DEVELOPMENT AND VALIDATION OF A MATLAB SOLVER FOR TURBULENT FLOW OVER A SQUARE BLOCK USING THE WILCOX (2006) k–ω MODEL

Alperen SOZER[1], **Daniel HEEREBOUT**[2], **Siraan MUNIYANDI**[3]

[1]student ID: 2346664 E-mail: a.sozer@student.tue.nl

[2]student ID: 2015900 E-mail: d.a.heerebout@student.tue.nl

[3]student ID: 2292580 E-mail: s.muniyandi@student.tue.nl

GROUP NUMBER: 12

ABSTRACT

This report presents a numerical investigation of two-dimensional, incompressible turbulent flow past a 2D square block, which acts as a bluff body, producing large-scale flow separation and periodic vortex shedding. The primary objective is to analyze key flow features, including the time-averaged velocity distribution, the structure of the recirculation zone, and the unsteady vortex shedding phenomenon downstream of the block. The Reynolds-Averaged Navier-Stokes (RANS) equations are solved using a custom MATLAB code. Turbulence is modeled using the "Standard" Wilcox (2006) $k-\omega$ model with a High-Re wall function approach, which simplifies the near-wall treatment.

The study includes systematic grid and time-step independence analyses to verify discretization accuracy. The resulting time-averaged velocity profiles and turbulent kinetic energy distributions demonstrate that the implemented model successfully captures the essential qualitative features of the separated flow wake, including the wake structure and recirculation zone. The performance of the turbulent model is assessed against available literature, highlighting its ability to reproduce key flow characteristics inherent to this geometry.

NOMENCLATURE

- U Mean flow velocity $[m/s]$
- u' Velocity fluctuations due to turbulence $[m/s]$
- p Pressure field $[Pa]$
- ρ Fluid density $[kg/m^3]$
- μ Molecular (dynamic) viscosity $[Pa \cdot s]$
- ν Kinematic viscosity $[m^2/s]$
- τ_{ij} Reynolds stress (modeled, via Boussinesq) $[N/m^2]$
- k Turbulent kinetic energy $[m^2/s^2]$
- ω Specific dissipation rate $[1/s]$
- $\tilde{\omega}$ Eddy-viscosity limiter [-]
- S_{ij} Mean strain-rate tensor $[1/s]$
- $\tilde{S_{ij}}$ Trace-free mean strain-rate tensor $[1/s]$
- Ω_{ij} Mean rotation-rate tensor $[1/s]$
- P Production of turbulent kinetic energy $[m^3/s^2]$
- X_r Reattachment length of the primary recirculation zone $[m]$
- D Block size $[m]$
- Re Reynolds number [-]
- M Mach number [-]
- y_p Distance from wall to the first cell center $[m]$
- C_w Constant for near-wall
- ϕ General transported variable
- Γ_ϕ Effective diffusion coefficient for ϕ $[\mathrm{m}^2/\mathrm{s}]$
- S_ϕ Total source/sink term for ϕ
- a_P, a_{nb} Finite volume matrix coefficients
- S_u Explicit part of the source term
- S_p Implicit part of the source term (Varies)
- F_f Mass flux across face f (convection) $\mathrm{kg}/(\mathrm{m}^2 \cdot \mathrm{s})$
- $D_{f,\phi}$ — Effective diff. conductance across face f $[\mathrm{kg}/\mathrm{s}]$
- ΔF_ϕ Transient term contribution $[\mathrm{kg}/\mathrm{s}]$
- p' Pressure correction variable $[\mathrm{N}/\mathrm{m}^2]$
- Δt Pseudo-time step $[\mathrm{s}]$
- V_P Volume of the computational cell P $[\mathrm{m}^3]$
- A_f Area of the control volume face f $[\mathrm{m}^2]$
- y^+ Dimensionless wall distance [-]
- I Turbulence intensity [-]
- Δ_τ Pseudo time steps $[\mathrm{s}]$

1. INTRODUCTION

Flow past a solid body, such as a square block, is a canonical problem in fluid dynamics and serves as a fundamental benchmark case for validating Computational Fluid Dynamics (CFD) codes. The simple geometry produces a rich set of complex flow phenomena, including flow separation at the sharp corners of the block, the formation of recirculation zones (or "bubbles") and eventual flow reattachment. These features are of great practical importance in engineering applications, such as in heat exchangers, electronic cooling, and general aerodynamics, where separation and reattachment significantly influence heat transfer and mixing characteristics .

This report presents a numerical investigation of two-dimensional, incompressible turbulent flow past a 2D square block placed within an air channel. The flow is simulated using a custom, self-implemented Reynolds-Averaged Navier-Stokes (RANS) solver developed in MATLAB.

1.1 Project objectives

The primary objective of this project is to analyze the complex, separated turbulent flow downstream of a square bluff body using a self-implemented MATLAB solver. The study

focuses on evaluating the predictive capability of the Wilcox (2006) k–ω turbulence model for flows involving separation and recirculation. To ensure numerical reliability, additional studies were carried out to verify grid and time-step independence. The specific objectives are:

1. **Model implementation and solver verification:** To successfully implement the High-Re "Standard" Wilcox (2006) $k-\omega$ turbulence model and verify the stability of the custom Finite Volume solver using the SIMPLE algorithm.
2. **Numerical independence studies:** To systematically conduct grid and time-step independence studies to establish an optimal, accurate, and computationally efficient mesh resolution and temporal discretization for the transient RANS simulation.
3. **Qualitative flow feature analysis:** To qualitatively analyze and demonstrate the principal hydrodynamic features of the flow wake, specifically:
    - The existence and structure of the recirculation zone (reverse flow region) behind the square block using time-averaged longitudinal velocity profiles.
    - The progressive development of the turbulent shear layer and boundary layer thickening along the block's surfaces.
    - The distribution of turbulent kinetic energy (k) in the wake region, confirming elevated turbulence levels consistent with flow separation.

1.2 Key research questions

The key research questions addressed in this report were:

- How accurately does the Wilcox (2006) k–ω model predict the time-averaged velocity field and recirculation region behind the square block compared with available literature data?
- How do mesh resolution and time-step size influence the stability and accuracy of the computed velocity profiles?
- Can the custom RANS solver qualitatively capture the characteristic features of bluff-body aerodynamics, including flow separation, recirculation, and wake recovery, as demonstrated by velocity profiles and flow field visualizations?
- Does the implemented solver achieve consistent convergence behavior across different grid and temporal resolutions?

2. COMPARISON OF TURBULENCE MODELS

In the Reynolds-Averaged Navier-Stokes (RANS) approach, the two-equation family of turbulence models, primarily $k-\varepsilon$ and $k-\omega$ variants, is widely employed. These models introduce transport equations for the turbulent kinetic energy k and a scale-determining variable to provide the turbulent eddy viscosity μ_t for closure via the Boussinesq hypothesis [5] [7].

A comparative assessment of the primary models is necessary to justify the selection for the flow regime involving separation and adverse pressure gradients inherent to the bluff body geometry. Further analysis of advantages and disadvantages of each model is shown in Appendix B.

2.1 The k- epsilon model

The standard $k-\varepsilon$ model is widely used due to its robustness and computational economy. It solves transport equations for k and the turbulent dissipation rate ε [2].

- **Governing equations:** Equations are implemented for k and ε to determine the Eddy viscosity $\mu_t = \rho C_\mu k^2/\varepsilon$ [7].
- **Benefits:** It is robust, computationally inexpensive, and performs well in free shear flows and wall-bounded flows with small-mild pressure gradients [7]. It is relatively insensitive to freestream conditions.
- **Limitations:** This model is known to perform poorly in flows with strong adverse pressure gradients or separation. It often overestimates turbulent mixing, thus suppressing or delaying flow separation and reattachment. Furthermore, its standard form requires the use of wall functions for the near-wall region, which limits accuracy when resolving the viscous sublayer is critical [7] [8] .

2.2 The k - omega family

The $k-\omega$ family, originally developed by D. Wilcox, addresses several limitations of the $k-\varepsilon$ model by solving for the specific dissipation rate ω (with dimensions $1/s$) instead of ε [6].

Original $k-\omega$ Model (1988)

Similar to the $k-\varepsilon$ model, it is a two-equation model. That means in addition to the conservation equations, it solves two transport equations (PDEs), which account for the history effects like convection and diffusion of turbulent energy.

- **Benefits:** It can be integrated directly to the wall without requiring damping functions, making it excellent for boundary-layer flows, especially those with adverse pressure gradients [8].
- **Limitations:** The main drawback is its extreme sensitivity to freestream or inlet boundary conditions for ω, which can lead to unpredictable results in external or open-channel flows [9]. It also lacks a robust mechanism to cap eddy viscosity in stagnant regions [4].

The "Standard" $k-\omega$ model (2006)

This model is a refinement of the original $k-\omega$ formulation, incorporating critical adjustments to improve its general applicability and robustness.

- **Refinements:** This version features revised model constants and, critically, includes a cross-diffusion term in the ω-equation (similar to $k-\varepsilon$ transformations) to improve behavior in the outer portion of boundary layers and free shear flows. It also features a shear stress limiter to prevent the over-prediction of eddy viscosity in stagnation regions [9].
- **Performance:** It provides a better balance of near-wall fidelity and far-field robustness compared to the original, yielding more accurate skin friction and separation predictions than the standard $k-\varepsilon$.

The $k-\omega$ SST model

The SST model is a hybrid approach designed to capture the best aspects of both $k-\varepsilon$ and $k-\omega$ models.

- **Mechanism:** It employs a blending function that smoothly transitions the calculation from the $k-\omega$ formulation near the wall (for accuracy) to the $k-\varepsilon$ formulation in the freestream (for reduced sensitivity) [4].
- **Benefits:** It is the most universally applicable, retaining accurate near-wall resolution while avoiding freestream sensitivity. Its shear stress limiter significantly improves prediction of flows involving separation and adverse pressure gradients [8].
- **Limitations:** It is the most complex of the models to implement, requiring additional blending functions and terms .

2.3 Model selection

The flow past a square block is characterized by flow separation, strong adverse pressure gradients and significant unsteady vortex shedding. Therefore, a model capable of accurately predicting separation and resolving the near-wall region is mandatory, ruling out the standard $k-\varepsilon$ model.

The $k-\omega$ (2006) and $k-\omega$ SST models are the strongest candidates. While the SST model is widely regarded as the most accurate general-purpose model for separated flow, the Wilcox (2006) standard $k-\omega$ model was selected for the following reasons:

1. **Completeness over original $k-\omega$:** It incorporates key mathematical and empirical refinements (cross-diffusion, stress limiter) necessary to mitigate the deficiencies of the original $k-\omega$ model in free shear layers and stagnant regions, making it a robust choice for the simulation complexity [9].
2. **Computational efficiency over SST:** The Wilcox (2006) model achieves high fidelity in the critical near-wall and separation regions with a less complex formulation than the hybrid SST model [8]. For a self-implemented, custom MATLAB solver, this reduced complexity translates to lower computational overhead and reduced computational time per iteration compared to the SST blending formulation.

Thus, the Wilcox (2006) $k-\omega$ model provides an optimal balance between the required accuracy for this class of separated turbulent flow and the need for computational efficiency within the constraints of the self-coded solver.

3. THEORETICAL BACKGROUND

The accurate computational modeling of fluid motion is founded upon the fundamental principles of mass and momentum conservation. This chapter introduces the differential equations used to mathematically describe the turbulent flow through the defined geometry.

Since the flow is characterized by chaotic fluctuations across a wide range of length and time scales, the instantaneous Navier-Stokes equations are deemed computationally intractable. A time-averaging approach is therefore employed to derive the Reynolds-Averaged Navier-Stokes (RANS) equations, which govern the mean flow variables. Subsequently, the underlying numerical approach utilizes the Finite Volume Method (FVM) with the SIMPLE algorithm for pressure–velocity coupling.

3.1 Governing assumptions

To simplify the Navier-Stokes equations, the following core assumptions are applied to the flow model:

- **Incompressible flow:** The fluid density (ρ) is assumed constant throughout the domain ($\partial\rho/\partial t = 0$). This is valid for air flow at low Mach numbers ($M \ll 0.3$), eliminating the need for an equation of state.
- **Isothermal flow:** The fluid temperature is assumed constant, meaning the energy equation is decoupled from the momentum equations. This simplifies the governing system by eliminating the need to calculate heat transfer or buoyancy effects.
- **Two-dimensional flow:** The simulation is simplified to two dimensions (2D), assuming the flow is uniform and fully contained along the spanwise direction.
- **Constant viscosity:** The molecular viscosity (μ), together with the fluid density (ρ) are considered constant properties of the system. Not so for μ_t, as it necessary to make the system non-linear.

3.2 Reynolds-Averaged Navier-Stokes (RANS)

Turbulent flow variables (velocity $\mathbf{U}$ and pressure p) are decomposed into a time-averaged mean component ($\overline{\mathbf{U}}$, $\overline{p}$) and a fluctuating component ($\mathbf{u}'$, p'), satisfying the Reynolds decomposition: $\mathbf{U} = \overline{\mathbf{U}} + \mathbf{u}'$ and $p = \overline{p} + p'$. Applying this decomposition to the instantaneous Navier–Stokes equations and averaging in time yields the RANS equations for incompressible, constant-property flow.

Continuity equation

The mean continuity equation is identical to the instantaneous form for incompressible flow:

$$\frac{\partial \overline{U}_i}{\partial x_i} = 0$$

Momentum equation

The mean momentum equation incorporates the effects of turbulence through the Reynolds stress tensor, $-\overline{\rho u_i' u_j'}$:

$$\rho\frac{\partial U_i}{\partial t} + \rho U_j \frac{\partial U_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \rho\overline{u_i' u_j'}\right]$$

where,

- U = mean flow velocity
- μ = molecular viscosity
- $\rho U_j \frac{\partial \overline{U}_i}{\partial x_j}$ = Convection term
- $\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right)\right]$ = Diffusion term

3.3 Turbulence Closure: Boussinesq Hypothesis and k–ω (2006) Model

The Reynolds stress tensor is modeled using the Boussinesq hypothesis, which relates the Reynolds stresses (τ_{ij}) to the mean strain rate through an eddy (turbulent) viscosity (μ_t):

$$-\rho\overline{u_i' u_j'} = \mu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) - \frac{2}{3}\rho k \delta_{ij}$$

where k is the turbulent kinetic energy and δ_{ij} is the Kronecker delta.

The **Standard Wilcox (2006) k–ω Model** determines μ_t and closes the system by introducing two transport equations for the turbulent kinetic energy (k) and the specific dissipation rate (ω). The turbulent viscosity is defined as [9]:

$$\mu_t = \frac{\rho k}{\omega} \tag{1}$$

Transport Equation for Turbulent Kinetic Energy (k)

$$\rho \frac{\partial k}{\partial t} + \underbrace{\rho U_j \frac{\partial k}{\partial x_j}}_{\text{Convection}} = \underbrace{\frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right]}_{\text{Diffusion}} + P_k - \underbrace{\rho \beta^* k \omega}_{\text{Destruction}}$$

where P_k is the production of turbulent kinetic energy, defined as:

$$P_k = \tau_{ij} \frac{\partial U_i}{\partial x_j} = \mu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \frac{\partial U_i}{\partial x_j} \tag{2}$$

Transport Equation for Specific Dissipation Rate (ω)

$$\rho \frac{\partial \omega}{\partial t} + \underbrace{\rho U_j \frac{\partial \omega}{\partial x_j}}_{\text{Convection}} = \underbrace{\frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\omega} \right) \frac{\partial \omega}{\partial x_j} \right]}_{\text{Diffusion}} + \underbrace{\alpha \frac{\omega}{k} P_k}_{\text{Product.}} - \underbrace{\rho \beta \omega^2}_{\text{Destruct.}} + D_w$$

where D_w is the cross-diffusion term, defined as:

$$D_w = \frac{\sigma_d}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \tag{3}$$

Model Constants (Wilcox 2006)

The key model constants used in the transport equations are [5]:

$$\beta_0 = 0.0708, \quad f_\beta = \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega}, \quad \chi_\omega \equiv \frac{\left| \Omega_{ij} \Omega_{jk} S_{ki} \right|}{(\beta\,\omega)^3}$$

$$\varepsilon = \beta^* \omega k, \qquad \ell = \frac{k^{1/2}}{\omega}, \quad \Omega_{ij} = \tfrac{1}{2}\left(\frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i} \right)$$

$$\alpha = \frac{13}{25}, \qquad \beta = \beta_0 f_\beta, \qquad \beta^* = \frac{9}{100}$$

$$\sigma_\omega = \frac{1}{2} \quad \sigma_k = \frac{3}{5} \qquad \sigma_d = \begin{cases} 0, & \dfrac{\partial k}{\partial x_j}\dfrac{\partial \omega}{\partial x_j} \le 0, \\ 1/8, & \dfrac{\partial k}{\partial x_j}\dfrac{\partial \omega}{\partial x_j} > 0. \end{cases}$$

The values for σ_k and σ_ω may vary depending on the specific Wilcox revision adopted in the custom solver, emphasizing that consistency with the numerical code implementation must be maintained.

3.4 Pressure-Velocity coupling: SIMPLE Algorithm

For incompressible flows, the pressure field is implicitly linked to the velocity field through the continuity constraint. The Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm is a widely used iterative procedure designed to enforce this coupling between pressure and velocity [8].

1. **Guess:** A pressure field (p^*) and initial velocities (U^*, V^*) are guessed.
2. **Momentum solution:** The discretized momentum equations are solved using the guessed pressure field (p^*) to obtain an intermediate velocity field (U^*, V^*) that may not yet satisfy continuity.
3. **Pressure correction equation:** A pressure correction (p') equation is derived by substituting the corrected velocities into the continuity equation. This equation is then solved to obtain p'.
4. **Correction:** The pressure and velocity fields are updated as
$$p = p^* + \alpha_p p', \quad U = U^* + U', \quad V = V^* + V'$$
5. **Iteration:** The process is repeated until a defined convergence criterion is met, ensuring that both the momentum and continuity equations are satisfied simultaneously.

The SIMPLE algorithm ensures that the final calculated velocity field satisfies continuity, which is paramount for an accurate numerical solution of incompressible flow problems.

4. NUMERICAL SETUP AND DISCRETIZATION

This chapter outlines the physical constraints of the flow domain, the spatial and temporal discretization techniques, and the robust solution control methods implemented in the custom MATLAB solver. These choices, together with a good implementation, are essential for achieving the required stability, convergence and model accuracy.

4.1 Computational domain and boundary conditions

The simulation analyzes turbulent flow past a square block of side D. The limits and constraints imposed on the domain attempt to accurately represent the flow physics while minimizing numerical interference.

Geometry and dimensions

The flow field is modeled within a structured, two-dimensional rectangular domain. The dimensions are chosen large enough to minimize the physical blockage effects and avoid spurious boundary reflections affecting the wake.

- The overall length is determined by summing the inlet length, block size, and outlet length, making a total of $24D$.
- The channel height was chosen to ensure a low blockage ratio to maintain conditions resembling an internal flow channel. This followed a 14:1 ratio, where 14D is height relative to 1D block height, as indicated in Fig (1).

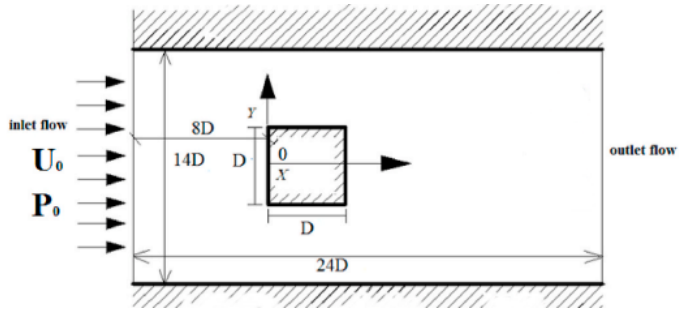


Figure 1: Computational domain geometry, showing key dimensions (D, 14D) and boundary condition locations. Reproduced from [3].

Boundary conditions

- **Inlet:** A uniform velocity profile is prescribed at the inlet boundary, where the velocity is constant across the channel height, $U = U_{IN}$. The vertical velocity component is set to $V = 0$. The turbulence quantities k_{in} and ω_{in} are initialized based on a selected turbulence intensity (commonly $I = 0.05$) and a representative turbulence length scale related to the inlet height and block height. This uniform profile represents a standard, simplified inflow condition that improves numerical stability in the initial simulation phases.
- **Solid walls:** All walls, including the faces of the created block geometry, are treated as no-slip boundaries, enforcing $U = V = 0$ at the wall. The turbulent kinetic energy is set to $k = 0$ directly at the wall.
- **Near-Wall treatment (High-Re):** Since the final stable solution utilizes a High-Reynolds number formulation of the Wilcox $k-\omega$ model, wall functions are explicitly employed to bridge the inner, viscous sublayer region. This approach assumes the near-wall boundary layer is in equilibrium, avoiding the need to resolve the steep velocity gradients immediately adjacent to the wall. Consequently, the boundary conditions for the turbulence quantities (k and ω) are applied at the first cell center, ensuring y^+ values are maintained within the range $30 < y^+ < 100$ [1].
- **Outlet:** A zero-gradient (Neumann) boundary condition is applied to all transported variables (U, V, k, ω) at the outlet, such that $\frac{\partial \Phi}{\partial x} = 0$. This condition assumes that the flow is fully developed [8]. The static pressure at the outlet is set to a fixed reference value, which provides a stable pressure reference and ensures numerical convergence of the pressure-velocity coupling.

Meshing and Near-wall Resolution

The primary constraint for this model implementation is the near-wall meshing: the mesh must ensure the dimensionless wall distance y^+ of the first cell center is within the log-layer region, $y^+ \geq 30$. This coarser resolution near the wall is appropriate because the chosen High-Re $k-\omega$ model utilizes the aforementioned wall functions, which approximate the physics in the viscous sublayer.

Modeling methodology note:

The initial project objective, detailed in *Appendix A*, was to implement a Low-Reynolds (wall-integrating) $k-\omega$ formulation. However, due to numerical instabilities encountered at the sharp corners of the bluff body, the final stable results presented in this report were necessarily obtained using the more numerically robust High-Reynolds formulation with wall functions.

The overall system of algebraic equations is solved iteratively using the Tri-Diagonal Matrix Algorithm (TDMA), applied line-by-line across the computational domain for each variable transport equation, enabling efficient computation of the solution vector.

4.2 Spatial discretization and meshing strategy

The continuous RANS and $k-\omega$ transport equations (*Chapter 3*) are transformed into a system of coupled linear algebraic equations using the Finite Volume Method (FVM).

Governing algebraic equations

The general steady transport equation for a scalar ($\phi \in \{k, \omega\}$) in a control volume P is discretized into the form [8]:

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + S_U + S_P \phi_P$$

This is rearranged for solution stability by incorporating the implicit source term (S_P) into the diagonal:

$$\underbrace{(a_P - S_P)}_{\text{Diagonal}} \phi_P - \sum_{nb} a_{nb} \phi_{nb} = S_U$$

Spatial Schematization (Hybrid Differencing)

The coefficient a_{nb} combines convection (F_{nb}) and diffusion (D_{nb}) terms using a *Hybrid Schematization*. The coefficients are defined generally and using the maximum function:

$$\text{Model specific} = \begin{cases} a_E & = \max\left(-F_e, D_e - F_e/2, 0\right) \\ a_W & = \max\left(F_w, D_w + F_w/2, 0\right) \end{cases}$$

where F_f represents the convective mass flux and D_f is the diffusive conductance across face f.

This blend ensures stability by defaulting to Upwind when convection dominates ($|F_{nb}| > D_{nb}$).

The diffusive conductance D_f for face f uses the effective kinematic diffusivity Γ_ϕ:

$$D_f = \Gamma_\Phi|_f \frac{A_f}{\delta_f}$$

4.3 SOURCE TERM LINEARIZATION

This section addresses the transformation of the PDEs into the final linear algebraic systems solved at each time step.

Linearization

The source terms S_ϕ for the k and ω transport equations contain non-linear, inter-dependent components (e.g., the production term P_k depends on μ_t, which depends on ω). To improve the iterative stability and accelerate convergence, the source term is linearized into an explicit component (S_u) and an implicit component ($S_p \phi_P$):

$$S_\phi = S_u + S_p \phi_P$$

where S_p is designed to be stabilizing and ϕ_P is the current time step's unknown value at the cell center P.

- **Implicit stabilization:** The implicit coefficient S_p is constructed to be stabilizing (negative or zero) to reinforce the diagonal dominance of the system matrix. For instance, the destruction term in the k-equation $(-\beta^*\omega k)$ is linearized as:

$$S_P(k) = -\beta^*\omega$$

  The negative S_p term is then transferred to the diagonal coefficient a_P (as per Equation in the Theory section).

- **Explicit source (S_u):** The explicit component S_u is primarily composed of the Production of Turbulent Kinetic Energy (P_k). For instance, the production term (P_k) is added entirely to S_u (the right-hand side) since it is a positive-definite term that drives the turbulence.

4.4 Solution control and Convergence

Since the RANS equations were solved using a pseudo-transient time marching scheme to aid stability and convergence towards a steady-state solution, the process requires temporal discretization.

- **Temporal discretization:** The equations are integrated over time using the First-Order Implicit Euler Scheme, chosen for its inherent unconditional stability.

- **Pseudo-time step ($\Delta\tau$):** The time-like step size, $\Delta\tau$, is treated as a numerical parameter to control the stability of the iterative process [8]. While the simulation uses this transient approach, the focus is on achieving the final steady-state solution rather than capturing the exact physical transient phenomena.

- **Under-relaxation:** Under-relaxation factors are applied explicitly within the SIMPLE loop to manage the solution stability and prevent oscillations during the pressure-velocity coupling iterations, which is essential for ensuring robust iterative convergence.

- **Convergence criteria:** The solver is designed to meet a strict final tolerance of 10^{-5} for all primary variables (U, V, P, k, ω). However, due to the inherent unsteadiness of the turbulent flow, the primary criteria for solution legitimacy and data extraction relied on the strict mass balance targets, requiring the continuity residuals to stabilize below $S_{max} \leq 10^{-8}$ and $S_{avg} \leq 10^{-9}$.

5. RESULTS & DISCUSSIONS

This chapter presents the transient and time-averaged results obtained from the RANS simulation using the implemented Wilcox (2006) k–ω model and validates these results against the objectives established in *Chapter 1*.

The numerical results presented below are produced by the custom MATLAB solver created and make use of the values described in *Table 1*.

Table 1: Turbulence-model constants used in the solver [8].

Constant	Value	Constant	Value
σ_k	2	σ_ω	2
κ	0.4187	γ_1	0.553
B^*	0.09	B_1	0.075

5.1 Time-averaged centreline velocity profile

Fig (2) shows the dimensionless, time-averaged longitudinal velocity profile along the channel centerline, normalized by the inlet velocity U_{in}. Data points were extracted from the MATLAB output file `velu.txt` and averaged over the total physical sampling period.

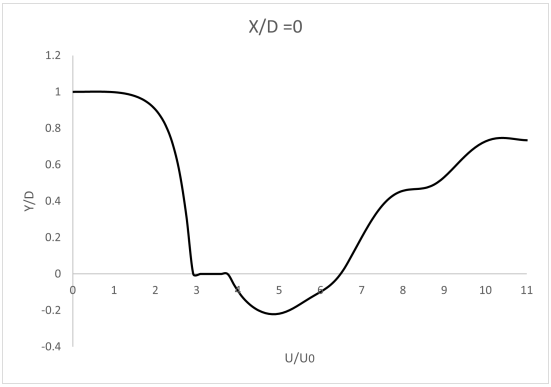


Figure 2: Dimensionless longitudinal velocity along the centerline, $\overline{U}/U_{\text{in}}$ vs. x/D.

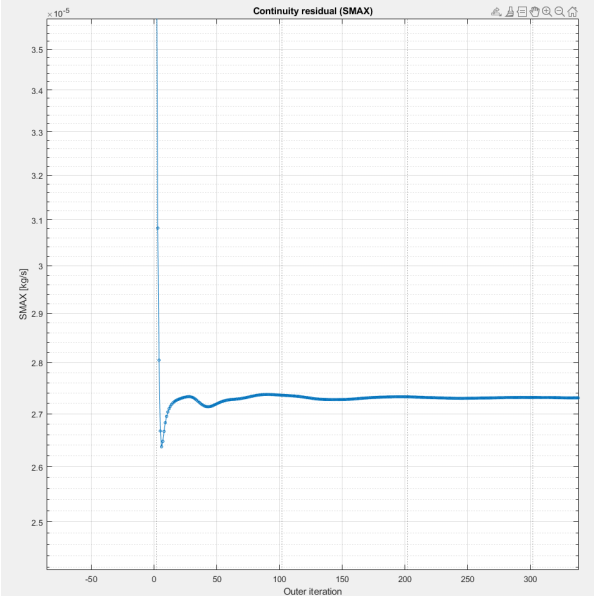
The resulting profile shows good agreement with reference data from literature, confirming that the implemented k–ω model accurately captures the velocity recovery and wake structure behind the bluff body [3].

5.2 Residuals and convergence

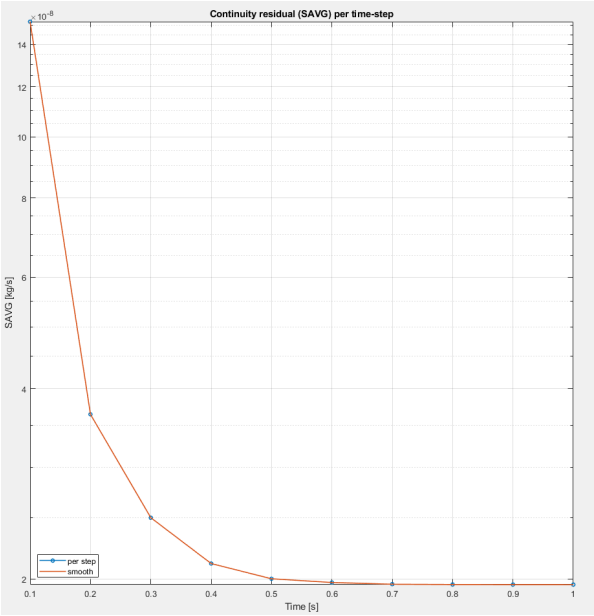
Residuals are used to assess the numerical stability achieved during each physical time step. In this study, the maximum and average residuals of the continuity equation (S_{max} and S_{avg}) are monitored throughout the simulation.

The maximum residual, S_{max}, represents the highest local mass imbalance, while the average residual, S_{avg}, gives the overall mean imbalance. As the inner SIMPLE loop iterates up to 10 times per time step, the following behavior is observed:

- **Continuity stabilization:** Fig (3a) and Fig (3b) show a clear initial reduction followed by a stabilized plateau. This confirms that the mass balance is satisfied and stabilized, ensuring a valid solution for the specific time step used.



(a) Maximum residual $S_{\max}$ versus iteration.



(b) Average residual S_{avg} versus time step.

Figure 3: Residual histories showing convergence behavior of the transient SIMPLE algorithm.

5.3 Independence studies

To ensure the time-averaged results presented in this chapter are robust and physically meaningful, different studies were performed. These verification analyses ensure that the captured flow features are genuine and not influenced by the discretization choices in space or time, thus validating the numerical fidelity of the solver.

Grid independence study

A grid independence study was performed to ensure that the numerical results are not influenced by the mesh resolution. Three structured meshes were tested, consisting of 120×70, 240×140, and 480×280 control volumes. The longitudinal velocity profiles were evaluated at the beginning of the square block for each grid and compared to assess mesh sensitivity.

Figure 4 shows that the medium and fine grids (240×140 and 480×280) produced nearly identical velocity distributions, while the coarsest grid (120×70) slightly mismatches the velocity gradients near the wall. Therefore, the 240×140 mesh was selected for all subsequent simulations as it provides a good balance between computational cost and accuracy. The grid can be visualize in Figure 6, as each quiver vector is represent a mesh cell.

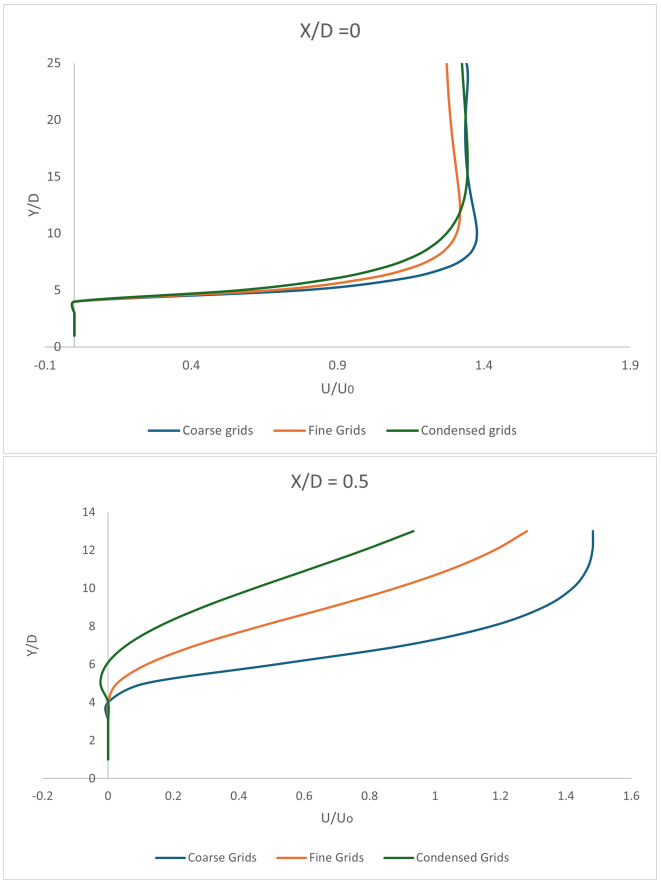


Figure 4: Comparison of longitudinal velocity profiles at different stages and for different grid resolutions.

Time independence study

A physical time step (Δt) independence study was conducted to verify that the unsteady results are independent of the chosen time step size. Simulations were performed up to $t = 1s$ using three different time step values, $\Delta t = 0.1$, $\Delta t = 0.05$, and $\Delta t = 0.01$.

The longitudinal velocity profiles obtained with all three time steps showed negligible variation, indicating that the temporal discretization is sufficiently fine to capture the evolution of the flow field accurately. As shown in Fig (5), the profiles overlap closely at both the beginning and the middle of the square block. This confirms that $\Delta t = 0.1$ is sufficiently small to capture the transient flow behavior accurately without introducing significant temporal discretization errors. Consequently, the values, which are summarized in *Table 2*, was adopted for all subsequent unsteady simulations.

Table 2: Numerical setup and solver controls (current run).

Item	Value / Setting
Domain size ($X_{max} \times Y_{max}$)	0.56×0.11 m
Mesh (current) ($N_x \times N_y$)	240×70 (60*3, 35*3)
Inlet mean velocity U_{in}	0.1 m s^{-1}
Time step Δt & total time	0.01 s, $t_{end} = 1$ s
Outer iterations (max)	MAX_ITER = 30
Momentum iters per sweep	U_ITER = 1, V_ITER = 1
Pressure-correction iters	PC_ITER = 30
k, ω, T iters per sweep	K_ITER = 1, Ω_ITER = 1, T_ITER = 1
Mass-balance targets	$S_{max} \leq 10^{-8}$, $S_{avg} \leq 10^{-9}$

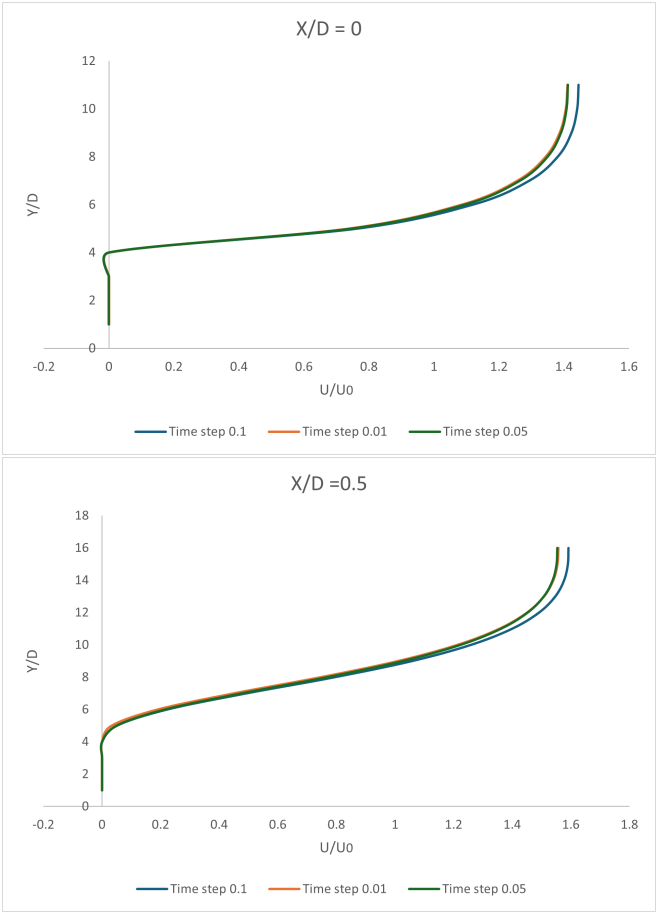


Figure 5: Comparison of longitudinal velocity profiles at different stages and for different time-step values ($\Delta t = 0.1$ and $\Delta t = 0.01$).

5.4 Development of transverse velocity along the upper surface

To analyze the evolution of the flow over the square block, time-averaged longitudinal velocity (u/u_0) profiles were extracted at several streamwise locations along the upper surface of the block. Fig (7) shows the non-dimensional velocity distribution at four representative positions: at the leading edge of the square surface, at the midpoint of the surface, and further downstream at 25% and 50% of the overall domain length.

Near the front edge of the block, the flow remains relatively uniform, and the velocity gradient normal to the wall is small, indicating the presence of a thin developing boundary layer. As the flow progresses downstream, the boundary layer thickens and the velocity profile becomes increasingly skewed. This behavior is characteristic of a shear layer developing over a bluff body, where momentum is transferred from the free stream to the wall region through viscous and turbulent processes.

At approximately the midpoint of the square surface, the velocity gradient near the wall becomes more pronounced, while the external flow experiences mild acceleration due to local pressure redistribution around the obstacle. Further downstream (25%-50% of the domain length), the velocity profiles begin to approach an asymptotic form, indicating that the flow gradually recovers and transitions toward a more stable boundary layer after passing over the block.

Overall, these results demonstrate that the numerical solution captures the progressive thickening of the boundary layer and the characteristic development of shear flow over a square bluff surface. The observed trends are consistent with classical boundary layer theory and align with expected physical behavior reported in literature [3] for similar geometries.

5.5 Velocity Magnitude and Vector Field Distribution

Fig (6) presents the instantaneous velocity vector field superimposed on the velocity magnitude contours with time steps $\Delta t = 0.01$. The quiver plot clearly illustrates the formation of a well-defined shear layer originating from the leading edge of the square block. As the flow interacts with the upstream face of the block, a region of reduced velocity develops immediately behind it, forming a wake characterized by recirculation and low-momentum fluid.

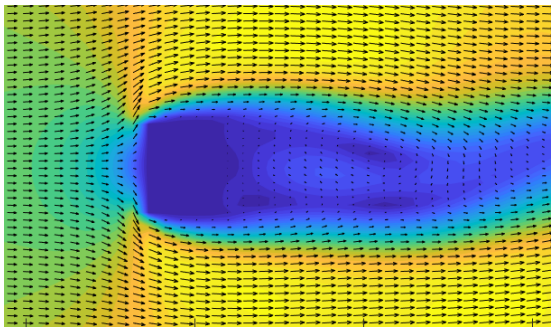
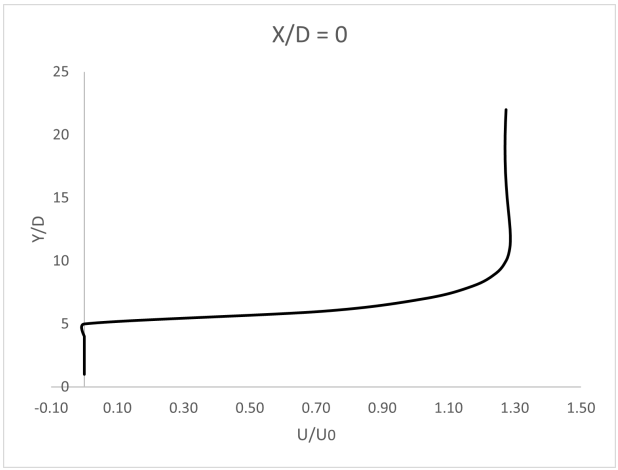


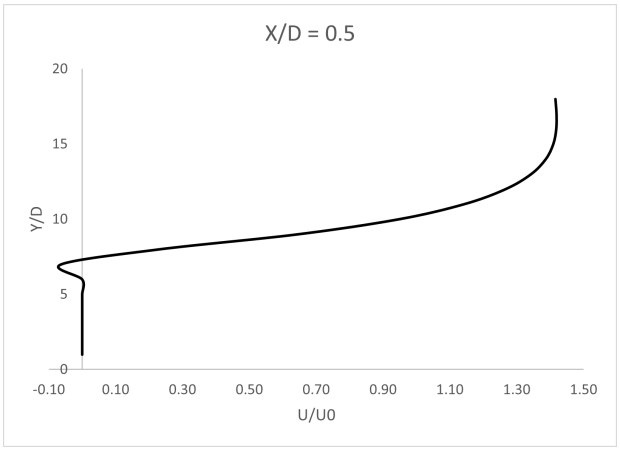
Figure 6: Instantaneous velocity vector field and corresponding velocity magnitude contours using time steps $\Delta t = 0.01$. The arrows illustrate the local flow direction and strength, while the color map represents the magnitude of the velocity field.

The high-velocity streamlines are observed to deviate around the block, generating strong velocity gradients near the upper and lower surfaces. Downstream of the obstacle, the wake region gradually widens and the velocity deficit is progressively recovered as mixing and turbulent diffusion transport momentum back into the wake. This behavior is consistent with classical bluff-body aerodynamics, where flow separation and vortex shedding lead to a distinct wake profile.

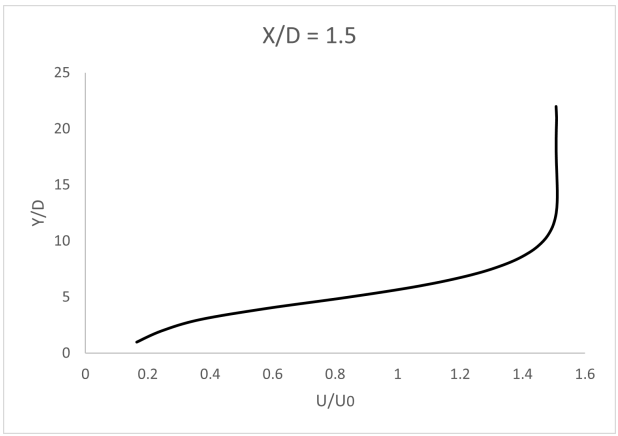
The averaged flow field demonstrates smoother spatial gradients compared to the instantaneous velocity snapshots, indicating that the unsteady fluctuations are largely suppressed when time-averaged. This confirms that the turbulent structures in the wake are inherently transient, while the mean flow retains the expected large-scale features of separated flow past a square obstacle.



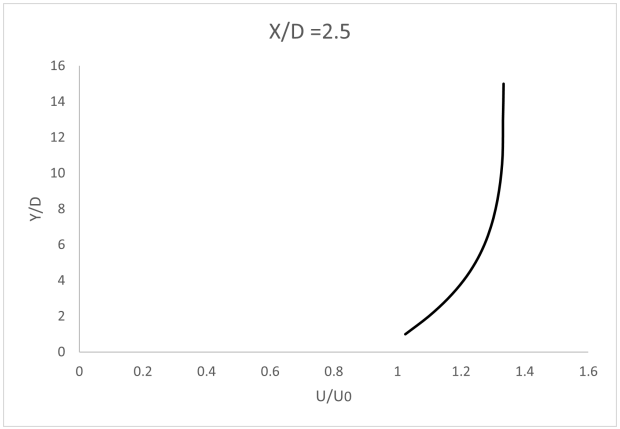
(a) u/u_0 at the beginning of the square surface



(b) u/u_0 at the middle of the square surface



(c) u/u_0 at 25% of the domain length



(d) u/u_0 at 50% of the domain length

Figure 7: Time-averaged transverse velocity profiles at various streamwise positions along the upper surface of the square block.

5.6 Wake Structure and Vortex Formation Behind the Square Block

Fig (8) illustrates the velocity magnitude field with superimposed streamlines around the square block. Immediately downstream of the block, a separated shear layer is observed to form, resulting in the development of a well-defined recirculation region. The closed streamline patterns within this region indicate the presence of eddies and reverse flow, which are characteristic features of bluff-body wakes.

The size and shape of the wake gradually evolve as the flow is convected downstream. The shear layers originating from the upper and lower surfaces of the block interact and periodically roll up, generating vortex structures that are advected into the wake. Further downstream, these vortices dissipate and the flow slowly re-establishes into a more uniform velocity profile as momentum is transferred back into the wake through turbulent mixing.

This phenomenon aligns with classical bluff-body flow dynamics, where flow separation at sharp corners leads to vortex shedding and a pronounced velocity deficit behind the obstacle.

5.7 Turbulent Kinetic Energy (K) Distribution

Fig (9) shows the contour of turbulent kinetic energy (K) around the square obstacle within the flow domain. The plot clearly illustrates that the region immediately downstream of the obstacle exhibits the highest kinetic energy levels, indicating intense turbulence and vortex shedding. The bright yellow region behind the obstacle represents the strong mixing zone where flow separation occurs. This is consistent with the expected wake formation behavior in bluff-body flows, where the shear layers from the obstacle's edges interact and generate vortical structures, leading to elevated turbulence levels in the near-wake region.

When compared with the velocity magnitude and streamline plot (figure 8), the turbulent kinetic energy distribution exhibits complementary behavior. Regions of low velocity and strong recirculation behind the obstacle correspond to zones of high kinetic energy, confirming that turbulence is most significant where the flow detaches and reattaches. Further downstream, the kinetic energy gradually dissipates, and the contours become smoother, showing the decay of turbulence intensity as the flow stabilizes. This pattern validates that the flow transitions from a highly turbulent wake near the obstacle to a more uniform and laminar region farther downstream.

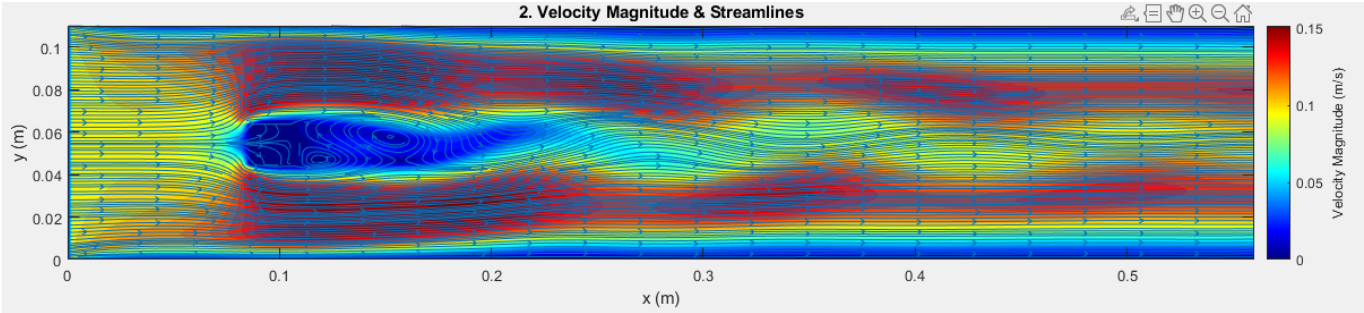


Figure 8: Velocity magnitude contours with streamlines showing flow separation and vortex formation downstream of the square block. The recirculation region and wake development are clearly visible behind the obstacle.

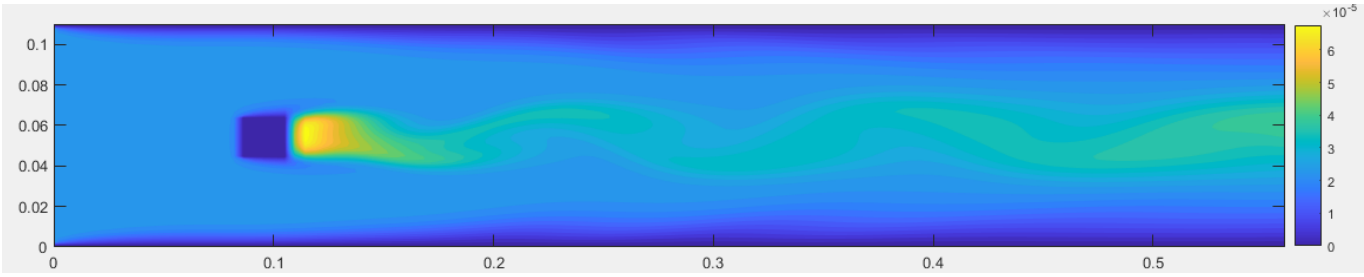


Figure 9: Turbulent Kinetic Energy (K) contour plot around the square obstacle.

6. CONCLUSIONS

This study numerically investigated the complex turbulent flow over a square bluff body using a custom Reynolds-Averaged Navier-Stokes (RANS) solver developed in MATLAB. The initial objective of implementing a wall-resolving, Low-Reynolds number $k-\omega$ formulation with the Backward-Facing Step (BFS) geometry proved computationally intractable due to numerical instabilities (*Appendix A*).

Consequently, the project successfully pivoted to the configuration used for the final analysis, thereby fulfilling Objective 1 (model implementation). This successful configuration utilized the more robust High-Reynolds number Wilcox (2006) $k-\omega$ model with wall functions and the simpler square block geometry.

The solver was then rigorously verified against Objective 2 (numerical independence studies). The systematic grid and physical time step (Δt) independence studies confirmed that the medium mesh (240×140) and a time step of $\Delta t = 0.1$ provided a resolution sufficient to isolate the physical flow dynamics from discretization artifacts. The continuity residuals (S_{max}, S_{avg}) representing mass imbalance achieved a stabilized state. This stability confirmed the solution represented a physically transient flow state, suitable for time-averaged analysis, rather than an artificial numerical divergence.

The results successfully achieved Objective 3 (qualitative flow feature analysis). The time-averaged velocity profiles, flow visualizations, and Turbulent Kinetic Energy (K) distributions demonstrated that the implemented High-Re model captures the essential features of bluff-body aerodynamics:

- **Recirculation zone:** The centerline velocity profile confirmed the existence and recovery of the wake behind the obstacle.
- **Shear layer development:** The longitudinal velocity profiles along the wall demonstrated the expected thickening and skewing of the turbulent shear layer.
- **Turbulence production:** The K contour plot localized the highest kinetic energy levels precisely in the turbulent mixing zone immediately downstream of the bluff body.

Overall, the project validated the custom MATLAB solver's robustness in handling the complex High-Re RANS formulation and demonstrated the critical ability to adapt the modeling strategy in response to numerical challenges. Future work should focus on extending the stability of the custom solver to lower Reynolds numbers or employing more advanced closures such as the $k-\omega$ SST model for broader applicability.

REFERENCES

[1] CFD-Online Wiki (n.d.). "Near-wall treatment for k-ω models". URL https://www.cfd-online.com/Wiki/Near-wall_treatment_for_k-omega_models.

[2] CFD-Wiki, CFD-Online (n.d.). "K-epsilon models". Content licensed under GFDL 1.2, URL https://www.cfd-online.com/Wiki/K-epsilon_models.

[3] MALIKOV, Z. and MADALIEV, M. (2022). "Numerical simulation of separated flow past a square cylinder based on a two-fluid turbulence model". *Journal of Wind Engineering and Industrial Aerodynamics*, **231**, 105171. URL https://www.sciencedirect.com/science/article/pii/S0167610522002677.

[4] MENTER, F.R. (1992). "Improved two-equation *k*-ω turbulence models for aerodynamic flows". NASA Technical Memorandum NASA-TM-103975, NASA, Ames Research Center, Moffett Field, CA. Often cited as the foundational report preceding the SST model, URL https://ntrs.nasa.gov/citations/19930013620.

[5] NASA Langley Research Center (n.d.). "The wilcox k-ω turbulence model". URL https://turbmodels.larc.nasa.gov/wilcox.html.

[6] SimScale Documentation (n.d.). "k-ω sst". URL https://www.simscale.com/docs/simulation-setup/global-settings/k-omega-sst.

[7] SimScale GmbH (n.d.). "K-epsilon turbulence model | global settings". URL https://www.simscale.com/docs/simulation-setup/global-settings/k-epsilon/.

[8] VERSTEEG, H.K. and MALALASEKERA, W. (2007). *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. 2nd ed. Pearson Education, Harlow. URL https://www.scribd.com/document/553752831/Versteeg-Malalasekera-2ed.

[9] WILCOX, D.C. (2006). *Turbulence Modeling for CFD*. 3rd ed. DCW Industries, Inc., La Cañada, CA.

APPENDIX A

In this appendix section the long journey followed to achieve a functional matlab turbulent model is explained, narrating the aim to achieve certain project objectives that were changed and updated based on the iterative progress on the model

DEVELOPMENT AND DEBUGGING OF THE *k*–ω MATLAB SOLVER

The development of a stable and accurate CFD solver based on the Wilcox (2006) *k*–ω model required a rigorous, iterative process of implementation, testing, and debugging. The evolution from the base *k*–ε solver to a physically consistent, numerically stable *k*–ω formulation involved challenges at both the model-physics level and the code-architecture level. This chapter summarizes the major stages of development, highlighting key difficulties and the systematic modifications that led to the final working solver.

Model conversion and implementation

The initial objective was to transform the transient *k*–ε solver provided in the course materials into a steady-state *k*–ω solver following the Wilcox (2006) formulation. This process required fundamental structural changes:

- **Model equations:** The ε transport equation was replaced with a new ω equation, introducing distinct production and destruction terms and a non-linear cross-diffusion term. The eddy viscosity definition and *k*-equation source terms were also redefined to match the *k*–ω framework.
- **Geometry definition:** The original 2D channel flow was replaced with a backward-facing step (BFS) configuration. An index-based cell-blocking system was implemented to deactivate solid cells within the computational domain.
- **Numerical schemes:** The algebraic solver was upgraded to a Tri-Diagonal Matrix Algorithm (TDMA) line-by-line method for efficient matrix solution. The Hybrid Differencing Scheme was applied to all transport equations to enhance numerical stability under convection-dominated conditions.

While these changes successfully reformulated the mathematical structure of the solver, they also introduced multiple instability sources, including geometry misidentification, wall-boundary inconsistencies, and unbounded turbulence production.

Initial debugging and solver instabilities

The first execution attempts resulted in catastrophic divergence within the first few iterations. The maximum mass imbalance (`SMAX`) exceeded 10^3, indicating severe non-physical behavior. Investigation revealed that the issue originated not from the turbulence model but from a coding flaw within the SIMPLE velocity-correction step. The `velcorr` function was mistakenly applying velocity corrections to solid cells on the step wall, creating an artificial mass injection point. The fix involved explicitly excluding solid-face indices from correction loops, which restored global mass conservation.

Subsequent simulations exposed a geometry logic error: the step was not properly defined, and the flow behaved as if the domain were a flat channel, producing a parallel velocity field (Figure 10). The cause was traced to incorrect logical conditions used to mark solid cells (`if J < Jstep_s`) without accounting for the horizontal index I. This caused all cells below the step height to be treated as solid across the entire length.

To eliminate this fragility, the geometry logic was refactored using three global Boolean masks: *isFluid_s*, `isFluid_u`, and `isFluid_v`. These matrices were defined during initialization and served as the single reference for solid/fluid status across all coefficient and boundary functions. This "single source of truth" eliminated geometry inconsistencies across solver routines.

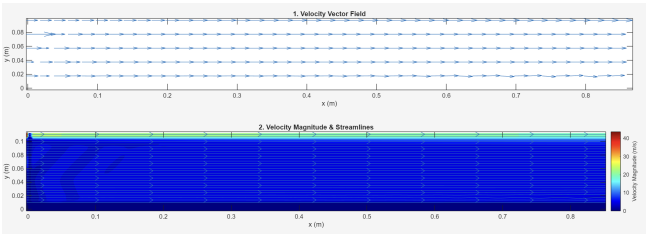


Figure 10: Incorrect flow path during early development stages due to faulty geometry logic.

Physical model conflicts and transition to Low-reynolds formulation

Once the geometry was correctly implemented, the flow began to separate at the step corner. However, this triggered a new instability: rapid numerical blow-ups at the re-entrant corner (Figure 11). The High-Reynolds-number version of the k–ω model proved incompatible with the singular velocity gradients at the sharp corner. The wall functions assumed an equilibrium turbulent boundary layer, which does not exist in the separated region. Consequently, the turbulence production term ($P_k \propto |\nabla U|^2$) grew uncontrollably, causing the turbulent viscosity μ_t and velocities to diverge.

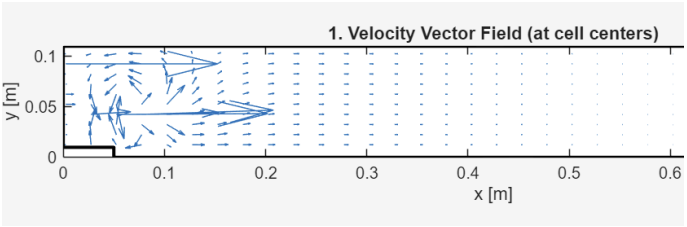


Figure 11: Numerical instability originating at the re-entrant corner of the step.

The resolution required a full transition from a High-Re to a Low-Reynolds-number formulation. This involved:

- **Grid refinement:** Increasing mesh density (e.g., from 170×22 to 440×70) to ensure that the first grid point satisfied $y^+ \leq 1$ and resolved the viscous sublayer.
- **Boundary condition revision:** Replacing wall functions with explicit no-slip and Low-Re near-wall conditions:

$$U = 0, \quad V = 0, \quad k = 0, \quad \omega = \frac{6\mu}{\rho\beta(dy)^2}.$$

- **Removal of wall-function logic:** Deleting all empirical High-Re wall models from `ucoeff` and `omegacoeff` functions to maintain physical consistency.

This transition eliminated the unbounded growth at the corner and improved overall model robustness, but it also exposed residual implementation errors in boundary loops and solver routines.

Final stabilization and code refinement

The Low-Re formulation stabilized the turbulence field but uncovered localized numerical spikes in pressure, particularly near the step's upper boundary (Figure 12). These were traced to conflicts between the boundary condition logic and the new Boolean masking. The `bound.m` function was incorrectly applying no-slip conditions to cells marked as fluid near the step, effectively overconstraining the system. Adjusting these loops to respect the mask definitions resolved the issue.

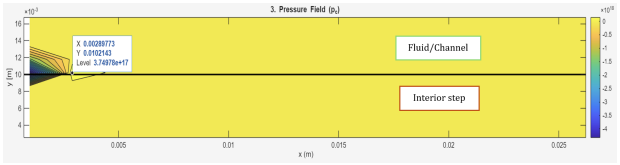


Figure 12: Pressure singularities observed near the step's top wall before final boundary logic corrections.

At the same time, several syntax and indexing errors in the solver functions were identified. A loop in `vcoeff` was iterating one index too far, attempting to solve for v on the physical top wall, while typos in the `solve_scalar` (TDMA) routine caused dimension mismatches. These were corrected, and the solver was further stabilized by removing the transient time-stepping loop. The switch from a transient to a steady-state SIMPLE-based solver allowed direct convergence toward the final steady solution, bypassing the violent initial shear-layer transients that had previously caused the simulation to diverge.

Conclusion of changes

The complete debugging process evolved from fixing basic logical errors to resolving deeper inconsistencies between the turbulence model, grid resolution, and physical flow conditions. The progression from High-Re to Low-Re modeling, along with systematic refactoring of the boundary condition logic, resulted in a solver that was numerically stable for simple flow cases but failed catastrophically when applied to the sharp corners and strong pressure gradients of the original Backward-Facing Step (BFS) geometry.

The path to a stable simulation was an exercise in systematic debugging and model re-evaluation. The initial goal of resolving the viscous sublayer using a Low-Reynolds number $k-\omega$ formulation with the complex BFS geometry proved computationally intractable given the stability issues inherent to a custom MATLAB solver.

Consequently, the project pivoted to the configuration used for the results in *Chapter 5*:

1. **Geometry change:** The flow was modeled around a simpler square block in a horizontal air channel.
2. **Model configuration:** The model reverted to the more numerically robust High-Reynolds number formulation

with wall functions ($y^+ > 30$) to bypass the need for fine near-wall grid resolution that had caused the earlier blow-ups.

This final configuration enabled the team to achieve a numerically stable state. While the stringent convergence criteria (L2-norm residuals of 10^{-5}) were not ultimately fulfilled, the model exhibited sufficient stability to perform the necessary numerical independence studies and successfully generate the time-averaged flow fields required to fulfill the primary project objective of analyzing the fundamental turbulent flow features (wake, recirculation, and shear layers) downstream of a bluff body.

This entire process highlights that in CFD, the numerical method, the grid resolution, and the physical model must all be consistent and appropriate for the specific flow problem to achieve a stable and meaningful result.

APPENDIX B

This appendix section is intended to directly show a series of advantages and disadvantages that each solver model has but which didn't fit with the flow of report, being displayed here.

Summary of Turbulence models

Standard $k-\varepsilon$ ***model***

Advantages:

- Simple, robust, and widely validated for engineering flows.
- Performs well for fully turbulent, high-Reynolds-number flows.
- Relatively insensitive to freestream turbulence levels.
- Low computational cost.
- Works well for free shear flows (jets, mixing layers) and simple wall-bounded flows with mild pressure gradients.
- Compatible with wall functions (does not require near-wall resolution).

Disadvantages:

- Poor performance in flows with strong adverse pressure gradients or separation (e.g., backward-facing step).
- Assumes isotropic turbulence; inaccurate for swirling or curved flows.
- Requires wall functions or damping corrections for near-wall modeling.
- Not ideal for low-Reynolds-number regions or laminar–turbulent transition.

Original $k-\omega$ ***model (1988)***

Advantages:

- Accurate near-wall behavior — can be integrated down to the wall (no wall functions needed).
- Predicts boundary layer separation and reattachment more accurately than *k*–ε.
- Performs well in flows with adverse pressure gradients.
- Good for aerodynamic and turbomachinery applications.

Disadvantages:

- Highly sensitive to freestream or inlet ω values — may yield very different results with different far-field turbulence levels.
- Can overpredict turbulence in free shear layers.
- Slightly more computationally expensive than *k*–ε.
- Still assumes isotropic turbulence (like all eddy-viscosity models).

Wilcox (2006) "Standard" $k-\omega$ ***model***

Advantages:

- Improved over the original *k*–ω: revised constants and cross-diffusion term for better accuracy in free shear and boundary layer flows.
- Includes eddy viscosity limiters to prevent unrealistically high turbulence in stagnation regions.
- Good near-wall accuracy and improved stability.
- Performs well for adverse pressure gradients and moderate separation.

Disadvantages:

- Still shows some freestream sensitivity (though less than the original model).
- Computationally more complex due to additional terms.
- May not perform as well in pure free-shear regions as *k*–ε.

$k-\omega$ ***SST Model (Menter's Shear Stress Transport)***

Advantages:

- Blends *k*–ω near walls and *k*–ε in the freestream, combining their strengths.
- Eliminates freestream sensitivity by reverting to *k*–ε behavior far from walls.
- Excellent prediction of flow separation, adverse pressure gradients, and stalled aerodynamic flows.
- Includes shear stress transport limiter, improving realism in separated and stagnation regions.
- Can be integrated to the wall (no wall functions required).
- Generally the most accurate and reliable two-equation RANS model for a broad range of engineering flows.

Disadvantages:

- Slightly higher computational cost than standard models.
- More complex formulation (blending functions and limiters).
- Still assumes isotropic turbulence (limitations of eddy-viscosity concept).
- Occasionally, the original *k*–ω may outperform it for very specific benchmark cases (e.g., some backward-facing step experiments).