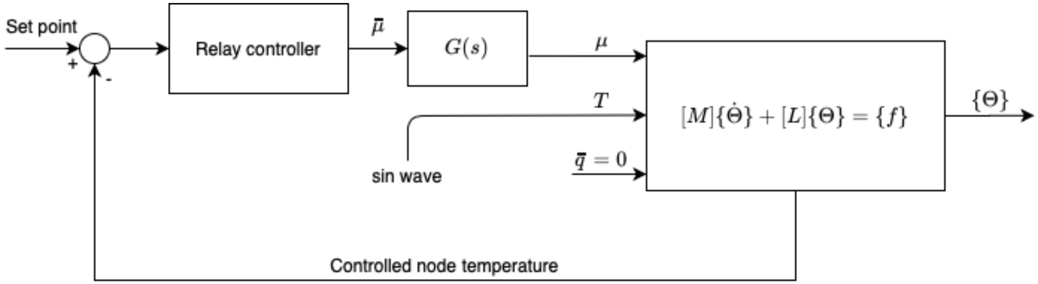




Simulation of external temperature variation

Simulation results with the control action will be shown.

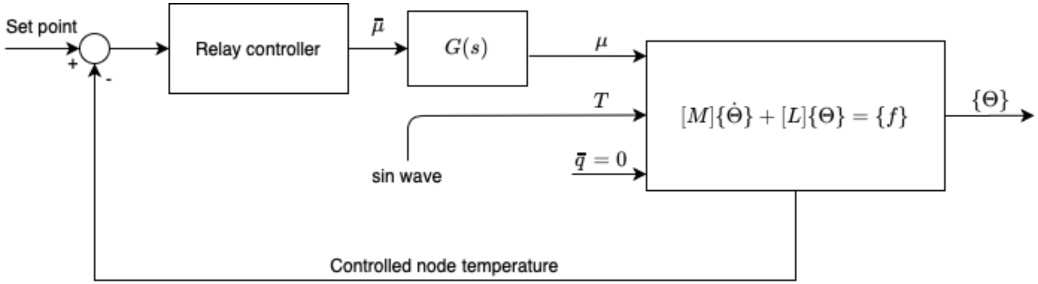
The proposed control scheme is in figure.



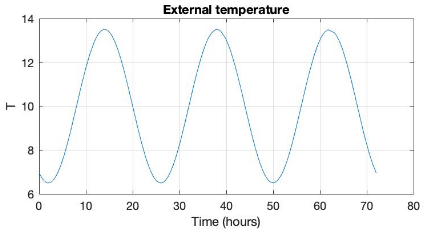
Simulation of external temperature variation

Simulation results with the control action will be shown.

The proposed control scheme is in figure.

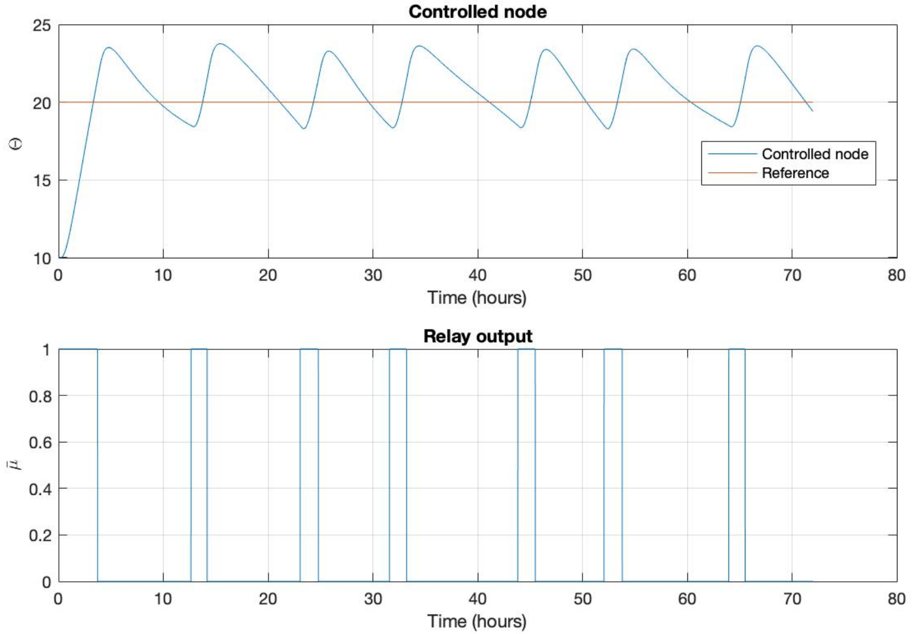


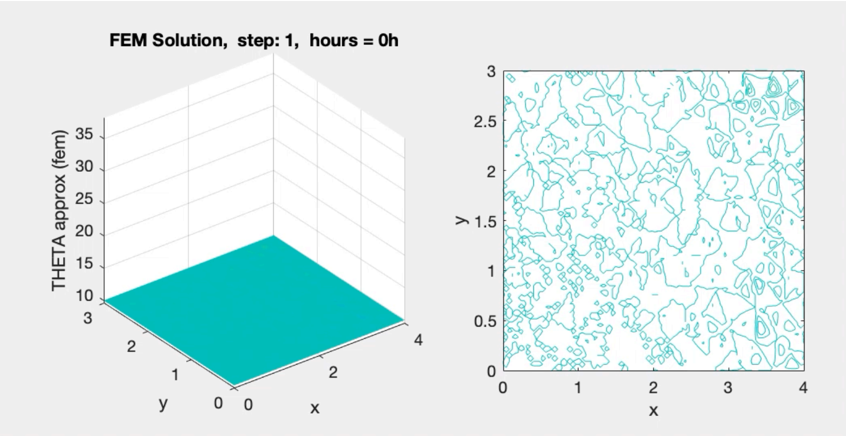
It is supposed that the external temperature *T* vary as a sinusoidal wave with a period of one day, reaching the higher value at 3pm and the lowest value at 3am.



Control and simulations

Control result





Conclusions

Further developments and references

Luigi D'Amico
Alessandro Melone

Further developments

- Parallelize the f.e.m. matrix computation in order to exploit multi-core CPU
- Estimate the observed order of accuracy also for the time-stepping algorithms.

Luigi D’Amico
Alessandro Melone

References

- Notes from the course “Mathematical methods for the field”, Prof. Guglielmo Rubinacci, University Federico II of Naples
- Lecture notes from the course “Mathematical methods for the field”, Prof. Giovanni Miano, University Federico II of Naples
- F. Trevisan, F. Villone, Modelli numerici per campi e circuiti, 2003
- Ioannis Koutromanos, James McClure, Christopher Roy, 2018, Fundamentals of Finite Element Analysis, chap.5,15,16,18
- Rajendra Karwa, 2017, Heat and Mass Transfer, chap.1
- William L. Oberkampf, 2010, Verification and Validation in Scientific Computing, chap.5

Thank you for the attention!

Course of “Mathematical methods for fields”
2019/2020

Prof. Guglielmo Rubinacci

Luigi D’Amico
Alessandro Melone