

Dinàmica de Gasos i Transferència de Calor i Massa

Anàlisi de la transferència de calor en un conducte cilíndric en règim transitori

Alumne: Pedro López Sancha

Professor: Carlos-David Pérez Segarra

Enginyeria en Tecnologies Aeroespacials

Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

Universitat Politècnica de Catalunya

25 de maig de 2020



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

**Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa**

Índex

1	Introducció	2
2	Anàlisi del problema	3
2.1	Hipòtesis simplifcatòries	3
2.2	Equacions de discretització	3
2.2.1	Equació de discretització del node extrem $i = 0$	3
2.2.2	Equació de discretització dels nodes interns $i = 1 \div N$	4
2.2.3	Equació de discretització del node extrem $i = N + 1$	6
2.3	Algoritme de resolució	7
3	Resolució del problema	8
3.1	Dades físiques i numèriques	8
3.2	Gràfiques	8
A	Algoritme TDMA	10
B	Algoritme de resolució	10

1 Introducció

Es considera un tub de radi interior R_1 , radi exterior R_2 i longitud H . La conductivitat tèrmica i la calor específica del material del tub són $\lambda_t(T)$ i $c_p(T)$, respectivament. A més, aquest disposa d'unes fonts internes $\dot{q}_v(t)$. Per la cavitat interior circula un fluid A a temperatura $T_A(t)$ i amb coeficient de transferència de calor $\alpha_A(t)$. L'exterior del tub està en contacte amb un fluid B a temperatura $T_B(t)$ i amb coeficient de transferència de calor $\alpha_B(t)$. La geometria del problema es mostra a la figura 1.

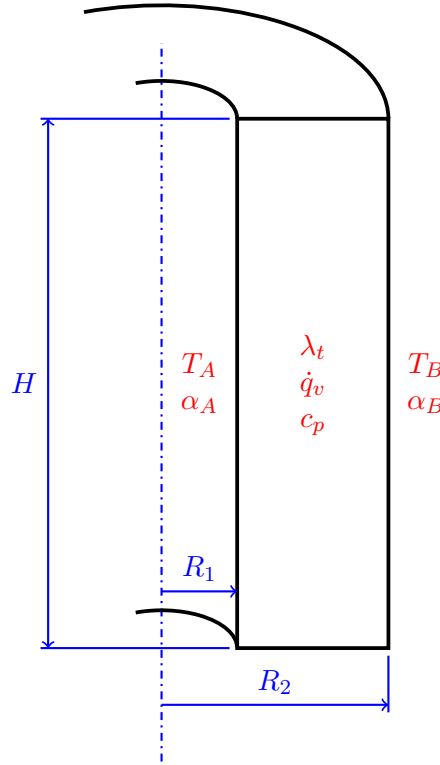


Figura 1. Geometria i propietats termofísiques.

Les temperatures dels fluids A i B varien segons les següents lleis:

$$T_A(t) = T_{A0} + \Delta T_A \sin(2\pi f_A t) \quad (1)$$

$$T_B(t) = T_{B0} + \Delta T_B \sin(2\pi f_B t) \quad (2)$$

on T_{A0} i T_{B0} són les temperatures mitges, ΔT_A i ΔT_B són les amplituds d'oscil·lació, i f_A i f_B són les freqüències d'oscil·lació.

2 Anàlisi del problema

2.1 Hipòtesis simplificatòries

Plantegem una sèrie d'hipòtesis per simplificar l'estudi:

- Considerarem que les funcions $\lambda_t(T)$, $c_p(T)$, $\dot{q}_v(t)$, $\alpha_A(t)$ i $\alpha_B(t)$ són constants.
- Les cares superior i inferior del tub són adiabàtiques.
- Donada la simetria de revolució, la distribució de temperatures $T(t, r, \theta, z)$ no depèn de la direcció angular, i.e., $T(t, r, \theta, z) = T(t, r, z)$. Com el tub té longitud finita i hem imposat que els extrems siguin adiabàtics, la temperatura al llarg de la coordenada z no serà uniforme. No obstant això, per simplificar l'estudi, assumirem $T(t, r, z) \approx T(t, r)$.

2.2 Equacions de discretització

Discretitzem el domini d'estudi per nodes centrats, de manera uniforme, tal com suggereix la figura 2.

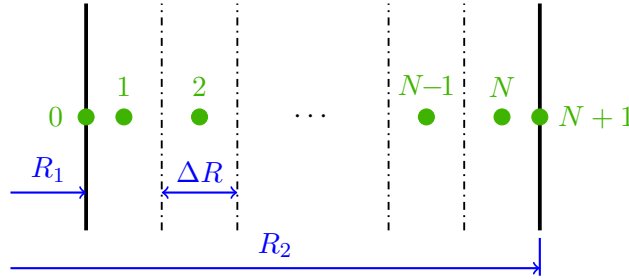


Figura 2. Discretització espacial per nodes centrats.

Al llarg de la direcció radial, dividim el tub en N volums de control, cadascun d'amplada $\Delta R = (R_2 - R_1)/N$. A les cares externes del tub situem els nodes externs 0 i $N + 1$. Al centre de cada volum de control col·loquem el node de discretització i , amb $i = 1 \div N$.

La discretització en el temps també es uniforme, amb un Δt adient a les freqüències f_A i f_B . La integració temporal segueix l'esquema de Crank-Nicolson, i.e., $\beta = \frac{1}{2}$.

2.2.1 Equació de discretització del node extrem $i = 0$

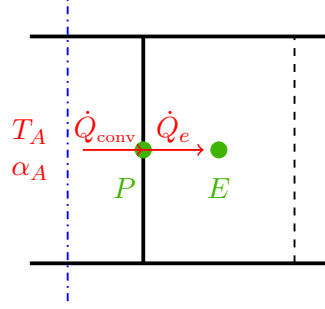
El node $i = 0$ és a la cara interna del cilindre, en contacte amb el fluid A . El diagrama de fluxos de calor es mostra a la figura 3.

Comencem plantejant el primer principi de la Termodinàmica en règim transitori sobre el node P , sabent que $V_P = 0$.

$$\frac{\partial}{\partial t} \int_{V_P} \rho u \, dV = \sum \dot{Q}_P = 0 \quad (3)$$

Integrant el terme dret de 3 entre els instants t^n i t^{n+1} , obtenim:

$$\int_{t^n}^{t^{n+1}} \sum \dot{Q}_P \, dt = \left(\beta \sum \dot{Q}_P^{n+1} + (1 - \beta) \sum \dot{Q}_P^n \right) \Delta t = 0 \quad (4)$$

**Figura 3.** Fluxos de calor en el node $i = 0$.

on el flux de calor net a través de P en un instant qualsevol t^n és

$$\dot{Q}_P^n = \dot{Q}_{\text{conv},A}^n - \dot{Q}_e^n = \alpha_A^n (T_A^n - T_P^n) S_{\text{conv},A} + \lambda_e^n \frac{T_E^n - T_P^n}{d_{PE}} S_e \quad (5)$$

Introduint (5) en (4) i desenvolupant, obtenim

$$\beta \left(\frac{\lambda_e^{n+1} S_e}{d_{PE}} + \alpha_A^{n+1} S_{\text{conv},A} \right) T_P^{n+1} = \beta \frac{\lambda_e^{n+1} S_e}{d_{PE}} T_E^{n+1} + \beta \alpha_A^{n+1} T_A^{n+1} S_{\text{conv},A} + (1 - \beta) \sum \dot{Q}_P^n \quad (6)$$

Definim els següents coeficients de discretització pel node d'estudi:

$$a_W[0] = 0 \quad (7)$$

$$a_E[0] = \beta \frac{\lambda_e^{n+1} S_e}{d_{PE}} \quad (8)$$

$$a_P[0] = a_W[0] + a_E[0] + \beta \alpha_A^{n+1} S_{\text{conv},A} \quad (9)$$

$$b_P[0] = \beta \alpha_A^{n+1} T_A^{n+1} S_{\text{conv},A} + (1 - \beta) \sum \dot{Q}_P^n \quad (10)$$

Les posicions dels nodes són $R[0] = R_A$ i $R[1] = R_A + \frac{\Delta R}{2}$. A partir d'aquestes, tenim

$$S_e = S_{\text{conv},A} = 2\pi R_A H \quad d_{PE} = R[1] - R[0] = \frac{\Delta R}{2}$$

L'equació de discretització que obtenim pel node $i = 0$ és

$$a_P[0] T^{n+1}[0] = a_W[0] T^{n+1}[-1] + a_E[0] T^{n+1}[1] + b_P[0] \quad (11)$$

2.2.2 Equació de discretització dels nodes interns $i = 1 \div N$

Els nodes interns són aquells amb índexs $1 \leq i \leq N$. El diagrama de fluxos de calor per aquests nodes es mostra a la figura 4.

Per trobar l'equació de discretització, comencem plantejant el primer principi de la Termodinàmica en règim transitori sobre el volum de control del node P :

$$\frac{\partial}{\partial t} \int_{V_P} \rho u dV = \sum \dot{Q}_P \quad (12)$$

No coneixem l'expressió de l'energia interna en el VC P . Per aquest motiu definim l'energia interna mitja del volum de control com

$$\bar{u}_P = \frac{1}{\rho_P V_P} \int_{V_P} \rho u dV \quad (13)$$

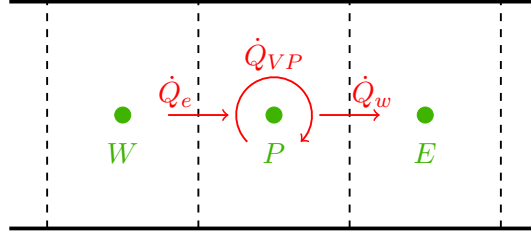


Figura 4. Fluxos de calor en els nodes interns del tub ($i = 1 \div N$).

Atès que ρ_P i V_P són constants, podem escriure (12) com:

$$\frac{\partial(\rho_P V_P \bar{u}_P)}{\partial t} = \rho_P V_P \frac{\partial \bar{u}_P}{\partial t} = \sum \dot{Q}_P \quad (14)$$

Integrant entre dos temps t^n i t^{n+1} tenim

$$\int_{t^n}^{t^{n+1}} \rho_P V_P \frac{\partial \bar{u}_P}{\partial t} dt = \int_{t^n}^{t^{n+1}} \sum \dot{Q}_P dt \quad (15)$$

Desenvolupem primerament el terme esquerre de (15). Sabem que l'energia interna és una funció de punt, per tant la integral de $\partial \bar{u}_P / \partial t$ entre els instants t^n i t^{n+1} no depèn del procés seguit sino dels estats inicial i final. Podem aproximar el sòlid per un sòlid semiperfecte i fer $\bar{u}_P \approx u_P$. Així mateix, sabem que $du = c_p dT$. Amb aquestes deduccions, el terme esquerre de (15) queda

$$\begin{aligned} \int_{t^n}^{t^{n+1}} \rho_P V_P \frac{\partial \bar{u}_P}{\partial t} dt &= \rho_P V_P \int_{t^n}^{t^{n+1}} \frac{\partial \bar{u}_P}{\partial t} dt = \rho_P V_P (\bar{u}_P^{n+1} - \bar{u}_P^n) \\ &\approx \rho_P V_P (u_P^{n+1} - u_P^n) = \rho_P V_P \bar{c}_{pP} (T_P^{n+1} - T_P^n) \end{aligned} \quad (16)$$

on \bar{c}_{pP} és

$$\bar{c}_{pP} = \frac{1}{T_P^{n+1} - T_P^n} \int_{T_P^n}^{T_P^{n+1}} c_p(\tau) d\tau \approx c_p$$

El terme dret de (15) desenvolupat queda

$$\int_{t^n}^{t^{n+1}} \sum \dot{Q}_P dt = (\beta \sum \dot{Q}_P^{n+1} + (1 - \beta) \sum \dot{Q}_P^n) \Delta t \quad (17)$$

Introduint (16) i (17) en (15) obtenim

$$\rho_P V_P \bar{c}_{pP} \frac{T_P^{n+1} - T_P^n}{\Delta t} = \beta \sum \dot{Q}_P^{n+1} + (1 - \beta) \sum \dot{Q}_P^n \quad (18)$$

on el flux de calor net en un instant general t^n és

$$\sum \dot{Q}_P^n = \dot{Q}_w^n - \dot{Q}_e^n + \dot{Q}_{VP}^n \approx -\lambda_w^n \frac{T_P^n - T_W^n}{d_{PW}} S_w + \lambda_e^n \frac{T_E^n - T_P^n}{d_{PE}} S_e + \dot{q}_V^n V_P \quad (19)$$

Desenvolupant (18) arribem a

$$\begin{aligned} \left[\beta \left(\frac{\lambda_w^{n+1} S_w}{d_{PW}} + \frac{\lambda_e^{n+1} S_e}{d_{PE}} \right) + \frac{\rho_P V_P c_p}{\Delta t} \right] T_P^{n+1} &= \\ &= \beta \frac{\lambda_w^{n+1} S_w}{d_{PW}} T_W^{n+1} + \beta \frac{\lambda_e^{n+1} S_e}{d_{PE}} T_E^{n+1} + \beta \dot{q}_V^{n+1} V_P + \frac{\rho_P V_P c_p}{\Delta t} T_P^n + (1 - \beta) \sum \dot{Q}_P^n \end{aligned} \quad (20)$$

Definim els següents coeficients de discretització

$$a_W[i] = \beta \frac{\lambda_w^{n+1} S_w}{d_{PW}} \quad (21)$$

$$a_E[i] = \beta \frac{\lambda_e^{n+1} S_e}{d_{PE}} \quad (22)$$

$$a_P[i] = a_W[i] + a_E[i] + \frac{\rho_P V_P c_p}{\Delta t} \quad (23)$$

$$b_P[i] = \beta \dot{q}_V^{n+1} V_P + \frac{\rho_P V_P c_p}{\Delta t} T_P^n + (1 - \beta) \sum \dot{Q}_P^n \quad (24)$$

Les posicions dels nodes involucrats són $R_W = R[i - 1]$, $R_P = R[i] = R_A + \frac{2i-1}{2} \Delta R$ i $R_E = R[i + 1]$. Les posicions de les cares w i e són $R_w = R[i] - \frac{\Delta R}{2}$ i $R_e = R[i] + \frac{\Delta R}{2}$. A partir d'aquestes tenim les següents superfícies i distàncies

$$\begin{aligned} S_w &= 2\pi R_w H & d_{PW} &= R[i] - R[i - 1] \\ S_e &= 2\pi R_e H & d_{PE} &= R[i + 1] - R[i] \\ V_P &= \pi(R_e^2 - R_w^2)H \end{aligned}$$

Així, l'equació de discretització dels nodes interns queda:

$$a_P[i] T^{n+1}[i] = a_W[i] T^{n+1}[i - 1] + a_E[i] T^{n+1}[i + 1] + b_P[i], \quad i = 1 \div N \quad (25)$$

2.2.3 Equació de discretització del node extrem $i = N + 1$

El node $i = N + 1$ és a la cara externa del cilindre, en contacte amb el fluid B . El diagrama de fluxos de calor sobre aquest node es mostra a la figura 5.

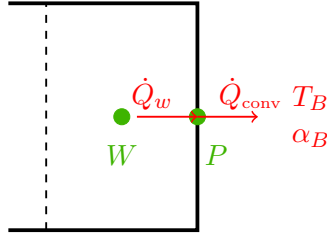


Figura 5. Fluxos de calor en el node $i = N + 1$.

L'obtenció de l'equació de discretització d'aquest node és similar a la del node $i = 0$. Partim de l'equació (4), on en aquest cas, el flux de calor net sobre P en un instant qualsevol t^n és

$$\sum \dot{Q}_P^n = \dot{Q}_w^n - \dot{Q}_{\text{conv},B}^n = -\lambda_w^n \frac{T_P^n - T_W^n}{d_{PW}} S_w - \alpha_B^n (T_P^n - T_B^n) S_{\text{conv},B} \quad (26)$$

Introduint els fluxos de calor net en t^n i t^{n+1} en (4) arribem a

$$\beta \left(\frac{\lambda_w^{n+1} S_w}{d_{PW}} + \alpha_B^{n+1} S_{\text{conv},B} \right) T_P^{n+1} = \beta \frac{\lambda_w^{n+1} S_w}{d_{PW}} T_W^{n+1} + \beta \alpha_B^{n+1} T_B^{n+1} S_{\text{conv},B} + (1 - \beta) \sum \dot{Q}_P^n \quad (27)$$

Definim els següents coeficients de discretització:

$$a_W[N+1] = \beta \frac{\lambda_w^{n+1} S_w}{d_{PW}} \quad (28)$$

$$a_E[N+1] = 0 \quad (29)$$

$$a_P[N+1] = a_W[N+1] + a_E[N+1] + \beta \alpha_B^{n+1} S_{\text{conv},B} \quad (30)$$

$$b_P[N+1] = \beta \alpha_B^{n+1} T_B^{n+1} S_{\text{conv},B} + (1 - \beta) \sum \dot{Q}_P^n \quad (31)$$

Introduint-los a 27, obtenim l'equació de discretització del node $i = N + 1$:

$$a_P[N+1] T^{n+1}[N+1] = a_W[N+1] T^{n+1}[N] + a_E[N+1] T^{n+1}[N+2] + b_P[N+1] \quad (32)$$

2.3 Algoritme de resolució

A continuació mostrem l'algorisme de resolució del problema plantejat.

Algoritme 1 Algoritme de resolució

1. Entrada de dades:

- Dades físiques: $R_1, R_2, H, \dot{q}_v, \lambda_t, c_p, T_A(t), \alpha_A, T_B(t), \alpha_B$.
- Dades numèriques: $N, \delta, \beta, \Delta t, T_0^*$.

2. Càlculs previs: posició dels nodes, càlcul de superfícies i volums.

3. Mapa inicial de temperatures: $T^n[i] = T_0(r[i]) \ i = 0 \div N + 1$.

4. Càlcul del nou instant $t^{n+1} = t^n + \Delta t$:

4.1. Estimació del mapa inicial: $T^{n+1,*}[i] = T^*[i] \ i = 0 \div N + 1$.

4.2. Càlcul dels coeficients de discretització: $a_W[i], a_E[i], a_P[i], b_P[i] \ i = 0 \div N + 1$.

4.3. Resolució de les equacions de discretització

$$a_P[i] T^{n+1}[i] = a_W[i] T^{n+1}[i-1] + a_E[i] T^{n+1}[i+1] + b_P[i] \ i = 0 \div N + 1$$

amb TDMA.

4.4. És $\max_{0 \leq i \leq N+1} \{ |T^{n+1}[i] - T^{n+1,*}[i]| \} < \delta$?

- Sí: Continua.
- No: $T^{n+1,*}[i] = T^{n+1}[i] \ i = 0 \div N + 1$, torna a 4.2.

5. Nou instant de temps t^{n+1} ?

- Sí: $T^n[i] = T^{n+1}[i]$, torna a 4.
- No: Continua.

6. Càlculs finals i impressió de resultats.

7. Fi

3 Resolució del problema

3.1 Dades físiques i numèriques

La geometria del problema ve donada per

$$R_1 = 0.10 \text{ m} \quad R_2 = 0.10 \text{ m} \quad H = 1 \text{ m}$$

Les propietats termofísiques del cilindre són:

$$\dot{q}_v = 10^4 \frac{\text{W}}{\text{m}^2} \quad \rho = 2700 \frac{\text{kg}}{\text{m}^3} \quad c_p = 900 \frac{\text{J}}{\text{kg K}} \quad \lambda = 50 \frac{\text{W}}{\text{m K}}$$

Les propietats termofísiques dels fluids són les següents:

$$\begin{aligned} \alpha_A &= 1300 \frac{\text{W}}{\text{m}^2} & T_A(t) &= 30 + 20 \sin\left(\frac{2\pi}{3600}t\right) ^\circ\text{C} \\ \alpha_B &= 120 \frac{\text{W}}{\text{m}^2} & T_B(t) &= 30 + 5 \sin\left(\frac{2\pi}{24 \cdot 3600}t\right) ^\circ\text{C} \end{aligned}$$

Per últim, les dades numèriques són $N = 50$, $\delta = 10^{-6}$, $\beta = \frac{1}{2}$, $\Delta t = 1 \text{ s}$ i $T_0^* = 30 ^\circ\text{C}$.

3.2 Gràfiques

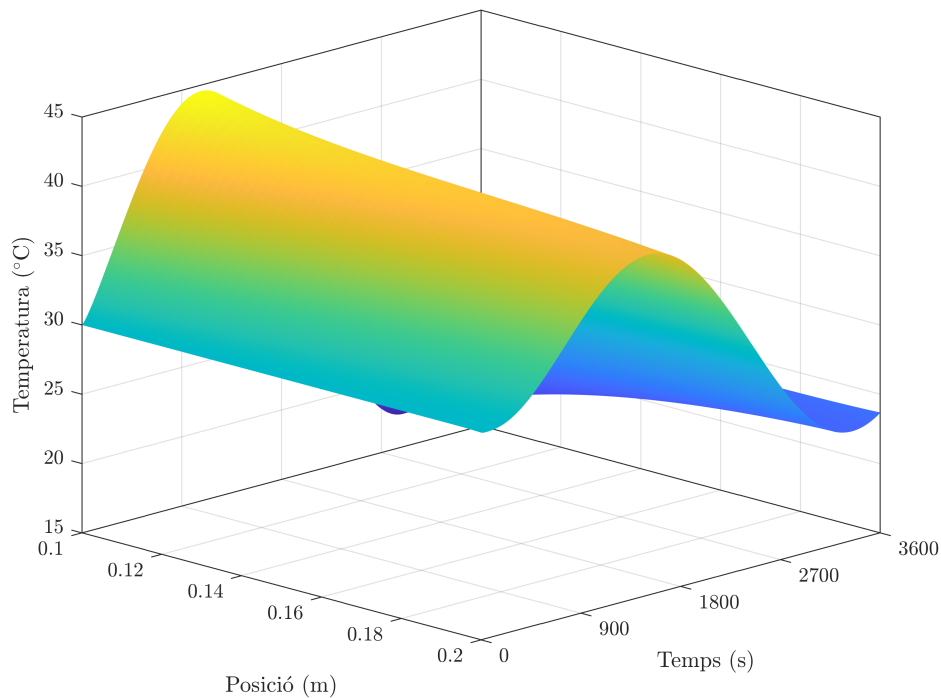


Figura 6. Superfície Temperatura–Posició–Temps

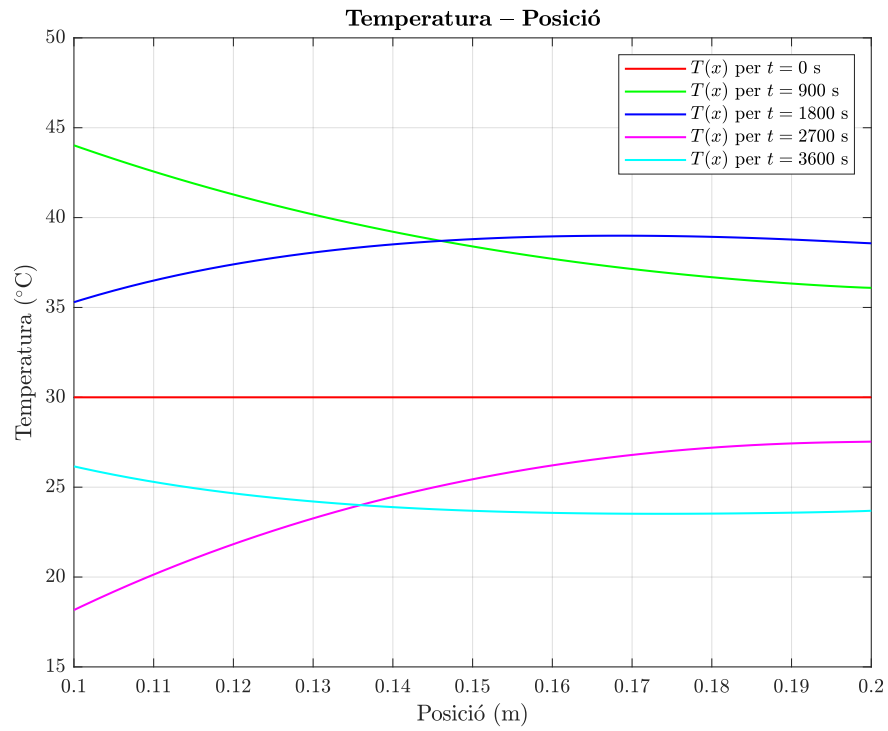


Figura 7. Temperatura en funció de la posició ($T(x)$) pels temps 0 s, 900 s, 1800 s, 2700 s i 3600 s.

A Algoritme TDMA

```

1 function [x] = tdma(A, b)
2 %-----
3 % Description: this function computes the solution to the linear system
4 % A x = b, where A is a tri-diagonal matrix
5 %-----
6 % Inputs: solution vectors
7 %   - A      [n x 3] matrix, where col 1 is lower-diagonal, col 2 is
8 %             the main diagonal and col 3 is the upper-diagonal
9 %   - b      Linear system vector      [n x 1]
10 %-----
11 % Outputs:
12 %   - x      Solution to A x = b
13 %-----
14
15 n = length(b); % Linear system dimension
16
17 % P coefficients vector
18 P = zeros(n, 1);
19 P(1) = A(1,3)/A(1,2);
20 for i = 2:n-1
21     P(i) = A(i,3)/(A(i,2) - A(i,1)*P(i-1));
22 end
23
24 % R coefficients vector
25 R = zeros(n, 1);
26 R(1) = b(1)/A(1,2);
27 for i = 2:n
28     R(i) = (b(i) + A(i,1)*R(i-1))/(A(i,2) - A(i,1)*P(i-1));
29 end
30
31 % Linear system solution
32 x = zeros(n, 1);
33 x(n) = R(n);
34 for i = n-1:-1:1
35     x(i) = P(i)*x(i+1) + R(i);
36 end
37
38 end

```

B Algoritme de resolució

```

1 clear;
2 close all;
3 clc;
4
5 %% 1. DATA INPUT
6 % 1.1 Physical data
7 % Geometrical data
8 R1 = 0.10;           % Cylinder internal radius [m]
9 R2 = 0.20;           % Cylinder external radius [m]
10 H = 1;               % Cylinder height [m]
11
12 % Cylinder thermophysical data
13 qv = 1e4;             % Internal heat source [W/m^3]
14 rho = 2700;           % Density [kg/m^3]

```

```

15 cp0 = 900;           % Specific heat           [J/kg K]
16 lambda0 = 50;        % Thermal conductivity   [W/m K]
17
18 % External fluids thermophysical data
19 alphaA0 = 1300;       % Heat transfer coefficient of fluid A [W/m^2 K]
20 alphaB0 = 120;        % Heat transfer coefficient of fluid B [W/m^2 K]
21
22 TA0 = 30;             % Average temperature of fluid A      [Celsius]
23 TB0 = 30;             % Average temperature of fluid B      [Celsius]
24
25 DeltaTA = 20;         % Amplitude of temperature oscillation of fluid A [Celsius]
26 DeltaTB = 5;          % Amplitude of temperature oscillation of fluid A [Celsius]
27
28 fA = 1/3600;           % Oscillation frequency for fluid A    [1/s]
29 fB = 1/(24*3600);      % Oscillation frequency for fluid B    [1/s]
30
31 TA = @(t) TA0 + DeltaTA*sin(2*pi*fA*t); % Fluid A temperature [Celsius]
32 TB = @(t) TB0 + DeltaTB*sin(2*pi*fB*t); % Fluid B temperature [Celsius]
33
34 lambda = @(T) 0*T+lambda0;
35 cp = @(T) 0*T+cp0;
36 alphaA = @(T) 0*T+alphaA0;
37 alphaB = @(T) 0*T+alphaB0;
38
39 % 1.2 Numerical data
40 N = 50;
41 Δ = 1e-6;
42 beta = 0.5;
43 t_step = 1;
44 T0_star = 30;
45
46 %% 2. PREVIOUS COMPUTATIONS
47 % 2.1 Nodes position
48 DeltaR = (R2-R1)/N;
49 r_node = zeros(N+2,1);
50 r_node(1) = R1;
51 r_node(2:end-1) = [R1+DeltaR/2:DeltaR:R2-DeltaR/2];
52 r_node(end) = R2;
53
54 % 2.2 Distances
55 d = zeros(N+1,1); % d(i) is the distance between nodes i and i+1
56 for i = 1:N+1
57     d(i) = r_node(i+1)-r_node(i);
58 end
59
60 % Surfaces
61 r_surf = [R1:DeltaR:R2]';
62 S = 2*pi*r_surf*H;
63
64 % Volumes
65 V = zeros(N+2,1);
66 for i = 2:N+1
67     V(i) = pi*(r_surf(i)^2-r_surf(i-1)^2)*H;
68 end
69
70 %% 3. INITIAL MAP
71 t_max = 3600; % Maximum time
72 M = t_max/t_step; % Number of time instants
73 T = zeros(N+2,M+1); % Temperature matrix
74 T(:,1) = T(:,1) + T0_star; % Initial map
75 Q = zeros(N+2,M+1); % Heat flows

```

```

76 Q(:,1) = 0;
77
78 %% 4. COMPUTE NEXT INSTANT OF TIME
79 n = 1;
80 while n < M+1
81     % 4.1 ESTIMATION OF THE INITIAL TEMPERATURE MAP
82     T_est = T(:,n);
83     error = intmax;
84     while error > Δ
85         % 4.2 COMPUTATION OF THE DISCRETIZATION COEFFICIENTS
86         A = zeros(N+2,3);
87         b = zeros(N+2,1);
88         % Node 1
89         A(1,1) = 0;
90         A(1,3) = beta*lambda(T_est(2))*S(1)/d(1);
91         A(1,2) = A(1,1) + A(1,3) + beta*alphaA(T_est(1))*S(1);
92         QPn = alphaA(T(1,n))*(TA((n-1)*t_step)-T(1,n))*S(1)+...
93             lambda(T(2,n))*(T(2,n)-T(1,n))*S(1)/d(1);
94         b(1) = beta*alphaA(T_est(1))*TA(n*t_step)*S(1)+(1-beta)*QPn;
95         % Internal nodes 2 ≤ i ≤ N+1
96         for i = 2:N+1
97             lambda_w = d(i-1)/((r_surf(i-1)-r_node(i-1))/lambda(T_est(i-1))...
98                 + (r_node(i)-r_surf(i-1))/lambda(T_est(i)));
99             A(i,1) = beta*lambda_w*S(i-1)/d(i-1);
100             lambda_e = d(i)/((r_surf(i)-r_node(i))/lambda(T_est(i)) + ...
101                 (r_node(i+1)-r_surf(i))/lambda(T_est(i+1))));
102             A(i,3) = beta*lambda_e*S(i)/d(i);
103             A(i,2) = A(i,1) + A(i,3) + rho*V(i)*cp(T_est(i))/t_step;
104
105             lambda_w = d(i-1)/((r_surf(i-1)-r_node(i-1))/lambda(T(i-1,n)) + ...
106                 (r_node(i)-r_surf(i-1))/lambda(T(i,n))));
107             lambda_e = d(i)/((r_surf(i)-r_node(i))/lambda(T(i,n)) + ...
108                 (r_node(i+1)-r_surf(i))/lambda(T(i+1,n))));
109             Qpn = -lambda_w*(T(i,n)-T(i-1,n))*S(i-1)/d(i-1) + ...
110                 lambda_e*(T(i+1,n)-T(i,n))*S(i)/d(i) + qv*V(i);
111             b(i) = beta*qv*V(i) + rho*V(i)*cp(T(i,n))*T(i,n)/t_step + (1-beta)*QPn;
112         end
113         % Node N+2
114         A(N+2,1) = beta*lambda(T_est(N+1))*S(N+1)/d(N+1);
115         A(N+2,3) = 0;
116         A(N+2,2) = A(N+2,1) + A(N+2,3) + beta*alphaB(T_est(N+2))*S(N+1);
117         QPn = -lambda(T(N+1,n))*(T(N+2,n)-T(N+1,n))*S(N+1)/d(N+1) - ...
118             alphaB(T(N+2,n))*(T(N+2,n)-TB((n-1)*t_step))*S(N+1);
119         b(N+2) = beta*alphaB(T_est(N+2))*TB(n*t_step) + (1-beta)*QPn;
120         % 4.3 SOLVE THE LINEAR SYSTEM USING TDMA
121         T(:,n+1) = tdma(A, b);
122         % 4.4 CHECK CONVERGENCE
123         T_est = T(:,n+1);
124         error = max(abs(T(:,n+1)-T_est));
125     end
126     %
127     % 5. NEXT INSTANT OF TIME
128     n = n + 1;
129 end
130
131 %% 5. PRINT RESULTS
132
133 % 5.1 Set interpreter latex
134 set(groot,'defaultAxesTickLabelInterpreter','latex');
135 set(groot,'defaulttextinterpreter','latex');
136 set(groot,'defaultLegendInterpreter','latex');

```

```

137
138 % 5.2 3D Plot
139 [X,Y] = meshgrid(r_node, [0:t_step:t_max]);
140
141 figure(1);
142 hold on;
143 s = surf(X,Y,T');
144 s.EdgeColor = 'none';
145 xlabel("Posici\ 'o $(\mathrm{m})$");
146 ylabel("Temps $(\mathrm{s})$");
147 zlabel("Temperatura $(^{\circ}\mathrm{C})$");
148 set(gcf, 'units', 'pixels', 'position', [1000,400,1.2*560,1.2*420]);
149 ylim([0 3600]);
150 yticks([0:900:3600]);
151 box on;
152 grid on;
153 view([45 20]);
154 hold off;
155
156 % 5.3 T plot for different instants of time
157 t_plot = [0:900:3600]+1;
158 legend_str = cell(1,length(t_plot));
159 [asd, index] = max(max(T));
160 h3 = figure(3);
161 title("\textbf{Temperatura -- Posici\ 'o}");
162 hold on;
163 for i = 1:length(t_plot)
164     legend_str(i) = {sprintf("$T(x)$ per $t = %d$ \ \mathrm{s}$", t_plot(i)-1)};
165 end
166 plot(r_node, T(:,t_plot(1)), 'r', 'LineWidth', 1);
167 plot(r_node, T(:,t_plot(2)), 'g', 'LineWidth', 1);
168 plot(r_node, T(:,t_plot(3)), 'b', 'LineWidth', 1);
169 plot(r_node, T(:,t_plot(4)), 'm', 'LineWidth', 1);
170 plot(r_node, T(:,t_plot(5)), 'c', 'LineWidth', 1);
171 xlabel("Posici\ 'o $(\mathrm{m})$");
172 ylabel("Temperatura $(^{\circ}\mathrm{C})$");
173 set(gcf, 'units', 'pixels', 'position', [1000,400,1.2*560,1.2*420]);
174 xlim([0.1 0.2]);
175 ylim([15 50]);
176 legend(legend_str);
177 box on;
178 grid on;
179 hold off;
180
181 % 5.4 Save plot as pdf
182 set(h3, 'Units', 'Centimeters');
183 pos = get(h3, 'Position');
184 set(h3, 'PaperPositionMode', 'Auto', 'PaperUnits', 'Centimeters', ...
185     'PaperSize', [pos(3), pos(4)]);
186 print(h3, 'figures/temperatura_posicio', '-dpdf', '-r0');

```