



MTF072 Computational Fluid Dynamics

H. Nilsson, D. Ghosh, K. Konstantinidis

November 17, 2022

Task 2

In Task 1 you worked with the diffusion equation in 2-D. Now we add convective terms to it (diffusion terms on the RHS and convection terms in the LHS). This Equation 1 is called the convection-diffusion equation, or the transport equation for temperature. In this task, you will write a program in Python to solve Eq. 1 in 2-D using finite volume methods. Use the hybrid upwind/central differencing scheme, see Eq. 5.43 in [1]. **Note that we set the continuity error in the convectives terms to be zero ($\Delta F = 0$) since we provide you with a fixed velocity field and there is no possibility of correcting it.**

There are 25 unique cases that will be provided and your group number defines the case you should pick. For example, group 20 should choose case 20, group 30 should pick case 5, and group 50 should choose case number 25, and so on. The velocity field and the grid are given for each group and are available in the course homepage (in Canvas). This task should also be carried out in groups, and should be presented in the form of a report.

The two-dimensional transport equation for temperature reads

$$\frac{\partial}{\partial x}(\rho UT) + \frac{\partial}{\partial y}(\rho VT) = \frac{\partial}{\partial x}\left(\Gamma \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma \frac{\partial T}{\partial y}\right) + S \quad (1)$$
$$\Gamma = \frac{k}{c_p}.$$

Where k and C_p varies according to your particular case.

The algebraic equation system should be solved using both:

- Gauss-Seidel
- TDMA

How to proceed

- In order to help you with the task, a templates is given in Python (`task_2_template.py`) that contains two extra functions. These functions don't need to be modified:

`ReadDataAndGeometry` and `PlotVectorField`. **Make sure to use this template and to stick to the same nomenclature** so that getting help from the assistants becomes easier.

- **Work in Kelvin instead of degrees Celsius!**
- **The heat fluxes exiting the domain are considered negative while the ones entering it are positive. This applies both for the boundary conditions as well as when computing global conservation later on.**
- Fetch your grid and velocity field (based on the case number) from the file `data.zip` at course home page . **You do not have to create the grid. Download the respective data folder from Canvas and extract it in the same directory as your `.py` file so that the provided code will work.**
- Plot the velocity field using the `PlotVectorField` function defined inside the Python template. Using the provided velocity field you can see the location and length of the inlet and outlet for your case (i.e. the walls are located at the points where both the velocity components are zero, or a extremelly small number such as 10^{-10}). Implement a code that identifies the boundary nodes and indices that correspond to an inlet or outlet.
- Write your program to solve Eq. 1
- First, use a coarse grid when you are writing/debugging your program and implement the residual criteria as suggested in next section. Later, when the program is fully functional, get updated results using the fine grid.
- Use $\partial T / \partial n = 0$ (n denotes the coordinate normal to the wall) as the boundary condition if not specified otherwise (**check your respective case**).
- A few groups will have a non-homogeneous Neumann condition. Consider how such a boundary condition should be implemented in an implicit way.

Convergence

It is very important to verify that a converged solution has been obtained. At each iteration compute the residual as

$$\varepsilon = \frac{1}{F} \left(\sum_{\text{all cells}} |a_P T_P - (a_E T_E + a_W T_W + a_N T_N + a_S T_S + S_u)| \right) \quad (2)$$

where F is a temperature flux used to normalize the residual. The temperature flux F should be a representative flux in your domain. In the present

work it is suitable to take F as the inlet convective heat flux minus the outlet convective heat flux, i.e.

$$F = abs \left(\sum_{inletBoundary} |(\rho U_i \Delta x_i) T| - \sum_{outletBoundary} |(\rho U_i \Delta x_i) T| \right), \quad (3)$$

where T is the temperature of each cell at the inlet or outlet. The solution is considered as converged if $\varepsilon < 0.001$. You can of course change this threshold to see if any differences are visible.

Reporting of the work

The work should be presented with a short report (max 12 pages without taking into account the appended code. **This is a strict limit**). The code can be included as an appendix at the end of the report but it is also needed to be uploaded in the files. It is important to try to discuss your results from a physical and a numerical point of view. Showcase relevant results for example as contour plots of the temperature etc. to support your findings. The report must include the following parts:

- Sensitivity to boundary conditions. Make (an interesting) change of one boundary condition (Similar to Task 1).
- Sensitivity to the solver. Which one is faster? Have you tried all possible directions while using TDMA?
- Sensitivity to convergence. Does the temperature change if you use another convergence criterion? Test $\varepsilon = 0.01$ and $\varepsilon = 0.0001$.
- Sensitivity to heat conductivity. Increase and decrease k by a factor of 100. Discuss how the results are affected and why.
- Plot the heat flux (along a Dirichlet boundary) or the wall temperature (along a Neumann boundary) as a function of x or y (it depends which wall are you considering). Describe what you see.
- Check if you approach global conservation, i.e. does the heat flux through all boundaries (inlet, outlet and walls) sum up to zero. Remember that you have two kinds of fluxes on the boundaries; convective fluxes (along inlet and outlet) and diffusive fluxes (along Dirichlet boundaries).
- Seldom will you end up with 0 (for the global conservation), due to the fact that you are not solving the velocity field, or using approximations for the coefficients (designed for equidistant mesh). Compare the net flux just calculated with the total entering fluxes or the sum of the absolute value of fluxes along the boundaries. A ratio lower than 10% is expected.

The general geometry

The general description of the geometry and boundaries for all the cases is presented below (c.f. Fig 1). The capital letters A , B , C and D , indicates the position of the inlet/outlet. Cannot find your inlet and outlets? Plot the velocity field to check where you have inlet and outlet conditions.

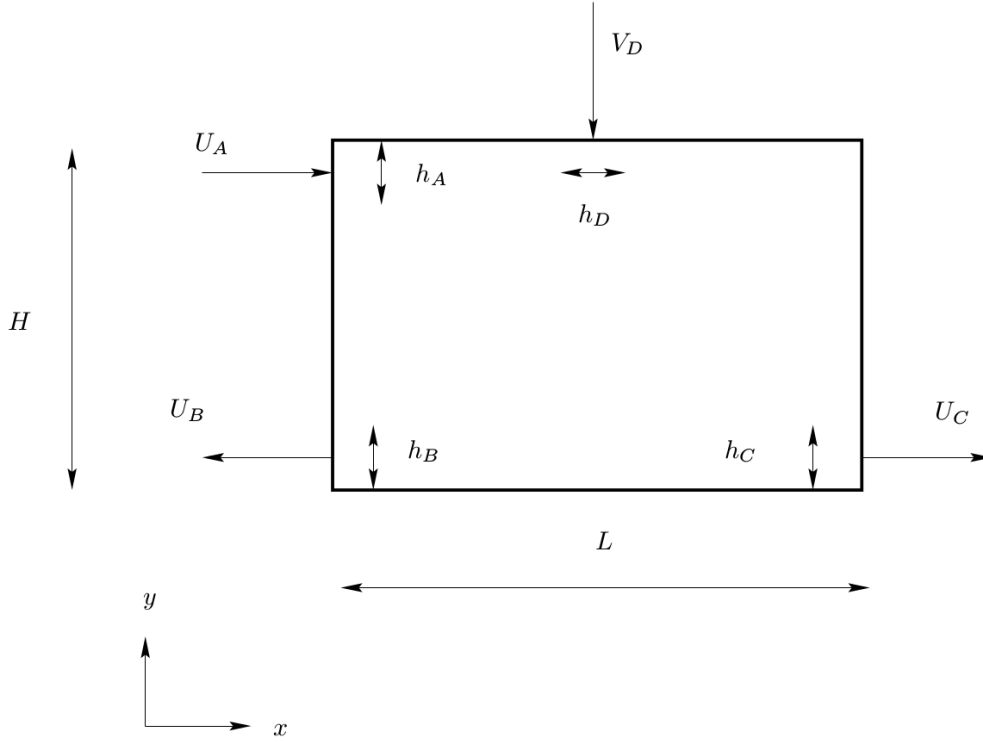


Figure 1: Configuration. The extent in the third coordinate direction is 1.

GRID 1: Case 1 to 5

Physical data: $\rho = 1$, $k = 1$ and $c_p = 500$.

Inlet boundary conditions: **Dirichlet** $T_A = 20^\circ C$.

Outlet boundary conditions: **Neumann**, $\partial T / \partial n = 0$.

Case	b.c.
1	$y = H \rightarrow T = 10$
2	$y = H \rightarrow q = 100 \text{ W/m}^2$
3	$x = L \rightarrow T = 10$
4	$x = L \rightarrow T = 50$
5	$y = 0 \rightarrow q = 100 \text{ W/m}^2$

Table 1: Boundary conditions for Case 1 to 5.

GRID 2: Case 6 to 10

Physical data: $\rho = 1$, $k = 1$ and $c_p = 500$.

Inlet boundary conditions: **Dirichlet** $T_A = 20^\circ C$.

Outlet boundary conditions: **Neumann**, $\partial T / \partial n = 0$.

Case	b.c.
6	$x = 0 \rightarrow T = 10$
7	$x = 0 \rightarrow T = 5$
8	$x = 0 \rightarrow q = 100 \text{ W/m}^2$
9	$x = 0, y/H \geq 0.5 \rightarrow T = 10$
10	$x = 0, y/H \leq 0.5 \rightarrow T = 10$

Table 2: Boundary conditions for Case 6 to 10.

GRID 3: Case 11 to 15

Physical data: $\rho = 1$, $k = 1$ and $c_p = 200$.

Inlet boundary conditions: **Dirichlet** $T_D = 10^\circ C$.

Outlet boundary conditions: **Neumann**, $\partial T / \partial n = 0$.

Case	b.c.
11	$x = 0 \rightarrow T = 20; x = L \rightarrow T = 20; y = 0 \rightarrow T = 0$
12	$x = 0 \rightarrow T = 0; x = L \rightarrow T = 20; y = 0 \rightarrow T = 0$
13	$x = L \rightarrow T = 20; y = 0 \rightarrow T = 10$
14	$x = 0 \rightarrow T = 20; x = L \rightarrow T = 20$
15	$y = 0 \rightarrow q = 100 \text{ W/m}^2$

Table 3: Boundary conditions for Case 11 to 15.

GRID 4: Case 16 to 20

Physical data: $\rho = 1$, $k = 1$ and $c_p = 200$.

Inlet boundary conditions: **Dirichlet** $T_D = -10^\circ C$.

Outlet boundary conditions: **Neumann**, $\partial T / \partial n = 0$.

Case	b.c.
16	$x = 0 \rightarrow T = 0; x = L \rightarrow T = -20; y = 0 \rightarrow T = 0$
17	$x = L \rightarrow T = 20; y = 0 \rightarrow T = 0$
18	$x = 0 \rightarrow T = 0; x = L \rightarrow q = -100 \text{ W/m}^2; y = 0 \rightarrow T = 10$
19	$x = 0 \rightarrow T = 50, x = L \rightarrow T = -20; y = 0 \rightarrow T = 10$
20	$x = 0 \rightarrow T = 20; y = 0 \rightarrow q = 50 \text{ W/m}^2$

Table 4: Boundary conditions for Case 16 to 20.

GRID 5: Case 21 to 25

Physical data: $\rho = 1$, $k = 1$ and $c_p = 200$.

Inlet boundary conditions: **Dirichlet** $T_D = 0^\circ C$.

Outlet boundary conditions: **Neumann**, $\partial T / \partial n = 0$.

Case	b.c.
21	$x = 0 \rightarrow T = 10$; $y = 0 \rightarrow T = 0$
22	$x = 0 \rightarrow T = 0$; $x = L \rightarrow T = 20$; $y = 0 \rightarrow T = 0$
23	$x = 0 \rightarrow T = 10$; $x = L \rightarrow T = -10$; $y = 0 \rightarrow T = 0$
24	$x = L \rightarrow T = 20$; $y = 0 \rightarrow T = 20$
25	$x = 0 \rightarrow T = 0$; $x = L \rightarrow q = 50W/m^2$; $y = 0 \rightarrow T = 30$

Table 5: Boundary conditions for Case 21 to 25.

References

- [1] H. Vesteg and W. Malalasekera, “An introduction to computational fluid dynamics,” 1995.