Numerical flow simulation
ME-474

Assignment 1

Flow through a plane channel

Groupe 27

Michele Poli 342543
Alexandro Majocchi 346190
Sarah Naili 363174
Miguel Moor 329379

Autumn semester 2025

École Polytechnique Fédérale de Lausanne

Table des matières

1 Introduction 3

2 Theoretical analysis and problem definition 4
2.1 Problem statement . . . 4
2.2 Q1 — Expression of the velocity field . . . 4
2.3 Q2 — Integral form over an inner control volume . . . 5
2.4 Q3 & Q4 — Discretization of diffusion and convection terms . . . 6
2.4.1 Diffusive term — Central Differencing (CD) . . . 6
2.4.2 Convective term — Upwind Differencing (UD) . . . 6
2.4.3 Discrete balance for one inner CV . . . 7
2.4.4 Dimensional consistency check . . . 7
2.5 Q5 — Indexing convention for the structured mesh . . . 8
2.6 Q6 — Numbering of degrees of freedom (DOF) . . . 8

3 Implementation of boundary conditions. Questions 7 → 9 9
3.1 Q7 — Boundary condition at the inlet (Dirichlet) . . . 9
3.2 Q8 — Outlet boundary condition (Neumann) . . . 9
3.3 Q9 — Isothermal wall boundary condition (Dirichlet) . . . 10

4 Physical parameters and preliminary verification. Questions 10 → 12 11
4.1 Q10 — Physical parameters and Péclet number calculation . . . 11
4.2 Q11 — First solution on the base mesh and visuals . . . 12
4.3 Q12 — A posteriori verification of boundary conditions . . . 13

5 Analysis of results and derived quantities. Questions 13 → 15 14
5.1 Q13 — Outlet profile, centerline and mean temperatures, and entrance length x_e . . 14
5.2 Q14 — SOR iterative solver and convergence behavior . . . 15
5.3 Q15 — Local Nusselt number along the lower wall . . . 16

6 Study of convergence and numerical accuracy. Questions 16 → 19 18
6.1 Q16 — Mesh refinement and grid independence study . . . 18
6.2 Q17 — Convection schemes comparison (UD vs QUICK) . . . 19
6.3 Q18 — Effect of mesh size on x_e and $\min_x Nu_T(x)$ . . . 21
6.4 Q19 — Effect of the Péclet number on $T(x, y)$ . . . 22

7 Advanced boundary conditions and energy balances. Questions 20 → 22 23
7.1 Q20 — Neumann wall boundary conditions and heat–flux formulation . . . 23
7.1.1 Temperature field . . . 23
7.1.2 Discrete wall equation . . . 23
7.1.3 Local Nusselt number . . . 24
7.1.4 Mesh–refinement analysis . . . 25
7.1.5 Verification of boundary conditions . . . 25
7.2 Q21 — Global energy balance . . . 25
7.3 Q22 — Local bottom-wall heating and target outlet temperature . . . 26
7.3.1 Discussion and checks . . . 27

8 Annexes 29

Table des figures

1 Structured mesh indexing and neighbor notation used in the finite–volume formulation. 8
2 Temperature field $T(x, y)$ (color map). 12
3 Top left : inlet profile $T(0, y) = 50°\mathrm{C}$. Top right / bottom left : bottom and top wall temperatures, showing the imposed isothermal wall at $T_{\mathrm{wall}} = 100°\mathrm{C}$ except at the inlet corners (treated as inlet). Bottom right : outlet check, confirming $\partial T/\partial x \approx 0$ (zero difference between the last two streamwise stations). 14
4 (left) Outlet temperature profile $T_o(y)$; (middle) evolution of $T_c(x)$ and $T_{\mathrm{mean}}(x)$; (right) normalized centerline temperature showing $x_e \approx 3.32$ m where $T_c = 0.9\,T_{\mathrm{wall}}$. 15
5 Convergence history of the SOR solver. Solid lines : normalized residual $r^{(k)}$. Dashed lines : relative error $\varepsilon^{(k)}$. Blue / orange : $\omega = 1.0$ (Gauss–Seidel limit). Yellow / purple : $\omega = 1.5$ (over–relaxation). 16
6 Local Nusselt number $\mathrm{Nu}_T(x)$ along the lower wall. 17
7 Comparison of the outlet profile $T_o(y)$, the axial evolution of $T_c(x)$ and $T_{\mathrm{mean}}(x)$, the local Nusselt number $\mathrm{Nu}_T(x)$, and the thermal entrance length x_e for four grids. . . 19
8 (left) x_e versus n_x for UD and QUICK–DC ; (right) relative error of x_e with respect to the finest grid. 20
9 Effect of mesh size on x_e and $\min_x Nu_T(x)$: variation with n_y (left) and with n_x (right). 21
10 Temperature field $T(x, y)$ for two Péclet numbers (UD, 200 × 21). 22
11 Temperature field $T(x, y)$ with Neumann wall boundary conditions ($q_{\mathrm{wall}} = 10\ \mathrm{W/m}^2$). 24
12 Local Nusselt number $Nu_q(x)$ for $(n_x, n_y) = (50, 5)$. 24
13 Local Nusselt number $Nu_q(x)$ for $(n_x, n_y) = (200, 21)$. 25
14 Local heating on the bottom wall ($x \in [2, 5]$ m). Top : temperature field $T(x, y)$. Bottom : mean temperature $T_{\mathrm{mean}}(x)$ with outlet target shown by a dashed green line. 27

1 Introduction

The purpose of this project is to develop and validate a finite volume method for solving a steady two–dimensional convection–diffusion problem inside a plane channel. The study focuses on the thermal field of a laminar, fully developed flow between two parallel plates, where temperature acts as a passive scalar, meaning it does not influence the velocity field. The main goal is to build a Matlab code which compute the temperature distribution $T(x, y)$, to test different numerical schemes, and to study how mesh refinement and boundary conditions affect the final results.

The domain is a rectangular channel, haveing coordinates going from $x = 0$ to $x = L$ and from $y = 0$ to $y = H$, with $L = 10$ m and $H = 1$ m. The flow is steady, incompressible and laminar, with a parabolic velocity profile, corresponding to a fully developed Poiseuille flow. The governing equation for the temperature field is the steady convection–diffusion equation,

$$\nabla \cdot (\rho \mathbf{u} T) = \nabla \cdot (\Gamma \nabla T),$$

where ρ is the fluid density, $\mathbf{u}$ is the velocity vector, and $\Gamma = k/c_p$ is the thermal diffusivity. The physical properties of the fluid are constant and given by $\rho = 1$ kg/m^3, $c_p = 10$ J/(kg $\cdot$ K), $k = 0.12$ W/(m $\cdot$ K), and $\nu = 10^{-2}$ m^2/s, which lead to a value of $\Gamma = 0.012$ kg/(m $\cdot$ s). The mean velocity is defined from the target Péclet number as $Pe = 2\rho H u_{\text{mean}}/\Gamma$, ensuring that the flow regime and the heat transfer are fully characterized by dimensionless quantities.

The finite volume method is applied on a uniform, structured mesh that covers the entire domain. Diffusion is discretized using the central differencing (CD) scheme, which is second–order accurate and conservative, while convection is first solved using the upwind differencing (UD) scheme for stability. Later, the QUICK scheme is tested to reduce numerical diffusion and improve accuracy. The system of linear equations is solved in Matlab, first using a direct solver and then with an iterative SOR method, in order to compare convergence behavior and computational efficiency.

Boundary conditions play a key role in the simulation. At the inlet of the channel, a fixed temperature $T = T_{\text{inlet}}$ is imposed, while at the outlet a zero–gradient condition $\partial T/\partial x = 0$ ensures a physically consistent outflow. The upper and lower walls can be either isothermal or adiabatic, depending on the specific question studied. These configurations allow the investigation of both Dirichlet and Neumann type boundary conditions within the same framework.

Beyond the numerical implementation, the project also aims to extract and interpret physically meaningful quantities. From the computed temperature field, the local and average Nusselt numbers are evaluated to characterize heat transfer along the channel. The thermal entrance length is determined to quantify the distance required for the temperature profile to become fully developed. Additional analyses include grid convergence studies, tests at higher Péclet numbers, and verification of the global energy balance. Finally, the case of a locally heated wall is examined to explore the system's response to spatially varying thermal fluxes.

Overall, this work provides an implementation and interpretation of the finite volume approach applied to a convection–diffusion problem. It combines theoretical understanding, numerical discretization and physical analysis, highlighting how different modeling choices—such as mesh density, boundary conditions and discretization schemes—affect both the stability and the physical accuracy of the computed solution.

2 Theoretical analysis and problem definition

2.1 Problem statement

We consider a steady, incompressible, laminar flow between two parallel plates separated by a distance H. The plates are placed horizontally at $y = 0$ and $y = H$, and the flow moves in the x direction only. The velocity field is $\mathbf{u} = (u_x(y), u_y(y))$. Because the flow is fully developed and two–dimensional, the velocity depends only on y, and $u_y = 0$. Gravity will be neglected.

2.2 Q1 — Expression of the velocity field

The flow satisfies the incompressible steady Navier–Stokes equations. For a Newtonian fluid with constant viscosity μ, these are :

$$\rho(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \mu \nabla^2 \mathbf{u}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{2}$$

Because the flow is steady and fully developed :

$$\frac{\partial u_x}{\partial x} = 0,$$

so for the continuity equation we can deduce $u_y = 0$. The pressure only changes in the x direction, which means that $p = p(x)$. We now consider the x–component of the momentum equation. Since $\partial u_x / \partial x = 0$ and $u_y = 0$, the convective terms vanish :

$$0 = -\frac{dp}{dx} + \mu \frac{d^2 u_x}{dy^2}.$$

This equation describes the balance between the pressure gradient and the viscous diffusion of momentum. Integrating twice with respect to y gives :

$$u_x(y) = \frac{1}{2\mu} \frac{dp}{dx} y^2 + C_1 y + C_2.$$

At the two stationary walls we apply the no-slip conditions $u_x(0) = 0$, and $u_x(H) = 0$. The first condition ($y = 0$) gives $C_2 = 0$. The second condition ($y = H$) gives $C_1 = -\frac{1}{2\mu}\frac{dp}{dx}H$. Substituting both constants back :

$$u_x(y) = \frac{1}{2\mu} \frac{dp}{dx} (y^2 - Hy).$$

In a channel flow, the pressure decreases along x, so $\frac{dp}{dx} < 0$, so, defining a positive pressure gradient $G = -\frac{dp}{dx} > 0$, the velocity becomes :

$$u_x(y) = \frac{G}{2\mu}(Hy - y^2). \tag{3}$$

This parabolic profile is the Poiseuille flow solution. It satisfies both the no-slip boundary conditions and a constant pressure gradient. In the next parts of the assessment, this analytical profile will be

used to define the convective velocity in the heat transfer equation.
The velocity is maximum at the centerline, where $\frac{du_x}{dy} = 0$, hence at $y = \frac{H}{2}$. So,

$$u_{max} = \frac{G}{2\mu}\left(H\frac{H}{2} - \frac{H^2}{4}\right) = \frac{GH^2}{8\mu}. \tag{4}$$

Substituting in the formula for $u_x(y)$, we find

$$u_x(y) = 4u_{max}\frac{y}{H}\left(1 - \frac{y}{H}\right). \tag{5}$$

The mean velocity across the channel height is defined as :

$$u_{mean} = \frac{1}{H}\int_0^H u_x(y)\,dy = \frac{GH^2}{12\mu}. \tag{6}$$

Integrating Equation (3), we find the following result :

$$u_x(y) = 6u_{mean}\frac{y}{H}\left(1 - \frac{y}{H}\right), \tag{7}$$

2.3 Q2 — Integral form over an inner control volume

We start from the steady heat equation for a passive scalar :

$$\nabla \cdot (\rho\,\mathbf{u}\,T) \;=\; \nabla \cdot \big(\Gamma\,\nabla T\big). \tag{8}$$

Consider a rectangular inner control volume (CV) $\Omega_{i,j}$ centered at (x_i, y_j), with width Δx and height Δy (uniform structured mesh). Let its boundary $\partial\Omega_{i,j}$ have four faces : west (W), east (E), south (S), north (N), with outward unit normals $\mathbf{n}_W = (-1, 0)$, $\mathbf{n}_E = (+1, 0)$, $\mathbf{n}_S = (0, -1)$, $\mathbf{n}_N = (0, +1)$. Face lengths are $|\mathrm{E}| = |\mathrm{W}| = \Delta y$ and $|\mathrm{N}| = |\mathrm{S}| = \Delta x$.
Integrating Equation (8) over $\Omega_{i,j}$ and applying the divergence theorem yields

$$\int_{\Omega_{i,j}} \nabla\cdot(\rho\mathbf{u}T)dA = \int_{\Omega_{i,j}} \nabla\cdot(\Gamma\nabla T)dA \Longleftrightarrow \oint_{\partial\Omega_{i,j}} \rho T\mathbf{u}\cdot\mathbf{n}dS = \oint_{\partial\Omega_{i,j}} \Gamma\nabla T\cdot\mathbf{n}dS.$$

We now write the boundary integrals as the sum of the four face contributions :

$$\begin{aligned}\rho\big[(u_xT)_E\,\Delta y - (u_xT)_W\,\Delta y + (u_yT)_N\,\Delta x - (u_yT)_S\,\Delta x\big]&\\ = \Gamma\big[(\partial_xT)_E\,\Delta y - (\partial_xT)_W\,\Delta y + (\partial_yT)_N\,\Delta x - (\partial_yT)_S\,\Delta x\big]&.\end{aligned} \tag{9}$$

Where the term on the left side of the equal sign is convective flux out of $\Omega_{i,j}$ and the term on the right side is diffusive flux out of $\Omega_{i,j}$. In our channel flow, the velocity field is $\mathbf{u} = (u_x(y),\,0)$, hence $u_y = 0$. Therefore, the normal convective fluxes across the horizontal faces vanish and the equation simplifies the conservative integral form over one inner CV :

$$\rho\Big[(u_xT)_E - (u_xT)_W\Big]\Delta y \;=\; \Gamma\Big[(\partial_xT)_E - (\partial_xT)_W\Big]\Delta y \;+\; \Gamma\Big[(\partial_yT)_N - (\partial_yT)_S\Big]\Delta x. \tag{10}$$

2.4 Q3 & Q4 — Discretization of diffusion and convection terms

The goal of this step is to transform the integral balance of Equation (10) into a discrete algebraic equation that relates the temperature in a control volume to the temperatures of its neighbors. The finite volume formulation must be conservative, bounded and transportive.

- **Conservativeness** : flux leaving one CV through a face must enter the adjacent CV through the same face, ensuring global energy conservation.
- **Boundedness** : internal values of T must remain between the boundary values (no overshoots or undershoots). This requires that all coefficients in the discrete equations be positive and that the matrix A be diagonally dominant.
- **Transportiveness** : the influence of neighboring cells must depend on the flow direction ; for pure convection only upstream nodes should contribute.

We now discretize separately the diffusive and convective terms following these principles.

2.4.1 Diffusive term — Central Differencing (CD)

Diffusion represents a symmetric process, therefore a CD scheme is used. According to the lecture, the diffusive flux across a face is

$$(\Gamma \nabla T) \cdot \mathbf{n}_f \;\approx\; \Gamma \, \frac{T_{nb} - T_P}{\delta_f},$$

where T_P and T_{nb} are the temperatures at the center of the current and neighboring CVs, and δ_f is the distance between them. For a uniform rectangular mesh,

$$D_e = \Gamma \, \frac{\Delta y}{\Delta x}, \quad D_w = \Gamma \, \frac{\Delta y}{\Delta x}, \quad D_n = \Gamma \, \frac{\Delta x}{\Delta y}, \quad D_s = \Gamma \, \frac{\Delta x}{\Delta y}.$$

The CD scheme is second–order accurate and naturally conservative, because the flux through a common face is computed once and shared by both CVs.

2.4.2 Convective term — Upwind Differencing (UD)

Convection transports information along the flow direction and therefore requires an asymmetric discretization. The UD scheme evaluates the variable at each face using the value from the upstream node :

$$T_e = \begin{cases} T_P, & u_e > 0, \\ T_E, & u_e < 0, \end{cases} \qquad T_w = \begin{cases} T_W, & u_w > 0, \\ T_P, & u_w < 0. \end{cases}$$

This guarantees positive coefficients even if, as will be discussed later, the Péclet number is higher than 2 (boundedness) and respect of the flow direction (transportiveness), although it is only first–order accurate and introduces some numerical diffusion when convection dominates. The mass fluxes across the vertical faces are

$$F_e = \rho \, u_e \, \Delta y, \qquad F_w = \rho \, u_w \, \Delta y,$$

and the net convective flux in CV(i, j) is

$$F_e T_e - F_w T_w.$$

As noted in the lecture, UD is very stable and always satisfies the Scarborough criterion, which ensures convergence of iterative solvers such as SOR.

2.4.3 Discrete balance for one inner CV

Combining CD for diffusion and UD for convection leads to an algebraic equation of the form

$$a_P T_P = a_E T_E + a_W T_W + a_N T_N + a_S T_S, \tag{11}$$

where the coefficients are obtained from the flux balances :

$$\begin{aligned} a_E &= D_e + \max(-F_e, 0) = D_e, \\ a_W &= D_w + \max(F_w, 0) = D_w + F_w, \\ a_N &= D_n, \\ a_S &= D_s, \\ a_P &= a_E + a_W + a_N + a_S. \end{aligned}$$

This expression is conservative by construction and guarantees boundedness for all positive Γ, ρ, and u_x, and since $u_x > 0$ (from flow from West to East), the upstream cell is W, thus $F_e > 0$ and $F_w > 0$. Consequently, in the UD scheme $a_E = D_e$ and $a_W = D_w + F_w$. Results are summarised in Table 1.

TABLE 1 – Discretization coefficients for the diffusion (CD) and convection (UD) schemes.

Scheme	$\mathbf{a_W}$	$\mathbf{a_E}$	$\mathbf{a_S}$	$\mathbf{a_N}$	$\mathbf{a_P}$
CD (diffusion only)	D_w	D_e	D_s	D_n	$a_W + a_E + a_S + a_N$
CD (convection + diffusion)	$D_w + \frac{1}{2}F_w$	$D_e - \frac{1}{2}F_e$	D_s	D_n	$a_W + a_E + a_S + a_N$
UD (convection + diffusion)	$D_w + F_w$	D_e	D_s	D_n	$a_W + a_E + a_S + a_N$

$F_e = \rho\, u_e\, \Delta y,\ F_w = \rho\, u_w\, \Delta y,\ D_W = D_e = \Gamma\, \frac{\Delta y}{\Delta x},\ D_S = D_n = \Gamma\, \frac{\Delta x}{\Delta y}.$

This relation represents the discrete energy conservation for one control volume. Each coefficient accounts for both diffusion (through D_f) and convection (through F_f) across the corresponding face f. In matrix form, the equations for all control volumes can be written as $\mathbf{A}\,\mathbf{T} = \mathbf{b}$, where $\mathbf{A}$ is the global coefficient matrix, $\mathbf{T}$ the vector of unknown temperatures, and $\mathbf{b}$ the source term vector.

2.4.4 Dimensional consistency check

In the present formulation, the governing equation has been divided by the heat capacity c_p, so the thermal diffusivity is defined as $\Gamma = k/c_p$ and treats T as a scalar quantity proportional to the specific enthalpy per unit mass. To recover the correct physical heat fluxes, proper dimensional factors must be reintroduced.

— The diffusive flux across a face is $q_{\text{diff}} = \Gamma\, \frac{\partial T}{\partial n} A_f$, where $\Gamma = k/c_p$ has units [kg/(m·s)], $\partial T/\partial n$ has units [K/m], and the face area A_f has units [m^2]. Therefore, q_{diff} has units [kg·K/s]. Multiplying by c_p [J/(kg·K)] restores the physical units of heat flux, [J/s] = [W].

— The convective flux is $q_{\text{conv}} = \rho\, u_f\, T_f\, A_f$, where ρ [kg/m^3], u_f [m/s], and A_f [m^2]. Hence, $\rho u_f A_f$ has units [kg/s]. Multiplying by $c_p T$ [J/(kg·K)·K] gives again [J/s] = [W].

Both diffusive and convective contributions represent energy transfer rates with units of power [W]. Dimensional homogeneity is preserved once the appropriate scaling by c_p is applied when interpreting the results in terms of real heat fluxes. This confirms that the discretized equation remains dimensionally consistent and physically meaningful.

2.5 Q5 — Indexing convention for the structured mesh

Since the computational grid is structured, it is convenient to assign two integer indices (i, j) to each CV :

— i : row index, increasing from bottom to top along the y–direction ;
— j : column index, increasing from inlet to outlet along the x–direction.

The adopted convention is illustrated in Figure 1, which corresponds to the diagram provided in the course slides.

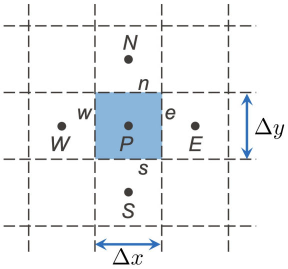


FIGURE 1 – Structured mesh indexing and neighbor notation used in the finite–volume formulation.

With this convention the inlet is at $j = 1$ and the outlet at $j = n_x$. The bottom is at $i = 1$ and the top at $i = n_y$. For an inner control volume (i, j), the neighboring cells are

$$(i, j-1) = W, \qquad (i, j+1) = E, \qquad (i-1, j) = S, \qquad (i+1, j) = N.$$

In our Matlab implementation, each control volume is associated with a unique linear index n, defined as $n = (j-1)\, n_y + i$, which increases first along y (rows) and then along x (columns). This convention allows mapping the 2D grid onto a 1D vector of unknowns.
The indices of the four neighboring cells are therefore :

$$n_W = n - n_y, \qquad n_E = n + n_y, \qquad n_S = n - 1, \qquad n_N = n + 1.$$

This mapping ensures that each temperature value T_n corresponds uniquely to one control volume, simplifying the assembly of the global sparse matrix $\mathbf{A}$ and the vector $\mathbf{b}$ in the finite volume formulation.

2.6 Q6 — Numbering of degrees of freedom (DOF)

The numbering convention introduced in Q5 allows each CV to be identified by its indices (i, j), and to be associated with a single algebraic unknown T_n. This linear numbering is necessary to convert the two–dimensional temperature field $T(x, y)$ into a one–dimensional vector $\mathbf{T}$ used in the matrix equation $\mathbf{AT} = \mathbf{b}$.

As defined previously, the index i increases from bottom to top along the y–direction and j from inlet to outlet along the x–direction. The corresponding linear index n is therefore $n = (j-1)\, n_y + i$.

The indices of boundary CVs and neighbouring cells are resumed in the following table :

Table 2 – Indices of boundary and neighboring control volumes.

Location / Neighbor	**Index relation**	**Description**
Inlet boundary	$j = 1 \Rightarrow n = i$	First column of CVs (flow entry)
Outlet boundary	$j = n_x \Rightarrow n = (n_x - 1)n_y + i$	Last column of CVs (flow exit)
Bottom wall	$i = 1 \Rightarrow n = (j-1)n_y + 1$	First row of CVs (lower wall)
Top wall	$i = n_y \Rightarrow n = j\, n_y$	Last row of CVs (upper wall)
West neighbor (W)	$n_W = n - n_y$	CV located at $(i, j-1)$
East neighbor (E)	$n_E = n + n_y$	CV located at $(i, j+1)$
South neighbor (S)	$n_S = n - 1$	CV located at $(i-1, j)$
North neighbor (N)	$n_N = n + 1$	CV located at $(i+1, j)$

Part of the used code is reported in Annex 1

3 Implementation of boundary conditions. Questions 7 → 9

3.1 Q7 — Boundary condition at the inlet (Dirichlet)

At the inlet of the channel ($j = 1$), the temperature of the incoming fluid is imposed and kept constant. This corresponds to a Dirichlet boundary condition of the form $T(x = 0, y) = T_{\text{inlet}}$.
For every control volume located on the inlet boundary, the temperature is prescribed and does not depend on neighboring values. In the algebraic finite–volume formulation, this condition is enforced by replacing the general balance equation described in Equation (11) with the simple constraint

$$T_P = T_{\text{inlet}}.$$

This is equivalent to setting $a_P = 1$, $b = T_{\text{inlet}}$ and all the others factor equal to zero. By doing so, the Dirichlet condition is applied directly in the matrix system $\mathbf{AT} = \mathbf{b}$, ensuring that the inlet cells remain fixed at the prescribed temperature.
Physically, this boundary condition represents the entrance of fluid with a uniform and known temperature T_{inlet}. It defines the upstream reference for all subsequent temperature variations due to convection and diffusion along the channel. The use of a Dirichlet condition ensures a stable and well–posed numerical problem at the inlet.

3.2 Q8 — Outlet boundary condition (Neumann)

At the outlet of the channel ($j = n_x$), a zero–gradient (Neumann) boundary condition is imposed, meaning that temperature respect

$$\frac{\partial T}{\partial x} = 0$$

at $x = L$. Physically, this means that the flow leaving the domain is thermally fully developed, so the temperature does not vary in the streamwise direction. In the finite–volume formulation, this condition is applied by requiring that the temperature in the outlet control volume equals that of its immediate upstream neighbor. On a uniform mesh, this follows from a backward–difference approximation of the derivative :

$$\frac{\partial T}{\partial x} \approx \frac{T_{(j=n_x)} - T_{(j=n_x-1)}}{\Delta x} = 0.$$

In algebraic form, for each outlet control volume, this becomes

$$1 \cdot T_P - 1 \cdot T_W = 0,$$

which can be directly written in the global matrix as

$$A_{n,n} = 1, \qquad A_{n,n_W} = -1, \qquad b_n = 0.$$

with $n = (n_x - 1) \cdot ny + i$, and i going form 1 to n_y. This treatment ensures numerical stability and physical consistency, since it allows the temperature field to exit the domain smoothly without introducing artificial gradients or reflections.

3.3 Q9 — Isothermal wall boundary condition (Dirichlet)

We impose isothermal walls at the top and bottom of the channel :

$$T(x, 0) = T_{\text{wall}}, \qquad T(x, H) = T_{\text{wall}}.$$

In the finite volume formulation, this is implemented by replacing the balance equation for each wall control volume with the constraint $T_P = T_{\text{wall}}$. In matrix form, for every wall node n,

$$A_{n,n} = 1, \qquad A_{n,m} = 0 \ \ (m \neq n), \qquad b_n = T_{\text{wall}}.$$

This enforces the temperature value, ensuring stability and boundedness (the diagonal remains dominant and interior values are bounded by the boundary).

Identification of wall nodes. Unknowns are numbered column by column according to

$$n = (j-1)\, n_y + i, \qquad i = 1, \dots, n_y, \quad j = 1, \dots, n_x,$$

with $i = 1$ (bottom wall $y = 0$) and $i = n_y$ (top wall $y = H$). The nodes subject to the Dirichlet condition are therefore

$$\{\, n = (j-1)\, n_y + 1 \,\}_{j=1}^{n_x} \quad \text{and} \quad \{\, n = j\, n_y \,\}_{j=1}^{n_x}.$$

In practice, the n-th row of the linear system $A\,T = b$ is overwritten to directly enforce $T_n = T_{\text{wall}}$.

Remarks. At corner nodes where multiple boundary conditions meet (inlet/outlet vs wall), the priority is determined by the implementation order (in this case, inlet is applied before walls). In our case at the corner the boundary condition expressed in Capter 3.1 is applied.

Part of the used code is reported in Annex 2.

4 Physical parameters and preliminary verification. Questions 10 → 12

4.1 Q10 — Physical parameters and Péclet number calculation

Before assembling the linear system, we define all physical and geometrical parameters of the problem, that are resumend in Table (3) :

Table 3 – Physical and geometrical parameters used in the simulation.

Parameter	**Symbol**	**Value**
Density	ρ	1 kg/m^3
Specific heat capacity	c_p	10 J/(kg $\cdot$ K)
Thermal conductivity	k	0.12 W/(m $\cdot$ K)
Kinematic viscosity	ν	1×10^{-2} m^2/s
Channel length	L	10 m
Channel height	H	1 m

The thermal diffusivity is computed from

$$\Gamma = \frac{k}{c_p} = 0.012 \frac{kg}{ms}.$$

A target Péclet number is chosen, $Pe_{\text{target}} = 16.5$, and is related to the mean velocity u_{mean} by

$$Pe = \frac{2\rho H u_{mean}}{\Gamma}.$$

From this, we can compute the required mean velocity :

$$u_{mean} = \frac{Pe\Gamma}{2\rho H} \approx 9.9 \cdot 10^{-2} \frac{m}{s}.$$

The Reynolds number is evaluated to check the flow regime :

$$Re = \frac{2Hu_{mean}}{\nu} \approx 20 \ll 2300.$$

which confirms a laminar flow.

Remarks. (i) Choosing Pe indirectly controls the convection-to-diffusion ratio in the numerical problem. (ii) Keeping Re low ensures the velocity profile remains laminar and fully developed. (iii) These physical parameters influence the stability and accuracy of the numerical solution through the relative magnitude of diffusive and convective fluxes.

4.2 Q11 — First solution on the base mesh and visuals

We now assemble the global linear system $\mathbf{AT} = \mathbf{b}$ using the discretization described above : CD for diffusion, UD for convection, Dirichlet conditions at the inlet and on both walls, and a zero–gradient (Neumann) condition at the outlet. The physical parameters are those listed in Table 3, and the mean velocity u_{mean} is chosen to correspond to the target Péclet number $Pe = 16.5$.

For this first test we use a relatively coarse mesh with $(n_x, n_y) = (50, 5)$. The unknowns are ordered column by column according to the indexing convention described earlier, and the system is solved directly in Matlab using the built–in backslash operator,

$$\mathbf{T} = \mathbf{A} \backslash \mathbf{b}.$$

This gives us a 2D temperature field $T(x, y)$ over the whole channel.

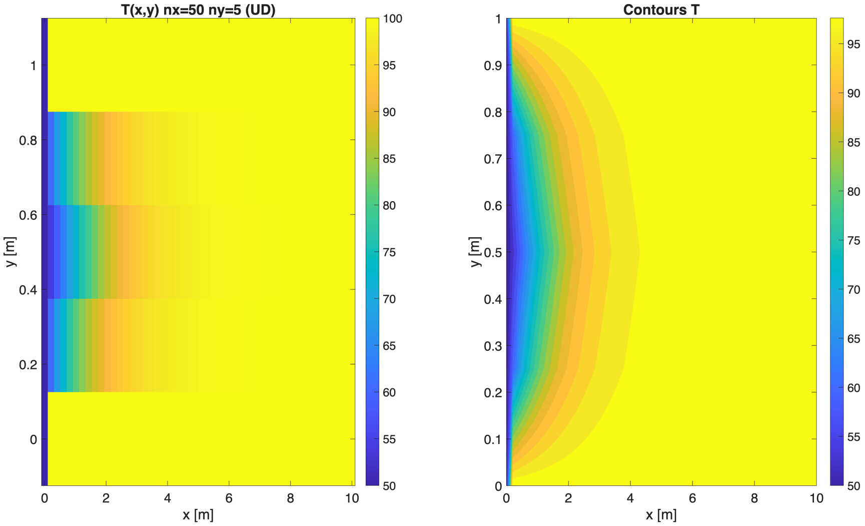


FIGURE 2 – Temperature field $T(x, y)$ (color map).

The result is shown in Figure 2, where we plot the temperature as a colormap over the domain. The inlet on the left is at the prescribed temperature $T_{\text{inlet}} = 50°\text{C}$, while the upper and lower walls are fixed at $T_{\text{wall}} = 100°\text{C}$. We see that near the inlet the fluid is still "cold" in the core of the channel, and heat diffuses from the hot walls into the flow. Further downstream (larger x), the interior temperature becomes more uniform and approaches the wall temperature, as expected for a long heated channel.

The right panel of Figure 2 shows the same solution using temperature contours. The isotherms are initially strongly curved near the inlet, where the thermal boundary layers grow from the two walls. Moving downstream, the contours become flatter and more horizontal, which indicates that the profile is becoming almost fully developed in y.

Overall, this first solution looks physically reasonable : heat is transferred from the walls to the fluid, the profile evolves smoothly in the streamwise direction, and there are no obvious oscillations or nonphysical values. This is consistent with the fact that we used UD for the convective term, which is very stable (although slightly diffusive).

4.3 Q12 — A posteriori verification of boundary conditions

After solving the linear system on the base mesh $(n_x, n_y) = (50, 5)$, we verify that all boundary conditions are correctly enforced in the numerical solution. This verification is done by extracting the temperature values along each boundary and comparing them with the imposed conditions.

At the inlet ($x = 0$), the temperature is fixed to $T_{\text{inlet}} = 50°\text{C}$. The code reports :

$$\text{Inlet min/max } T = [50.000,\ 50.000]\ °\text{C},$$

confirming that the Dirichlet condition is applied exactly.

At the top and bottom walls, the target value is $T_{\text{wall}} = 100°\text{C}$. The printed values are :

$$\text{Bottom wall min/max } T = [50.000,\ 100.000]\ °\text{C}, \qquad \text{Top wall min/max } T = [50.000,\ 100.000]\ °\text{C}.$$

The minimum of 50°C occurs at the inlet corners, because in our implementation the corner control volumes are treated as inlet cells (Dirichlet $T = T_{\text{inlet}}$) rather than wall cells. Apart from these corner nodes, all wall control volumes are exactly at $T_{\text{wall}} \approx 100°\text{C}$, confirming that the isothermal (Dirichlet) wall condition is satisfied.

At the outlet ($x = L$), a Neumann condition is imposed to model a fully developed thermal field. Numerically, the last column of control volumes satisfies $T(L, y) = T(L - \Delta x, y)$. The code reports :

$$\max_y \big| T(L, y) - T(L - \Delta x, y) \big| = 0.000 \times 10^0\ °\text{C},$$

which indicates that the zero-gradient condition is perfectly respected within numerical precision.

Boundary	**Check**	**Result from code**
Inlet	$T(0, y) = T_{\text{inlet}}$	min/max = [50.000, 50.000]
Walls	$T = T_{\text{wall}}$	min/max = [50.000, 100.000] (corners at inlet)
Outlet	$\partial T/\partial x = 0$	$\max \lvert T(L) - T(L - \Delta x)\rvert = 0.000$

TABLE 4 – Q12 — A posteriori verification of boundary conditions.

In summary, all boundary conditions are correctly implemented : the inlet is fixed at $T_{\text{inlet}} = 50°\text{C}$, the top and bottom walls are isothermal at $T_{\text{wall}} = 100°\text{C}$ except for the inlet corners (by design), and the outlet satisfies $\partial T/\partial x = 0$.

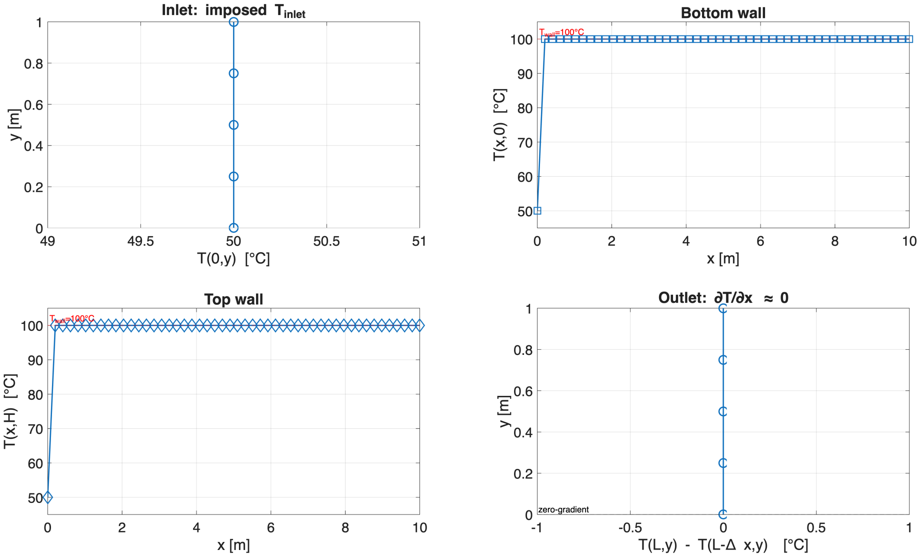


FIGURE 3 – Top left : inlet profile $T(0, y) = 50°\text{C}$. Top right / bottom left : bottom and top wall temperatures, showing the imposed isothermal wall at $T_{\text{wall}} = 100°\text{C}$ except at the inlet corners (treated as inlet). Bottom right : outlet check, confirming $\partial T/\partial x \approx 0$ (zero difference between the last two streamwise stations).

5 Analysis of results and derived quantities. Questions 13 → 15

5.1 Q13 — Outlet profile, centerline and mean temperatures, and entrance length x_e

From the computed field $T(x, y)$ showed in Figure 2, three quantities are extracted : (i) the outlet profile $T_o(y) = T(L, y)$, (ii) the centerline temperature $T_c(x) = T(x, H/2)$, and (iii) the velocity-weighted mean temperature

$$T_{\text{mean}}(x) = \frac{\int_0^H u_x(y)T(x, y)\, dy}{\int_0^H u_x(y)\, dy}. \tag{12}$$

The outlet profile $T_o(y)$ is nearly uniform and very close to $T_{\text{wall}} = 100°\text{C}$, showing that the fluid is almost thermally equilibrated at the exit.

Both $T_c(x)$ and $T_{\text{mean}}(x)$ rise monotonically from the inlet temperature $T_{\text{inlet}} = 50°\text{C}$ toward T_{wall} as heat diffuses from the walls into the core. The mean temperature is slightly higher than the centerline value, as expected in a laminar heating process.

The thermal entrance length x_e is defined where the centerline temperature reaches 90% of the wall temperature :

$$T_c(x_e) = T_{\text{inlet}} + 0.9(T_{\text{wall}} - T_{\text{inlet}}). \tag{13}$$

From the numerical data we obtain $x_e \approx 3.32$ m, meaning that the flow becomes thermally developed after about three channel heights. Figure 4 summarizes the three plots and the definition of x_e.

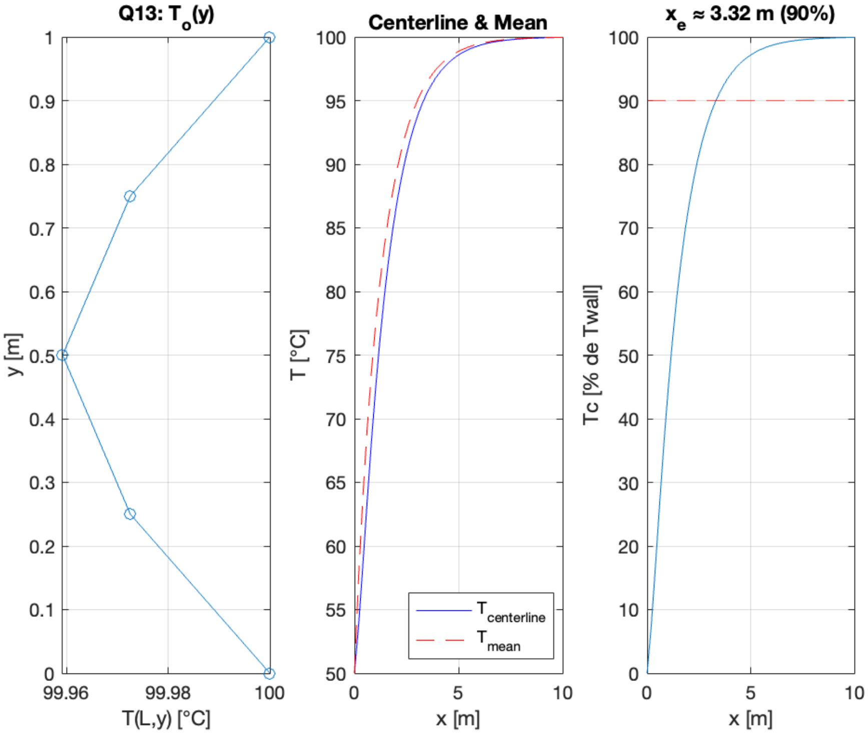


FIGURE 4 – (left) Outlet temperature profile $T_o(y)$; (middle) evolution of $T_c(x)$ and $T_{\text{mean}}(x)$; (right) normalized centerline temperature showing $x_e \approx 3.32$ m where $T_c = 0.9\,T_{\text{wall}}$.

5.2 Q14 — SOR iterative solver and convergence behavior

In this step we solve the linear system $\mathbf{AT} = \mathbf{b}$ using an iterative Successive Over–Relaxation (SOR) method. The update for each unknown T_i follows the standard relaxed Gauss–Seidel form used in the lecture :

$$T_i^{(k+1)} = (1-\omega)\,T_i^{(k)} + \frac{\omega}{A_{ii}}\Big(b_i - \sum_{j<i} A_{ij}T_j^{(k+1)} - \sum_{j>i} A_{ij}T_j^{(k)}\Big),$$

where ω is the relaxation factor. For $\omega = 1.0$ the method reduces to classical Gauss–Seidel, while $\omega > 1$ corresponds to over–relaxation.

To monitor convergence we record, at each iteration k, the relative error

$$\varepsilon^{(k)} = \frac{\|\mathbf{T}^{(k)} - \mathbf{T}^{(k-1)}\|_2}{\|\mathbf{T}^{(k-1)}\|_2}$$

and the normalized residual

$$r^{(k)} = \frac{\|\mathbf{A}\mathbf{T}^{(k)} - \mathbf{b}\|_2}{\|\,\mathrm{diag}(\mathbf{A})\,\mathbf{T}^{(k)}\|_2}.$$

The iterations are stopped when both fall below 10^{-5}. The solver is started from a uniform initial guess $T_0 = 50°\mathrm{C}$ everywhere in the domain.

Figure 5 shows the evolution of $\varepsilon^{(k)}$ (dashed curves) and $r^{(k)}$ (solid curves) for $\omega = 1.0$ and $\omega = 1.5$. For $\omega = 1.5$ we observe an initial overshoot of the error : the over–relaxation applies stronger corrections at each step, so the field first moves away from the initial guess before settling. After this transient, both the residual and the error decay and it converges in 32 iterations.

An interesting point is that, in our case, the total number of iterations needed to reach the tolerance is of the same order for $\omega = 1.0$ (36 iterations) and $\omega = 1.5$. A plausible explanation is that our initial guess ($T = 50°\mathrm{C}$ everywhere) is already quite close to the final temperature field in most of the channel, so even the $\omega = 1.0$ run starts from a state that is not very far from the converged solution.

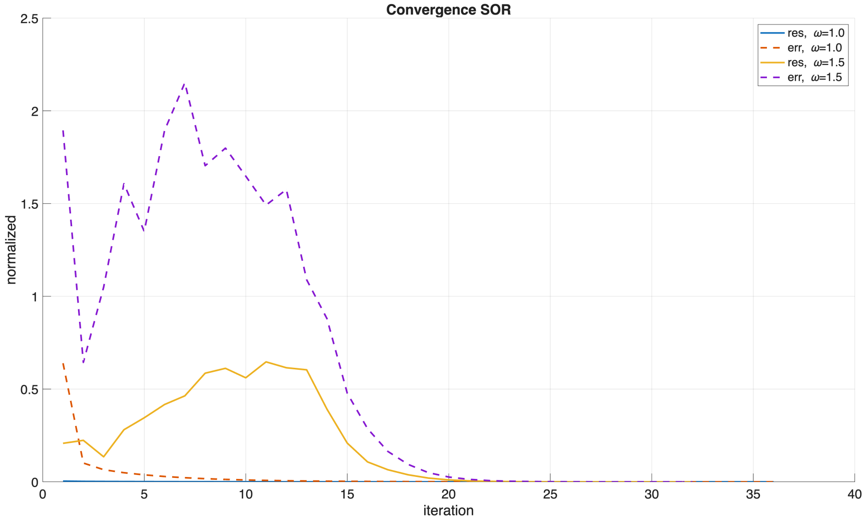


FIGURE 5 – Convergence history of the SOR solver. Solid lines : normalized residual $r^{(k)}$. Dashed lines : relative error $\varepsilon^{(k)}$. Blue / orange : $\omega = 1.0$ (Gauss–Seidel limit). Yellow / purple : $\omega = 1.5$ (over–relaxation).

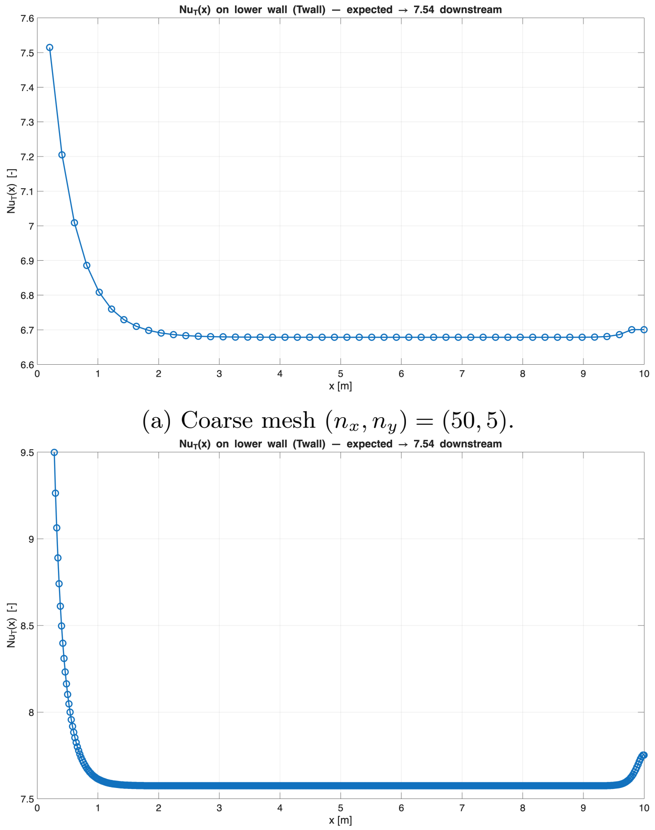
5.3 Q15 — Local Nusselt number along the lower wall

We evaluate the local heat transfer along the heated lower wall by defining the local Nusselt number as

$$\mathrm{Nu}_T(x) = \frac{2H\,q(x)}{k\,[T_{\mathrm{wall}} - T_{\mathrm{mean}}(x)]}, \qquad q(x) = k\,\frac{\partial T}{\partial n}, \tag{14}$$

where $q(x)$ is the wall heat flux, $T_{\mathrm{mean}}(x)$ is the velocity-weighted mean temperature at station x. For a thermally fully developed laminar flow between two parallel isothermal plates, theory predicts a constant value $\mathrm{Nu}_T = 7.54$ downstream.

Figure 6 shows the computed $\mathrm{Nu}_T(x)$ on the lower wall for two grids : a coarse mesh $(n_x, n_y) = (50, 5)$ and a refined mesh $(500, 51)$.



(a) Coarse mesh $(n_x, n_y) = (50, 5)$.

(b) Refined mesh $(n_x, n_y) = (500, 51)$.

FIGURE 6 – Local Nusselt number $\mathrm{Nu}_T(x)$ along the lower wall.

In both cases the qualitative behavior is physically correct : $\mathrm{Nu}_T(x)$ is very large near the inlet, where the thermal boundary layer is still extremely thin and the wall-normal temperature gradient is very steep. Then $\mathrm{Nu}_T(x)$ decays rapidly and eventually tends to an almost constant value far downstream. This plateau region indicates that the temperature field has become thermally fully developed.

However, the value of the downstream plateau depends strongly on grid resolution :

— On the coarse mesh $(50, 5)$, the plateau settles around $\mathrm{Nu}_T \approx 6.7$, which is below the theoretical 7.54.

— On the refined mesh $(500, 51)$, the plateau is $\mathrm{Nu}_T \approx 7.56$, essentially matching the analytical value 7.54.

The reason for this discrepancy is mainly the wall-normal resolution n_y. The Nusselt number depends directly on the wall heat flux $q(x)$, which in turn depends on the wall gradient

$$\left.\frac{\partial T}{\partial y}\right|_{\text{wall}} \approx \frac{T(y_2, x) - T(y_1, x)}{\Delta y},$$

where y_1 is the wall node and y_2 is the first fluid node above the wall. In the coarse mesh ($n_y = 5$), there are only five cells across the full channel height. This means that the first off-wall temperature point is far from the wall in physical terms : the near-wall thermal boundary layer is not resolved, and the jump in T between the wall and the fluid is large. As a consequence, the numerical derivative $(T(y_2) - T(y_1))/\Delta y$ underestimates the true wall gradient. If the wall gradient is underestimated, the computed heat flux $q(x)$ is too low, and therefore $\mathrm{Nu}_T(x)$ is too low.

When we refine the mesh to $(500, 51)$, we add many more points in the wall-normal direction. Now the first interior node is much closer to the wall, Δy is much smaller, and the temperature jump between the wall and the fluid is properly captured. The numerical estimate of $\partial T/\partial y$ at the wall becomes much more accurate, so the wall heat flux $q(x)$ increases to its correct value, and the resulting Nusselt number converges to $\mathrm{Nu}_T \approx 7.56$ downstream, in excellent agreement with the theoretical $\mathrm{Nu}_T = 7.54$.

In addition, the finer mesh in the streamwise direction ($n_x = 500$ instead of 50) helps us clearly identify a long fully developed region before the outlet. With the coarse mesh, the outlet boundary condition ($\partial T/\partial x = 0$) can still influence the last part of the channel, slightly contaminating the "asymptotic" Nusselt number. With the refined mesh, the thermally developed plateau is cleaner and extends over a larger portion of the domain.

6 Study of convergence and numerical accuracy. Questions 16 → 19

6.1 Q16 — Mesh refinement and grid independence study

To verify the spatial accuracy of the finite volume solver and assess grid independence, the simulation was repeated on four structured meshes of increasing resolution : $(n_x, n_y) = (50, 5)$, $(100, 11)$, $(200, 21)$ and $(400, 41)$. The same physical parameters and boundary conditions were kept for all cases (Dirichlet at inlet and walls, Neumann at the outlet, UD for convection and CD for diffusion).

The following quantities were extracted from each simulation :

- the outlet temperature profile $T_o(y) = T(x = L, y)$,
- the centerline and mean temperature evolutions $T_c(x)$ and $T_{\text{mean}}(x)$,
- the local Nusselt number $\mathrm{Nu}_T(x)$ on the lower wall,
- and the thermal entrance length x_e defined by

$$\frac{T_c(x_e) - T_{\text{inlet}}}{T_{\text{wall}} - T_{\text{inlet}}} = 0.9.$$

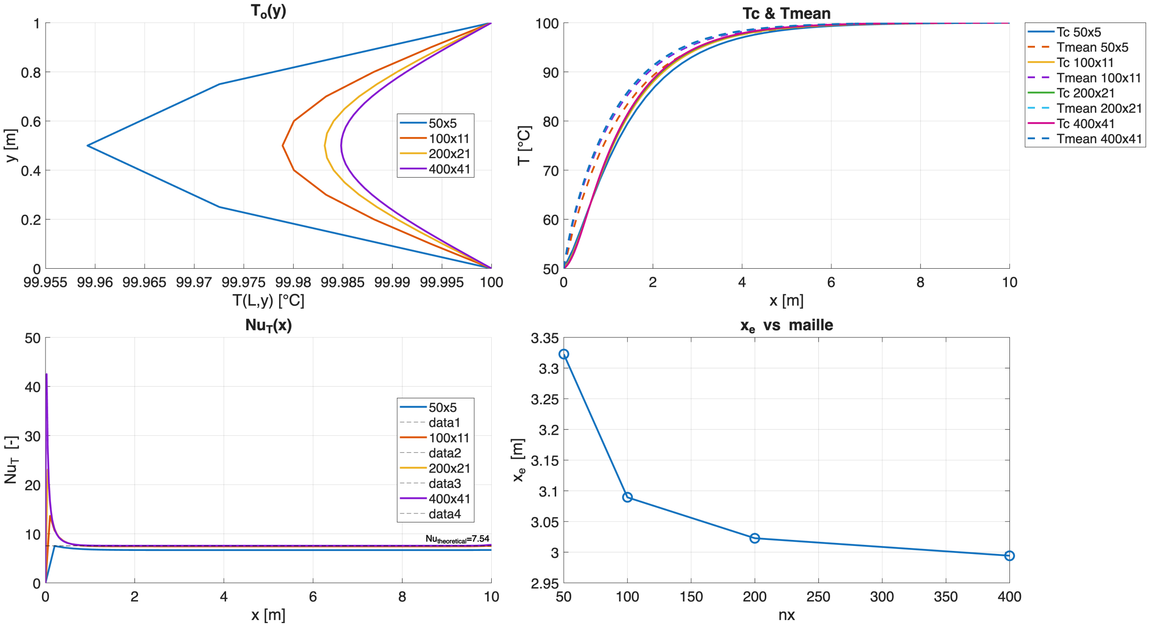


FIGURE 7 – Comparison of the outlet profile $T_o(y)$, the axial evolution of $T_c(x)$ and $T_{\text{mean}}(x)$, the local Nusselt number $\text{Nu}_T(x)$, and the thermal entrance length x_e for four grids.

Figure 7 summarizes the results. In the top-left plot, the outlet temperature profiles $T_o(y)$ progressively collapse as the grid is refined : the difference between the 200×21 and 400×41 meshes is negligible, indicating convergence of the crosswise temperature field. The same trend is seen in the top-right plot, where both $T_c(x)$ and $T_{\text{mean}}(x)$ overlap for all fine grids, confirming consistency of the solution.

The bottom-left panel shows the local Nusselt number $\text{Nu}_T(x)$ along the lower wall. On the coarsest grid (50×5), Nu_T is underpredicted due to the poor resolution of the thermal boundary layer and the resulting underestimation of the wall temperature gradient. As the grid is refined, $\text{Nu}_T(x)$ approaches the analytical asymptotic value $\text{Nu}_T = 7.54$, and the downstream plateau becomes perfectly constant for meshes finer than 200×21. This confirms that the heat flux and near-wall gradients are correctly resolved.

Finally, the bottom-right plot shows the thermal entrance length x_e as a function of n_x. The value decreases from about 3.3 m on the coarse grid to about 3.0 m on the finest grid, then remains practically unchanged, proving that the estimated entrance length is grid-independent.

In summary, all monitored quantities exhibit convergence with mesh refinement. A grid of 200×21 cells is sufficient to achieve accurate and stable results, since further refinement does not alter the solution significantly.

6.2 Q17 — Convection schemes comparison (UD vs QUICK)

We compare the influence of the convection scheme on the thermal entrance length x_e for meshes $(n_x, n_y) \in \{(50, 5), (100, 11), (200, 21), (400, 41)\}$. The flow is laminar and fully developed, diffusion is treated with a CD scheme, and the boundary conditions are identical to previous cases : Dirichlet at the inlet, isothermal walls, and a zero–gradient (Neumann) condition at the outlet. The QUICK

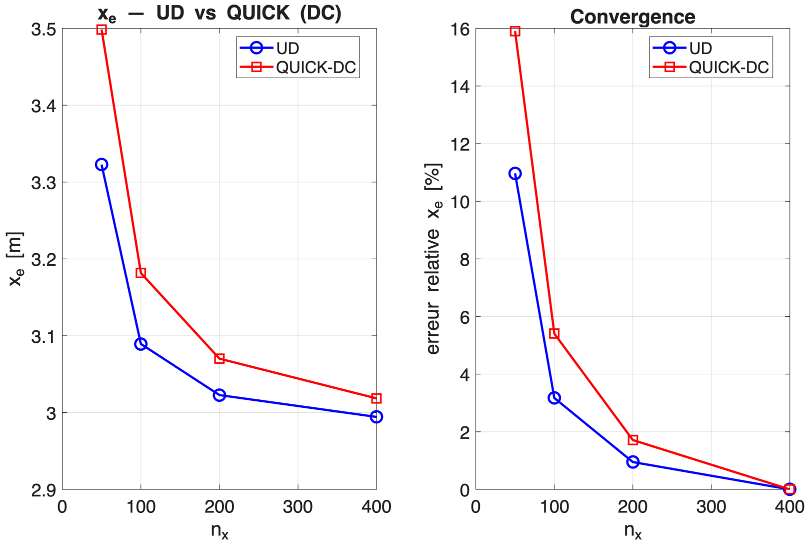


FIGURE 8 – (left) x_e versus n_x for UD and QUICK–DC ; (right) relative error of x_e with respect to the finest grid.

scheme approximates the face values of T using a quadratic interpolation based on two upstream nodes and one downstream node. Since $F_e > 0$ and $F_w > 0$:

$$T_e = \tfrac{3}{8}T_E + \tfrac{6}{8}T_P - \tfrac{1}{8}T_W, \qquad T_w = \tfrac{3}{8}T_P + \tfrac{6}{8}T_W - \tfrac{1}{8}T_{WW}.$$

This third–order scheme significantly reduces numerical diffusion, but it is not bounded — on coarse meshes, overshoots or undershoots may occur since coefficents may be negative.

The QUICK scheme can be implemented in a bounded way by rewriting the discretized equation as :

$$a_P T_P = a_W T_W + a_E T_E + b + \left[a_W^\star T_W^\star + a_E^\star T_E^\star - a_P^\star T_P^\star\right],$$

where the coefficients with the superscript ($\star$) and the corresponding temperatures $T^\star$ are evaluated from the previous iteration or from the upwind solution. The correction term is therefore added to the right–hand side of the upwind system, while the main matrix remains that of the UD scheme. This ensures diagonal dominance and stability while progressively incorporating the higher–order QUICK contribution.

The thermal entrance length is defined in Equation 13, and the relative variation with respect to the value obtained on the finest mesh is :

$$\varepsilon_{\text{rel}}(n_x) = \frac{|x_e(n_x) - x_e(400)|}{x_e(400)} \times 100\ [\%].$$

Figure 8 shows that, on coarse meshes, the bounded QUICK–DC formulation is less diffusive than the UD scheme, resulting in a slightly larger thermal entrance length x_e. This behavior is expected, since the third–order interpolation used in QUICK better preserves the sharp temperature gradients near the inlet, while the first–order UD scheme introduces additional numerical diffusion that smooths the profile. As the grid is refined, both schemes gradually converge toward the same value of x_e, demonstrating that the QUICK–DC formulation is consistent and that both discretizations tend to the same physical solution in the limit of vanishing grid spacing.

Conclusion. The QUICK scheme with bounded deferred correction combines the robustness of UD with the accuracy of a third–order interpolation. It improves the resolution of the thermal boundary layer, reduces numerical diffusion on coarse meshes, and converges to almost the same fully developed profile as the UD scheme on fine meshes.

6.3 Q18 — Effect of mesh size on x_e and $\min_x Nu_T(x)$

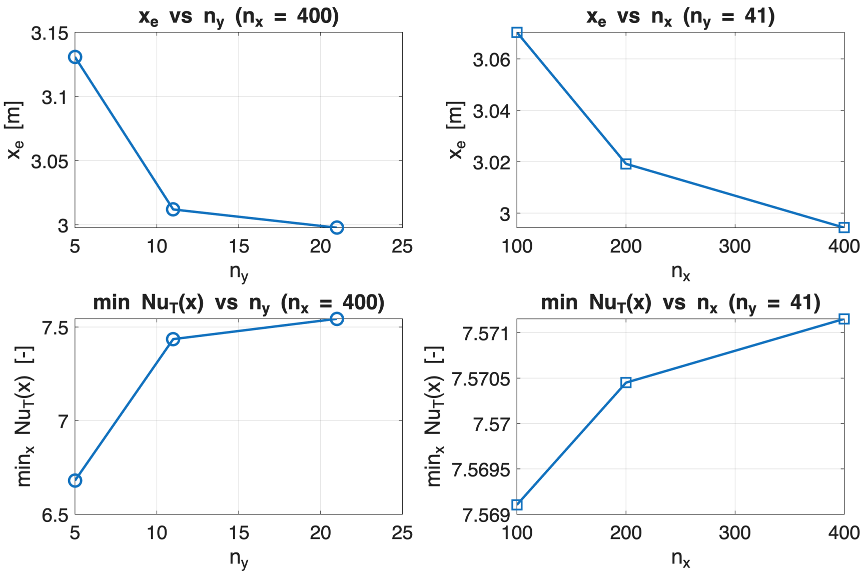


FIGURE 9 – Effect of mesh size on x_e and $\min_x Nu_T(x)$: variation with n_y (left) and with n_x (right).

In this question we studied how the mesh sizes affects the entrance length x_e and the local Nusselt number. Two mesh families were tested : (i) fixed $n_x = 400$ and $n_y = \{5, 11, 21\}$, and (ii) fixed $n_y = 41$ and $n_x = \{100, 200, 400\}$. For each case we computed the entrance length x_e and the minimum value of the local Nusselt number defined in Equation 14.

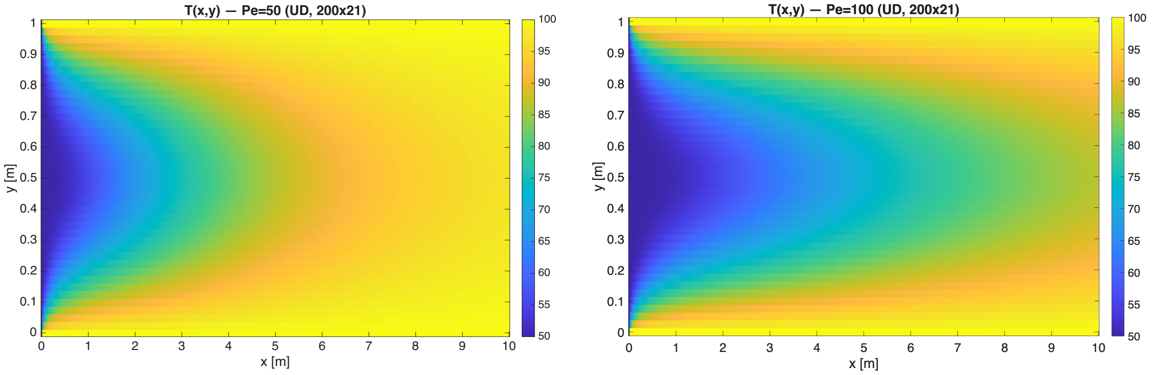
When refining in y, the entrance length decreases and the minimum Nu_T gets closer to 7.54. This makes sense because smaller Δy improves the estimation of the wall gradient $\partial T/\partial y$, which directly determines the heat flux and therefore Nu_T. When refining in x, instead, Δy stays the same, so the dominant error on the wall gradient does not change. The field becomes smoother along x, but the near–wall heat flux remains almost the same. Because of that, the minimum Nu_T may stay constant or even slightly move away from 7.54. The outlet Neumann condition $\partial T/\partial x = 0$ can also flatten the downstream profile and keep Nu_T a bit higher.

The values of x_e are slightly different from the previous questions because here we use UD (more diffusive) and anisotropic meshes. The larger numerical diffusion of UD leads to a thicker thermal layer and thus a longer entrance region. The definition of x_e also depends on the discrete centerline profile, so small changes in mesh or interpolation can shift the 90% location.

Conclusion. At the same total number of cells, it is generally better to refine in the y direction since it directly improves the resolution of the wall gradients and the Nusselt number. Refining in x mostly helps the smoothness of the streamwise field but has little impact on the wall heat–transfer accuracy.

6.4 Q19 — Effect of the Péclet number on $T(x, y)$

We assess the influence of the streamwise Péclet number by varying u_{mean} while keeping all other parameters and boundary conditions unchanged. Two cases are shown in Figure 10 : $Pe = 50$ and $Pe = 100$. The 200×21 grid was chosen as a good compromise between precision and cost, since previous tests carried out in Chapter 6.1 showed that beyond this level of refinement, precision improved only slightly.



(a) $Pe = 50$

(b) $Pe = 100$

FIGURE 10 – Temperature field $T(x, y)$ for two Péclet numbers (UD, 200×21).

As shown in the results, the thermal entrance length approximately doubles when the Péclet number increases from $Pe = 50$ to $Pe = 100$, which is consistent with the growing dominance of convection over diffusion. When Pe increases (here through a higher mean velocity u_{mean}), the thermal boundary layers near the walls become thinner, and heat penetrates more slowly toward the channel center. As a result, the centerline temperature remains lower further downstream at $Pe = 100$, and the overall entrance length x_e increases. This behavior reflects the shorter diffusion time available as convection becomes stronger. All temperature values remain bounded between T_{in} and T_{wall}, confirming the monotonicity of the numerical solution and the absence of spurious oscillations. Overall, the results are physically consistent and validate the correct implementation of the convection–diffusion solver.

Numerical challenge at high Péclet numbers. When the Péclet number becomes large, convection dominates over diffusion, and the temperature field develops very steep spatial gradients near the walls and at the inlet. Capturing these thin thermal layers requires a sufficiently fine mesh ; otherwise, the discrete convective fluxes cannot balance diffusion, leading to oscillations or excessive numerical diffusion. Low–order upwind schemes remain stable but become over diffusive, while schemes such as QUICK may lose boundedness if not carefully implemented. Therefore, high Pe convection–diffusion problems are numerically challenging because they demand both fine spatial resolution and stable, non–oscillatory high–order discretizations.

7 Advanced boundary conditions and energy balances. Questions 20 → 22

7.1 Q20 — Neumann wall boundary conditions and heat–flux formulation

In this configuration, the walls are no longer isothermal : a constant and uniform wall heat flux is imposed instead,

$$q_{\text{wall}} = k\frac{\partial T}{\partial n} = 10\ \frac{\text{W}}{\text{m}^2},$$

while the inlet and outlet remain Dirichlet and Neumann conditions, respectively. This setting corresponds to a physical case of uniform wall heating.

7.1.1 Temperature field

Figure 11 shows the temperature field obtained on the refined mesh $(n_x, n_y) = (200, 21)$. The temperature increases monotonically along the flow direction, as expected for a constant heat input. The contours are smooth and symmetric, with higher temperatures near the outlet and approximately linear gradients in the y direction. No non–physical oscillations are observed, meaning that the Neumann boundary condition is correctly implemented and that the flux is properly conserved at both walls.

7.1.2 Discrete wall equation

For the top and bottom wall control volumes, the temperature is not fixed but determined by the imposed normal heat flux. In finite–volume form, the Neumann boundary condition is discretized at the wall as

$$\pm k\,\frac{T_P - T_{nb}}{\Delta y} = q_{\text{wall}},$$

where the upper sign (+) corresponds to the bottom wall ($y = 0$) and the lower sign (-) to the top wall ($y = H$). Rearranging, the discrete algebraic equations for the boundary nodes become

$$\begin{aligned} \text{Top wall :} \quad & -\,k\,T_P + k\,T_N = q_{\text{wall}}\,\Delta y, \\ \text{Bottom wall :} \quad & +\,k\,T_P - k\,T_S = q_{\text{wall}}\,\Delta y. \end{aligned}$$

This formulation ensures that the imposed flux appears as a source term on the right–hand side, while maintaining the diagonal dominance of the coefficient matrix.

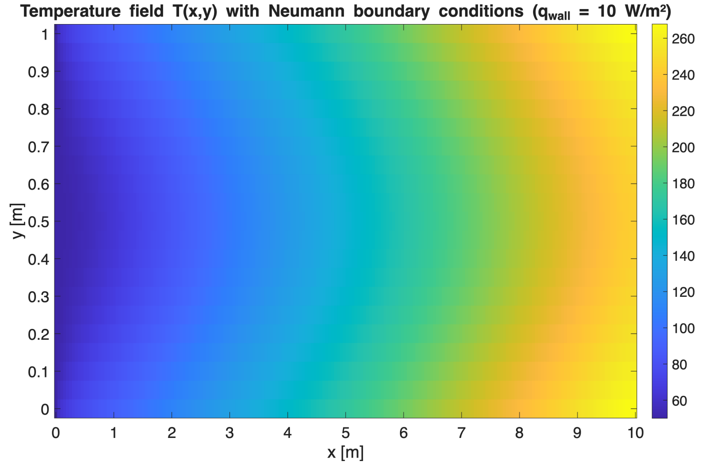


FIGURE 11 – Temperature field $T(x, y)$ with Neumann wall boundary conditions ($q_{\text{wall}} = 10$ W/m^2).

7.1.3 Local Nusselt number

The local Nusselt number is defined as

$$Nu_q(x) = \frac{2H\, q_{\text{wall}}}{k\left(T_w - T_{\text{mean}}\right)},$$

where T_{mean} is the weighted mean temperature defined in Equation (12). Figures 12 and 13 report the evolution of $Nu_q(x)$ for two grids, $(50, 5)$ and $(200, 21)$. In both cases, $Nu_q(x)$ is high near the inlet, where the temperature difference between wall and bulk fluid is small, and rapidly decreases before reaching a nearly constant value downstream. The fully developed region yields an asymptotic value close to the theoretical reference $Nu_q \approx 8.24$ for laminar channel flow with constant wall heat flux.

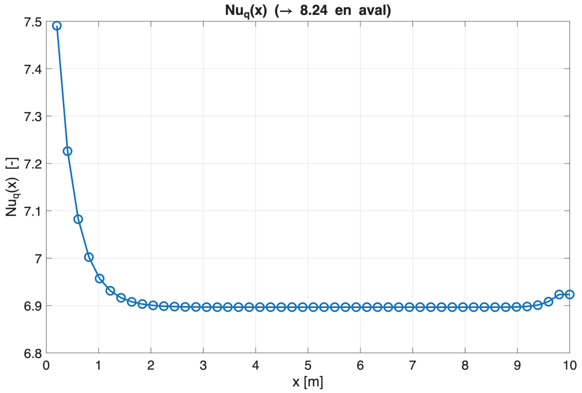


FIGURE 12 – Local Nusselt number $Nu_q(x)$ for $(n_x, n_y) = (50, 5)$.

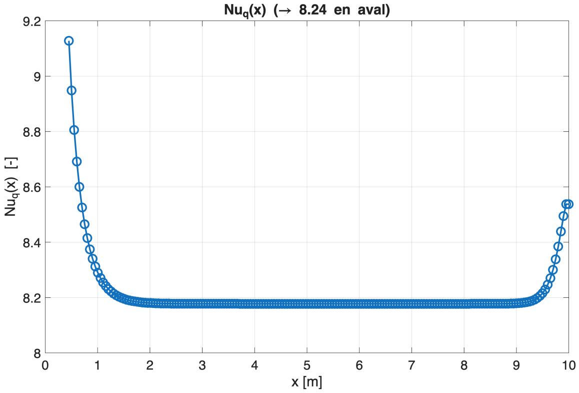


FIGURE 13 – Local Nusselt number $Nu_q(x)$ for $(n_x, n_y) = (200, 21)$.

7.1.4 Mesh–refinement analysis

As the grid is refined, the computed $Nu_q(x)$ profile becomes smoother and the asymptotic value approaches the theoretical limit more closely : for the coarse mesh $(50, 5)$, Nu_q stabilizes around ≈ 6.9, whereas for the refined mesh $(200, 21)$, the value increases to ≈ 8.2, close to the theoretical value of 8.24. This demonstrates the expected grid convergence of the finite–volume solver and confirms that numerical diffusion is reduced on finer meshes.

An increase in the downstream Nu_q values is observed on the most refined grid. This minor rise can be attributed to the outlet Neumann condition ($\partial T/\partial x = 0$), which tends to flatten the temperature gradient near the channel end, producing a small overestimation of T_{mean}, which is closer to T_w, and hence a modest increase in the computed Nusselt number.

7.1.5 Verification of boundary conditions

The a posteriori verification of wall fluxes confirms that the computed fluxes match the imposed value $q_{\text{wall}} = 10\ \text{W/m}^2$ with negligible relative error. No unphysical oscillations are observed, and the outlet condition $\partial T/\partial x = 0$ is correctly satisfied, as the streamwise temperature gradient vanishes near the exit. Therefore, the Neumann flux boundary conditions are properly implemented and numerically consistent within the finite–volume formulation.

So we can say that the simulation correctly reproduces the behavior of a laminar flow with constant wall heat flux : the temperature increases linearly along the channel, and the local Nusselt number approaches the theoretical fully developed value as the mesh is refined. The solution is stable, bounded, and physically consistent, confirming the correct implementation of the Neumann–flux boundary condition.

7.2 Q21 — Global energy balance

We verify that the total energy added to the fluid equals the energy leaving the domain. With a spanwise thickness equal to one (per unit depth), the energy balance over the control volume reads

$$\dot{m}\, c_p \big(T_{\text{out}} - T_{\text{in}}\big) = 2\, q_{\text{wall}}\, L \qquad \text{with} \qquad \dot{m} \;=\; \rho \int_0^H u(y)\, dy, \tag{15}$$

where T_{in} and T_{out} are defined with T_{mean}, which is the weighted mean temperature defined in Equation (12). In discrete form, we report the three power terms and the residual

$$Q_{\text{in}} = \dot{m}c_p\, T_{\text{in,bulk}}, \qquad Q_{\text{out}} = \dot{m}c_p\, T_{\text{out,bulk}}, \qquad Q_{\text{walls}} = 2\, q_{\text{wall}}\, L,$$

and define the imbalance

$$I \;=\; Q_{\text{in}} + Q_{\text{walls}} - Q_{\text{out}}.$$

With $q_{\text{wall}} = 10$ W/m^2, $L = 10$ m, $\rho = 1$, $c_p = 10$ and $(n_x, n_y) = (200, 21)$ we obtain :

Quantity	Value [W]	Note
Q_{in}	49.38	inlet bulk enthalpy flow
Q_{walls}	200	$2\, q_{\text{wall}} L$
Q_{out}	245.21	outlet bulk enthalpy flow
$I = Q_{\text{in}} + Q_{\text{walls}} - Q_{\text{out}}$	4.17	residual

The relative imbalance is

$$\frac{|I|}{Q_{\text{in}} + Q_{\text{walls}}} \;\approx\; 1.67\% \qquad \text{and} \qquad \frac{|I|}{Q_{\text{out}}} \;\approx\; 1.70\%.$$

A positive I of 4.17 W means that the outgoing enthalpy flux is slightly lower than the total input (inlet + walls).

A residual of $\sim 1\%$ is consistent with the discretization choices : (i) first–order UD convection on a finite grid (numerical diffusion), (ii) outlet Neumann condition $\partial T/\partial x = 0$ that can slightly flatten the last temperature column and bias T_{mean} at the outlet and (iii) trapezoidal integration of uT. In practice, the imbalance decreases by refining (n_x, n_y).

The discrete energy balance is nearly satisfied : the error is small and well explained by numerical effects at the outlet and by grid resolution. Overall, the implementation of the Neumann wall flux and the bulk–temperature integrals is consistent with the theoretical balance in Equation (15).

7.3 Q22 — Local bottom-wall heating and target outlet temperature

We replace the isothermal walls by a heat flux on the bottom wall, applied only in $2 \leq x \leq 5$, with inlet Dirichlet $T_{\text{in}} = 50$°C, outlet $\partial T/\partial x = 0$, and top wall adiabatic. On the heated segment we impose

$$k\,\frac{\partial T}{\partial n} = q_{\text{wall}} \quad \text{for } 2 \leq x \leq 5, \qquad k\,\frac{\partial T}{\partial n} = 0 \quad \text{elsewhere},$$

while the top wall has $k\,\partial T/\partial n = 0$.

In Matlab, the wall Neumann condition is enforced at the wall : for a bottom-wall node $(1, j)$,

$$\frac{T_{1,j} - T_{2,j}}{\Delta y} k = q_{\text{wall}}(x_j),$$

with $q_{\text{wall}}(x_j) = q_0$ if $x_j \in [2m, 5m]$ and 0 otherwise. To obtain the reference solution corresponding to a target outlet temperature, the code performs an iterative search on the imposed wall heat flux q_{wall}. The procedure starts by defining two bounds, $q_{\min} = 0$ and $q_{\max} = 100$ W/m^2, and evaluates a mid–value $q_{\text{test}} = \frac{q_{\min}+q_{\max}}{2}$. A steady–state solution is computed with this boundary condition, and the resulting T_{out} is compared with the desired target temperature T_{target}. If $T_{\text{out,mean}} < T_{\text{target}}$, the lower bound is increased ($q_{\text{min}} = q_{\text{test}}$) ; otherwise, the upper bound is reduced ($q_{\text{max}} = q_{\text{test}}$). The algorithm iterates until convergence, thereby determining the required wall heat flux that achieves the specified outlet temperature.

Figure 14 shows $T(x, y)$ and the axial evolution of $T_{\text{mean}}(x)$ for three targets $(60, 70, 80°\text{C})$, using $(n_x, n_y) = (200, 21)$. The dashed red lines denote the heated window $[2, 5]$ m. We observe a linear rise of $T_{\text{mean}}(x)$ across the heated zone and a plateau downstream.

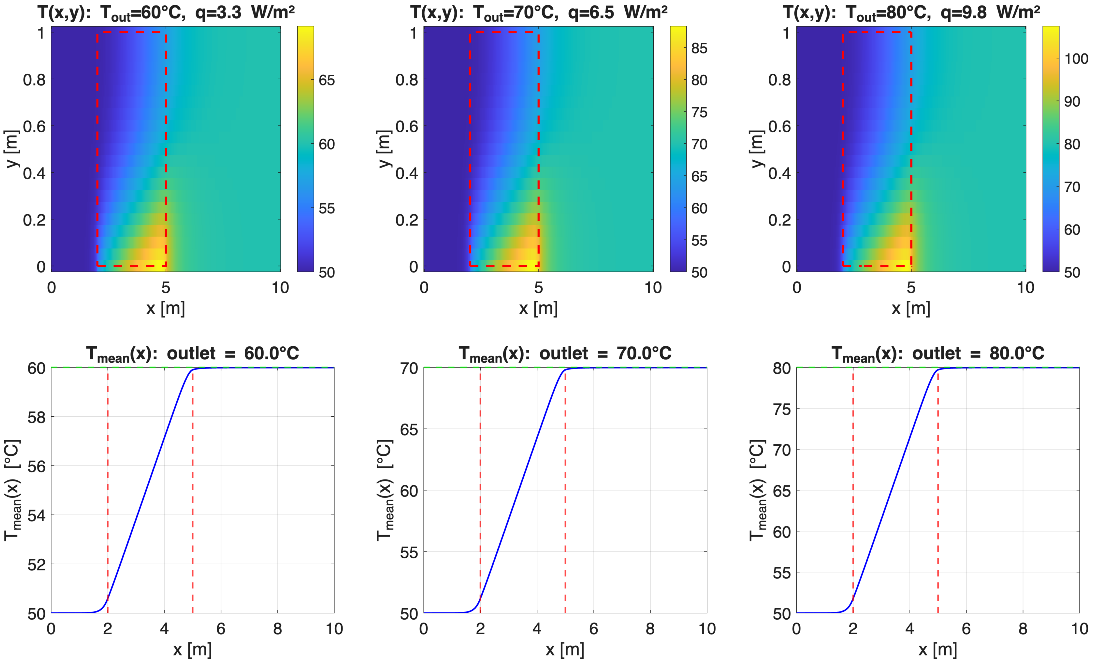


FIGURE 14 – Local heating on the bottom wall ($x \in [2, 5]$ m). Top : temperature field $T(x, y)$. Bottom : mean temperature $T_{\text{mean}}(x)$ with outlet target shown by a dashed green line.

The required flux levels obtained are :

ΔT [°C]	Target T_{out} [°C]	q_{wall} [W/m^2]
+10	60	3.27
+20	70	6.54
+30	80	9.81

TABLE 5 – Required bottom-wall flux to hit the target outlet bulk temperature.

7.3.1 Discussion and checks

- **Near-linearity.** The three values of q_{wall} scale almost exactly with the requested ΔT, confirming the expected linear response.

- **Energy consistency.** The global balance from Q21 (using the same configuration) shows $Q_{\text{out}} \approx Q_{\text{in}} + 2\int q_{\text{wall}}\,dx$ up to a few watts, consistent with discretization and outlet-Neumann truncation errors.
- **Boundary conditions.** The solution remains bounded and smooth ; no spurious oscillations appear. The wall–flux is correctly enforced at the DOFs (the numerical gradient at the walls matches the imposed q_{wall}).

8 Annexes

```matlab
function [A,b] = assembleSystem_Q1toQ6(nx,ny,dx,dy,u_prof,rho,Gamma)
N = nx*ny;
A = spalloc(N,N,5*N);
b = zeros(N,1);
for j = 2:(nx-1)
    for i = 2:(ny-1)
        n  = (j-1)*ny + i;
        nW = n - ny; nE = n + ny; nS = n - 1; nN = n + 1;
        De = Gamma*dy/dx; Dw = De;
        Dn = Gamma*dx/dy; Ds = Dn;
        uP = u_prof(i);
        Fe = rho*uP*dy; Fw = Fe;
        aW = Dw + max(Fw,0);
        aE = De + max(-Fe,0);
        aS = Ds; aN = Dn;
        aP = aW + aE + aS + aN;
        A(n,n ) = aP;
        A(n,nW) = -aW;
        A(n,nE) = -aE;
        A(n,nS) = -aS;
        A(n,nN) = -aN;
    end
end
end
```

Listing 1 – Assembly of the algebraic system up to Question 6

```matlab
for j = 1:nx
    for i = 1:ny
        n = (j-1)*ny + i;

        isInlet  = (j == 1);
        isOutlet = (j == nx);
        isBottom = (i == 1);
        isTop    = (i == ny);

        % --- Inlet (Dirichlet) ----------------------------
        if isInlet
            A(n,:) = 0;
            A(n,n) = 1;
            b(n)   = Tinlet;
            continue;
        end

        % --- Outlet (Neumann: dT/dx = 0) -------------------
        if isOutlet
            nW = n - ny; % west neighbour
            A(n,:) = 0;
            A(n,n)  =  1;
            A(n,nW) = -1;
            b(n)    =  0;
            continue;
        end

        % --- Bottom and top walls (Dirichlet) ---------------
        if isBottom || isTop
            A(n,:) = 0;
            A(n,n) = 1;
            b(n)   = Twall;
            continue;
        end

        % (interior cells are treated in the main assembly loop)
    end
end
```

Listing 2 – Implementation of boundary conditions (Questions 7–9)

Références

[1] Slides from the ME-474 Numerical Flow Simulation course,
École Polytechnique Fédérale de Lausanne (EPFL), Fall 2025.

[2] Patankar, S. V. (1980). *Numerical Heat Transfer and Fluid Flow.*
Hemisphere Publishing Corporation.

[3] Bejan, A. (2013). *Convection Heat Transfer*, 4th Edition.
John Wiley & Sons.

[4] Rohlfs, W., & Lienhard, J. H. (2016). Entrance length effects on Graetz number scaling in laminar duct flows with periodic obstructions.
International Journal of Heat and Mass Transfer.

[5] Versteeg, H. K., & Malalasekera, W. (2007). *An Introduction to Computational Fluid Dynamics : The Finite Volume Method*, 2nd Edition.
Pearson Education.

[6] OpenAI, *ChatGPT : language model for content assistance and text refinement*,
`https://chat.openai.com`, accessed on 23 May 2025.