The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center ASEE
Portland, OR . June 23 - 26, 2024 Paper ID #44153

Integrating Theory and Practice: A CFD Education Approach

Dr. MEHMET Nasir SARIMURAT, Syracuse University

Mehmet Nasir Sarimurat earned his Ph.D. from Syracuse University in Syracuse, NY, USA, in 2008. He held positions as a Senior and Staff Engineer at United Technologies Carrier Corporation in East Syracuse, NY, USA, from 2007 to 2018. In 2018, he made the transition to the Department of Mechanical and Aerospace Engineering at Syracuse University. Currently, he serves as an Associate Teaching Professor and also holds the role of Undergraduate Program Director for Mechanical Engineering. His research is primarily focused on fluid mechanics and turbomachinery, with a particular emphasis on model-based simulation, design, and high-performance computing.

©American Society for Engineering Education, 2024

Integrating Theory and Practice: A CFD Education Approach

Mehmet Sarimurat
mnsarimu@syr.edu
Mechanical and Aerospace Engineering Department
Syracuse University

Abstract

This paper introduces the design and implementation of a course in computational fluid dynamics (CFD) offered to senior undergraduate students at Syracuse University. The course utilizes a novel structure that integrates theoretical foundations with practical,“tutorial-based” experiences. The curriculum balances theoretical fundamentals, solidified through numerical solution implementation in Python, with hands-on experience using industry-standard Ansys Fluent software. Notably, the use of Python in the introductory phase prepares students for the increasing utilization of Python for customization and optimization within commercial CFD packages. Furthermore, the second part of the course adopts a unique problem-solving approach where students actively replicate pre-recorded tutorials, fostering deeper understanding compared to traditional lecture formats. This comprehensive and student-centered curriculum prepares future engineers with the critical skills and software proficiency required to address contemporary fluid mechanics challenges.

Introduction

In the context of rapid technological progress, the widespread use of computers and their substantial computing power has led to the recognition of Computational Fluid Dynamics (CFD) as a crucial tool in the design and analysis of engineering applications. This technology has permeated various sectors, including aerospace design, chemical processing, oil drilling, biotechnology, and energy generation. CFD’s transformative power lies in its ability to significantly reduce the need for expensive physical prototypes. By virtually simulating and visualizing complex fluid flows, engineers can identify design flaws, optimize performance, and accelerate the development of cutting-edge solutions.

As CFD becomes more prevalent and indispensable in many industries, there is an increasing demand for competent CFD practitioners. To address this demand, many universities have incorporated CFD education into their undergraduate mechanical and aerospace engineering curriculum[1,2,3]. A comprehensive review of existing undergraduate CFD courses can be found in Li and Cheung[4].

A key challenge in educating future computational fluid dynamics (CFD) professionals lies in bridging the gap between theoretical understanding and practical application. Traditional undergraduate CFD courses often lean heavily on instructor-led lectures, leaving limited space for student engagement through hands-on learning. This stands in stark contrast to the experiences of industry professionals, who often acquire their skills outside of formal classroom settings. This suggests that active learning approaches, which engage students beyond passive information consumption, can foster valuable skill development even without direct classroom instruction[4]. Supporting this view, a study by Adair et al.[5] found that a significant majority (73%) of CFD students preferred learning through tutorials, lab activities, or collaborative group work, compared to only 11% who favored traditional lectures. These findings highlight the need for a shift towards more active learning approaches in CFD education.

This paper introduces a novel undergraduate CFD course specifically designed to overcome this challenge. The course curriculum blends a rigorous theoretical foundation with a unique ”tutorial-based” practical experience. In the first segment, the fundamentals of CFD are explored, emphasizing the numerical solution of differential equations using the finite volume method. Drawing inspiration from the renowned book of Versteeg and Malalasekera[6], “An Introduction to Computational Fluid Dynamics: The Finite Volume Method,” students gain hands-on experience implementing numerical solutions for 1D and 2D diffusion and convection-diffusion equations through Python programming.

Transitioning from theoretical foundations to real-world applications, the second part of the course equips students with practical experience using industry-standard commercial simulation software, Ansys Fluent. In this section, students tackle diverse engineering scenarios like supersonic wind tunnels, laminar and turbulent flow simulations, and convective cooling systems. This section adopts a unique learning approach: instead of passively receiving instruction on how to use the CFD code, students actively solve problems by watching pre-recorded tutorials, replicating the presented scenarios, and submitting their work as quizzes. It’s important to note that the problems covered in the videos are first discussed in detail in class. Students then transition to the computer lab equipped with the CFD software, where they watch the videos and recreate the covered concepts with the instructor readily available to answer any questions that arise during the simulations.

The course’s comprehensive nature, covering both theoretical fundamentals and practical experience with industry-standard software, aims to prepare students to address contemporary fluid mechanics challenges. The teaching methodology integrates theory and ”tutorial-based” practice, aligning with the evolving landscape of engineering education, which emphasizes hands-on experience and practical skills. “The Applications of Computational Fluid Mechanics” course, outlined in this paper, serves as a model for imparting essential CFD skills, contributing to the advancement of engineering education.

Governing Equations for Fluid Flow

The Applications of Computational Fluid Dynamics course begins by deriving the fundamental equations for fluid flow, which include the conservation of mass, momentum, and energy. Notably, this section is both challenging and crucial for understanding the course content. The governing equations for fluid flow can be intricate, often posing a complexity that students may find challenging. However, a comprehensive grasp of these equations is essential for students to numerically solve and interpret the results effectively. Consequently, significant emphasis is placed on thoroughly deriving these equations in the initial classes. Key references for this derivation include CFD books by Anderson[7], as well as Versteeg and Malalasekera[6]. Due to space limitations, the detailed derivation of the conservation equations is omitted, and only the final form of the equations is presented in the vector form as follows:

Continuity:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \tag{1}$$

x-momentum:

$$\frac{\partial (\rho u)}{\partial t} + \nabla \cdot (\rho u \mathbf{V}) = -\frac{\partial p}{\partial x} + \nabla \cdot (\mu \nabla u) + S_{Mx} \tag{2}$$

y-momentum:

$$\frac{\partial (\rho v)}{\partial t} + \nabla \cdot (\rho v \mathbf{V}) = -\frac{\partial p}{\partial y} + \nabla \cdot (\mu \nabla v) + S_{My} \tag{3}$$

z-momentum:

$$\frac{\partial (\rho w)}{\partial t} + \nabla \cdot (\rho w \mathbf{V}) = -\frac{\partial p}{\partial z} + \nabla \cdot (\mu \nabla w) + S_{Mz} \tag{4}$$

Energy:

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{V}) = -p \nabla \cdot \mathbf{V} + \nabla \cdot (k \nabla T) + \Phi + S_e \tag{5}$$

Several points related to Equations (1)-(5) above:

- In Equations (2)-(4), S_M represents the momentum source, encompassing both the body force and viscous stress terms that account for compressibility. This serves to further streamline and simplify the momentum equations.
- In Equation (5), Φ denotes the dissipation function, responsible for capturing the energy dissipation resulting from viscous stress, and S_e represents the energy source term.
- By combining Equations (1)-(5) with the equations of state for perfect gases, which are:

$$p = \rho R T \tag{6}$$

and

$$e = C_v T \tag{7}$$

and assuming known values for the viscosity μ, conduction coefficient k, and specific heat constant C_v for a given fluid, a system of seven equations and seven unknowns is obtained. The unknowns include the density ρ, the three velocity components (u, v, w), pressure p, temperature T, and internal energy e. Consequently, particularly for laminar flows, the mathematical problem is closed. Given appropriate initial and boundary conditions, solving these seven equations provides a solution for the seven unknown flow properties.

- Equations (1)-(5) form a set of coupled, nonlinear partial differential equations. It is not possible to solve these equations analytically, except for a narrow range of very simple problems. For more complicated real-world engineering problems, these equations can only be solved numerically and this is where the need for computational fluid dynamics comes in.

Transport Equation: The Model Equation for Fluid Flow

Examining Equations (1)-(5), one can discern a similarity in their formulation. This commonality allows for the characterization of these equations under the term known as the transport equation. The general transport equation for a fluid property ϕ in its differential form is considered:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho\phi\mathbf{V}) = \nabla \cdot (\Gamma\nabla\phi) + S_\phi \tag{8}$$

In the above equation, property ϕ can represent any fluid properties, such as velocity, temperature, or pollutant concentration. The first term on the left-hand side denotes the temporal change (rate of increase or decrease) of the property ϕ, while the second term represents the net rate of flow of the property in or out of the fluid element. On the right-hand side, the first term characterizes the rate of change (increase or decrease) of the property ϕ due to diffusion, and the second term indicates the production or destruction of the property.

In Equation (8), by selecting appropriate values for the diffusion coefficient Γ and source terms, one can set ϕ equal to the following quantities (see Table 1) to obtain the special forms of conservation equations as presented in Eqs. (1)-(5).

Table 1: Correspondence between Transport Equation and Conservation Laws

$\phi = 1$	Continuity
$\phi = u$	x-momentum
$\phi = v$	y-momentum
$\phi = w$	z-momentum
$\phi = T$	Energy equation

As is commonly practiced in introductory Computational Fluid Dynamics (CFD) courses, this class utilizes the Finite Volume Method (FVM) to solve the integral form of the transport equation presented in Eq.(8). This approach introduces students to the numerical methods used in solving the governing equations of fluid flow, avoiding the complexities associated with attempting to solve the entire set of conservation laws presented in Eqs. (1)-(5)

We start by integrating Eq. (8) over a random differential control volume $d\forall$ in the form:

$$\int_{CV} \frac{\partial(\rho\phi)}{\partial t}\, d\forall + \int_{CV} \nabla \cdot (\rho\phi\mathbf{V})\, d\forall = \int_{CV} \nabla \cdot (\Gamma\nabla\phi)\, d\forall + \int_{CV} S_\phi\, d\forall \tag{9}$$

By applying the Gauss Divergence Theorem, which states that for an arbitrary function $\mathbf{F}$

$$\int_{CV} \nabla \cdot \mathbf{F}\, d\forall = \int_{CS} \mathbf{F} \cdot\, d\mathbf{A} \tag{10}$$

the transport equation takes on the following form:

$$\frac{\partial}{\partial t}\int_{CV} (\rho\phi)\, d\forall + \int_{CS} (\rho\phi\mathbf{V}) \cdot\, d\mathbf{A} = \int_{CS} (\Gamma\nabla\phi) \cdot\, d\mathbf{A} + \int_{CV} S_\phi\, d\forall \tag{11}$$

Introduction to the Finite Volume Method

The finite volume method offers a structured approach to solving partial differential equations, consisting of the following five fundamental steps.

Step 1: Mesh Generation

Initially, the domain is subdivided into smaller sections, a process commonly referred to as mesh generation.

Step 2: Deriving the integral form of the differential equations that need to be solved

The second step involves ensuring that students grasp the given differential equations, integrate the equation, and convert volume integrals to surface integrals using the Gauss divergence theorem where applicable.

Step 3: Discretization for Internal Cells

Subsequently, the integral form of the equation derived in Step 2 is discretized for internal cells.

Step 4: Discretization for Boundary Cells

Here, the integral form of the equation derived in Step 2 is discretized for the boundary cells.

Step 5: Matrix Setup

Once the differential equation is discretized for both internal and boundary cells, the final task is to set up the matrix system and solve it to determine the fluid property (e.g., the variable in the differential equation) at each cell.

To illustrate the solution of a differential equation using the finite volume method, we select the 1D convection-diffusion equation with a source term, expressed as:

$$\frac{d}{dx}(\rho\phi u) = \frac{d}{dx}\left(\Gamma\frac{d\phi}{dx}\right) + S_\phi \qquad 0 \leq x \leq L \tag{12}$$

This equation signifies the transportation of the property ϕ, as depicted in the schematic in Figure 1. The domain size is denoted by L, and the values of ϕ at the left and right boundaries are predetermined. Furthermore, there is a production of ϕ within the control volume, as represented by the source term S_ϕ.

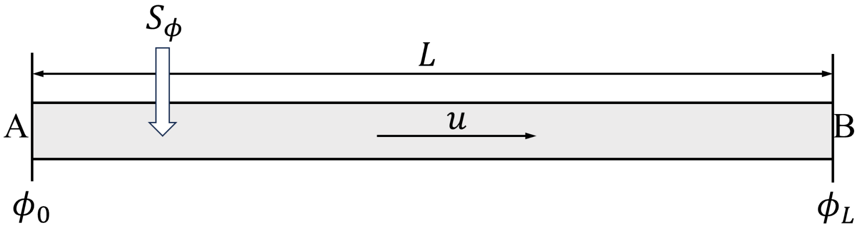

Figure 1: One-dimensional domain for transportation of property ϕ with source

This problem will be solved numerically using the finite volume method and the five steps presented above.

Step 1: Mesh Generation

Initially, the domain is divided into a finite number of N cells, and N control volumes are created. For instance, $N = 5$ can be chosen, resulting in the generation of five finite control volumes, as illustrated in Figure 2.

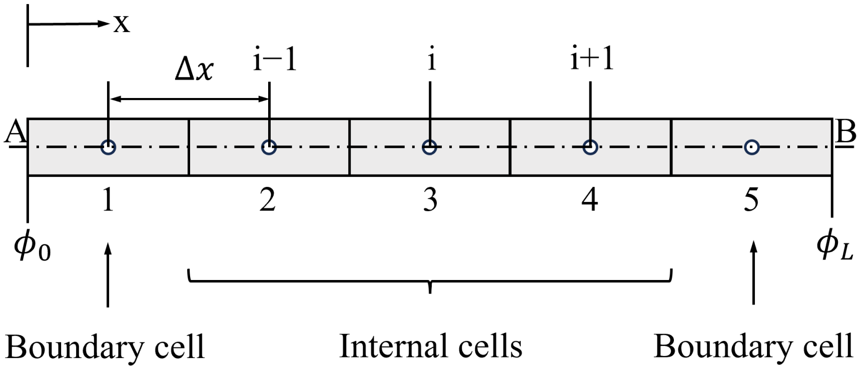

Figure 2: Mesh generation

Step 2: Deriving the integral form of the differential equations that need to be solved:

The differential equation given in Eq.12 needs to be solved numerically using FVM. Since the finite volume method solves the integral form of the differential equations, we begin by integrating the equation over an arbitrary control volume, yielding:

$$\int_{CV} \frac{d}{dx}(\rho\phi u)d\mathcal{V} = \int_{CV} \frac{d}{dx}\left(\Gamma\frac{d\phi}{dx}\right)d\mathcal{V} + \int_{CV} S_\phi d\mathcal{V} \tag{13}$$

Applying the Gauss divergence theorem:

$$\int_{CS} (\rho\phi u)dA = \int_{CS} \left(\Gamma\frac{d\phi}{dx}\right)dA + \int_{CV} S_\phi d\mathcal{V} \tag{14}$$

Step 3: Discretize the equation for the internal cells

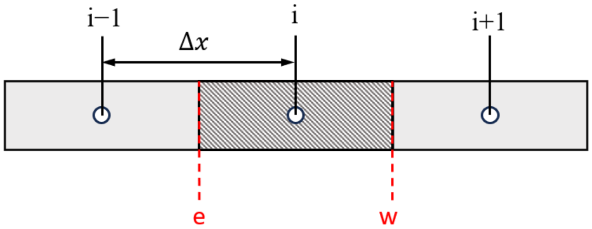

Figure 3: Internal cell used for discretizing the equation

For an arbitrary internal cell (i) shown in Fig. 3 with the hatched lines and cell boundaries (w) to the west and (e) to the east, the integral in Eq. 14 can be carried out as:

$$(\rho\phi u)\, A|_e^w = \left(\Gamma\frac{d\phi}{dx}\right) A|_e^w + S_\phi\mathcal{V} \tag{15}$$

or, since ρ, u, and Γ are constant, and $\mathcal{V} = \Delta x A$:

$$\rho u A\left(\phi_w - \phi_e\right) = \Gamma A\left[\left(\frac{d\phi}{dx}\right)_w - \left(\frac{d\phi}{dx}\right)_e\right] + S_\phi\Delta x A \tag{16}$$

The value of ϕ and the gradient (dT/dx) at the west and the east boundaries can be calculated using linear approximation, or the central differencing as:

$$\rho u\left(\frac{\phi_{i+1}+\phi_i}{2} - \frac{\phi_i+\phi_{i-1}}{2}\right) = \Gamma\left(\frac{\phi_{i+1}-\phi_i}{\Delta x} - \frac{\phi_i-\phi_{i-1}}{\Delta x}\right) + S_\phi\Delta x \tag{17}$$

Defining:

$$F \equiv \rho u \qquad \text{and} \qquad D \equiv \frac{\Gamma}{\Delta x} \tag{18}$$

Eq. 17 can be simplified as:

$$\left(-D - \frac{F}{2}\right)\phi_{i-1} + 2D\phi_i + \left(-D + \frac{F}{2}\right)\phi_{i-1} = S_\phi \Delta x \tag{19}$$

Equation 19 is valid for all internal cells.

Step 4: Discretize the equation for the boundary cells

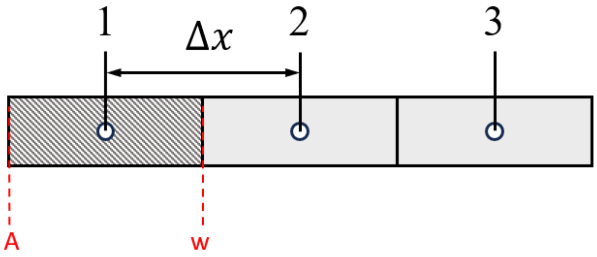

Figure 4: Cell at the left boundary

Starting with the left boundary shown in Figure 4 with the boundaries being (A) and (w), integration of Eq. 14 over this control results in

$$(\rho\phi u)\, A|_A^w = \left(\Gamma \frac{d\phi}{dx}\right) A|_A^w + S_\phi V \tag{20}$$

which can be written as:

$$\rho u\,(\phi_w - \phi_A) = \Gamma \left[\left(\frac{d\phi}{dx}\right)_w - \left(\frac{d\phi}{dx}\right)_A\right] + S_\phi \Delta x \tag{21}$$

Analyzing each term in the equation above, ϕ_w can be calculated as the average of ϕ at the two neighboring nodes, and ϕ_A is already known since $\phi_A = \phi_0$. Moving on to the terms on the left-hand side: the gradient $(d\phi/dx)_w$ can be calculated using central difference approximation, and $(d\phi/dx)_A$ can be determined using forward differencing approximation. Utilizing this information, Eq. 21 becomes:

$$\rho u \left(\frac{\phi_2 + \phi_2}{2} - \phi_0\right) = \Gamma \left(\frac{\phi_2 - \phi_1}{\Delta x} - \frac{\phi_1 - \phi_0}{\Delta x/2}\right) + S_\phi \Delta x \tag{22}$$

Using the definitions provided in Eq.18, the discretized form of the above equation becomes:

$$\left(3D + \frac{F}{2}\right)\phi_1 + \left(-D + \frac{F}{2}\right)\phi_2 = (F + 2D)\phi_0 + S_\phi \Delta x \tag{23}$$

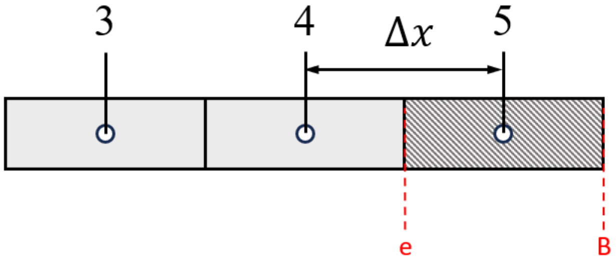

Figure 5: Cell at the right boundary

Moving on to the right boundary shown in Figure 5 with the boundaries being (e) and (B), integration of Eq. 14 over this control volume yields:

$$(\rho\phi u)\, A|_e^B = \left(\Gamma\frac{d\phi}{dx}\right) A|_e^B + S_\phi V \tag{24}$$

which can be written as:

$$\rho u\,(\phi_B - \phi_e) = \Gamma\left[\left(\frac{d\phi}{dx}\right)_B - \left(\frac{d\phi}{dx}\right)_e\right] + S_\phi \Delta x \tag{25}$$

Analyzing each term in the equation above, ϕ_B is known since $\phi_B = \phi_L$, and ϕ_e can be calculated as the average of ϕ at the two neighboring nodes. Moving on to the terms on the left-hand side: the gradient $(d\phi/dx)_B$ can be calculated using backward difference approximation, and $(d\phi/dx)_e$ can be determined using central differencing approximation. Utilizing this information along with the definitions provided in Eq. 18, the discretized form of Eq. 25 becomes:

$$\left(-D - \frac{F}{2}\right)\phi_4 + \left(3D - \frac{F}{2}\right)\phi_5 = \left(\frac{-F}{2} + 2D\right)\phi_L + S_\phi \Delta x \tag{26}$$

Step 5: Setting up the matrix

Once the differential equation that needs to be solved is discretized for the internal and boundary cells, the next step is to set up the matrix system and solve it to obtain the fluid property, in this case, ϕ, at each node.

Table 2: Discretized algebraic equations for each cell

Cell Number	Discretized equation
1	$\left(3D+\frac{F}{2}\right)\phi_1+\left(-D+\frac{F}{2}\right)\phi_2=(F+2D)\phi_0+S_\phi\Delta x$
2	$\left(-D-\frac{F}{2}\right)\phi_1+2D\phi_2+\left(-D+\frac{F}{2}\right)\phi_3=S_\phi\Delta x$
3	$\left(-D-\frac{F}{2}\right)\phi_2+2D\phi_3+\left(-D+\frac{F}{2}\right)\phi_4=S_\phi\Delta x$
4	$\left(-D-\frac{F}{2}\right)\phi_3+2D\phi_4+\left(-D+\frac{F}{2}\right)\phi_5=S_\phi\Delta x$
5	$\left(-D-\frac{F}{2}\right)\phi_4+\left(3D-\frac{F}{2}\right)\phi_5=\left(\frac{-F}{2}+2D\right)\phi_L+S_\phi\Delta x$

Table 2 summarizes the algebraic equations obtained from discretizing the differential equation for the boundary and internal cells, as presented above in Eq. (22) for the cell at the left boundary, Eq. (26) for the cell at the right boundary, and Eq. (19) for internal cells. Note that, although the domain is divided into five cells for this problem, Eqs. (19), (22), and (26) are independent of the number of cells. We could have divided the domain into 1000 cells, and still, Eq. (22) would have been valid for the left cell (cell number 1), Eq. (26) would have been valid for the right cell (cell number 1000), and Eq. (19) would have been valid for the remaining 998 internal cells. An algebraic equation for each internal cell could have been obtained by replacing (i) in this equation with the corresponding cell number.

Next, two numerical examples are presented to reinforce the use of the finite volume method in the solution of the differential equations. It is noted that these two examples are modified from Ref.[6].

Example 1: 1D diffusion Equation with the Source Term

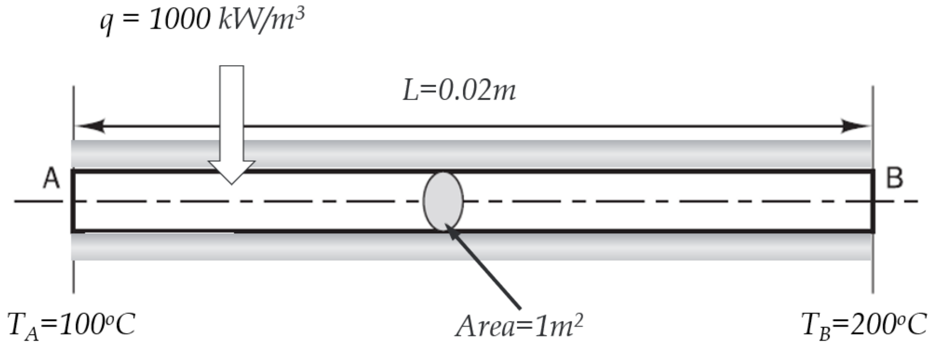

Figure 6: Heat conduction in 1D rod with the heat source

Consider the problem of heat conduction in an insulated rod, as illustrated in the figure. 6. The rod has a length of $L = 2$ cm, a constant thermal conductivity $k = 0.5$ W/mK, an area of $A = 1$ m^2, and uniform heat generation $q = 1000$ kW/m^3. Faces A and B are at temperatures $T_A = 100$ °C and $T_B = 200$ °C, respectively. The governing equation is given by:

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right)+q=0 \tag{27}$$

(a) Solve this problem analytically and find the temperature distribution $T(x)$ along the rod.

(b) Utilize the finite volume method with 5 cells ($N=5$) to calculate the steady-state temperature distribution in the rod.

To solve this problem analytically, Eq. 27 can be integrated twice, and boundary conditions can be utilized, yielding an analytical solution for $T(x)$ as

$$T(x)=\left[\frac{T_B-T_A}{L}+\frac{q}{2k}\left(L-x\right)\right]. \tag{28}$$

To solve this problem numerically using the Finite Volume Method with 5 cells, we follow the same steps covered earlier for the solution of Eq. 12. By setting $u=0$, $\phi=T$, $S_\phi=q$, and $\Gamma=k$, we can rewrite Eq. 12 to obtain Eq. 27. Additionally, we have $\phi_0=T_A$ and $\phi_L=T_B$. With these substitutions, the discretized algebraic equations in Table 2 become:

Table 3: Discretized algebraic equations for 1D diffusion problem

Cell Number	Discretized equation
1	$3DT_1-DT_2=2DT_0+q\Delta x$
2	$-DT_1+2DT_2-DT_3=q\Delta x$
3	$-DT_2+2DT_3-DT_4=q\Delta x$
4	$-DT_3+2DT_4-DT_5=q\Delta x$
5	$-DT_4+3DT_5=(2D)T_L+q\Delta x$

Equations in the above table can be put in the matrix as:

$$\begin{bmatrix} 3D & -D & 0 & 0 & 0 \\ -D & 2D & -D & 0 & 0 \\ 0 & -D & 2D & -D & 0 \\ 0 & 0 & -D & 2D & -D \\ 0 & 0 & 0 & -D & 3D \end{bmatrix}\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix}=\begin{bmatrix} 2DT_0+q\Delta x \\ q\Delta x \\ q\Delta x \\ q\Delta x \\ (2D)T_L+q\Delta x \end{bmatrix} \tag{29}$$

By solving this matrix system, the temperature at each node can be calculated. This process is implemented through a Python code, as illustrated in Figure 7. The results obtained from the numerical calculations are then plotted against the exact solution of the differential equation in Figure 8. The numerical results, along with the exact solution, are tabulated in Table 4, where the percent error due to approximations in the numeric solution is also presented. As shown in this table, when using 5 cells in this particular case, the error is within the range of 3% or less.

One crucial point to emphasize here is the importance of programming and knowledge of a programming language in the numerical solutions of engineering problems. While the matrix system presented in Eq. (29) is relatively straightforward and could be solved using an engineering calculator, real-world engineering problems often involve considerably more complexity. To illustrate this, consider a simple modification to this problem: increasing the number of cells from 5 to 15. Solving the resulting matrix manually on a calculator would become cumbersome and time-consuming. In contrast, with Python code, this change is effortless. By simply adjusting the value of N to 15 and rerunning the program, we can efficiently obtain the numerical solution for this problem with 15 cells as shown in Fig. 9.

```
import numpy as np
import matplotlib.pyplot as plt

TA, TB, q, L, k, N = 100, 200, 1e6, 0.02, 0.5, 5
dx = L / N
D=k/dx

x = np.linspace(dx / 2, L - dx / 2, N)

a = (np.diag(-2 * np.ones(N)) +
     np.diag(np.ones(N - 1), 1) +
     np.diag(np.ones(N - 1), -1))
a[0, 0], a[-1, -1] = 3D, 3D

b = -q * dx * np.ones(N)
b[0] -= 2 *D* TA
b[-1] -= 2 *D* TB

tempr = np.linalg.solve(a, b)

x_analytic = np.linspace(0, L, 100)
t_analytic = ((TB - TA) / L +
              q / (2 * k) * (L - x_analytic)) * x_analytic + TA

plt.plot(x_analytic, t_analytic, ’k’, label=’Analytic’)
plt.plot(x, tempr, ’ro’, label=’Numeric’)
plt.legend()
plt.locator_params(axis=’x’, nbins=5)
plt.xlabel(’$x(m)$’)
plt.ylabel(’$T(^oC)$’)
```

Figure 7: Python code for FVM solution of 1D diffusion with sourcse

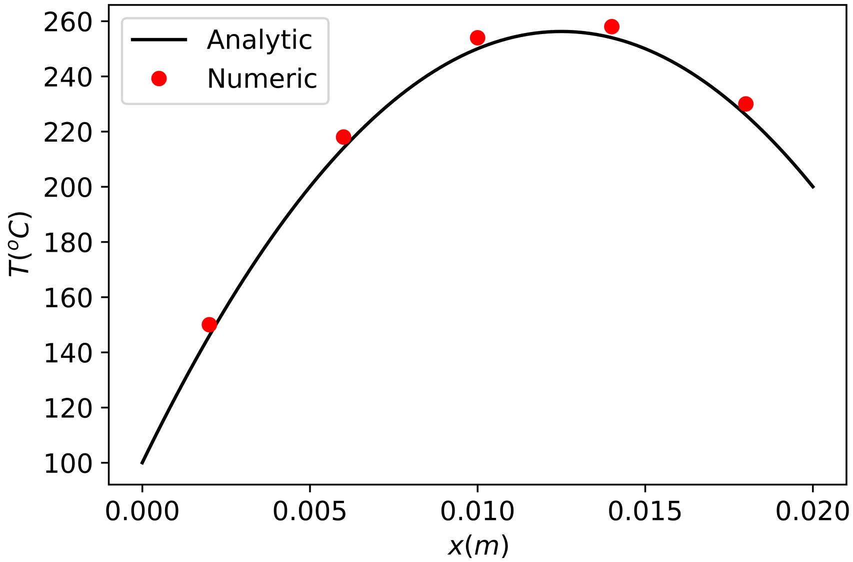

Figure 8: Exact vs FVM (numeric) solution ($i_{\max} = 5$)

Table 4: Comparison of Finite Volume and Exact Solutions ($i_{\max} = 5$)

Node Number	1	2	3	4	5
x (m)	0.002	0.006	0.01	0.014	0.018
Finite Volume Solution	150	218	254	258	230
Exact Solution	146	214	250	254	226
Percentage Error (%)	2.73	1.86	1.60	1.57	1.76

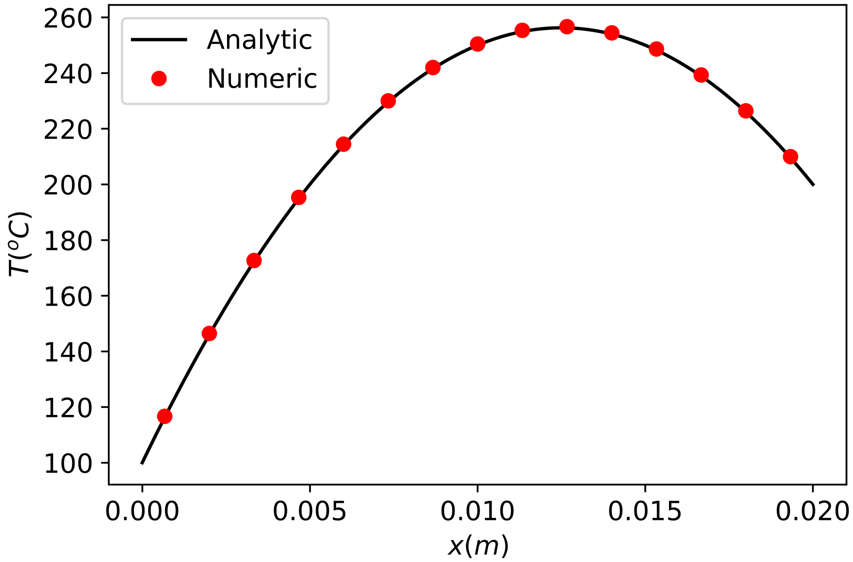

Figure 9: Exact vs FVM (numeric) solution ($i_{\max} = 15$)

Figures 9 also illustrate the importance of mesh refinement in numerical methods. In Fig. 8, with only 5 cells, the numerical solution deviates from the analytical result. However, refining the mesh to 15 cells in Fig. 9 leads to perfect agreement, highlighting the impact of cell size on accuracy. This is a key concept in computational fluid dynamics (CFD): mesh independence. As the number of grid points increases and cell size Δx decreases, the error in the numerical solution shrinks, leading to better agreement with the exact solution. Achieving mesh independence involves systematically refining the mesh and monitoring key parameters until the solution does not change with further refinement of the mesh. This ensures that the chosen mesh is sufficiently fine for reliable CFD simulations.

Example 2: 1D Convection Diffusion Equation

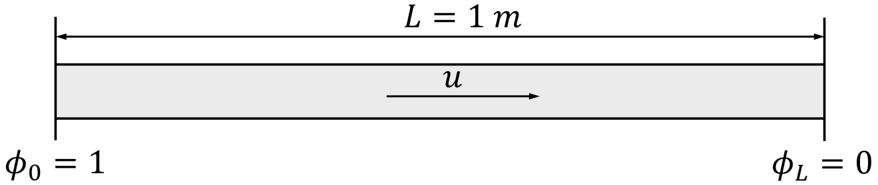

Figure 10: One-dimensional domain for transportation of property ϕ. with no source

Consider the source free 1D convection-diffusion equation in the form

$$\frac{d}{dx}\left(\rho\phi u\right) = \frac{d}{dx}\left(\Gamma\frac{d\phi}{dx}\right) \qquad 0 \leq x \leq 1 \tag{30}$$

which represents the transport of the property ϕ through convection and diffusion in one-dimensional domain shown in Figure 10. The boundary conditions are: $\phi(x=0) = \phi_0 = 1$ and $\phi(x=L) = \phi_L = 0$.

Solve this equation numerically using the Finite Volume Method and the following parameters: $u = 0.1$ m/s, $L = 1.0$ m, $\rho = 1.0$ kg/m^3, $\Gamma = 0.1$ kg/m.s. Compare the results against the exact solution of:

$$\phi(x) = \phi_0 + (\phi_L - \phi_0)\frac{\exp\left(\rho u x/\Gamma\right) - 1}{\exp\left(\rho u L/\Gamma\right) - 1} \tag{31}$$

To solve this problem numerically using the Finite Volume Method with 5 cells, we follow the same steps covered earlier for Eq.12. Comparing Eqs.12 and 30, we see that they are identical except for the absence of a source term in Eq. 30. Additionally, the domain size is set to $L = 1$ m, and the boundary conditions are specified as $\phi_0 = 1$ and $\phi_L = 0$. With these substitutions, the discretized algebraic equations in Table 2 become:

Table 5: Discretized algebraic equations, ($S_\phi = 0$, $\phi_0 = 1$ and $\phi_L = 0$)

Cell Number	Discretized equation
1	$\left(3D + \frac{F}{2}\right)\phi_1 + \left(-D + \frac{F}{2}\right)\phi_2 = F + 2D$
2	$\left(-D - \frac{F}{2}\right)\phi_1 + 2D\phi_2 + \left(-D + \frac{F}{2}\right)\phi_3 = 0$
3	$\left(-D - \frac{F}{2}\right)\phi_2 + 2D\phi_3 + \left(-D + \frac{F}{2}\right)\phi_4 = 0$
4	$\left(-D - \frac{F}{2}\right)\phi_3 + 2D\phi_4 + \left(-D + \frac{F}{2}\right)\phi_5 = 0$
5	$\left(-D - \frac{F}{2}\right)\phi_4 + \left(3D - \frac{F}{2}\right)\phi_5 = 0$

The discretized equations in Table 5 can be put in the matrix as follows:

$$\begin{bmatrix} 3D + F/2 & -D + F/2 & 0 & 0 & 0 \\ -D - F/2 & 2D & -D - F/2 & 0 & 0 \\ 0 & -D - F/2 & 2D & -D + F/2 & 0 \\ 0 & 0 & -D - F/2 & 2D & -D + F/2 \\ 0 & 0 & 0 & -D - F/2 & 3D + F/2 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} F + 2D \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{32}$$

By solving this matrix system, the property ϕ at each node can be calculated. This process is again implemented through Python code, as illustrated in Figure 11. The results obtained from the numerical calculations are then plotted against the exact solution of the differential equation in Figure 12.

These examples provide a solid foundation for students to grasp the fundamentals of the Finite Volume Method (FVM). Notably, several additional examples are covered in class before transitioning to applications with commercial CFD software, Fluent. These include an example demonstrating the upwind scheme for obtaining a converged solution and another exploring the solution of the 2D diffusion equation. These are omitted here to maintain brevity.

Two homework sets are assigned to test student comprehension of the FVM. These problems mirror those covered in class and require students to use Python for programming. Recognizing the diverse Python experience among students, the code structure is provided, with students only required to modify specific sections relevant to the given problem. It's important to emphasize that Python use in this CFD class is a unique and contemporary feature. This is because many commercial CFD software packages allow user customization and optimization, which is often done in Python. Therefore, having students write simple Python solutions for differential equations prepares them for future endeavors as CFD engineers in industry, where basic Python knowledge is increasingly valuable.

```python
import numpy as np
import matplotlib.pyplot as plt


# Constants
rho, u, gamma, L = 1, 0.1, 0.1, 1
Fi0, FiL = 1, 0
N = 5
dx = L / N
F = rho * u
D = gamma / dx



# Analytical solution
x_analytic = np.linspace(0, L, 100)
Fi_analytic = (
    Fi0 + (FiL - Fi0) * (np.exp(F * x_analytic / gamma) - 1)
    / (np.exp(F * L / gamma) - 1)
)


# Numeric solution
x = np.linspace(dx / 2, L - dx / 2, N)
a = (
    np.diag((2 * D) * np.ones(N))
    + np.diag((-D - F / 2) * np.ones(N - 1), -1)
    + np.diag((-D + F / 2) * np.ones(N - 1), 1)
)
a[0, 0] += D + F / 2
a[-1, -1] += D - F / 2

b = np.zeros(N)
b[[0, -1]] = [F + 2 * D, 0]

Fi = np.linalg.solve(a, b)


# Plotting
plt.plot(x_analytic, Fi_analytic, 'k', label='Analytical')
plt.plot(x, Fi, '--ro', label='Numerical')
plt.legend(loc='best', shadow=True, fontsize='x-large')
plt.xlabel('x (m)')
plt.ylabel('$\phi$')
```

Figure 11: Python code for FVM solution of 1D convection diffusion equation

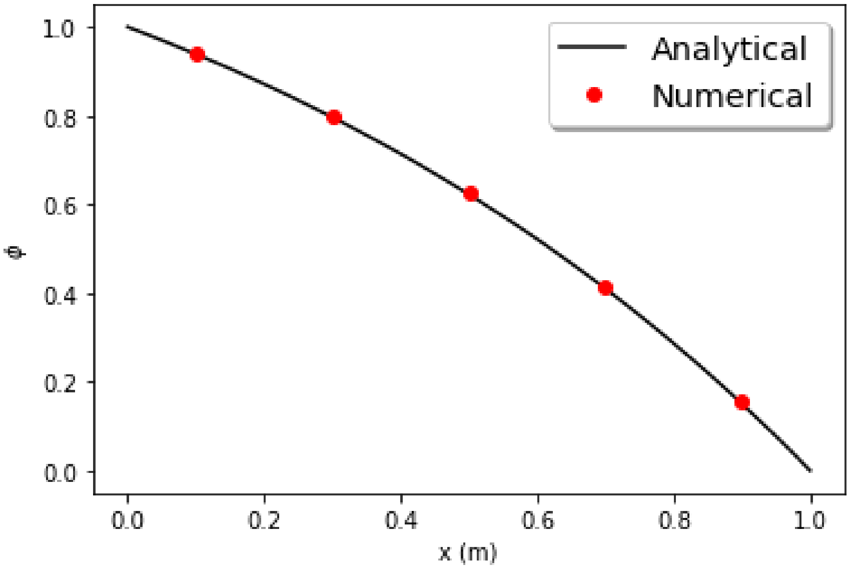

Figure 12: Exact vs FVM (numeric) solution ($N = 5$)

Applications of CFD Using Ansys Fluent

This CFD course stands out for its focus on real-world applications using the commercial code Ansys Fluent, comprising 60$\%$ of the class. Before diving into the commercial code, students are introduced to the fact that CFD codes, like Ansys Fluent, follow steps similar to those discussed in class for solving conservation equations of fluid flow using the finite volume method. Specifically, the process involves mesh generation, discretization of relevant equations using the finite method, and solving the discretized equations to obtain flow properties at each node.

A key distinction between Fluent and the class approach lies in how the solution to discretized equations is achieved. In class, matrix algebra is utilized, while commercial codes opt for iterative schemes due to their computational efficiency.

The class meets twice a week for 80 minutes each session. During the first 80 minutes, the class delves into an in-depth discussion of the selected problem. In the subsequent 80 minutes, students convene in a computer lab equipped with the Ansys Fluent CFD code. They watch a pre-recorded video illustrating the simulation of the particular problem, then recreate the simulation and submit it as a quiz. The instructor is available during this lab time to address any queries from students. Appendix-1 includes links to video recordings for simulations of each problem covered in the CFD class, and a brief discussion of two of these problems follows.

Example 1: Inviscid Flow in a supersonic wind tunnel

The application section of the course begins with the simulation of inviscid flow inside a supersonic wind tunnel designed for testing air vehicles capable of exceeding the speed of sound (Fig. 13).

Concentrating on the 2D midsection of such a tunnel as shown in Fig. 14, the flow at the inlet of the wind tunnel comes from a reservoir where the pressure and temperature are P_0 and T_0

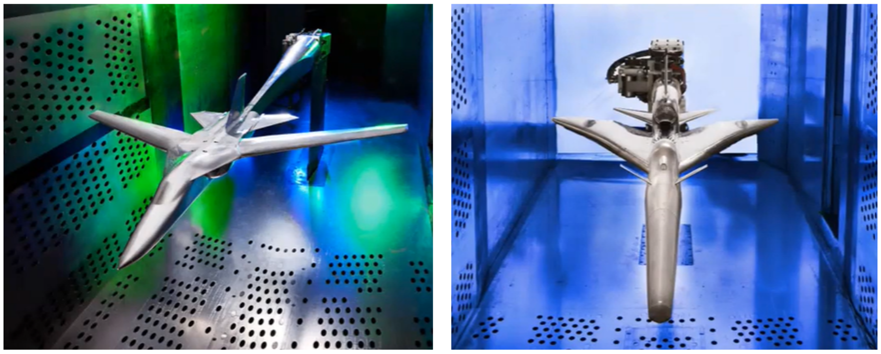

Figure 13: supersonic wind tunnel[8]

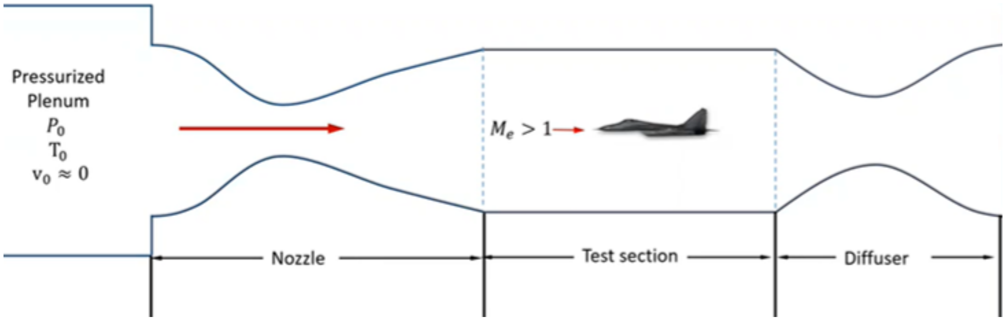

Figure 14: 2D cross section of a supersonic wind tunnel

respectively. The reservoir's large size ensures minimal flow velocity within, making P_0 and T_0 the stagnation pressure and temperature. The flow then proceeds through a convergent-divergent nozzle, expanding to supersonic flow before entering the test section. Further information on the workings of supersonic wind tunnels can be found on the NASA website[9].

For this exercise, the simulation focuses on the flow inside the converging/diverging nozzle preceding the test section shown in Fig. 15. Only half of the nozzle is simulated due to symmetry. The simulation assumes inviscid flow, as viscous effects are confined to narrow regions along the walls and have a limited impact on predicting flow properties within the test section.

The choice of this exercise as the initial class assignment is rooted in its status as a classic textbook problem with an analytical solution. Consequently, students can compare CFD results against analytical outcomes to assess the accuracy of their simulations. Flow properties for comparison include the development of Mach number and static pressure across the convergent-divergent nozzle, represented by the equations:

$$\left(\frac{A}{A_t}\right)^2 = \frac{1}{M^2}\left[\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M^2\right)\right]^{\frac{\gamma+1}{\gamma-1}} \tag{33}$$

$$\frac{p}{P_0} = \left(1+\frac{\gamma-1}{2}M^2\right)^{-\frac{\gamma}{\gamma-1}} \tag{34}$$

This problem can be solved numerically using Ansys Fluent by applying the same steps that were

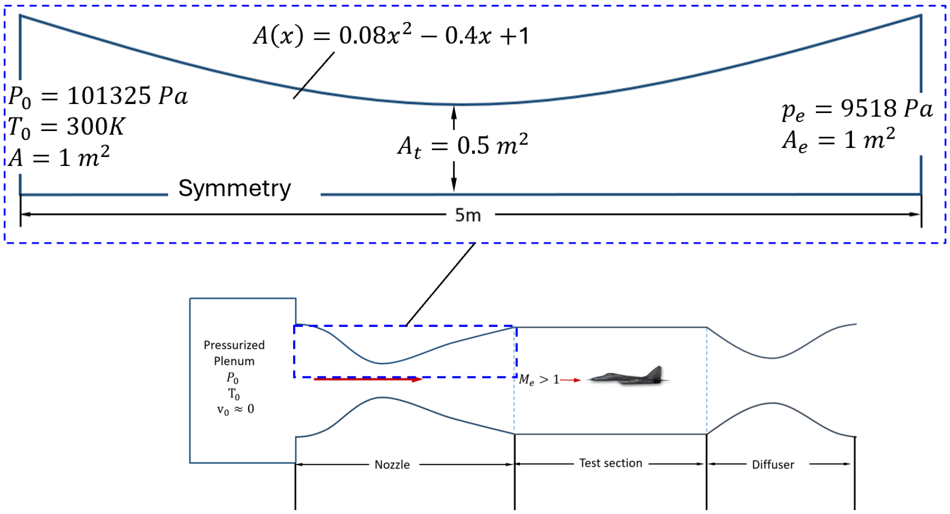

Figure 15: Converging diverging section of the nozzle simulated using CFD

discussed earlier when the Finite Volume Method was discussed, which are:

Step 1: Mesh Generation

The first step is dividing the domain into smaller subdomains, which is also known as the mesh generation. This is done in Ansys Meshing, and the resultant mesh is shown in Fig. 16.

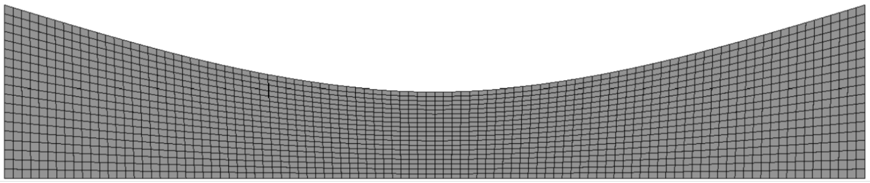

Figure 16: Mesh generation

Step 2: Deriving the integral form of the differential equations that need to be solved

The second step is to ensure that students understand which equations Ansys Fluent will solve. Because this is a 2D, steady, inviscid flow problem, the conservation equations presented in Eqs (1)-(5) undergo significant simplification and adopt the following form:

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{35}$$

$$\frac{\partial(\rho u^2)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} = -\frac{\partial p}{\partial x} \tag{36}$$

$$\frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2)}{\partial y} = -\frac{\partial p}{\partial y} \tag{37}$$

$$c_V \left[\frac{\partial(\rho u T)}{\partial x} + \frac{\partial(\rho v T)}{\partial y} \right] = -p \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \tag{38}$$

The above equations together with the state equation of

$$p = \rho R T \tag{39}$$

are integrated over each control volume and integral form of these equations are obtained.

Steps 3-5: Discretization and solution of the equations

Students are informed that in the subsequent stages, Ansys Fluent utilizes the finite volume method to discretize the governing equations. The resulting set of algebraic equations is then solved using an iterative scheme. Figures (17)-(20) display some results obtained from the solution of this problem by Ansys Fluent using the Finite Volume Method.

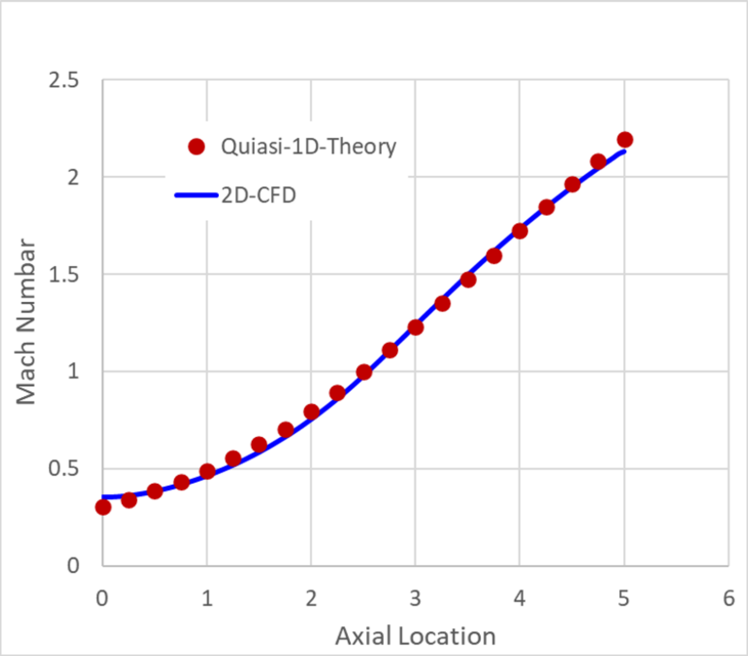

Figure 17: Mach Number Plot at the Symmetry-line (CFD vs 1D theory)

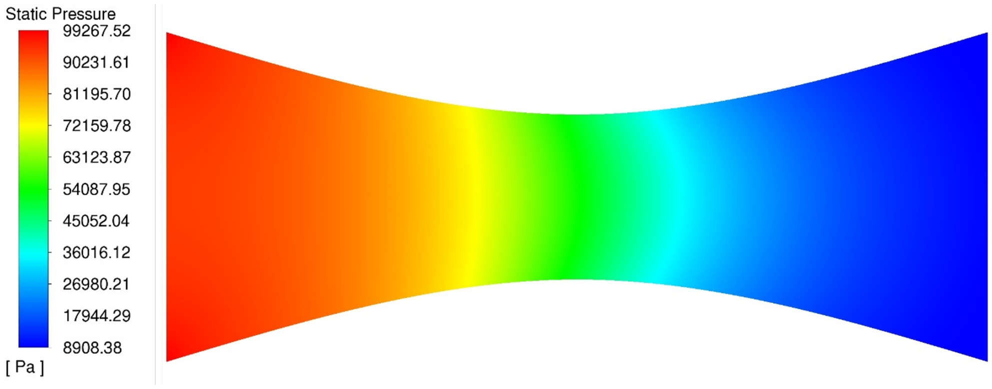

Figure 18: Contours of static pressure

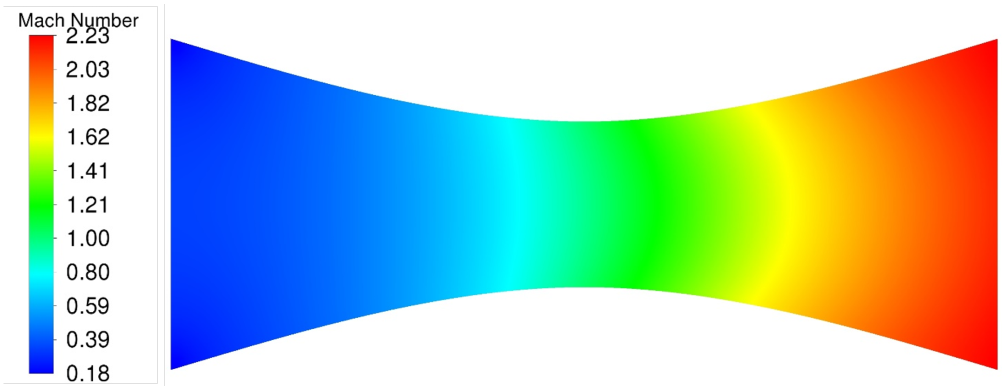

Figure 19: Contours of Mach Number

Example 2: Laminar Flow around a 2D Cylinder

For the second problem, we simulate 2D flow around a cylinder at different Reynolds Numbers. The Reynolds Numbers selected are 2, 10, 20, and 40. The cylinder diameter is 1 meter, fluid density is 1 kg/m^3, and dynamic viscosity is 0.05 kg/(m·s). The velocity is varied to achieve the desired Reynolds Number. It is noted that this problem is modified from an Ansys innovation course[10]. The problem schematic and boundary conditions are shown in Figure (20).

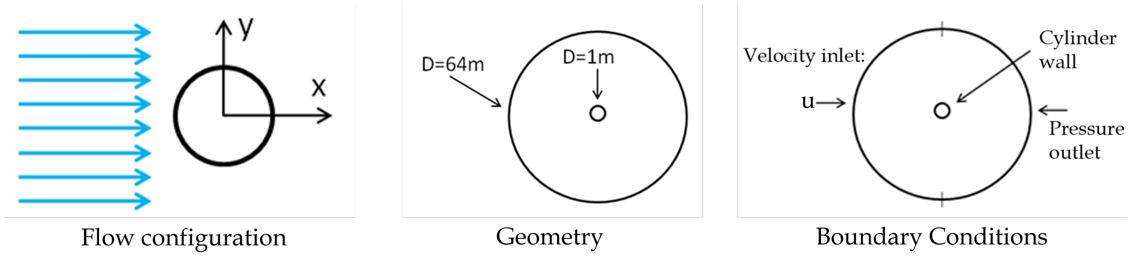

Figure 20: Problem schematic and boundary conditions (Modified from[10])

Step 1: Mesh Generation

The meh generated for this problem using Asnys meshing is shown in Fig. 21.

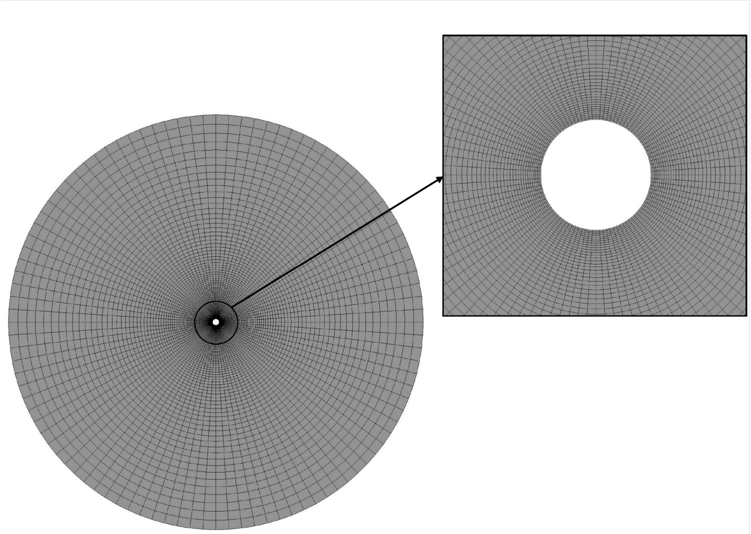

Figure 21: Mesh generation

Step 2: Deriving the integral form of the differential equations that need to be solved

Similar to the previous problem, the second step is to ensure that students understand which equations Ansys Fluent will solve. This problem involves 2D, steady, incompressible, laminar flow. Additionally, there is no need to solve the energy equation since temperature is not of interest, and thus, the problem can be assumed isothermal. Under these assumptions, the conservation equations presented in Eqs (1)-(5) take on the following form:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{40}$$

$$\rho \left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} \right) = -\frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{41}$$

$$\rho \left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} \right) = -\frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \tag{42}$$

Above equations are integrated over each control volume and integral form of these equations are obtained.

Steps 3-5: Discretization and solution of the equations

In the subsequent steps, Ansys Fluent utilizes the finite volume method to discretize the integral form of the equations presented above. The resulting set of algebraic equations is then solved using an iterative scheme. It is noted that for this case, by solving the above three equations, the commercial code will provide three flow properties at each node: pressure p, x-component of velocity u, and y-component of velocity v. These properties can be used to calculate other parameters of interest, such as drag force or drag coefficient. Figure (22) displays students' work, depicting pathlines colored by velocity magnitude, while Table 6 presents students' work comparing the drag coefficient as predicted by CFD versus experimental data for different Reynolds numbers.

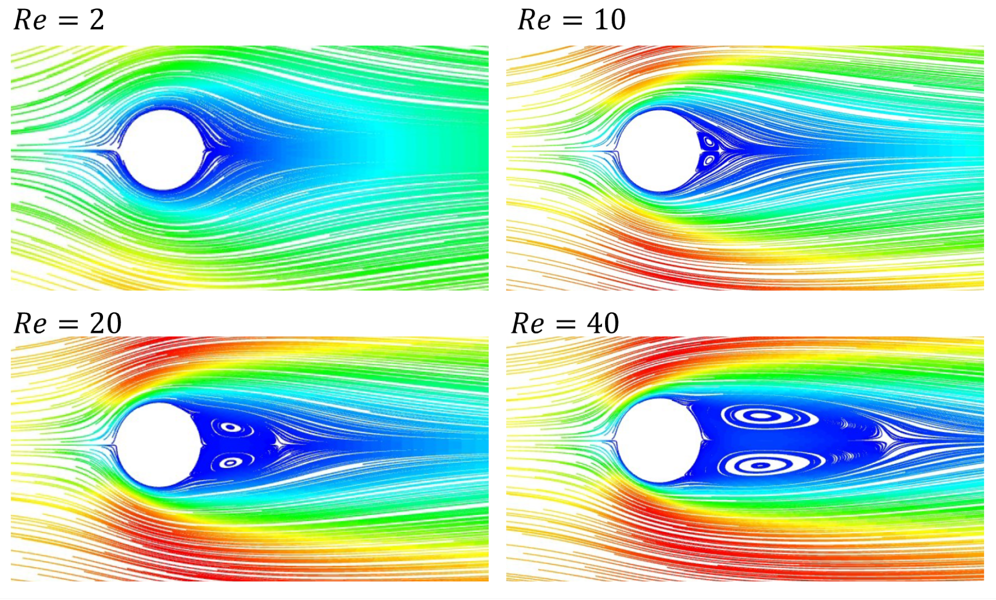

Figure 22: Pahlines around 2D cylinder as function of Reynolds number

Table 6: Comparison of Drag Coefficients

Reynolds Number	Coefficient of Drag (CFD)	Experimental	Percent Difference
2	7.65	7.3	4.8%
10	2.82	2.8	0.9%
20	2.04	2.0	2.1%
40	1.53	1.5	2.1%

Beyond the two exemplar cases presented above, several other cases are covered in the full course. These additional cases include:

- CFD simulation of Turbulent Flow in a mixing elbow
- Convective Cooling of a heat sink
- CFD simulation of unsteady flow around a 2D Cylinder
- Design and optimization of a high-speed compressor stage using Ansys turbo tools and CFD

While the details of these cases are not presented here, recordings are accessible through the links provided in the Appendix-1.

It is important to note that before delving into the CFD solutions for turbulent flow cases, two lectures are dedicated to providing a foundational understanding of turbulent flows. These lectures cover:

- Introduction to turbulent flow

- Using the Reynolds number to classify laminar vs. turbulent flow
- Modeling turbulent flow in CFD:
  - Selecting an appropriate turbulence model
  - Choosing methods for near-wall flow modeling
  - Specifying turbulence boundary conditions at inlets

Finally, in the lectures, the importance of validation and verification, the sensitivity of simulations to turbulence models, and the significance of mesh-independent studies in CFD are discussed. This is accomplished by exploring various technical publications in these areas. For instance, the importance of validation and verification is highlighted using resources such as the AIAA guide on validation and clarification of CFD[11], and students are introduced to the NASA website[8].

Additionally, the sensitivity of solutions to turbulence models and the necessity of mesh-independent studies are explored by discussing publications authored by the instructor[12,13,14,15] and other researchers[16,17]. These documents are also provided for students to review independently.

Conclusion

The paper has presented the design and implementation of a unique ”Applications of Computational Fluid Mechanics” course tailored for senior undergraduate students. Its innovative structure integrates theoretical foundations with a practical, ”tutorial-based” experience, equipping students with both the knowledge and skills necessary to tackle real-world fluid mechanics and heat transfer challenges.

The course strikes a balance between theoretical fundamentals, covered through numerical solution implementation in Python, and hands-on experience using industry-standard Ansys Fluent software. By incorporating Python in the introductory phase, students are prepared for future industry endeavors, given Python’s increasing utilization for customization and optimization in commercial CFD packages. Additionally, the course’s second part adopts a unique problem-solving approach where students actively replicate pre-recorded tutorials, fostering deeper understanding and engagement.

The comprehensive and student-centered approach of this course prepares future engineers with the critical skills and software proficiency required to address contemporary fluid mechanics challenges. By integrating Python usage and active learning strategies, this course serves as a model for equipping students with essential CFD skills, contributing to the advancement of engineering education.

References

[1] Kendrick Aung. Design and implementation of an undergraduate computational fluid dynamics (cfd) course. In *2003 Annual Conference*, pages 8–367, 2003.

[2] Rajesh Bhaskaran. Strategies for the integration of computer-based simulation technology into the engineering curriculum. In *2007 Annual Conference & Exposition*, pages 12–1303, 2007.

[3] Steven AE Miller. A contemporary course on the introduction to computational fluid dynamics. *International Journal of Mechanical Engineering Education*, 48(4):315–334, 2020.

[4] Xiangdong Li and Sherman CP Cheung. A learning-centered computational fluid dynamics course for undergraduate engineering students. *International Journal of Mechanical Engineering Education*, 2024.

[5] Desmond Adair, Zhumabay Bakenov, and Martin Jaeger. Building on a traditional chemical engineering curriculum using computational fluid dynamics. *Education for Chemical Engineers*, 9(4):e85–e93, 2014.

[6] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education, 2007.

[7] John David Anderson, Gérard Degrez, Erik Dick, and Roger Grundmann. *Computational fluid dynamics: an introduction*. Springer Science & Business Media, 2013.

[8] NASA Glenn Research Center. Overview of cfd verification validation, . URL `https://www.grc.nasa.gov/www/wind/valid/tutorial/overview.html`. Accessed February 8, 2024.

[9] NASA Glenn Research Center. Facilities — glenn research center — nasa, . URL `https://www1.grc.nasa.gov/facilities/`.

[10] Ansys Inc. Steady-flow past a cylinder. URL `https://courses.ansys.com/index.php/courses/steady-flow-past-a-cylinder/`. Accessed February 7, 2024.

[11] Computational Fluid Dynamics Committee. Guide: Guide for the verification and validation of computational fluid dynamics simulations (aiaa g-077-1998 (2002)), 1998.

[12] Mehmet N Sarimurat and Thong Q Dang. Shock management in diverging flow passages by blowing/suction, part 2: applications. *Journal of Propulsion and Power*, 28(6):1230–1242, 2012.

[13] Mehmet N Sarimurat and Thong Q Dang. An analytical model for boundary layer control via steady blowing and its application to naca-65-410 cascade. *Journal of Turbomachinery*, 136(6):061011, 2014.

[14] Mehmet N Sarimurat. Assessment of a correlation based transitional model for compressor cascade performance prediction.

[15] Mehmet Sarimurat. A comprehensive analysis of flow blowing and its effects on change in main flow conditions and loss generation in compressible flows. *Aerospace Science and Technology*, 130:107905, 2022.

[16] Robin Blair Langtry, FR Menter, SR Likki, YB Suzen, PG Huang, and S Vo¨ lker. A correlation-based transition model using local variables: Part ii—test cases and industrial applications. In *Turbo Expo: Power for Land, Sea, and Air*, volume 41693, pages 69–79, 2004.

[17] Robin Langtry, Janusz Gola, and Florian Menter. Predicting 2d airfoil and 3d wind turbine rotor performance using a transition model for general cfd codes. In *44th AIAA aerospace sciences meeting and exhibit*, page 395, 2006.

Appendix-1: Link to the pre-recorded videos on CFD applications

1. **Application 1:** CFD simulation of a supersonic wind tunnel
   Link: `https://video.syr.edu/media/t/1_tfphjkw8`

2. **Application 2:** CFD Simulation of a flow around 2D Cylinder
   Link: `https://video.syr.edu/media/t/1_ovwrgu1t`

3. **Application 3:** CFD Simulation of turbulent flow in a mixing elbow
   Link: `https://video.syr.edu/media/t/1_phu3e71t`

4. **Application 4:** Convective cooling of a heat sink
   Link: `https://video.syr.edu/media/t/1_phu3e71t`

5. **Application 5:** Aeromechanical-design-of-a-compressor-stage
   Link: `https://video.syr.edu/media/t/1_whegz752`